



# LUND UNIVERSITY

## IT-kvalitet i praxis: systemutvecklarens kunskap om och syn på kvalitet

Steen, Odd

2002

[Link to publication](#)

*Citation for published version (APA):*

Steen, O. (2002). *IT-kvalitet i praxis: systemutvecklarens kunskap om och syn på kvalitet*. [Doktorsavhandling (monografi), Institutionen för informatik]. Department of Informatics, Lund University.

*Total number of authors:*

1

### General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117  
221 00 Lund  
+46 46-222 00 00

**Lund Studies in Information and Computer Sciences, ISSN 0283-6386**

1. Olerup, Agneta (1982). "A Contextual Framework for Computerized Information Systems. An exploratory study of computerized information systems in nine organizations with regard to technological and environmental factors". Nyt Nordisk Forlag Arnold Busk, Kobenhavn.
2. Wormell, Irene (1985). "SAP Subject Access Project. Improved retrieval for monographic publication ", Dept. of Information & Computer Science, Lund University, Lund.
3. Sandström, Gunhild (1985). "Towards Transparent Data Bases - How to interpret and act on expressions mediated by computerized information systems". Chartwell-Bratt Studentlitteratur, Lund.
4. Baark, Erik (1986). "The Context of National Information Systems in Developing Countries. India and China in a comparative Perspective". Research Policy Institute, Lund University, Lund.
5. Flensburg, Per (1986). "Personlig databehandling - introduktion, konsekvenser, möjligheter", Chartwell-Bratt Studentlitteratur, Lund.
6. Friis, Siv. (1991). "User Controlled Information Systems Development - problems and possibilities towards Local Design Shops". Dept. of Information & Computer Science, Lund University, Lund.
7. Carlsson, Sven (1993). "A Longitudinal Study of User Developed Decision Support Systems". Dept. of Informatics, Lund University, Lund.
8. Sagheb-Tehrani, Medhi (1993). "Expert Systems Development: Some problems, motives and issues in an exploratory study", Dept. of Informatics, Lund University, Lund.
9. Lindh, Jörgen (1993). "Datorstödd undervisning i skolan - möjligheter och problem". Studentlitteratur, Lund.
10. Hägerfors, Ann (1994). "Co-learning in Participative Systems Design", Dept. of Informatics, Lund University.
11. Ingman, Sissi (1997). "Förtroende och datorbruk", Dept. of Informatics, Lund University, Lund.
12. Eriksén, Sara (1998). "Knowing and the Art of IT Management", Dept. of Informatics, Lund University, Lund.
13. Zhang, Xiu Hua (1999). "User Participation in Object-Oriented Contexts", Dept. of Informatics, Lund University, Lund.
14. Lannér, Olof (1999). "Datorstöd i skrivandet - En longitudinell studie på grundskolan och gymnasieskolan", Dept. of Informatics, Lund University, Lund.
15. Messeter, Jörn (2000). "Operatörens blick - om inplacering av IT-stöd i erfarenhetsöverföring inom en lokal praxis". Dept. of Informatics, Lund University, Lund.

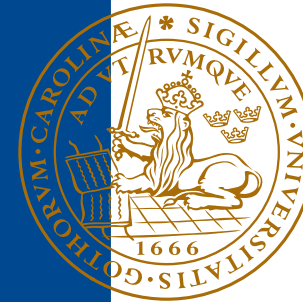
**Lund Studies in Informatics, ISSN 1651-1816**

1. Meggerle, Theis & Steen, Odd (2002), "IT-kvalitet i praxis: systemutvecklares kunskap om och syn på kvalitet", Dept. of Informatics, Lund University, Lund.



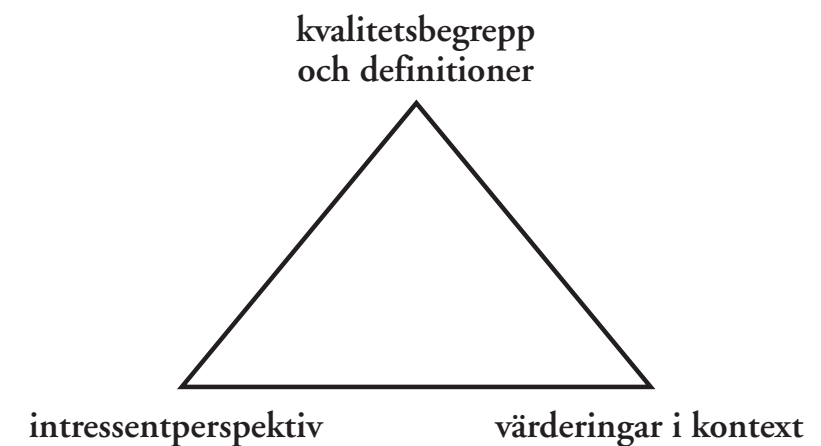
**EKONOMIHÖGSKOLAN**  
Lunds universitet

IT-kvalitet i praxis • Theis Meggerle & Odd Steen • Lund



# IT-kvalitet i praxis

systemutvecklares kunskap  
om och syn på kvalitet



Theis Meggerle & Odd Steen  
Institutionen för Informatik  
Lunds universitet

# IT-kvalitet i praxis

systemutvecklarens kunskap om och syn på kvalitet

Av

**Theis Meggerle och Odd Steen**



**EKONOMIHÖGSKOLAN**

Lunds universitet

**AKADEMISK AVHANDLING**

som för avläggande av filosofie doktorsexamen i Informatik vid  
samhällsvetenskapliga fakulteten, Lunds universitet,

kommer att offentligen försvaras i

hörsal 101

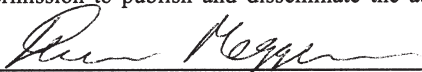
Ekonomicentrum II, Ole Römers väg 6, Lund

torsdagen den 6 juni 2002, kl. 13.00

|  |  |                       |
|--|--|-----------------------|
| Organisation<br>LUND UNIVERSITY<br>Department of Informatics<br>Ole Römers väg 6<br>S-223 63 Lund, Sweden  | Document name<br>DOCTORAL DISSERTATION |                       |
|  | Date of issue<br>2002-06-06            |                       |
| Authors<br>Theis Meggerle<br>Odd Steen   | CODEN:                                 |                       |
|  | Sponsoring organisation                |                       |
| Title and subtitle<br>IT-kvalitet i praxis: systemutvecklares kunskap om och syn på kvalitet   |  |                       |
| <p>This work is about IT quality and systems developers' knowledge and view of it. IT quality is considered as the characteristics of a system or program that influence e.g. the experience of using it, the easiness to learn and understand it, how it could be customised according to new demands, and the possibility for those who have developed it to understand how it works and can be enhanced. There are at least three good reasons for studying the concept of IT quality and systems developers' knowledge and view of it. Firstly, everybody has sometimes experienced that computer systems and programs do not work, as we would like them to do. Secondly, the use of information technology is continuously increasing and it is becoming more and more a part of the ordinary day. Consequently, the quality of information technology affects a lot of people. And thirdly, systems developers' knowledge about and view of IT quality is, to our knowledge, an almost unexplored issue.</p> <p>In this thesis we study IT quality as a concept and as a question of knowledge and knowing. Our perspective of knowledge is not only knowledge that can be verbalised and expressed as clear logical statements and rules, but also experience-based knowledge that is expressed in actions in practical situations. This implies both a theoretical and practical focus on systems developers' knowledge and a focus on professional systems developers in practice.</p> <p>With the point of departure in the concept of IT quality, theoretical and practical knowledge and practice we explore systems developers knowledge about IT quality. We have asked questions like; How do systems developers apprehend quality? Which concepts do they use when talking about quality? How do they judge quality? How can they act for better quality?</p> <p>Our empirical study, comprising 19 in-depth interviews, shows that the concept of IT quality is inherently complex and that it in itself contains a variety of qualities that together constitute the meaning of IT quality. Therefore, a common definition of IT quality as a point of departure for actual use is meaningless. Moreover, systems developers' judgements of IT quality are often based on values and comparisons. Hence, the ability to judge quality rests on personal experience, and therefore evaluations of quality can only in part be regulated by rules such as measurements, calculations and standards. Furthermore, to reach high quality is to exceed beyond the customer's requirements. Therefore, the ability to reach high quality rests on personal experiences from practice and "fingerspitzgefühle", i.e. practical knowledge and the ability to judge.</p> <p>The conclusion we draw about IT quality in practice is that concepts and definitions can pinpoint important aspects, but that quality always is somebody's evaluation from a perspective. In that sense, quality cannot be operationally defined. IT quality as a phenomenon is constituted by values, perspectives and contexts. This means that interested parties choose qualities, that can partly be expressed by concepts and definitions, and that judgements of these always are carried out in context and always are valuations.</p> |  |                       |
| Key words<br>IT quality, knowledge in practice, IT artefact, IT quality assessment, judgement ability  |  |                       |
| Classification system and /or index terms (if any)   |  |                       |
| Supplementary bibliographical information  |  | Language<br>Swedish   |
| ISSN and key title<br>1651-1816 Lund Studies in Informatics no. 1  |  | ISBN<br>91-628-5277-9 |
| Recipient's notes  | Number of pages<br>265                 | Price                 |
|  | Security classification                |                       |

Distribution by (name and address)

We, the undersigned, being the copyright owners of the abstract of the above-mentioned dissertation, hereby grant to all reference sources permission to publish and disseminate the abstract of the above-mentioned dissertation

Signature 

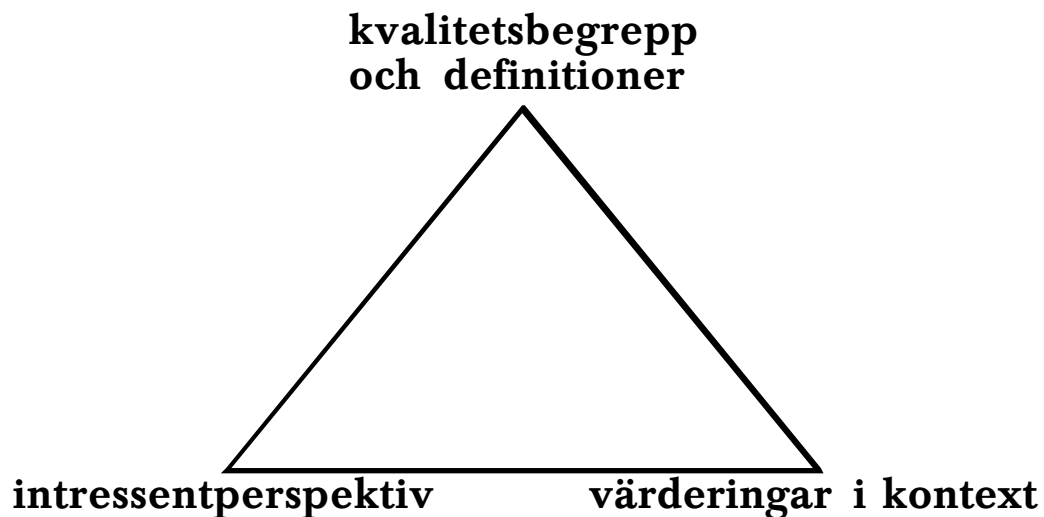
Date 2002-04-29

Signature 

Date 2002-04-29

# IT-kvalitet i praxis

systemutvecklarens kunskap om och syn  
på kvalitet



©Theis Meggerle & Odd Steen  
Institutionen för Informatik  
Lunds Universitet

ISSN 1651-1816 Lund Studies in Informatics no. 1  
ISBN 91-628-5277-9



## Förord

Just nu i den sista skälvande stunden av skrivande kan vi inte låta bli att reflektera över vilka problem vi har haft i detta arbete. Nog har vi haft problem med avhandlingen som sådan, men det är snarare skrivandet som vi här tänker på. Vi har naturligtvis använt ett ordbehandlingsprogram, vilket många gånger har varit värdefullt. Användningen har dock inte helt och hållit varit bekymmersfri. Formatproblem mellan olika datorer, innehållsförteckningar som inte stämmer och referenser i texter som är felaktiga är några av de problem vi har brottats med (mestadels Odd). Förvisso hade vi inte klarat att skriva avhandlingen utan detta verktyg (tror vi), men samtidigt har vi också lagt ner onödigt arbete på att lösa problem. Tid som skulle ha kunnat användas till själva författandet istället. Vi verkar inte heller vara ensamma om denna erfarenhet. För hur många av oss har inte varit med om att dator så att säga krånglar och inte fungerar som vi har tänkt?

Generellt sett handlar detta om IT-kvalitet. Mer specifikt handlar det om egenskaper hos produkter som påverkar vår upplevelse av att använda dem, hur lätta de är att förstå och lära sig, vilken nytta man kan ha av dem och möjligheten att förändra dem till nya krav. Det man nu, med kritiken som utgångspunkt, kan undra är: Hur uppfattar de som utvecklar mjukvaruprodukter IT-kvalitet?

Det är just detta som denna avhandling belyser. Vi har undersökt förståelsen för kvalitet i systemutvecklarens praxis och dragit slutsatser utifrån detta.

I detta arbete vill vi naturligtvis tacka vår handledare, professor Pelle Ehn. Pelle har med sina kunskaper och visioner om design, format vår förståelse för ämnet och därigenom påverkat föreliggande avhandling. Vi vill också tacka Pelle för konstruktiva men ibland "förgörande" kommentarer. Andra som vi också riktar ett varmt tack till är professor Gunhild Agnér-Sigbo, professor Agneta Olerup och doktoranden Jonas Hedman vid institutionen för In-

formatik, Lunds Universitet. De har med tålamod läst avhandlingen och kommit med värdefulla omdömen. Vi vill också tacka Jon Steen för all korrekturläsning. Jon har sett till att de värsta "språkgrödorna" har försvunnit i den svenska texten. Skuggan av alla språkliga fel faller naturligtvis på oss.

Arbetet har fördelats enligt följande:

Prologen, kapitel fyra och fem samt den populärvetenskapliga sammanfattningen har Odd skrivit.

Kapitel ett, två och den engelska sammanfattningen har Theis skrivit.

Kapitel tre, sex, sju, åtta och de avslutande kommentarerna har skrivits av Odd och Theis tillsammans.

Personligen skulle jag, Theis, vilja tacka min familj – Siv, Oskar och Sofia – för att de har stått ut med deras trötta och förvirrade sambo och pappa (jag kan inte lova att jag blir bättre).

Jag, Odd, riktar mitt tack till nära och kära som stått ut med en litan i flera år och ändå varit uppmuntrande.

Till sist vill även vi ge varandra en klapp på axeln för det stöd vi har givit varandra i stunder när motivation och lust inte har varit de rätta. Vår vänskap har varit avgörande i detta arbete.

Lund, den 29/4 – 2002.

Theis Meggerle och Odd Steen





## Innehållsförteckning

|  |           |
|--|-----------|
| <b>Prolog</b> .....  | <b>13</b> |
| <b>1 Systemutvecklare och IT-kvalitet</b> .....  | <b>25</b> |
| 1.1 Hur ser egentligen systemutvecklare på kvalitet? .....   | 26        |
| 1.2 Systemutvecklarens bristande design- och<br>bedömningsförmåga .....  | 28        |
| 1.3 IT-kvalitet och kunnande – en problematisering (eller IT-<br>kvalitet och personerna bakom hantverket) ..... | 31        |
| 1.3.1 Kvalitet som begrepp .....   | 34        |
| 1.3.2 Kvalitet och kunnande .....  | 35        |
| 1.4 Disposition .....  | 38        |
| <b>2 IT-kvalitet och kvalitetsbegrepp</b> .....  | <b>40</b> |
| 2.1 Kvalitetssystem .....  | 41        |
| 2.1.1 Kvalitetsdefinitioner .....  | 44        |
| 2.2 Mjukvarukvalitet – En kort historik .....  | 46        |
| 2.2.1 Kvalitetsbegrepp och mjukvarukvalitet .....  | 52        |
| 2.3 Gränssnittskvalitet .....  | 53        |
| 2.3.1 Kvalitetsbegrepp och gränssnittskvalitet .....   | 57        |
| 2.4 Sammanfattning och reflektion .....  | 61        |
| <b>3 Kunskap och kunnande</b> .....  | <b>65</b> |
| 3.1 Kunskap och kompetens enligt Johannessen .....   | 67        |
| 3.1.1 Språk och begrepp .....  | 70        |
| 3.1.2 Påstående-, färdighets- och förtrogenhetskunskap .....   | 71        |
| 3.1.3 De tysta inslagen .....  | 73        |
| 3.1.4 Exempels makt .....  | 74        |
| 3.1.5 Sammanfattning Johannessen .....   | 75        |
| 3.2 Kunskap och kompetens enligt Rolf .....  | 76        |
| 3.2.1 Tyst kunskap enligt Polanyi .....  | 77        |
| 3.2.2 Personlig kunskap .....  | 78        |
| 3.2.3 Praktisk kunskap .....   | 79        |
| 3.2.4 Skicklighet, know-how och kompetens .....  | 81        |
| 3.2.5 Reflektion och kommunikation .....   | 84        |

|          |   |            |
|----------|---|------------|
| 3.2.6    | Sammanfattning Rolf.....  | 85         |
| 3.3      | Kunskap och kunnande enligt Johannessen och Rolf.....                         | 86         |
| <b>4</b> | <b>Undersökning av systemutvecklares syn på IT-kvalitet.....</b>              | <b>91</b>  |
| 4.1      | Utgångspunkter för undersökningen.....  | 92         |
| 4.1.1    | Vad är systemutvecklare i vår undersökning? .....                             | 92         |
| 4.1.2    | Systemutvecklaren ur två perspektiv .....                                     | 94         |
| 4.2      | Utgångspunkter för metodval.....  | 96         |
| 4.2.1    | Observation som undersökningsmetod .....                                      | 98         |
| 4.2.2    | Intervjuer som undersökningsmetod.....  | 100        |
| 4.3      | Val av metod och undersökningens genomförande .....                           | 103        |
| 4.3.1    | Problematiskt att finna företag .....   | 103        |
| 4.3.2    | Datainsamling.....  | 105        |
| 4.3.3    | Analys av intervjuerna.....   | 110        |
| 4.3.4    | Kvaliteten på undersökningen.....   | 120        |
| 4.3.5    | Att presentera empirin.....   | 126        |
| <b>5</b> | <b>Systemutvecklares kvalitetssyn – ett samtal<br/>med Eva och Adam .....</b> | <b>133</b> |
| <b>6</b> | <b>IT-kvalitet i praxis – en diskussion .....</b>                             | <b>165</b> |
| 6.1      | Begreppet kvalitet och dess betydelse .....                                   | 166        |
| 6.2      | Hur kvalitet bedöms .....   | 172        |
| 6.3      | Bedömningsförmågans innehåll.....   | 175        |
| 6.4      | Bedömningsförmågans utveckling och grund .....                                | 178        |
| 6.5      | Reflektionens roll i utvecklingen av<br>bedömningsförmågan .....              | 180        |
| 6.6      | Sker en kommunikation kring IT-kvalitet och<br>bedömning?.....                | 185        |
| 6.7      | Sammanfattning .....  | 187        |
| <b>7</b> | <b>De två projekten .....</b>   | <b>190</b> |
| 7.1      | Projektet IT-designkvalitet – paradigmatisksk form och<br>funktion.....       | 190        |

|  |            |
|--|------------|
| 7.2 Den första artefaktmodellen .....                              | 197        |
| 7.2.1 Värderingsperspektiv som kan kopplas till modellen.....      | 198        |
| 7.3 Den andra artefaktmodellen .....                               | 200        |
| 7.4 Stilar och stilteori .....                                     | 203        |
| 7.4.1 Och vad blev det av det... ?.....                            | 208        |
| 7.5 Projektet Kvaliteket.....                                      | 210        |
| 7.5.1 Ett medium för reflektion kring kvalitet i bruk.....         | 211        |
| 7.5.2 Ett exempel på en representation i Kvaliteket.....           | 213        |
| 7.5.3 Några reflektioner om varför det inte lyckades.....          | 215        |
| <b>8   Projektet och IT-kvalitet i praxis – en reflektion.....</b> | <b>218</b> |
| 8.1 Att definiera kvalitet.....                                    | 218        |
| 8.1.1 Att använda exempel.....                                     | 221        |
| 8.2 IT-kvalitet i praxis .....                                     | 224        |
| <b>9   Avslutande kommentarer .....</b>                            | <b>229</b> |
| <b>Populärvetenskaplig sammanfattning.....</b>                     | <b>234</b> |
| <b>English summary.....</b>  | <b>245</b> |
| <b>bilaga 1.....</b>   | <b>249</b> |
| <b>bilaga 2.....</b>   | <b>252</b> |
| <b>Referenser .....</b>  | <b>255</b> |

## Figurförteckning

|  |     |
|--|-----|
| Figur 4.1: Kategorier för aspekten påståendekunskap .....  | 111 |
| Figur 4.2: Kategorier för aspekten färdighetskunskap.....  | 112 |
| Figur 4.3: Kategorier för aspekten förtrogenhetskunskap.....   | 115 |
| Figur 4.4: Tillagda kategorier för övrigt intressant i<br>förhållande till kvalitet.....                     | 116 |
| Figur 4.5: Utdrag från sammanställning under kategorin<br>FÄ-BED-TYK, intervju 7. ....                       | 118 |
| Figur 4.6: Utdrag från meningskoncentrering under<br>kategorin FÄ-BED-TYK, intervju 7.....                   | 119 |
| Figur 4.7: Utdrag från sammanfattning/tolkning, intervju 7. ....   | 120 |
| Figur 7.1: projektets metamodell för artefaktaspekter<br>och kvalitetsperspektiv relaterade till dessa.....  | 199 |
| Figur 7.2: Den andra artefaktmodellen<br>(modifierad efter Ehn et al., 1996a, s. 22).....                    | 202 |
| Figur 7.3: Sammanställning av några stildefinitioner<br>(bearbetad och översatt från Ehn et al., 1997a)..... | 209 |
| Figur 8.1:Fenomenet IT-kvalitet som tre aspekter.....  | 228 |
| <br>   |     |
| Bild 7.1: Startbilden till bidraget om Ikea-MHS.....   | 214 |



## Prolog

Lars: Välkomna Eva och Adam till det här samtalet som skall handla om hur systemutvecklare ser på kvalitet på de produkter som de utvecklar. Jag tänkte börja med frågan – Vad är kvalitet?

Eva: Det var ju ingen lätt fråga! Kvalitet är ju väldigt mycket och kan betyda nästan vad som helst egentligen – det beror så mycket på vad man lägger i begreppet, så på det viset är kvalitet något subjektivt och något som ligger i betraktarens öga.

Adam: För att kunna säga vad kvalitet är måste man också säga ur vems perspektiv kvalitet skall ses. Kunden är en intressent vars perspektiv man kan anta. Man skulle ju också kunna tänka sig att anta användarnas perspektiv, eller utvecklarnas, eller kanske systemförvaltarnas. Man kan nog till och med se olika perspektiv hos utvecklarna – är det en programmerare, en funktionsdesigner eller kanske en databasdesigner?

Lars: Men vad betyder i såfall kvalitet om man utgår från kundens perspektiv?

Eva: Kunder är ju väldigt olika och har olika önskemål beroende på deras verksamhet. Men det är häri det ligger samtidigt – ett system skall vara ett verksamhetsstöd, det skall bidra med något positivt i verksamheten så att den blir effektivare, eller att man kan hantera sina kunder bättre, eller vad det nu kan vara. Det skall kort sagt ha den funktionalitet som efterfrågas.

Adam: Men ett program eller system måste vara användbart och fungera – om det hänger sig hela tiden eller räknar fel, eller om databasen blir inkonsistent, så fungerar inte systemet eller programmet som det skall och då är det inget stöd i verksamheten. Men samtidigt är kvalitet en resursfråga och i det perspektivet får vissa kvaliteter, exempelvis underhållsbarhet, ibland stryka på foten till förmån för att systemet skall vara billigt eller att det skall utvecklas och implementeras snabbt.

Lars: Så underhållsbarhet är en kvalitet som du upplever som viktig men som beställaren inte alltid värderar?

Adam: Ja, ibland så får man kompromissa med den kvaliteten för det kostar ju naturligtvis mer att göra system underhållsbara, även om det i långa loppet kan bli billigare om systemet är underhållsbart.

Lars: Men går det då att definiera kvalitet och säga att kvalitet är X?

Eva: Ja, på ett sätt går det ju genom att använda ISOs definition och säga att kvalitet är att uppfylla kundens uttalade och underförstådda behov.

Adam: Nej, jag tycker nog inte att man kan definiera kvalitet, däremot kan man tala om vilka kvaliteter som man tycker är viktiga – exempelvis underhållsbarhet, modularitet, förståelig dialogstruktur, läsbar och kommenterad kod, dokumenterade program.

Lars: Det verkar då som att man, även om man inte kan definiera kvalitet, kan använda olika begrepp för att karakterisera kvalitet.

Eva: Ja, det är nog så att det är enklare att nämna olika kvalitetsbegrepp av både teknisk och bruksmässig karaktär, än att slå fast en definition av kvalitet.

Lars: Är de begreppen ni använder uttryck för vad som är hög kvalitet på program och system?

Eva: Ja, så är det ju. Ett program av hög kvalitet skall ha flera av de egenskaper som vi har nämnt hittills, där kanske funktionalitet är det viktigaste.

Adam: Jag håller med Eva, men tycker också att hög kvalitet är mer än att uppnå det som kunden har satt upp i kravspecen.



Eva: Jag håller med dig där – hög kvalitet är faktiskt att lämna ett större bidrag än vad kunden egentligen hade förväntat sig. Men samtidigt skall man inte bygga in mer funktionalitet bara för att det går.

Lars: Så hög kvalitet är att man kan ge ett mervärde i förhållande till de önskemål och krav som kunden har, men att man samtidigt inte skall bygga in en massa som kunden eller beställaren egentligen inte har behov av?

Eva: Ja, det skulle jag vilja säga.

Lars: Ok, men låg kvalitet på ett program då, är det motsatsen till det ni precis har sagt?

Adam: Ja, det blir det ju. Ett program av låg kvalitet fungerar inte som det är tänkt att göra, det är inte stabilt och kraschar hela tiden, användaren förstår inte hur det skall användas och det kräver lång inläring, prestanda är dålig så användaren måste vänta länge på att till exempel få fram en kund på sin skärm.

Eva: Och med ett perspektiv som utvecklare så blir ju kvaliteten låg om programmen är illa skrivna och dokumenterade så att de blir svåra att sätta sig in och underhålla och förändra. Om man sedan lyfter blicken till system så är många av dessa kvaliteter desamma naturligtvis, men då tillkommer också att program skall kunna samverka på ett bra sätt i systemet och att system skall kunna samverka sinsemellan på ett smidigt sätt.

Adam: Det speciella med datasystem är ju också att de representerar en given bild vid ett givet tillfälle – de så att säga cementerar verksamheten, men verksamheten förändras ju hela tiden och då är det viktigt att systemen också går att förändra. Ju svårare det är att göra, desto lägre är kvaliteten tycker jag.

Lars: Hur avgör ni i såfall att ni har uppnått den kvalitet som ni skall uppnå? Mäter ni det eller uppskattas det på något annat sätt?

Eva: Mäter kvalitet gör vi ju väldigt lite, för lite egentligen kanske. Men jag tror också att det är oerhört svårt att sätta upp mätbara kriterier för särskilt många kvaliteter och sen också skapa metoder för att mäta dessa. Hur skall man till exempel kunna mäta kundtillfredsställelse eller enkelhet eller lättlärdhet? De här mjukare kvaliteterna tror jag inte att man kan skapa mått för.

Lars: Om man inte kan mäta kvalitet, hur avgör man i såfall kvaliteten? För den måste väl bedömas ändå?

Eva: Det är klart att på något sätt måste kvaliteten bedömas, annars har man ju väldigt svårt att se om man har uppfyllt de kvalitetskrav som man vill uppnå. Det finns flera andra sätt än mätning att bedöma kvaliteten och testning är ju väldigt viktigt och mycket använt. Sedan har man ju också metoder och standarder som stöd för hur program skall utformas, hur gränssnitt skall se, hur dokumentationen skall skrivas och så vidare. Kvalitet är också att följa metoder och standarder, att inte avvika från dem.

Adam: Sedan kan man ju också ha formella kodgenomgångar och andra typer av reviewer, där man jämför med kravspecen och standarder och på så vis bedömer kvaliteten.

Eva: Men samtidigt måste man se metoder och standarder som stöd, inte regelverk som man absolut inte får avvika ifrån.

Lars: Så man kan alltså behöva mer eller mindre bryta mot standarden till exempel, för att uppnå hög kvalitet?

Eva: Ja, fast man måste naturligtvis ha goda skäl till det. Man håller sig till standard, guide lines och metoden så länge det fungerar.

Lars: Men att veta när man skall avvika verkar ju förutsätta en sorts erfarenhetskunskap, en fingertoppskänsla?

Adam: Javisst, och det är ju också därför som Eva säger att metoder och standarder skall ses som stöd i systemutvecklingen. De

här strukturerade hjälpmedlen kanske räcker till 80% av kvaliteten, så den klassiska 80/20-regeln gäller även här.

Eva: I erfarenhetskunskapen ligger också att veta när man skall sluta designa något, för det ger sig ju inte av sig självt utan det kan alltid bli lite bättre eller snyggare.

Lars: Men när ni nu bedömer kvalitet, är det vissa kvaliteter som är viktigare än andra?

Eva: Ja, men det tangerar det som vi var inne på tidigare om perspektiv och nivåer. Men visst, jag tycker ju att funktionalitet är väldigt viktigt och att systemet fungerar som ett stöd i arbetet.

Adam: Jag tycker ju att dokumentation är viktigt, men naturligtvis är det som Eva säger att funktionaliteten, användbarheten och stödet i arbetet är det viktigaste. Underhållsbarheten är naturligtvis också viktig.

Lars: När det gäller vem som bedömer kvalitet, eller kanske snarare avgör kvaliteten, verkar det som om det är kunden eller användaren som gör det?

Eva: Nja, jag tycker att kvaliteten avgörs av de involverade parterna tillsammans – kvalitet är den sammanlagda bilden av allas åsikter.

Lars: Men då är det inte bara kunden som avgör kvaliteten?

Eva: Jo, slutligen är det så, men visst är det så att det handlar om bygga rätt system bra, inte bara bygga rätt system.

Lars: Hur viktigt är då kvalitet?

Adam: Ja, det är ju en märklig fråga att ställa! Kvalitet är ju naturligtvis oerhört viktigt, har vi ingen kvalitet så har vi heller inget system egentligen. Har vi ingen kvalitet så försvinner våra kunder och vi kan inte få nya. Utan kvalitet går ju företaget under.

Eva: Ja, fast kvalitet har inget egenvärde i sig heller, utan det handlar om att hantera förväntningar och det handlar om image delvis. Det är som med bilar – man förväntar sig högre kvalitet på en Mercedes än på en Lada och vill man ha en Lada så accepterar man lägre image och lägre kvalitet.

Lars: Men en Mercedes är ju betydligt dyrare än en Lada, så det är ju en kostnadsfråga också.

Eva: Javisst, men så är det ju med kvalitet på våra produkter också. Mer tid, mer folk och mer pengar resulterar ju alltid i bättre kvalitet, eller i alla fall i möjligheter till bättre kvalitet.

Adam: Ibland är det så bråttom med utvecklingen av systemet så att kunden är beredd att offra vissa kvaliteter för att få en snabb implementation. Då kanske sådana saker som bra och logisk uppbyggnad, underhållsbarhet och kommenterad kod kommer i andra hand.

Lars: Men då kanske kvaliteten blir låg.

Adam: Ja, det kan den ju bli då.

Lars: Men vad händer då?

Eva: Då ringer användare och är missnöjda.

Lars: Är inte det lite tråkigt?

Adam: Jo, det är det ju, men samtidigt kan man lära sig en hel del av det och få erfarenheter av vad som är dåligt och vad som inte fungerar så bra. Man får ju också erfarenhet av vad användare och kunder anser vara kvalitet, och den erfarenheten är väldigt bra att ha när man går vidare i andra projekt.

Lars: Så den negativa feedbacken är bra ur erfarenhetssynpunkt. Den positiva feedbacken, är den lika viktig för erfarenheten?

Eva: Den negativa kritiken hörs mest och ger fler detaljer. Den positiva däremot är mer tyst och sporadisk och mer av karaktä-

ren att det fungerar bra eller att det går mycket snabbare att arbeta med det. Det är inte några detaljer i den kritiken och då är det svårare att lära sig någonting konkret att ha med sig i sin erfarenhet.

Lars: Varför undersöker ni inte själva vad användarna tycker om kvaliteten när det inte förekommer några klagomål?

Adam: Tid helt enkelt, det finns ingen tid till det. När någonting fungerar bra och det inte kommer in så många klagomål så är det dags att ta itu med nästa projekt.

Eva: Det finns heller ingen tid till att reflektera över vad det kan vara som är bra och utan den reflektionen blir det ju svårt att förstå vad som kan vara bra.

Lars: Varför ges inte den tiden?

Adam: Vi är problemfokuserade helt enkelt. Problem och brister måste åtgärdas, och helst snabbt, medan vi är glada om det är tyst, för då måste det fungera och användarna klara sig själva, vilket vi också vill uppnå.

Eva: Tyvärr, är det så som Adam säger, men när den positiva kritiken faktiskt når oss kan den alltså bidra till en bättre bedömningsförmåga och en viktig erfarenhet.

Lars: Nu kom vi naturligt in på bedömningsförmåga. Tycker ni metoder och mätning räcker för att bedöma kvalitet, förutsatt att man då mäter?

Eva: Nej, det tycker inte jag att det gör. Att skapa mätmetoder och mått verkar väldigt svårt, särskilt när det då gäller de mjukare kvaliteterna. Jag har också svårt att se att mätning skulle kunna ge allt det man behöver veta för att avgöra om kvaliteten är tillräckligt bra. Samma sak gäller med metoder och standarder av olika slag – de räcker helt enkelt inte hela vägen. Vissa saker, till exempel gränssnittet, bedöms egentligen helt och hållet subjektivt och då finns det inga metoder som garanterar att man uppnår hög kvalitet.

Adam: Ja, man måste ha en fingertoppskänsla också för att kunna bedöma kvaliteten. Dessutom är ju kvalitet något föränderligt och målen måste flyttas framåt hela tiden – man kan inte vila på lagrarna så att säga.

Lars: Hur uppskattar man i såfall exempelvis underhållsbarheten nu?

Adam: Ja, det sker ju genom erfarenhet, att man har lärt sig att vissa sätt att göra saker och ting på är sämre än andra när det gäller underhållsbarheten.

Lars: Å andra sidan finns mått och mätmetoder inom exempelvis Software Engineering för att till exempel mäta komplexiteten. Hur kommer det sig att ni inte använder dem?

Adam: Några sådana känner inte jag till eller har hört talas om.

Lars: Ok, men ni tror i alla fall inte att metoder och så vidare räcker för att bedöma kvalitet. Kan man säga då att det också krävs en tyst, erfarenhetsbaserad kunskap?

Eva: Ja, definitivt och det har ju också framkommit i diskussionen flera gånger att erfarenhetskunskapen är väldigt viktig och utan den skulle det för övrigt bli svårt att utnyttja metoder på ett bra och vettigt sätt.

Lars: Vad består då den tysta kunskapen av?

Adam: Det handlar ju om erfarenheter man samlar på sig under åren i olika projekt – saker man har gjort som fungerat bra eller mindre bra, sådant som man har hört av andra med större erfarenhet. Man utvecklar med tiden en känsla för kvalitet och den är väl en del av den tysta kunskapen.

Eva: Ja, för kvalitetsbedömning är i mycket en personlig sak – jag har min syn på kvalitet och andra har en annan. Så man har personliga ramar för kvalitetsbedömning som formas av ens erfarenheter totalt sett, alltså inte bara inom yrket, och dessa ramar påverkar mycket hur man bedömer kvalitet.

Lars: Hur utvecklas och formas den här bedömningsförmågan?

Eva: Till viss del har man en grund i utbildning och kurser.

Adam: Fast jag måste säga att jag inte tycker av kvalitet togs upp särskilt bra i den utbildning jag gick, alltså systemvetenskap. Vi hade ingen kurs i kvalitet explicit, utan det smög sig nog mer in i andra kurser och hanterades inte som ett begrepp i sig, vilket jag tycker är synd.

Lars: Så kurser och utbildning ger en grund, men fokuserar sällan på kvalitet som sådant?

Eva: Ja, det kan man säga.

Lars: Erfarenheten är viktig säger ni och då borde väl även utbytet av erfarenheter vara viktigt?

Adam: Ja, naturligtvis är det utbytet väldigt viktigt. Det är ju bland annat genom att ta till sig av mer erfarna som man bygger upp sin egen erfarenhet.

Lars: Hur går det till, hur sker det här utbytet?

Eva: Det sker på många sätt. Till exempel när man utvecklar något specifikt, någon gränssnittskomponent kanske, så kan man ta kontakt med kollegor som har utvecklat något liknande tidigare för att se hur de gjorde och vilka tips de har.

Adam: Det kan också vara att man ser något som någon annan gör, men som man inte tycker är riktigt bra ur underhållssynpunkt till exempel, och då kan man prata om det och ge tips på hur det kan göras bättre. Eller att man själv är osäker och vill ha synpunkter från sina kollegor.

Lars: Det sker så att säga i det löpande arbetet?

Eva: Ja, men det kan också vara mer systematiskt genom att man har olika kompetensgrupper som bevakar olika områden och sedan sprider den kunskapen till andra.

Adam: Fast jag tycker att den systematiken du nämnde är rätt sällsynt och att det oftast handlar om att utbyta erfarenheter inom projektet och då kopplat till en viss produkt, det vill säga den man håller på att utveckla.

Eva: Sedan så har man ofta personer som fungerar som coacher i verksamheten som tar sig an de nyanställda och förmedlar arbetssätt och synsätt, exempelvis vad det innebär att vara en god konsult.

Lars: Så, bedömningsförmågan påverkas mycket av samarbete och utbyte med andra, främst kollegor, men det sker sällan särskilt strukturerat eller systematiskt?

Eva: Till viss del finns en systematik, vilket naturligtvis är olika för olika företag, men det mesta är nog ostrukturerat och osystematiskt ja.

Lars: Spelar andra faktorer in i utvecklandet av bedömningsförmågan, till exempel teorier om kvalitet, teknikutveckling, med mera?

Adam: För de allra flesta tror jag inte att kvalitetsteorier kommer in i bilden – för vissa kanske, men inte för de flesta. Teknikutvecklingen påverkar såtillvida att nya verktyg leder till nya möjligheter som kan vara bättre.

Lars: Men hur påverkar det bedömningsförmågan?

Adam: Nya möjligheter leder till nya lösningar och det i sin tur till nya sätt att bedöma kvalitet på sådana lösningar.

Lars: Jag tänkte också på det ni sade om hur viktigt det är med samarbete och utbyte med andra kollegor för forandet av bedömningsförmågan. Det borde ju innebära att ni talar med varandra om kvalitet och då undrar jag hur det går till.

Eva: Jag tycker väl egentligen inte att vi talar om kvalitet i den bemärkelsen.



Lars: Hur menar du då?

Eva: Jag menar att vi inte talar om kvalitet med det ordet eller om kvalitet som ett begrepp och vad det innebär. När man talar om kvalitet så gör man det i ett projekt i samband med en viss produkt, inte om kvalitet generellt. Det är ju inte så att vi står på fikarasten och pratar om kvalitet med varandra.

Lars: Hur kommer det sig?

Eva: Man gör inte det helt enkelt, men det kan ju hänga ihop med att man inte reflekterar tillräckligt över vad kvalitet är och inte heller har tid till det.

Lars: Håller du med om det Adam?

Adam: Ja, det gör jag nog. Jag tror inte att man är medveten om kvalitet på det viset, utan det kommer in på ett annat sätt. Men jag tror ändå att man reflekterar över kvalitet, men inte så att man sitter och tänker på vad det ordet kan betyda.

Lars: Vad skulle behövas tycker för att få en mer aktiv diskussion?

Eva: Ja, att man hade seminarier och diskussioner och kanske bjöd in gästföreläsare som talar om kvalitet och annat runt omkring det. Att man reflekterar mera naturligtvis.

Lars: Så, man talar inte om kvalitet på det viset riktigt, men om ni skall förmedla en kvalitetsuppfattning till någon annan, vilka ord och begrepp använder ni då?

Adam: Jag tycker inte att vi har speciella ord och begrepp som kan användas för att förmedla en kvalitetsuppfattning.

Lars: Men det tycker jag är lite märkligt eftersom ni använder en del ord för att tala om kvalitet. Det känns lite paradoxalt att det inte skulle finnas ord och begrepp.

Eva: Ja, fast vi tänker nog inte på dem som speciella ord och begrepp för just kvalitet. När vi talar om kvalitet med varandra

så sker det nog i ett mer vardagligt språk. Dessutom tycker jag att många ord och begrepp är för mångtydiga och otydliga för att man skall kunna vara precis i det man säger och veta att den man talar till också uppfattar det man menar.

Lars: Du sade språk där och jag undrar om ni tycker att det finns ett speciellt språk för att förmedla kvalitetsuppfattningar?

Eva: Det hänger väl i och för sig ihop med om vi har begrepp och ord, och jag tycker inte att vi har ett speciellt bra språk, om vi nu har något.

Adam: Jag tycker nog att det finns ett språk och det kan innehålla sådana saker som att om man följt standard, guidelines med mera. Men visst, det är kanske inte så specialiserat som du frågar efter.

Lars: Använder ni exempel som utgångspunkt för att förmedla en kvalitetssyn, exempelvis ett väldigt bra system ni har sett någonstans?

Eva: Till viss del använder vi exempel, men nog inte på något strukturerat och systematiskt sätt. Men om man skall tala om ett bra gränssnitt till exempel, då måste man ju visa det samtidigt.

Lars: Så, att förmedla intryck av ett system kan vara svårt?

Adam: Ja, det kan det, särskilt om man inte kan visa och provköra det.

Lars: Skulle ni vilja ha haft ett bättre språk med begrepp och exempel i såfall?

Eva: Ja, det hade ju varit bra och det hade kunnat skärpa diskussionen om kvalitet och göra det lättare att få fram vad man menar.

Lars: Ja då tackar jag er för att ni har tagit er tid att dela era tankar och erfarenheter kring kvalitet med oss.

# 1 Systemutvecklare och IT-kvalitet

Den här avhandlingen inleddes som synes med en prolog utformad som en diskussion mellan tre personer om olika aspekter på IT-kvalitet. Två av dessa personer, Eva och Adam, är systemutvecklare medan den tredje personen, Lars, är den som har inbjudit Eva och Adam till diskussionen och därmed är en sorts diskussionsledare.

Frågor man som läsare (antar vi) kan ställa sig är: Vad är egentligen meningen med den här diskussionen? Vilka är Eva och Adam? Vem är Lars? Har diskussionen verkligen ägt rum eller är den påhittad?

Svaret på den sista frågan är att diskussionen är påhittad och att den aldrig ägt rum i verkligheten. Den framstår också som väl tillrättalagd för att vara verklig. Dock är det så att prologen är en förkortad version av ett längre fiktivt samtal, vars innehåll och de synpunkter och åsikter som framförs i det, är förankrade i verkliga intervjuer med yrkesverksamma systemutvecklare kring deras uppfattningar om IT-kvalitet. Diskussionen i prologen är alltså det förenkla och förkortade resultatet av en empirisk undersökning som syftat till att belysa just dessa uppfattningar.

Efter genomgång av teori- och metodkapitel återkommer diskussionen som ett fylligare samtal i kapitel 5 "Systemutvecklarens kvalitetssyn – ett samtal med Eva och Adam", där vår förhoppning är att delar och helheter framträder tydligare mot en bakgrund av en djupare hantering av problemområdet. Detta samtal är som nämnts fiktivt men ändå troget det empiriska materialet och den analys av det som vi har genomfört, och därigenom är samtalet också troget de verkliga data som ligger till grund för det (se kapitel 4 "Undersökning av systemutvecklarens syn på IT-kvalitet", där vi också redogör närmare för Eva, Adam och Lars).

Prologen tjänar därmed som en sorts introduktion till avhandlingen och ger en första bild av det som vi har studerat och lägger därigenom en grund för en förståelse för problemområdet.

Den första av de ovan ställda frågorna, vad är egentligen meningen med den här diskussionen, är delvis redan besvarad. Meningen är att det fiktiva samtalet och därmed diskussionen i prologen skall sammanfatta våra analyser av empirin och därigenom frambära en bild av den del av IT-kvalitet som vi har valt att studera och vad detta betyder för systemutvecklare.

Men frågan rymmer också flera andra delfrågor av djupare karaktär som mer har att göra med syftet med det fiktiva samtalet. Varför finns det över huvud taget? Vilka frågor är det tänkt att besvara? Och dessa frågor leder i sin tur till frågor om varför den empiriska undersökningen genomförts och vilka frågeställningar som ligger till grund för den. Vi skall i resterande delen av det här kapitlet reda ut dessa förhållanden och nysta upp problemområdet, så att våra frågeställningar kommer att framstå så klara och tydliga som möjligt och vara kopplade till diskussionen.

## **1.1 Hur ser egentligen systemutvecklare på kvalitet?**

Diskussionen/samtalet utgår från sex teman som i sig rymmer frågan i rubriken ovan. Anledningen att över huvud taget ställa sig frågan "Hur ser egentligen systemutvecklare på kvalitet?" är att kvalitet under de senaste 10-20 åren hamnat i fokus och detta inte minst inom IT-området. Kvalitetssäkring, kvalitetssystem, ISO med mera ges en allt större betydelse. Samtidigt är strävan bakom ansatser, metoder, tekniker och verktyg inom systemutvecklingsområdet, vare sig dessa kommer från Software Engineering eller den skandinaviska skolan, att användningen av dem skall resultera i högre kvalitet både på processen systemutveckling och på produkten som resultatet av denna process. Att intressera sig för IT-kvalitet över huvud taget ligger därmed väl i fas med informatik såväl som i tiden. Kvalitetstänkande och kvalitetssystem är områden som tilldrar sig stort intresse.

Det finns dock ytterligare en väg in i detta för oss som hänger ihop med våra grundläggande studier på institutionen för Informatik vid Lunds Universitet och inte minst med magisterkursen. Vi båda hade redan innan vi började på institutionen, som då hette ADB-informationsbehandling, ett mer utvecklat intresse för programmering och, vad vi skulle vilja kalla det, den mer påtagliga och hantverksmässiga biten av systemutveckling. Vi hade från början också en ambition att studera en kortare tid för att sedan gå ut i näringslivet.

Under studiernas gång vaknade ett intresse för att studera vidare och lära mer. På högre nivåer i utbildningen kom vi i kontakt med kurser som präglades av en nyfikenhet inför ämnets och professionens grundläggande karaktär. Dessa kurser hade en inriktning mot design och designteori, med målet att slå konstruktiva broar mellan dessa områden och vårt eget ämne. Om man väljer att betrakta informatik som ett designämne kommer nya aspekter i fokus, aspekter som behandlats inom andra, äldre och mer utvecklade<sup>1</sup> designämnena, främst designteori och arkitektur, men även inom området industriell design.

Anledningen till att detta verkade så lockande på oss, hänger ihop med att "greppet" med design innebar att stifta bekantskap med områden som inte redan tidigare funnits med i utbildningen och som gav möjlighet att läsa och lära om insikter som inte direkt hänförs till systemutveckling som yrke i en snävare mening. Design som ord ledde våra tankar till ett "friare" skapande av produkter, än vad ordet systemutveckling gjorde. Dessutom fokuserade dessa kurser på designern som professionell praktiker (se ex. Jones, 1981; Schön, 1983; Lawson, 1990), vilket var sällsynt inom vårt eget område. Tankar och undersökningar kring hur en systemutvecklare tänker var inte något som vi stiftat någon större bekantskap med

---

<sup>1</sup> Med utvecklade menar vi här kunskapsområden och ämnen som dels har en längre historia, dels har en vetenskaplig behandling av ämnet som sådant, dess grunder och teorier. Vad vi menar är alltså att det inom ämnet ingår en vetenskaplig reflexion över ämnet, en sorts metavetenskap.

inom vårt ämne, medan det inom designområdet finns en väl utvecklad behandling av designers tänkande.

Sedan dess har design och designtänkande fortsatt att locka oss på olika sätt. Under utbildningen kom också tankarna på att gå ut i förvärvslivet i bakgrunden till förmån för ett intresse att lära mer, vilket ledde oss in på magisterutbildningen, vilken också hade en relativt stark prägel av design och designtänkande. Magisterkursen ledde i sin tur oss vidare in på forskarutbildningen och den här avhandlingen – ”IT-kvalitet i praxis – systemutvecklarens kunskap om och syn på kvalitet”

Vägen dit har emellertid varit ganska krokig och mödosam och har gått via två projekt som vi deltog i de första åren av forskarutbildningen, vilket vi kommer att berätta mer om i kapitel 7 ”De två projekten”. Utan att gå händelserna i förväg skall vi översiktligt ge en bakgrund till hur vårt intresse för IT-kvalitet i praxis uppstod.

## **1.2 Systemutvecklarens bristande design- och bedömningsförmåga**

Den allra första början på vår forskarutbildning är egentligen magisterkursen i informatik, där vi också för första gången kom i kontakt med professor Pelle Ehn och det projekt, ”IT-designkvalitet – paradigmisk form och funktion” (Ehn, 1994), som han hade startat tillsammans med en av våra andra kollegor Micke Svedemar. Pelle engagerade oss som magisterstudenter inom projektet och vi skrev också vår uppsats inom projektets ramar.

Detta projekt hade ett klart uttalat designtänkande genom att vara inspirerat av arkitektur och stilar, och därtill hörande paradigmiska exempel. Tanken var att förbättra systemutvecklarens förmåga att uppnå och bedöma brukskvalitet, vilken ansågs vara låg på de produkter som systemutvecklare framställde. Upphovet till detta intresse var bland annat följande uttalande av Mitchel Kapor, grundaren av Lotus Corporation och en av upphovsmännen bakom Lotus 1-2-3:

Despite the enormous outward success of personal computers, the daily experience of using computers far too often is still fraught with difficulty, pain, and barriers for most people ... The lack of usability of software and the poor design of programs are the secret shame of the industry ... Computing professionals themselves should take responsibility for creating a positive user experience. Perhaps the most important conceptual move to be taken is to recognize the critical role of design, as a counterpart to programming, in the creation of computer artifacts ... By training and inclination, people who develops programs haven't been oriented to design issues. This is not to fault the vital work of programmers. It is simply to say that the perspectives and skills that are critical to good design are typically absent from the development process, or, if present, exists only in an underground fashion (Kapor, 1996, s. 3).

För att beskriva och fånga brukskvalitet utvecklade också projektet en egen modell med rötter i arkitekturteori med syftet att kunna säga något meningsfullt om brukskvaliteten på vad vi kallade IT-artefakter. I förlängningen skulle detta leda till skapandet av en stilteori av förebildliga exempel för inspiration, diskussion och lärande som i sin tur skulle förbättra systemutvecklarens förmåga att såväl uppnå högre kvalitet som att bedöma den. Bristen stod enligt projektet inte systemutvecklingsmetoder i sig, utan i designförmågan hos praktiserande systemutvecklare. Alltså var inte svaret att utforma ännu en modell, utan svaret var att möjliggöra en utveckling av designförmågan.

Vi tog oss an den här uppgiften som en projektgrupp och gruppens första uppgift var att undersöka om den modell som tagits fram inom projektet för att bedöma brukskvalitet var användbar. Jag (Odd) och Theis provade den i en studie av ett säljstödssystem för Volvos återförsäljare och fann att den gick att använda. Ett par andra magisteruppsatser fann den också användbar (Ganjabadi & Henriksson, 1994; Eriksson, 1995). Alltså kunde projektgruppen gå vidare i arbetet att skapa en praktiskt användbar stilteori med utgångspunkt från modellen. Med modellens hjälp skulle gruppen undersöka och beskriva brukskvaliteten hos datasystem som var i

verklig användning i verkliga kontexter och sedermera inlemma dessa beskrivningar i en stilteori. För att göra stilteorin tillgänglig för de tänkta brukarna (sic!) av denna, tänkte vi inom projektet göra en sorts rapportserie på cd-rom med utgåvor av förebildliga exempel, gärna med ”körbara” delar av de olika representerade systemen för att möjliggöra en mer konkret upplevelse.

Något som emellertid snart visade sig vara ett problem, var att projektgruppen inte riktigt visste hur man skapade en stilteori, vilket blev en uppgift för mig och Theis att försöka reda ut. Med vår bristande förförståelse trodde vi att uppgiften inte skulle vara så krävande, vilket vi ganska snart fick erfara att den var. Stilbegreppet uppvisade helt enkelt en komplexitet som vi inte kunde hantera på något bra sätt och vi hittade ingen praktisk stildefinition som vi kunde anpassa.

Svaret på det problemet blev för projektgruppen, att återvända till exemplets makt med föresatsen att mängden exempel skulle leda till att det induktivt, men med stöd från modellen, skapades stilar. Problemet var bara att projektet var alltför litet, lokalt och hade för få deltagare för att detta skulle vara en såväl gynnsam som framkomlig väg. Alltså måste fler engageras i denna viktiga uppgift och den framstormande webben blev den tänkta lösningen. Ett forum på webben som alla kunde besöka och där diskussioner om kvalitet kunde äga rum, blev den internationella plattformen för publicering av exempel, från oss inom projektet och från andra, från forskare och praktiker. Idén till Kvaliteket var född och lanserades. Mottagandet blev varmt, men intresset att bidra var svalt och Kvaliteket blev aldrig den framgång som vi i projektgruppen hade hoppats på och föreställt oss.

Detta skapade problem eftersom Kvaliteket samtidigt var tänkt att tjäna som ett empiriskt verktyg för avhandlingar inom projektet. För min och Theis del ledde detta till en kris som tvingade oss att ta flera steg tillbaka och börja tänka om. I den processen började vi se flera svagheter i och tveksamheter med projektets utgångspunkter och frågeställningar, vilket efter mycket huvudbry ledde fram till



nya, men relaterade, frågeställningar som vi kunde arbeta vidare med.

### **1.3 IT-kvalitet och kunnande – en problematisering (eller IT-kvalitet och personerna bakom hantverket)**

IT-kvalitet framstod för oss fortfarande som ett intressant område, men vi återknöt kontakten med idéer om praktiserande designers, i det här fallet systemutvecklare, och deras tänkande kring sin praxis med fokus på just IT-kvalitet. Som vi skall visa i kapitel 2 "IT-kvalitet och kvalitetsbegrepp" är kvalitet ingalunda något nytt inom IT-området, vare sig när det gäller kvalitetsegenskaper eller sätt att försöka uppnå kvalitet. Åtminstone gäller detta i den teoretiska behandlingen av IT-kvalitet.

Däremot har vi hittat väldigt lite om IT-kvalitet ur ett praxisperspektiv, det vill säga ett perspektiv på systemutveckling som en yrkesverksamhet, och om systemutvecklarens syn på denna praxis. Två undantag från denna "regel" är dels Stoltermans (1991) avhandling om systemutveckling som en designpraxis, där det framkommer att systemutvecklare till stora delar saknar ett språk för att reflektera över IT-kvalitet, och dels Wilson & Hall (1998) som i en pilotstudie av skillnaden mellan kvalitetsuppfattningar mellan systemutvecklare och ledning, visar att en sådan skillnad föreligger och att ansatser till systematiserade kvalitetsförbättringar (kvalitetssystem) därmed misslyckas.

Stolterman intresserar sig i sin avhandling främst för systemutvecklaren som designer och hans/hennes tankeprocesser i förhållande till systemutvecklingsområdets uppgifter och metodanvändning, vilket resulterar i en ansats kallad idealorienterad systemutveckling som närmast är en sorts metodologisk grund för systemutvecklingsprocessen. Däremot finner vi att tänkandet kring kvali-

tet endast behandlas i en liten utsträckning hos Stolterman, även om han också för vissa resonemang kring detta<sup>2</sup>.

Wilson & Hall (1998) går däremot explicit in på systemutvecklarens uppfattningar om IT-kvalitet, sett i relation till ledningens perspektiv som ofta är att försöka standardisera och styra processen för att uppnå konformitet. Artikeln är dock endast en rapport om en pilotstudie och berör således bara ytan på den problematiken.

Frånsett dessa två arbeten har vi inte funnit några ytterligare undersökningar av hur IT-kvalitet hanteras, uppfattas och betraktas av dem som faktiskt utför själva arbetet med att utveckla system. Här finns alltså ett intressant och outforskat område.

För att återknyta till diskussionen och de sex teman som den är uppbyggd kring, är det just denna fråga den belyser – hur ser systemutvecklare på kvalitet? Som skall visas längre fram i avhandlingen är kvalitet dock ett vanskligt och mångtydigt begrepp, varför man inte kan ställa frågor om det hur som helst och därför vilar undersökningen och diskussionen på teman som syftar till att belysa olika sidor av IT-kvalitet (för det kompletta frågeformuläret, se bilaga 2).

Dessa sex teman är:

1. Definition av kvalitet
2. Karakteristika för hög respektive låg kvalitet
3. Graden av kvalitets betydelse
4. Bedömningsförmåga
5. Utvecklandet och forandet av bedömningsförmåga
6. Förmedling av bedömning

Temana behandlar dels kvalitet som en egenskap hos en produkt, i det här fallet datasystem och -program, och som begrepp, dels som ett praktiskt kunnande hos systemutvecklare och hur det kunnande

---

<sup>2</sup> I den redovisade empirin återfinns endast lite som hänför sig till systemutvecklarens kvalitetsuppfattning som sådan. (Stolterman, 1991b)

det utvecklas. Det bakomliggande intresset är alltså kvalitetsegenskaper och praktiskt kunnande.

För att fånga aspekterna av kvalitet som egenskap och begrepp, syftar tema 1, 2 och 6 (som också inbegriper språk och begrepp för kommunikation) främst till att undersöka hur och om systemutvecklare definierar kvalitet och vilka begrepp de använder för att tala om och beskriva kvalitet. Via dessa teman blir det alltså möjligt att få en bild av systemutvecklarens förståelse för kvalitet som begrepp och en inblick i hur kvalitet tolkas inom systemutvecklarens praxis. Vi kommer att längre fram betrakta detta som ett utslag för påståendekunskap om kvalitet, alltså verbaliserad och artikulerad kunskap (begreppet påståendekunskap diskuteras i kapitel 3). Detta kan sedan jämföras med hur kvalitet kommer till uttryck inom den teori vi tar upp.

De övriga fyra temana syftar främst till att fånga aspekter av praktisk, handlingsorienterad kunskap och i fokus är systemutvecklarens syn på vilka färdigheter och vilket kunnande, och därtill hörande erfarenhetskunskap, som behövs i relation till kvalitet, samt hur dessa utvecklas och förbättras. Här är det fråga om vad vi längre fram kommer att diskutera som färdighets- och förtrogenhetskunskap och personlig, praktisk kunskap och kompetens (se kapitel 3 där dessa begrepp diskuteras). Meningen är att försöka tränga djupare in i systemutvecklarens kunnande och syn på kvalitet genom att undersöka hur kvalitet hanteras, bedöms och kommuniceras inom systemutveckling, samt att försöka förstå vilken roll färdighet och förtrogenhet spelar för kvalitetsförståelse och bedömningsförmåga.

I prologen handlar de två första sidorna främst om möjligheten att definiera kvalitet och om olika begrepp för att beskriva kvalitetsegenskaper. De teman som berörs här är alltså framför allt tema 1, 2 och 6, det vill säga de teman eller de delar av teman som handlar om kvalitet som ett begrepp och som egenskaper.

Resterande delen av prologen handlar främst om systemutvecklarens kunnande, hur den utvecklas och påverkas. Här är delar av alla

teman utom tema 1 alltså grunden för diskussionen och i fokus är systemutvecklarnas personliga förmåga och kunskap i förhållande till IT-kvalitet, snarare än IT-kvalitet som en egenskap hos en produkt.

Detta leder i sin tur till att det diskussionen har behandlat dels är kvalitet som egenskap hos vissa IT-produkter, dels är kvalitet som ett kunnande hos praktiserande systemutvecklare, vilka alltså är våra frågeställningar i den här avhandlingen. Vi fortsätter därför med att utveckla den första frågeställningen – kvalitet som begrepp – för att sedan gå vidare med den andra – kvalitet och kunnande.

### *1.3.1 Kvalitet som begrepp*

Slår man upp begreppet kvalitet får man vanligen svaret att det är en egenskap. En egenskap antingen hos en artefakt eller ett naturligt objekt. I vårt fall är artefakten av informationsteknologisk karaktär och således ligger vår fokusering på begreppet IT-kvalitet.

IT-kvalitet ges också idag en mycket betydande roll. Det finns till och med de som, ur ett näringslivsperspektiv, hävdar att hög kvalitet är en nödvändighet för överlevnad på dagens konkurrensutsatta marknad. Med andra ord skulle det kunna uttryckas så här: 'är kvalitet inte en del av det man levererar går man en säker död till mötes'<sup>3</sup>. Trots att begreppet är förknippat med en slags företagsmässig överlevnadsstatus framstår ändå kvalitet som ett vanskligt och svårhanterligt begrepp (Dahlbom & Mathiassen, 1993; Kan, 1995; Eriksson & Törn, 1997).

Ett belysande exempel på denna vansklighet ger Garvin (1988) där olika definitioner av kvalitet grupperas om sex kategorier där respektive kategori exemplifieras med vardera två definitioner. Totalt blir det en fråga om tolv mer eller mindre olika definitioner av kva-

---

<sup>3</sup> Vilket borde vara en självklarhet. Men ser man till de produkter vi dagligen kommer i kontakt med kan man ställa sig något frågande. Boken "Design of everyday things" av Norman (1990) är en riklig illustration på att de produkter vi till vardags använder inte är av bästa brukskvalitet.

litet. Därtill görs försök att kombinera olika definitioner av kvalitet (Kan, 1995; Adelakun, 1998)<sup>4</sup>. Detta visar på att det råder en osäkerhet om kvalitetsbegreppets innebörd, samtidigt som det pekar på begreppets rikhet och komplexitet.

Trots denna osäkerhet finns det en rik flora av kvalitetsbegrepp eller kvalitetsrelaterade egenskapsbegrepp. De flesta återfinns inom Software Engineering och MDI<sup>5</sup>-området där respektive område i stor utsträckning fokuserar på mjukvara respektive gränssnitt.

Denna mångfald av begrepp som har uppstått i försöken att fånga betydelsen i IT-kvalitet och den något paradoxala osäkerheten i möjligheten att definiera begreppet, samtidigt som det framstår som så betydelsefullt, gör det till ett intressant område att studera.

### *1.3.2 Kvalitet och kunnande*

Som vi har diskuterat tidigare är en praxissyn på systemutveckling, d.v.s. en syn som utgår från att systemutveckling är en yrkesverksamhet med specifika kunskaper och regler, ett ovanligt synsätt i den forskning vi har stött på. Att anlägga ett praxisperspektiv på IT-kvalitet som tar sin utgångspunkt i systemutvecklarens egna uppfattningar, tolkade av forskare, är därför intressant dels för att det endast har gjorts i liten utsträckning och dels för att det har riktats och fortfarande riktas kritik mot IT-kvaliteten på de produkter som utvecklas.<sup>6</sup> En stor mängd definitioner och begrepp har tydligen inte resulterat i att kvaliteten har blivit bättre<sup>7</sup>. Därmed bör det praktiska kunnandet kring IT-kvalitet vara betydelsefullt. Begrepp

---

<sup>4</sup> Vi kommer att återkomma till dessa definitioner i kapitel 2.

<sup>5</sup> Människa-Dator Interaktion

<sup>6</sup> Det är viktigt att poängtera att det är systemutvecklarens praxis vi studerar och att vi gör en tolkning av denna. Vi utgör inte själva denna praxis och därför representerar vår studie ett utifrånperspektiv, baserat på analyser av systemutvecklarens inifrånperspektiv.

<sup>7</sup> Även om det kanske har lett till en större medvetenhet om vad som kan vara kvalitet.

och definitioner räcker uppenbarligen inte för hantera kvalitet, utan det behövs ett praktiskt kunnande.

När det gäller den bristande kvaliteten på de produkter som systemutvecklare framställer så har vi tidigare visat att Kapor (ibid.) är starkt kritisk. Även andra framhåller att kvaliteten är inte var den borde vara. Friedrich (1997) till exempel, är också kritisk och menar att det finns en kris både i praktik och teori. Systemutvecklingsmetoder bygger fortfarande på det mekanistiska arvet och "... normativism leaves the designer without any idea of criteria which should be considered to achieve 'good design.'" (ibid., s. 3). Med andra ord finns det brister i teorier om systemutveckling och det saknas goda förebilder i praktiken. Friedrich anser dessa förhållanden som så allvarliga att han benämner situationen som en andra mjukvarukris.

En kritik mer orienterad mot bruket ger Dahlbom & Mathiassen:

There you are putting some finishing touches to your paper before printing it, and since you have the time you decide to take a break and install the new version of the operating system on your personal computer. At the beginning things go smoothly, but all of a sudden you are caught in what seems to be an eternally branching questionnaire, having to answer endless questions about things that seems wholly irrelevant, and that you know nothing about. ...Half an hour later you finally get back to real work, only to find that under the new system you can no longer use your thesaurus. ...Suddenly you make a worse, indeed catastrophic, discovery: You can no longer print out documents from your text processing system. (Dahlbom & Mathiassen, 1993, s. 137)

Anledningarna bakom kritiken kan säkerligen ha många bottenar. Eventuellt beror det på, just som omnämnt tidigare, att det saknas en entydig definition av kvalitet. Eller, det kanske görs för lite mätningar både vad det gäller produkt och process? (Kahn ibid., Burr & Owen, 1996). Dagens kvalitetssystem kanske resulterar i likriktning

och är innovationskvävande? (Brunsson & Jacobsson, 1998). De modeller och metoder som används vid utveckling av system vilar kanske på grundvalar som är mindre lyckade i praktiken? (Ehn 1988, Stolterman 1991). Eller kan det vara bristande designförmåga? (Kapor, 1996).

Vi har valt den koppling som Kapor (ibid.) gör, d.v.s. att sätta designförmåga i relation till IT-kvalitet. Dock inte med utgångspunkten att förmågan skulle vara bristande eller att detta skulle vara anledningen till en berättigad kritik. Snarare har vi valt en mer undersökande utgångspunkt och istället använt kritiken som en utgångspunkt för frågeställningar. Till exempel har vi istället för att kritisera designförmågan ställt frågor om vad som karakteriserar denna förmåga. Hur utvecklas den? Vilka delar består den av? När och hur används den? Hur kommuniceras den?

Detta innebär för oss två saker; dels ett kunskapsperspektiv, dels ett praxisperspektiv. Genom att koppla IT-kvalitet till designförmåga har vi valt ett kunskapsperspektiv på IT-kvalitet. Men till skillnad från att enbart hantera förmågan att designa har vi valt att anlägga ett bredare perspektiv. Summariskt handlar detta om teoretisk kunskap och praktisk erfarenhet såväl som designförmåga (ett praktiskt kunnande). Då dessa begrepp utgör ett teoretiskt fundament i denna avhandling och således förtjänar ett större utrymme än vad som kan ges här, behandlas de i kapitel 3.

Intresset för praxis och den professionella praktikern har, som omnämnts tidigare, rötter i vår bakgrund. I någon mening har detta perspektiv från början varit en självklarhet. Men kunskapsperspektivet för också med sig ett praxisperspektiv. Ett kunskapsperspektiv som inbegriper ett fokus på begrepp som erfarenhet och förmåga, innefattar även ett fokus på personerna i fråga. I denna avhandling är det systemutvecklare.

Drivkraften i denna avhandling är således ett intresse för kvalitet i praxis hos systemutvecklare. Detta yttrar sig som två övergripande intresseområden, nämligen: kvalitet som en egenskap inom systemutveckling, vilket ger en objektsyn, och produktkvalitet som

en bedömning inom samma praxis, vilket ger en innebördssyn. Detta kopplar vi då dels till begreppet kvalitet inom IT-området, vilket inbegriper betydelser, definitioner med mera, dels till kunskap och kunnande i relation till detta. Det finns därför två teoretiska ingångar i detta som är av intresse; dels den teoretiska behandlingen av IT-kvalitet i litteratur och inom olika forskningsområden, dels teorier och tankar om kunskap och kunnande. Med detta som bakgrund kan vi senare precisera frågeställningen ytterligare.

## **1.4 Disposition**

Prologen inleder avhandlingen med en kortfattad och fiktiv diskussion mellan två systemutvecklare och en diskussionsledare kring systemutvecklare och IT-kvalitet och lägger en grund för det problemområde vi har studerat

I kapitel 1 tar vi upp de teman som diskussionen utspinner sig kring och sätter dessa i fokus och som utgångspunkt för ett resonemang om våra frågeställningar.

I kapitel 2 gör vi en i sammanhanget kortfattad genomgång av kvalitetsbegreppet som det gestaltar sig inom viktiga områden som behandlar produktkvalitet ur ett IT-perspektiv. Huvudsakligen rör vi oss inom Software Engineering- och MDI-områdena där man har myntat många kvalitetsbegrepp som relaterar till vårt intresse och våra frågeställningar. Vi visar också att det finns vissa bakomliggande perspektiv för vad som är viktiga kvaliteter och vi återger också vad de olika kvalitetsbegreppen står för.

Kapitel 3 går djupare in på tankar om kunskap och kunnande. Framför allt är det Bertil Rolfs tolkning av Polanyi och Kjell Johannessens tolkning av Wittgenstein som vi utgår ifrån, eftersom båda dessa intresserar sig för praktisk kunskap. Vi har valt att betrakta systemutveckling som en praxis där handlingskunskap och därmed regelföljande, reflektion, erfarenhet och lärande blir intressant. En viss skillnad i syn föreligger mellan polanyitraditionen och



wittgensteintraditionen, även om vi anser att det är betydligt mer som förenar än som skiljer åt.

I kapitel 4 redogör vi för planeringen och genomförandet av den empiriska undersökningen. Vi redovisar också hur analys- och tolkningsprocessen är uppbyggd och vilka olika steg som vi har genomgått för att åstadkomma en förhoppningsvis bra analys och givande tolkning av det empiriska materialet. Vi diskuterar också kvaliteten på undersökningen, samt olika sätt att presentera resultaten, där vi alltså har valt en berättelse i samtalsform mellan två systemutvecklare och en diskussionsledare.

Kapitel 5 är själva samtalet som täcker de teman kring systemutvecklarens kunskap och kunnande om kvalitet och syn på bedömningsförmåga som vi har undersökt.

Kapitel 6 är en diskussion och reflektion kring resultaten från den empiriska undersökningen kopplat till teorierna om IT-kvalitet och den mer praxisinriktade synen, där kunskap, kunnande och kompetens är viktiga inslag. Vi sätter alltså undersökningens resultat i perspektiv av teorin och drar slutsatser utifrån ett mer praxisbetonat synsätt.

I kapitel 7 berättar vi noggrannare historien om de två projekt vi deltog i under de första åren och som utgör utgångspunkten för våra egna frågeställningar. Det är från dessa som intresset för IT-produktkvalitet och bedömningsförmåga kommer. Vi diskuterar också de problem som vi efterhand fick med den modell som projektet utvecklade och vissa antaganden som var gjorda.

I kapitel 8 för vi en diskussion om hur våra slutsatser och tankar från kapitel 6 kan ses i ljuset av de projekt vi tidigare deltagit i. Detta kapitel är också avslutningen på avhandlingen sett ur ett vetenskapligt perspektiv.

Vi avslutar dock själva skriften med våra avslutande kommentarer i kapitel 9, där vi mer fritt spekulerar över våra slutsatser och hur vi kan föreställa oss att de skulle kunna användas för att förbättra systemutvecklarens praxis med avseende på IT-kvalitet.

## 2 IT-kvalitet och kvalitetsbegrepp

Vi skall i detta kapitel förmedla en förståelse för kvalitet. Kvalitet översätts ofta med ord som beskaffenhet, egenskap och värde. Beskaffenhet kan i sin tur betyda hur något är konstruerat, uppbyggt, vilka drag något har eller hur något är (upplevs). Egenskap framstår som ganska likt beskaffenhet men innebär kanske mer ett fokus på en del av något, som är utmärkande, särskild eller kännetecknande. Värde betecknar däremot ofta något som vackert eller fult, positivt eller negativt samt bra eller dåligt. I grova drag skulle det därför kunna påstås att kvalitet består av en bedömning av antingen en del av något eller en helhet. För att fortsätta med (den elementära) begreppsanalysen framstår det som att en bedömning alltid görs av någon eller något (en måttstock).

Kvalitet blir därigenom ett svårt begrepp. Beroende på vilken detaljgrad (del av något eller helhet) som studeras blir egenskaperna olika. Därför vilar det inte på en idé utan är multidimensionellt. Vidare görs inte bedömningar utan anledning. Antingen reagerar någon känslomässigt på något eller så finns det ett bakomliggande intresse av att bedöma. Bedömningen är därför beroende av detta intresse och därtill avgör intresset vad som bedöms. Följaktligen pratas det därför på olika sätt om kvalitet. Slutligen, framstår också omfånget av begreppet kvalitet som brett. För vad handlar inte om kvalitet?

Att skriva om kvalitet blir därför inte helt oproblematiskt. Begreppet måste avgränsas. En sådan avgränsning är redan gjord. Vi har tidigare avgränsat avhandlingen till att endast beröra produktkvalitet på IT-artefakter. Kvalitet på IT-artefakter eller IT-kvalitet innebär dock också ett stort omfång. För att göra det någorlunda hanterligt har vi valt att sammanställa kvalitetsbegreppet inom de områden där vi har funnit att det finns ett utbrett och medvetet tänkande kring kvalitet. Detta har inneburit att behandlingen av IT-kvalitet kommer att beröra mjukvarukvalitet och gränssnittskvalitet.

I föreliggande beskrivning görs dock ett undantag för kvalitetssystem och kvalitetsdefinitioner, med skälen att det skapar en större förståelse för begreppet som sådant och att det överlappar senare delar i beskrivningen av mjukvaru- och gränssnittskvalitet.

Vi går sedan över specifikt till IT-kvalitet. Respektive perspektiv, mjukvaru- och gränssnittskvalitet, börjar med en kort historik över kvalitetsbegreppet. Därefter följer en mer specifik representation av olika kvalitetsbegrepp inom områdena. Vi avslutar kapitlet med en sammanfattning och reflektion. Utan att gå händelserna i förväg kan det redan här vara värt att påpeka att till skillnad från inledningen ovan, där bedömningar anses göras av någon verkar den som bedömer vara bortglömd i den teoretiska behandlingen av IT-kvalitet.

## **2.1 Kvalitetssystem**

Kvalitetstänkande är en mångårig tradition som går långt tillbaka i historien. I en hantverkares professionalism ingick det att kontrollera varans kvalitet innan den såldes eller byttes bort. I dagens terminologi kallas detta för kvalitetsgranskning.

Det moderna kvalitetstänkandet har sina rötter inom tillverkningsindustrin i USA. När industrialiseringen tog vid och massproduktion var ett faktum blev det nödvändigt med mer formell granskning. Produkter började bestå av många detaljer som inte längre kunde sättas ihop direkt efter varandra; det krävdes mycket kompetent arbetskraft (till skillnad från tidigare då det krävdes en person) och utrustningen var mycket dyr. Denna press gav upphov till speciell tillverkningsutrustning som kontinuerligt var i behov av kalibrering. Kvalitet handlade under denna tid om att skapa en överensstämmelse med uppställda krav och detta säkerställdes genom manuell mätning, sortering och gradering.

Efterhand upptäckte man emellertid att detta inte var tillräckligt. Inte i meningen att det var förkastligt, utan snarare att man upptäckte att delar som producerades, oavsett tillvägagångssätt, alltid

skulle uppvisa större eller mindre variation<sup>8</sup>. Lösningen på problemet var statistiska metoder. Till en början syftade dessa metoder till att förutsäga sannolikheter för att ett visst objekt skulle ligga inom vissa gränser<sup>9</sup> (ibid.).

En bra bit efter andra världskriget började dock den framgångsrika amerikanska industrin få konkurrens, framförallt från Japan. Detta skapade nya krav. Den tidigare tillverkningsorienterade preventiva metoden var inte tillräcklig.

Även om ”nöjda kunder” var viktigt tidigare blev det än viktigare nu. Kundens behov, krav och förväntningar kom nu mer i fokus.

Samtidigt förändrades synen på kvalitet från den begränsade tillverkningsorienterade till att vara en ledningsfråga. Kvalitet skulle börja högst upp (d.v.s. i ledningen) och genomsyra hela organisationen. Kvalitet var allas arbete och skulle byggas in i både processer och i personalens medvetande – genom designprocessen, produktionskedjan, till marknaden. Detta ligger till grund för det som idag kallas ”Total Quality Control” och ”Total Quality Management” (Crosby, 1979).

Dessa tankar utgör utgångspunkten för kvalitetssystem inom vårt eget område. Precis som inom tillverkningsindustrin såg man nödvändigheten av att införa kvalitetssystem i processen att utveckla mjukvara.

Kvalitetssystem utgörs vanligen av vad som kallas kvalitetssäkring och kvalitetskontroll<sup>10</sup>. Kvalitetssäkring handlar om att definiera hur en organisation skall uppnå hög kvalitet. Denna definitionsakt innebär att definiera och välja standard. Dessa standarder tillämpas

---

<sup>8</sup> Beroende på att råmaterial, kompetens och utrustning alltid varierar över tiden.

<sup>9</sup> Till exempel genomsnittslängden eller diametern på en viss detalj.

<sup>10</sup> Hårdvaruindustrin använde tidigt olika kvalitetssystem. Då vårt eget område är en avknoppning av numerisk analys som låg nära datavetenskap (”computer science”) framstår det som rimligt att anta att inspiration även från detta området hämtades och inte enbart från området TQM.

sedan på utvecklingsprocessen. En internationell standard som vanligen förespråkas är ISO-9000 (Oskarsson & von Schantz, 1987; Sanders & Curran, 1994; Ince, 1995; Sommerville, 1996). ISO-9000 innefattar ett antal generella standarder där ISO-9001 berör design, utveckling och underhåll av produkter. ISO-9003 är en tolkning av ISO-9000 för utveckling av mjukvara. Därtill finns det även standarder för mjukvara och dokumentation (Sommerville, 1996).

Kvalitetskontroll handlar om att övervaka utvecklingsprocessen för att säkerställa att standarder följs. Det kanske vanligaste kontroll-sättet är granskningar. En granskning kan ha både formen av att granska produkten och av att granska processen. Den kan till exempel syfta till att upptäcka fel i utformningen (designen), att ta reda på hur långt processen är framskriden eller att finna tekniska avigheter i produkten.

På senare tid har det också ägnats en del intresse åt processförbättring (SEI – ”software process improvements”). Utgångspunkten är att det inte spelar någon roll hur väl kvalitetssäkrad processen är om processen i sig är felaktig för den produkt som utvecklas. Detta innebär att det på ett bättre sätt är nödvändigt att förstå existerande processer och förändra dessa efter kvalitets- och kostnadsmål (Sommerville, 1996). En av förgrundsfigurerna till dessa tankar är Humphrey (1989). För att skapa en förståelse för processers tillstånd argumenterar Humphrey för nödvändigheten av metrik<sup>11</sup>.

Sedan början av 90-talet har många också sökt sig till CMM för vägledning och struktur för programvarukvalitet. Med inspiration av Humphreys arbete skapades ”Capability Maturity Model” i slutet av 80-talet av ”Software Engineering Institute” på uppdrag av DoD (”Department of Defense”). Utgångspunkten för uppdraget var att DoD behövde en modell för att bättre bedöma leverantörer. Man upptäckte emellertid snabbt att CMM var användbar både

---

<sup>11</sup> Ett mycket bra exempel på användningen av metrik i processen är Kan (1995) och Burr & Owen (1996).

som bedömningsmodell och som en modell för att förbättra utvecklingsprocesser (Paulk, 1994).

CMM graderar en utvecklingsverksamhet från 1 till 5. Den första nivån, kallas för "Initial". På denna nivå kan alla utvecklingsverksamheter "kvala in" och karakteriseras av att vara kaotisk. Verksamheter på denna nivå lovar ofta för mycket, mer än vad de klarar av, och överger ofta projektplanen till förmån för programmering. På nivå 3, "the defined level", har utvecklingsverksamheten lyckats skapa en standardiserad process som är upprepningsbar. På nivå 5, "the optimizing level" kan verksamheten fokusera på kontinuerlig förbättring. Utvecklingsverksamheten har nu instrument för att identifiera svagheter, styrkor och har förmåga att agera proaktivt (ibid.)

Kvalitetssystem och kvalitetsgranskning tillämpas idag nog av de flesta större företagen. De större företagen kräver också ofta att deras underleverantörer använder och tillämpar kvalitetssystem. Därför breder användningen ut sig. Men något som är lite lustigt, är att det finns en mängd olika kvalitetsdefinitioner. Man skulle kunna tänka sig, att alla som arbetar med kvalitetssystem är överens om vad kvalitet är. För detta borde väl vara en förutsättning. Så verkar dock inte fallet vara. Vi skall i nästa del kapitel beskriva ett antal exempel på mer eller mindre olika definitioner av kvalitet.

### *2.1.1 Kvalitetsdefinitioner*

Garvin (ibid.) har kategoriserat definitioner utifrån vilket sätt som man har närmat sig begreppet. Dessa är det transcendentala, det produktorienterade, det användarbaserade, det tillverkningsorienterade och det värdebaserade. Vi skall här ta upp en definition från varje kategori.

Utifrån det transcendentala synsättet är kvalitet enligt Garvin (ibid.) en sorts naturlig excellens. Den kan universellt kännas igen och är av odiskutabelt hög kvalitet – den är något utöver det vanliga. Den är tidlös och innehar egenskaper som kan sägas gå utöver smak och

trend. Kvalitet är något vi känner igen, men som vi inte kan definiera:

Quality is neither mind nor matter, but a third entity independent of the two... even though quality cannot be defined, you know what it is. (Pirsig, 1974, ss. 185-213).

Till skillnad från den transcendentala ser den produktorienterade synen kvalitet som en precis och mätbar egenskap. Skillnader i kvalitet handlar alltså om kvantiteten av en viss egenskap eller ingrediens. En korv som innehåller mycket rent kött är alltså av hög kvalitet. Därmed kan olika produkter rangordnas efter hur mycket av en viss egenskap den innehar. Ett exempel på en sådan definition är följande:

Differences in quality amount to differences in the quantity of some desired ingredient or attribute. (Abbott, 1955, enl. Garvin, 1988).

Användarbaserade definitioner tar sin utgångspunkt i en köparens eller användares krav och behov. Kvaliteten ligger i betraktarens ögon och de produkter som bäst tillfredsställer en kund är de som anses inneha den högsta kvaliteten. Ett exempel som kanske många känner igen är Jurons definition: "Quality is fitness for use." (Juran, 1974, s. 2-2, enl. Garvin, 1988).

Medan användarbaserade definitioner tar sin utgångspunkt i en köparens eller användares preferenser tar de tillverkningsbaserade sin utgångspunkt i själva tillverkningsprocessen och i en kravspecifikation (som förvisso kan innehålla en köparens preferenser). Så fort en kravspecifikation är fastlagd ses varje skillnad mot denna som en kvalitetssänkning. Hög kvalitet är ekvivalent med att uppnå de uppställda kraven och göra det rätt första gången. Ett exempel på en tillverkningsorienterad definition är "conformance to requirements" (Crosby, 1979, s. 15).

Till sist, värdebaserade definitioner tar ovanstående idé ett steg längre genom att tillföra kund, kostnad och pris. Hög kvalitet är alltså en överensstämmelse med kraven till rätt pris eller kostnad.

Ett exempel på detta är Feigenbaums definition av kvalitet: "Quality means best for certain customer conditions. These conditions are (a) the actual use and (b) the selling price of the product." (1961, s. 1).

## **2.2 Mjukvarukvalitet – En kort historik**

I begynnelsen var datorer byggda av, förvaltade av och ämnade för ingenjörer med ingenjörproblem, i huvudsak militära sådana. Män i vita rockar använde de första datorerna för ballistiska beräkningar åt den amerikanska försvarsindustrin under andra världskriget (Langefors, 1979). Syftet var avgränsat och jämfört med idag var teknologin ganska osofistikerad. Huvudsyftet var att med små resurser få denna nya teknologi att fungera, vilket inte alltid var så lätt eftersom elektronrören så ofta gick sönder eller behövde bytas ut på grund av kortslutning.

Med undantag från de problem som fanns med hårdvaran, hade man också problem med mjukvaran. Med en brist på programmeringsspråk och ett konstant behov att optimera kod, blev programmering en esoterisk konst för ett fåtal människor med matematisk bakgrund (Dijkstra, 1979c, 1979b). Förutom de mekaniska problem man hade var kvalitet under denna tid lika med effektivitet.

Användningen av datorteknologi för administrativa uppgifter började så smått under 50-talet i England. Nyttjandet av datorer flyttades från laboratoriet till nya expanderande områden av samhället. Under dessa första decennier av datorisering och programmering var effektivt utnyttjande av primärminne och processortid i centrum. Som Dijkstra uttrycker det: "... the available equipment was a painfully pinching shoe..." (ibid., s. 115). Processortid var dyr, primärminnet var litet och detta drev fram "smarta" och effektiva lösningar (Dijkstra, 1979b). Det primära kvalitetsmåttet var effektivitet som uppskattades i termer av minnes- och processoranvändning (Budgen, 1994).



Under 60- och 70-talen växte antalet datorsystem rejält. Genombrott inom den digitala industrin gjorde datorer billigare och mer kraftfulla. Användningen av datorer spred sig nu än mer i samhället, främst bland företag. Multiprogrammering samt multianvändarsystem introducerades och nya begrepp som till exempel mänskliga-datorinteraktion (MDI) myntades. Realtidssystem kunde samla, analysera och omvandla data från en mängd olika källor, såsom "on-line"-databaser (Pressman, 1987).

Allt eftersom teknologin utvecklades, vilket innebar större och mer omfattande datorsystem, tillsammans med ny teknik för att kommunicera mellan maskiner, ökade drastiskt mjukvarans komplexitet. Vidare utvecklades mjukvara för stordatorer och minidatorer i så snabb takt att marknaden fullständigt översvämmades. Organisationer och användare stod i ett överflöd av mjukvara. Alla dessa program, tusentals rader kod, skulle rättas när felaktigheter upptäcktes, modifieras när omgivningen förändrade kraven och anpassas till den teknologiska utvecklingen. Dessutom var programmen fortfarande skrivna med effektivitet som ledstjärna, innehållande svårtolkade smarta lösningar och oöverblickbara strukturer. Begreppet underhåll var ett faktum. De nya kraftfulla och effektiva maskinerna som skulle lösa alla programmeringsproblem och bli en evig lycksalighet för programmerare, ledde istället till en mjukvarukris (Dijkstra, 1979b; Pressman, 1987). På en NATO-konferens 1967 med syftet att diskutera krisen föreslogs Software Engineering som en ny disciplin (Shapiro, 1997).

Ett av de första stegen från det odisciplinerade programmeringssättet var det som senare kom att kallas strukturerad programmering. Äran för starten (mitten på 60-talet) tillskrivs ofta Dijkstra, även om han själv har en mycket ödmjuk hållning till detta (Yourdon, 1979).

Dijkstra (1979a, 1979c) anförde redan 1965 argument mot användandet av "goto"-uttryck, idén om "top-bottom"-design och betonade korrekthet samt elegans som viktiga kvalitetskriterier. Med matematiken som analogi försökte han visa den "sanna naturen" kring programs kvalitet. Det första som Dijkstra diskuterar är kvaliteten på programs resultat, d.v.s. korrekthet. Varför litar vi på re-

sultatet och om vi gör det, hur kan vi mäta för att öka vår förtroende för att det resultat som produceras är det som var tänkt, är två frågor som han inledningsvis ställer. Dijkstra kommer här fram till att matematiken inte är till någon större hjälp eftersom ett programs korrekthet inte går att bevisa, utan det bästa som kan göras är att säga: ”Jag har inte upptäckt några misstag.” (Dijkstra, 1979a s. 5, vår översättning)<sup>12</sup>.

Han nedslås dock inte av detta utan går vidare till programs struktur. Inom matematiken har det sedan antikens dagar varit känt att komplexitet kan hanteras genom att bryta ned problemet i delar. Enligt Dijkstra är likheten mellan beviskonstruktion och programkonstruktion slående. I båda fallen är utgångspunkten och målen givna och metoden för att angripa komplexiteten är densamma. Dijkstra menar därför att det inte är orimligt, utan till och med lämpligt att programmeraren kan anamma samma tillvägagångssätt, med argumentet ”...I have only a very small head and must live with it. I, therefore, see the dissection technique as one of the rather basic patterns of human understanding...” (ibid. s. 6). Dessutom ökar det programmets läsbarhet, möjligheten till förståelse och granskning. Han ser därför också uttrycket ”goto” som en styggelse eftersom hopp bakåt och framåt i ett program inte kan anses som väl strukturerat och definitivt inte som läsvänligt.

Slutligen hävdar han också att, även om ett programs korrekthet inte går att bevisa är förståelse för korrektheten beroende på programmets klarhet och i vilken grad programmet reflekterar den utförda processens struktur.

Även om det tog ganska lång tid innan angreppssättet strukturerad programmering blev accepterat i näringslivet, har det förändrat kvalitetsbegreppet från att enbart innebära effektivitet till att även innebära läsbarhet, lätthet till förståelse, elegans och korrekthet.

---

<sup>12</sup> Uppfattningen att ett programs korrekthet inte matematiskt går att bevisa utsattes för kraftig kritik. I en senare artikel medger Dijkstra (1979c) att korrektheten går att bevisa men enbart på programtext eftersom möjligheten till olika typer av indata är så stor att en demonstration av ett givet programs korrekthet skulle bli alldeles för stort.

Dijkstras artikel skapade en ganska stor debatt kring programmering. Näringslivet tyckte till en början att det var trams, andra tyckte att det fanns undantag för uttrycket ”goto” (t.ex. Knuth, 1979). Men det viktigaste med artikeln är i vår mening, att den skapade ett nytt tänkesätt kring programmering och påverkade kvalitetsbegreppet, vilket i sin tur genererade nya idéer.

En sådan idé eller ett sådant tänkesätt är strukturerad design, eller mer specifikt, modultänkande. Begreppet strukturerad design myntades av Stevens et al. (1979), men grundtanken kan sägas tillhöra Parnas (1979).

Parnas (ibid.) ifrågasatte hela dunkelheten i modularisering genom att ställa frågor som: Vad är en modul? Vad skiljer en dålig modul ifrån en bra? Hur gör vi när vi modulariserar ett program? (Yourdon, 1979). Genom sitt berömda ”KWICK”-exempel visade han hur en modularisering är bättre än en annan<sup>13</sup>. Därigenom visade han också principer för bättre underhållsbarhet, återanvändbarhet och det som idag kallas för ”information hiding”<sup>14</sup>. Parnas (1979) argumenterade för att informationsmängden mellan olika moduler skulle vara så liten som möjligt. Poängen är att en viss moduls interna uppbyggnad och innehåll skall vara irrelevant för andra anropande moduler.

Erfarenheten av dessa (ibland elementära) begynnande idéer har lagt grunden till det som idag inbegrips i begreppet mjukvarukvalitet. Detta har efterhand givit upphov till en förfining av idéerna samt etablerandet av andra områden och nya kvalitetsbegrepp. En sådan förfining är till exempel underhållsbarhet där sammanhållning och koppling är viktiga begrepp. Vi återkommer till detta senare.

---

<sup>13</sup> Det som är slående är att man med utgångspunkt i den bättre lösningen undrar hur man överhuvudtaget har kunna tänka sig den sämre lösningen.

<sup>14</sup> Denna enkla historik fortsätter sedan med skapande av tekniker för att förbättra processen, till exempel strukturerad analys (DeMarco, 1979). Detta ligger dock utanför ramen för vår avgränsning och därmed utanför detta kapitel.

Ett område som dessa idéer och mjukvarukrisen har givit upphov till är disciplinen Software Engineering. Begreppet var medvetet valt för att vara provokativt, eftersom det implicerar ett behov för mjukvaruutveckling som är mer baserad på teoretiska och praktiska fundament inom andra etablerade ingenjörscienciner (Shapiro, 1997).

Det huvudsakliga problemet med systemutveckling ansågs vara avsaknaden av lämpliga teorier för att hantera design och utveckling av mjukvara. Problemet med komplexitet var mycket påtagligt i designprocessen och genererade många tankar, såsom modularitet, abstraktion, ledning och metrik.

Även om det går att skönja två läger inom Software Engineering, ett som ansågs mer vetenskapligt och med rötterna inom matematik och ett mer ingenjörsmässigt baserat på praktik, framstår det förstnämnda som dominerande<sup>15</sup>.

Synen på kvalitet kom att bli att den skall vara mätbar med syftet att göra den objektiv, i den meningen att den skall vara oberoende av ett subjekt och således tolkningsfri (Pressman, 1987; Kan, 1995; Sommerville, 1996). Detta bygger på tanken att man knappast vet något om det som inte går att uttrycka i siffror<sup>16</sup>.

When you can measure what you are speaking about and express it in numbers, you know something about it, but when you cannot measure, when you cannot express it in numbers,

---

<sup>15</sup> Trots att vi anser att det förstnämnda läget, d.v.s. det som anses som vetenskapligt, är dominerande, finns det de som argumenterar för att de problem som finns idag med mjukvara beror just på att "software engineering" är ovetenskapligt och därmed otillräcklig i sin tillämpning av matematik. Baber (1997) jämför historiskt disciplinen "software engineering" med andra ingenjörscienciner, såsom elektrisk telegrafi, och kommer fram till att "software engineering" är i behov av och med nödvändighet kommer att utvecklas i en riktning mot innebär regelbunden användning av prediktiva matematiska modeller.

<sup>16</sup> Dock inte utan protester. Många menade att mätning och matematiska formler aldrig kommer att fungera och därför kan de enbart vara ett komplement till mer praktiska metoder (Shapiro, 1997).

your knowledge is of a meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced to the stage of a science. (Lord Kelvin enl. Pressman, 1987).

I strävan efter kvalitet utvecklades således en rad olika metoder för mätning och matematiska modeller. Mills (1975, enl. Shapiro, 1997) presenterade till exempel en matematisk modell för strukturerad programmering med syftet att lyfta bort de instinktiva och intuitiva elementen i utvecklingsprocessen. Komplexitet som var mycket aktuellt under tiden och som ansågs avgörande för underhållsbarhet, skulle till exempel mätas och anges med Halsteads "cyclomatic number"<sup>17</sup> (Halstead, 1977 enl. Shapiro, 1997).

Det visade sig dock att det var problematiskt att mäta egenskaper hos mjukvara. Det andra lägret, d.v.s. de som mer förespråkade ett ingenjörsmässigt angreppssätt än ett matematisk, visade att de små program som utgjorde exempel för mätning var tillrättalagda och på inget sätt liknade verkligheten. Därför var dessa mått i praktiken bara till vissa delar användbara. En motsvarande situation gällde dock även för det andra lägret, alltså det ingenjörsmässigt praktiska. Att i interaktion med mjukvara testa fram till exempel korrektheten, tog oändligt lång tid och var osäkert. Därför kom de olika angreppssätten att komplettera varandra och ge upphov till vad vi idag ser som Software Engineering, även om disciplinen fortfarande har en förkärlek till mätning.

Under hela förloppet har många kvalitetsbegrepp bildats och diskuterats. Från att enbart omfatta effektivitet, består kvalitetsbegreppet idag av en hel flora av begrepp. Vi skall i det följande beskriva några av dessa.

---

<sup>17</sup> "Cyclomatic number" är funktion av hur många vägar en logisk exekvering kan vandra genom ett program.

### 2.2.1 *Kvalitetsbegrepp och mjukvarukvalitet*

Storskaliga industriella och kommersiella system har ofta en lång livslängd, ibland tjugo år eller mer. Många sådana system förblir inte statiska efter den ursprungliga utvecklingen, utan tenderar att utvecklas kontinuerligt tills de avvecklas. Existerande krav förändras och nya läggs till. Möjligheten att förändra systemet är därför en viktig kvalitetsegenskap. När det talas om förändringsbarhet ("modifiability") framstår det som viktigt att tänka på följande:

*Underhållsbarhet* ("maintainability") handlar om att kunna finna och åtgärda ett fel. (Sommerville, 1996).

*Tilläggsbarhet* ("extensibility") handlar om möjligheten att lägga till nya funktioner såväl som att ta bort gamla och lägga till förbättrade. (Bass et al., 1998).

*Omstrukturering* ("restructuring") handlar naturligtvis om omstrukturering av komponenter och deras relationer inom systemet, till exempel att flytta en komponent från ett delsystem till ett annat (ibid.).

*Flyttbarhet* ("portability") handlar om att anpassa systemet till en ny teknisk plattform, till exempel ny hårdvara, användargränssnitt, operativsystem, programmeringsspråk eller kompilatorer.

I relation till förändringsbarhet är också sammanhållning och koppling viktigt. I all korthet kan dessa två kvalitetsbegrepp beskrivas som: "coupling is the degree of interdependence between modules" (Page-Jones, 1988, s. 58) och "cohesion is the measure of the strength of functional relatedness of elements within a module" (ibid., s. 83).

System existerar inte heller ofta i isolering. Mer eller mindre samverkar de med omgivningen ("interoperability"). Denna samverkan kan vara allt från enkel dataöverföring till att mer vara ett delsystem i det "stora" systemet (vi tänker här på kommunikation mellan datoriserade system). Och skall man tro visionärerna kring den nya ekonomin kommer behovet av samverkan att öka.

Med dagens ständigt växande storlek och komplexitet på system, speciellt industriella system, blir *testning* ("testability") allt viktigare, men även svårare och dyrare.

*Tillförlitlighet* ("reliability") handlar om systemets generella duglighet att upprätthålla exekvering och funktionalitet när systemfel inträffar eller vid icke förväntad eller inkorrekt användning (Sommerville, 1996). Två aspekter av tillförlitlighet kan urskiljas:

*Feltolerans* ("fault tolerance") innebär strävan efter att säkerställa beteende vid uppträdandet av ett fel och interna reparation av detta, till exempel att förlora förbindelsen till en komponent i ett distribuerat system och således återknyta förbindelsen. (Bass et al., 1998).

*Robusthet* ("robustness") hanterar säkerheten i ett system, d.v.s. det skyddar systemet från inkorrekt och urartad användning samt behåller systemet i ett definierat tillstånd när ett fel inträffar. Till skillnad från feltolerans innebär inte robusthet möjligheten till fortsatt exekvering utan garanterar enbart att systemet avslutas i ett väldefinierat tillstånd (ibid.).

*Effektivitet* ("performance") slutligen, handlar om effektivt utnyttjande av den tekniska plattformens faciliteter.

## 2.3 Gränssnittskvalitet

Allt eftersom tekniken utvecklades började man under 60-talet också få problem med själva användningen. Från att vara en teknologi gjord för och använd av specialister, forskare och ingenjörer sträckte sig teknologin nu till att stödja andra typer av användare som inte förstod de avancerade maskinerna. Användbarheten var helt enkelt för dålig, d.v.s. datorsystemen var svåra att förstå, använda och interagera med.

Denna situation gav upphov till nya idéer kring hur datorer kunde användas. Banbrytande idéer såsom Ivan Sutherland's "Sketchpad" från 1962 och pionjärbetet på 60-talet av Douglas Engelbart och hans forskningsgrupp på Stanford University som resulterade i

ordbehandling och musen samt Alan Key's utveckling av verktygslådor för programmering (Smalltalk), skapades. Andra viktiga arbeten var användningen av fönstersystem, "WIMP"-gränssnitt, metaforer och direktmanipulering.

I all enkelhet kan man säga att denna utveckling gav ett ytterligare fokus - nämligen på människan. Istället för att anpassa människan till maskinen försökte man anpassa maskinen till människan. Detta skapade forskningsområdet människa-dator interaktion ("Human Computer Interaction"). Vi kommer i fortsättningen använda akronymen MDI för detta område.

Det framstår som möjligt att inom MDI-området identifiera två olika perspektiv på människan. Den tidigare forskningen domineras av en mer mekanistisk syn på människan och är mer riktad mot den biologiska mänskliga organismen som har vissa perceptiva och kognitiva egenskaper samt minnesförmågor. I denna mekanistiska anda var man primärt intresserad av att undersöka människan som en maskin och utveckla objektiva, observerbara, mätbara kvalitets-egenskaper för att få datorn att fungera optimalt med den mänskliga maskinen. Exempel på sådana tekniker som mäter vissa egenskaper är till exempel "Keystroke-Level Model" (Card et al., 1990) och GOMS (Preece et al., 1994). I termer av Löwgren (1995) kan denna utveckling omnämnas som "the general theory perspective".

Allt eftersom området utvecklades började detta behavioristiska och kognitiva angreppssätt att kritiseras. Synen på användare som en passiv och avpersonaliserad individ var alldeles för omänsklig och begränsande. Istället föreställde man sig användare som aktiv och kontrollerande (ibid.) Bannon (1991) föreslog ett fundamentalt skifte i synen, från mänskliga faktorer ("human factors") till mänskliga aktörer ("human actors"). En aktör som har kapaciteten att själv reglera och koordinera arbetet. Detta skapade ett intresse för hur människor agerar i verkliga arbetssituationer och hur ett datorsystem kunde vara ett stöd i dessa.

Detta skifte i fokus är också synligt i Löwgrems (1995) artikel om olika perspektiv på användbarhet. Tidigare nämnda "General the-



ory” har sina rötter i det experimentella paradigmet där laboratoriet fungerade som utgångspunkt för sökandet av objektiva fakta och generaliserbar kunskap om interaktion mellan dator och människa. Allt eftersom området utvecklades blev det uppenbart att många viktiga användbarhetsfrågor enbart är påtagliga i verkliga situationer. ”Usability engineering”<sup>18</sup> angriper dessa brister genom att tillämpa ingenjörsparadigmet, d.v.s. mer tillämpad vetenskap.

Intresset är inom detta område emellertid fortfarande riktat mot mätbara kvaliteter. Men till skillnad från det generella perspektivet är utgångspunkten en specifik användargrupp och specifika arbetsuppgifter, som systemet är tänkt att stödja. Således kan man säga att där det generella perspektivet söker kontextoberoende kvaliteter som ter sig gällande för gemene man, söker ”usability engineering” mätbara kriterier i den aktuella användningssituationen. Det är dock likväl en fråga om generaliserbarhet och objektivitet, men som sagt begränsat till användningssituationen. För att uppnå en sådan generalisering till användarpopulationen, används olika tekniker för att analysera användningskontexten (ibid.) Resultatet används sedan som en användbarhetspecification.

”Usability engineering” verkar dock inte bekymra sig för systems praktiska nytta, d.v.s. om ett systems tjänster är lämpliga eller inte. Om tjänsterna är effektiva och fria från felaktigheter anses systemet vara av god kvalitet. I den meningen, som Löwgren framhäver, separeras användbarhet från nytta.

Även om ”usability engineering” tar sin utgångspunkt i användningssituationen är intresset fortfarande riktat mot generaliserbarhet och objektivitet. I den meningen delar man synen med ”General theory”. Människan är fortfarande ett objekt med maskinliknande egenskaper som kan mätas och jämföras med objektiva kvaliteter. Det är således fortfarande en fråga om mänskliga faktorer och inte mänskliga aktörer.

---

<sup>18</sup> I avsaknad av en bra svensk översättning har vi här använt det engelska begreppet.

Denna separation, mellan nytta och användbarhet, kritiseras också av till exempel Whiteside & Wixon (1987, enl. *ibid.*). För dem är användbarhet primärt inte en fråga om mätbara användbarhetskriterier, utan de betonar snarare den kontinuerliga erfarenhet som folk har av att använda mjukvara. Användbarhet blir därmed subjektiv och relateras till interaktionen mellan användare och dator vid ett speciellt tillfälle i tid och rum. Detta innebär också att nytta och användbarhet fogas samman, för om användare inte uppfattar systemet ifråga som ”nyttigt” och meningsfullt kommer användbarheten i systemet också vara meningslös.

Vägen fram till en sådan uppfattning av användbarhet är enligt Löwgren (*ibid.*) kontextuell design. Med ett flertal tekniker, primärt baserade i etnografiska fälttekniker, skapar sig systemutvecklare en uppfattning om användare och kunder, som förhoppningsvis kommer att reflekteras i systemet. Fastän det finns skillnader i det ideologiska fundamentet är likheterna med participativ design och den skandinaviska skolan inom systemutveckling slående<sup>19</sup>.

Detta anser vi vara ett skifte i fokus från den biologiska människan till den ”mänskliga” människan. Människan betraktas i och med detta inte enbart som ett organiskt maskineri som kan förstås som en informationsprocessor eller ett behavioristiskt system vars gränssnitt till ett mekaniskt maskineri kan optimeras efter objektiva och logiska användbarhetsmått. Människan är ett subjekt som har fantasi, intellektuell kapacitet, känslor etc. och som ständigt utvecklas socialt och på andra sätt i privatlivet och i arbetet.

Men inte heller detta skifte från objektiva till subjektiva kvalitetsegenskaper spelar någon större roll när användbarhet är en del av ett statiskt system som inte förändras. Så ser knappast dagens arbetsituationer ut. Dessa karakteriseras snarare av både snabba och oförutsägbara förändringar och ständig utveckling. En specificering av användbarhet baserad på objektiva och subjektiva kvalitetsmål kommer därför inte att fånga användares framtida behov och ar-

---

<sup>19</sup> För participativ design se till exempel (Friis, 1991; Bødker et al., 1993; Ehn, 1993). För den skandinaviska skolan se Ehn (1988) och Bansler (1989).

betssituationer. Detta, och den mer humanistiska synen, kräver mer dynamiska system och mjukvara som kan anpassas till behov och arbetssituationer som ständigt är under förändring.

Svaret på detta inom MDI-området är, enligt Löwgren (ibid.), flexibilitetsperspektivet, både i process och produkt. En flexibel produkt låter användare förändra och modifiera mjukvaran och dess representation för att klara nya behov och arbetssituationer. En person som har möjligheten att konstant utvecklas i arbetet (vilket verkar vara ett krav idag) bör följaktligen också ha möjligheten att skraddarsy sin mjukvara i enlighet med den personliga utvecklingen. Detta framstår som ett steg fram emot mänskliga aktörer.

Det sista perspektivet som Löwgren (ibid.) tar upp är det sociala perspektivet. Om vi accepterar det humanistiska intresset, som vi gör när subjektiva kvaliteter är en del av synen, måste vi också ta hänsyn till den sociala aspekten av mänskligheten. Inom MDI, speciellt "General theory", har dock fokuseringen varit på den enskilde användaren, isolerad från den sociala kontexten. Arbete sker dock ofta i mindre sociala system, till exempel arbetsgrupper och projektgrupper. Användbarhet borde därför, inom detta perspektiv, också inkludera Kooperation och koordination<sup>20</sup>.

### *2.3.1 Kvalitetsbegrepp och gränssnittskvalitet*

I det föregående avsnittet är det möjligt att urskönja olika synsätt inom MDI-området. Eftersom användbarhet har varit den primära angelägenheten, finns det därför naturligtvis en mångfald av användbarhetsmål och kvalitetsmål inom ramen för detta begrepp. Enligt Dix et al. (1993) finns det tre huvudkategorier för användbarhet. Dessa är lärbarhet, flexibilitet och robusthet. Vi skall i det följande representera dessa.

*Lärbarhet* ("learnability") kan enligt Dix et al. (ibid.) sägas vara:

---

<sup>20</sup> Detta har i sin tur givet upphov till området Computer Supported Cooperative Work (se ex. Bannon, 1991).

1. Systemet måste kunna förstås av en nybörjare.
2. Systemet måste möjliggöra för användare att uppnå en maximal nivå av utförande.

Lärbarhet består enligt (ibid.) av fem kvalitetsbegrepp. Dessa är förutsägbarhet ("predictability"), syntesbarhet ("synthesizability"), förtrogenhet ("familiarity"), generaliserbarhet ("generalizability") och konsistens ("consistency").

En viktig aspekt av lärbarhet är att användare kan förutsäga vad som kommer att hända i systemet när en handling utförs. Kunskapen om interaktionsförloppet måste vara tillräcklig för att bestämma konsekvenserna av en ännu icke utförd handling. Om resultatet skulle vara förvånande eller slumpmässigt är det helt enkelt svårt för användare att lära sig systemet.

Intimt förknippat med förutsägbarhet är syntesbarhet. För att ett system skall kunna bli förutsägbart måste det vara möjligt att bygga upp en "mental modell" av systemets uppförande. För att möjliggöra detta behövs stöd i form av tillgång till föregående utförda operationer och deras relation till det aktuella tillståndet.

En person ackumulerar också, under årens lopp av användning, kunskap och erfarenheter. Dessa erfarenheter eller denna förtrogenhet tar man med sig in i nya situationer. Skiljer sig systemet mycket från denna förtrogenhet blir det svårare för användare att lära sig ett nytt system. Vice versa, blir det lättare för användaren om det finns en överensstämmelse mellan dennes förtrogenhet och det nya systemets handhavande och beteende.

För att hjälpa användare att få en mer komplett och förutsägbar modell är generaliserbarheten viktig. Detta innebär att det bör vara möjligt att återanvända erfarenheten av ett visst resultat i en annan situation, vilket gör inlärningen lättare.

En annan viktig kategori av användbarhetsprinciper är flexibilitet. Denna kategori rör de sätt som användare kan nyttja för att utbyta information med ett datorsystem. Inom ramen för flexibilitet anses

följande begrepp viktiga: dialoginitiativ ("dialogue initiative"), multitrådning ("multithreading"), uppgiftsöverförbarhet ("task migratability"), utbytbarhet ("substitutivity") och anpassningsbarhet ("customizability").

Om interaktionen mellan användare och datorsystem ses som en dialog mellan parter, blir initiativet en intressant aspekt. En användare kan å ena sidan ha initiativet och således vara fri inom ramen för de tjänster som systemet erbjuder att initiera vilken handling som helst, å andra sidan kan initiativet också helt och hållet ligga inom ramen för systemet, d.v.s. att en användares interaktion handlar om att svara på initiativ från systemet. Flexibiliteten blir naturligtvis större ju mer initiativet ligger hos användaren.

En annan aspekt av flexibilitet är multitrådning. En dialog kan ses som en sammansättning av trådar, där varje dialogs tråd är en sammanhängande del av den dialogen och varje tråd, utifrån en given uppgift, är en del av interaktionen mellan användare och systemet. Att tråda dialoger ger möjlighet att stödja utförandet av flera uppgifter samtidigt. Ett system är mer flexibelt om användare fritt kan byta mellan de uppgifter som skall utföras.

För att en uppgift skall bli överflyttningsbar, måste systemet tillåta användaren att flytta kontrollen av utförandeprocessen mellan användare och system. Därtill bör det också vara möjligt att dela en intern uppgift mellan användaren och systemet. Flexibiliteten kan således bli bättre genom att till exempel automatisera en krävande uppgift som stavningskontroll, men samtidigt låta användaren ta kontroll vid behov.

Utbytbarhet innebär att ekvivalenta uppgifter kan växlas emellan. I det ordbehandlingsprogram jag (Theis) använder, kan jag till exempel ange marginalen i dokumentet till tum eller centimeter. Jag kan också välja att byta ut en graf mot en numerisk presentation eller byta grafisk representation på grafen i sig.

Anpassningsbarhet handlar om möjligheten att kunna anpassa gränssnittet. Från ett användarperspektiv kan det till exempel innebära att in och utdata kan anpassas till att bättre passa användarens

behov. Ur ett systemperspektiv kan det till exempel innebära att inställningar i systemet automatiskt anpassas beroende på vilken användare det är.

Skälen till att en användare använder ett datorsystem är ofta att uppnå ett antal mål i arbetssituationen. Robusthet, som också är den sista huvudkategorin, handlar om de egenskaper i interaktionen som gör att dessa mål nås på ett bra sätt. Robusthet utgörs enligt Dix et al. (ibid.) av observerbarhet ("observability"), återställbarhet ("recoverability"), responstid ("responsiveness") och uppgiftsöverensstämmelse ("task conformance").

En användare har oftast en intention med att använda ett datorsystem. Denna intention översätts gärna till en handlingsplan i interaktionen<sup>21</sup>. För att användaren skall kunna veta om handlingsplanen är hållbar eller inte, måste det vara möjligt att utvärdera systemets interna tillstånd och jämföra detta med handlingsplanen. Således måste det interna tillståndet genom gränssnittet kunna observeras.

Som vi alla vet är det ganska lätt att göra misstag i användningen av datorsystem. Resultatet av misstag kan ibland vara irriterade. Men det kan också vara farliga misstag med ödesdiga konsekvenser (och borde därför vara omöjliga). Därför bör det i ett system som är robust vara möjligt att så att säga "återställa" från ett gjort misstag och kunna gå tillbaka till de föregående "friska" tillståndet.

När ett tillstånd (i datorsystemet) förändras måste detta på ett eller annat sätt uttryckas till en användare. Hur snabbt denna kommunikation sker mellan användare och system brukar kallas för responstid. Generellt sett brukar det vara önskvärt att ett system har så kort responstid och omedelbar verkan som möjligt. Men det finns tillfällen där man kan överväga ett sådant mål eftersom det kan stressa användare och till och med skapa förvirring (eftersom det går för snabbt).

---

<sup>21</sup> Se exempelvis Norman (1990).

Uppgiftsöverensstämmelse slutligen handlar om, om det finns en överensstämmelse mellan de mål som användare vill nå och de tjänster som systemet erbjuder. Jag (Theis) som just nu skriver ett delkapitel i en avhandling tycker det är önskvärt med rättstavning eftersom det är mitt mål att texten är fri från felstavningar. Frågan är då: Finns denna tjänst och kan jag styra den på önskat sätt?

## **2.4 Sammanfattning och reflektion**

För att återkoppla till problematiseringen, syftar denna avhandling till att undersöka fenomenet IT-kvalitet i relation till kunskap och kunnande. I detta kapitel har vi visat den teoretiska behandlingen av begreppet ifråga. Som kanske redan har uppmärksammats är det i presentationen ovan inte mycket eller många som explicit fokuserar på denna relation. Innan vi dock börjar diskutera denna, som vi anser, problematik kan en sammanfattning av kapitlet vara på sin plats.

Kapitlet började med att ge en inblick i hur kvalitetsbegreppet mer formellt började få fotfäste. Detta skedde inom tillverkningsindustrin med anledning av industrialiseringen. Synen på kvalitet och sättet att kontrollera visade sig efterhand inte vara tillräcklig och fick ge vika för en mer kundcentrerad syn. Samtidigt förändrades också ansvaret för kvalitet. Kvalitet var allas arbete och skulle genomsyra hela organisationen. Kvalitetssystem på alla nivåer i organisationen togs fram och tillämpades.

Dessa tankar kan grovt sägas ligga till grund för de kvalitetssystem som tillämpas vid utveckling av IT-artefakter. Kvalitetssystemet kontrollerar processen både genom att standardisera och mäta i process och produkt. Trots tillämpning av kvalitetssystem finns det alltid en risk att processen i sig är felaktig för den produkt som utvecklas. Det spelar alltså ingen roll hur mycket processen är kvalitetssäkrad om processer eller delprocesser inte är ändamålsenliga för den produkt som utvecklas. CMM (Capability Maturity Model) försöker hantera denna problematik genom statistiska mätningar i

processen som ligger till underlag för så att säga "finslipa" processerna.

Trots att många har ägnat mycket tid åt kvalitet och kvalitetssystem framstår det som mycket osäkert vilken innebörd begreppet bör ges. Ett belysande exempel på denna vansklighet ger Garvin (1988) där olika definitioner av kvalitet grupperas om sex kategorier där respektive kategori exemplifieras med två definitioner. Vi tog emellertid bara upp en från varje kategori.

Kapitlet fortsätter sedan med att förmedla en bild av mjukvarukvalitet. Beskrivningen tar sin utgångspunkt i när de första datorerna började användas. Kvalitet innebar till en början effektivitet, men har under historiens förlopp utvecklats till att betecknas med en hel flora av begrepp. Datoranvändningens utveckling har emellertid inte varit problemfri. Då datorer började användas till administrativa syften, översvämmades marknaden snart av programvara. Mängder av program, som dessutom inte höll någon högre kvalitet, skulle underhållas. Mjukvarukrisen var ett faktum och bland annat med anledning av denna skapades Software Engineering.

Denna nya disciplin försökte ge sig i kast med de föreliggande kvalitetsproblemen. Den dominerande synen var och är idag att de skulle lösas med hjälp av matematiska modeller och mätning av olika kvalitetsbegrepp.

Kapitlet fortsätter sedan med att beskriva gränssnittskvalitet. Återigen har vi försökt spegla den historiska utvecklingen vilket tar sitt uttryck i olika syner inom MDI-området. Den tidiga forskningen verkar vara dominerad av ett mekanistiskt, biologiskt och behavioristiskt intresse där människan ses som en informationsprocessor eller biologisk maskin. Detta intresse uppvisar stora likheter med många aspekter inom Software Engineering. Intresset präglas av ett sökande efter objektiva, generaliserbara och mätbara kvalitetsmål.

Denna syn på människan har starkt kritiserats av många, inte minst inom MDI-området. Mätbara aspekter är i sig inte tillräckliga för att förstå den komplexitet som finns i interaktionen mellan en användare och ett datorsystem. Detta har lett till uppkomsten av andra



perspektiv såsom det flexibla, det subjektiva och det sociala perspektivet. Icke desto mindre finns för varje perspektiv förespråkare respektive antagonister.

Vi avslutade detta kapitel med en introduktion till de kvalitetsbegrepp som har utvecklats inom ramen för gränssnittskvalitet.

För att återgå till inledningen av denna sammanfattning och nämnda problematik kan det framstå som att vi påstår att behandlingen av IT-kvalitet saknar ett kunskaps- och praxisperspektiv. Det gör det naturligtvis inte. Bakom varje val av definition, val av egenskap att fästa uppmärksamheten på, tillvägagångssätt att uppnå hög kvalitet o.s.v., döljs en syn på kunskap och praxis. Den formuleras eller uttrycks emellertid inte alltid explicit. Inom Software Engineering framskymtar två sådana perspektiv, dels det strängt matematisk-logiska och dels vad man skulle kunna uttrycka som ett mer pragmatiskt matematisk-logiskt perspektiv. Inom MDI-området identifierar Löwgren (ibid.) i relation till användbarhet ett flertal, såsom ”general theory”, det flexibla och det subjektiva perspektivet. Men överlag verkar explicita formuleringar av dessa perspektiv lysa med sin frånvaro. Dessa, i någon mån, kunskapsteoretiska perspektiv finns dock alltid och styr både innehåll i och på vilket sätt som IT-kvalitet betraktas.

Det som framstår som gemensamt för dessa något osynliga perspektiv, med en viss reservation för MDI-området, är att de inte inbegriper praktiserande systemutvecklare – det saknas ett praxisperspektiv. I fokuseringen på IT-kvalitet saknas i vår mening ofta den som faktiskt utvecklar systemen och den praxis där utvecklingen äger rum. Detta kan bero på många orsaker. En kan vara att den syn som anläggs helt enkelt inte omfattar praktikern och dennes kunskap. Till exempel är praktikern irrelevant ur ett matematiskt logiskt perspektiv, då strävan är att göra kunskapen om kvalitet objektiv och därmed frikoppla den från både subjekt och praxis. En annan kan vara att den generaliseras bort i själva framställningen av ett kvalitetsbegrepp. Därtill varierar synen på kunskap kraftigt, d.v.s. på vad som räknas som kunskap. Det matematisk-logiska

perspektivet accepterar inte kunskap som formellt inte går att bevisa eller härleda.

Som vi kommer att visa senare har vi valt ett bredare kunskapsperspektiv som tar sin utgångspunkt i praxis och den praktiserande systemutvecklaren. Denna kunskapssyn inkluderar både kunskap som är teoretisk och kunskap som mer är av handlingsorienterad karaktär. Den handlingsorienterade sidan innehåller till exempel begrepp som tillägning, bedömning, erfarenhet och kommunikation i relation till praktikern systemutvecklaren. Och det är just den handlingsorienterade sidan av kunskap som vi här anser saknas. Teorin lämnar här ett tomrum. För att vara mer specifika, saknas det i den teoretiska behandlingen av IT-kvalitet, en diskussion om hur systemutvecklare lär sig kvalitet, hur de bedömer kvalitet, hur de pratar om kvalitet, vilken roll erfarenheten spelar o.s.v.

Vi kommer i nästa kapitel diskutera dessa aspekter i mer detalj, d.v.s. begreppet kunskap och kunnande.

### 3 Kunskap och kunnande

Vad vi har presenterat ovan är behandlingen av IT-kvalitet som en egenskap och som begrepp. Men det finns ytterligare en aspekt som framstår som viktig när det gäller IT-kvalitet i förhållande till systemutveckling som för oss till stor del upplevs som en fråga om kunskap – det som vi har identifierat som en bortglömd aspekt. Det finns alltså en praxisdimension som är intressant att studera när det gäller kvalitet på dataprogram och -system, en dimension där systemutvecklarna utgör praktikerna som försöker uppnå hög IT-kvalitet. På så vis är IT-kvalitet inte endast en fråga om egenskaper, utan också om systemutvecklarens kunskap om kvalitet, kunskap för att bedöma kvalitet etc.

Detta leder sedan vidare till närbesläktade frågor om skicklighet, kompetens, tyst kunskap och lärande. Flera av dessa ord har i dagens samhälle en moderiktig klang i och med att det talas om ”kunskapssamhället” och ”kunskapsarbetare”. Kunskapssamhället anses efterträda industrisamhället och i det är kunskapsarbetare den nya sortens arbetare.

Frågan om kunskap är ju en klassisk filosofisk fråga som har behandlats under hela den västerländska filosofins historia – kunskapens väsen är helt enkelt ett av de stora centralproblemen. Kampen har bland annat stått mellan dem som hävdade att vår kunskap om vår värld och verklighet helt är avhängig av och uppkommen genom våra sinneserfarenheter, vilket empiristerna, exempelvis Locke med sin berömda liknelse med det oskrivna bladet – tabula rasa, hävdade. Rationalisterna, exempelvis Kant, å sin sida hävdade att vissa av våra kunskaper är medfödda eller oberoende av erfarenheten. Frågan om vem som har rätt låter sig nog aldrig besvaras tillfredsställande och uttömmande, om nu verkligheten över huvud taget är beskaffad på ett sådant sätt att ett korrekt svar existerar. Det är vilket fall ett kunskapsteoretiskt område där några väl kända åskådningar är positivism, hermeneutik, kritisk rationalism och

empirism<sup>22</sup>. Det finns således en rad olika föreställningar om vad det innebär att ha kunskap och om kunskapens anatomi<sup>23</sup>.

Vi gör emellertid inga anspråk på att vara filosofer eller kunskaps-teoretiker, så den filosofiska dimensionen i frågan om kunskap och kunnande lämnar vi i stort därefter och inriktar oss på översiktligare resonemang. Vårt intresse för kunskap och kunnande handlar i stor utsträckning om dessa begrepp kopplade till en praktisk verksamhet, alltså någon form av praxiskunskap eller handlingskunskap, där begreppet tyst kunskap intar en central plats och där erfarenhetsuppbyggnad, reflektion och kunskapsutbyte är viktiga ingredienser<sup>24</sup>.

Vi skall i det följande utgå från Johannessens (Johannessen i Johannessen & Rolf, 1990; Johannessen, 1999) respektive Rolfs (Rolf i Johannessen & Rolf, 1990; Rolf, 1995) uppfattningar om kunskap och kunnande och väljer att separera dem i redovisningen eftersom de uppenbarligen själva anser att deras synsätt är fundamentalt olika. Vi kommer dock att avsluta med en sammanfattning av dessa båda synsätt, som snarare tar fasta på likheter än skillnader.

---

<sup>22</sup> Se Alvesson & Sköldberg (1994) för en introduktion till dessa åskådningar.

<sup>23</sup> Det kan därför vara värt att redan här poängtera att vi inte kommer att argumentera för en kunskapsteoretisk position gentemot andra kunskapsteoretiska utgångspunkter. En sådan diskussion ligger utanför avhandlingens syfte och skulle med all säkerhet resultera i en avhandling i sig. Däremot kommer vi i beskrivningen av Johannessens kunskapssyn representera på ett mycket kortfattat sätt Johannessens kritik av den logiska positivismen. Detta gör vi för att skapa ett sammanhang och en bättre förståelse för Johannessens kunskapssyn.

<sup>24</sup> Vi har valt att endast utgå från Kjell Johannessen och Bertil Rolf i vår teoretiska behandling av praktisk kunskap. Andra referenser som hade varit intressanta för våra diskussioner är Schön (1983, 1987), som diskuterar den reflektive praktikern och nödvändigheten av reflektion, och Molander (1996) som vidareutvecklar vissa tankegångar om praktisk och tyst kunskap. Exempelvis Messeter (2000) använder dessa författare konstruktivt i sin avhandling. Vi refererar dock Schön i vår diskussion om reflektion i avsnitt 6.5, men inte i en sådan omfattning att vi anser att en teoretisk fördjupning är nödvändig. Reflektionens betydelse för utveckling av kunskap framgår även klart av Rolf (1995).

Ett sätt att se på kunskap är att den kan ses som tre olika typer eller aspekter<sup>25</sup>: påståendekunskap, färdighetskunskap och förtrogenhetskunskap (Nordenstam, 1983; Johannessen, 1999). Både Kjell Johannessen och Tore Nordenstam utgår från den wittgensteinska traditionen i sin syn på kunskap, och då framför allt praxis- eller handlingskunskap. I opposition till detta har Bertil Rolf lyft fram den ungerske samhällsvetaren Michael Polanyis idéer i samma ämne.

Även om Johannessen och Rolf, och kanske framför allt Rolf<sup>26</sup>, verkar tycka att det finns fundamentala skillnader dem mellan, finner vi att det finns stora likheter mellan Rolfs och Johannessens tankar, inte minst gäller det avståndstagandet från det positivistiska arvet – ett avståndstagande de delar med många ”postmoderna” tänkare, exempelvis Magee (1999) när det gäller filosofi och Kvale (1997) när det gäller undersökningsmetodik. Vi anser också att de två synsätten delar många likheter, bland annat inställningen till regelföljande inom en praxis och den sociala kontext som kunnande utövas inom.

Dessa olika teorier om kunskap och kunnande belyser viktiga aspekter av kvalitet i relation till det vi behandlar, inte minst gäller detta den undersökning vi genomfört och som presenteras i kapitel 4.

### **3.1 Kunskap och kompetens enligt Johannessen**

Enligt Johannessen (1999) har de logiska positivisternas kunskaps-teoretiska arv varit och är kanske fortfarande förhärskande. Detta arv, menar han, har utarmat våra föreställningar om vad det innebär att ha kunskap. Enligt den logiska positivismen kan vi enbart ha

---

<sup>25</sup> Enligt Rolf (1995) råder det vissa tveksamheter om det är aspekter eller typer som avses, men Nordenstam (1983, s. 21) skriver faktiskt ”... tredelning av olika slags kunskap... ”.

<sup>26</sup> Se exempelvis i (Johannessen & Rolf, 1990).

kunskap om sådant som kan formuleras och beläggas med empiriska metoder:

(1) Det måste vara möjligt att formulera vårt vetande på ett eller annat språk (ibid., s. 15).

(2) Den språkligt formulerade utgåvan av vårt vetande måste kunna beläggas empiriskt eller bevisas med formella metoder (ibid., s. 15).

Att ge vårt vetande eller vår kunskap en språklig dräkt och belägga den med empiriska observationer eller logisk-matematiska bevis framstår alltså med utgångspunkt från detta som ett nödvändigt inslag.

Att använda ett begrepp innebär således att vi verbalt måste kunna förklara och belägga eller bevisa det. Talar vi om en stol måste vi alltså kunna förklara vilket betydelseinnehåll uttrycket stol har. Om man definierar uttrycket skulle man kanske komma fram till följande: "Uttrycket 'stol' används vanligtvis på svenska för att beteckna 'sittmöbel med ryggstöd beräknad för en person'" (ibid., s. 96). Att tala om en stol innebär således att det måste handla om följande: (a) att det är en sittmöbel, (b) att den måste ha ett ryggstöd och (c), att den skall vara beräknad för en person. Dessa tre villkor utgör begreppets innehåll och skulle enligt ovanstående vara ett uttryck för vårt vetande om begreppet stol.

Detta den logiska positivismens krav anser Johannessen (ibid.) vara oroande, eftersom många typer av omdömen därmed hamnar utanför kunskapens eller vetandets område. Omdömen och mer handlingsorienterad kunskap karakteriseras just av att de inte kan "bevisas" empiriskt eller härledas matematiskt<sup>27</sup>. En stol har ju även en användningsdimension och ingår i en kontext som är olika från

---

<sup>27</sup> Johannessen påpekar att mindre generella värdenormer går att deducera ifrån allmänna normer av en viss typ i ett system. Men de normer som tillsammans antas vara grundnormen kan inte styrkas med empiriska eller formella metoder och därför hänger fortfarande hela systemet i luften.

individ till individ<sup>28</sup>. Vi skulle därför enligt Johannessen, d.v.s. om vi antar denna gränsdragning för kunskapsbegreppet, frånta oss möjligheten att prata om till exempel estetisk, etisk eller moralisk kunskap.

I opposition till den logiska positivismen representerar Johannessen ett vidare kunskapsbegrepp. Han menar att kunskap inte enbart låter sig fångas genom att ange ett språk, som innehåller logiska konstanter, empiriska variabler, formationsregler eller syntaktiska regler, härledningsregler och korrespondensregler. Det finns element i kunskapen som ligger utanför denna gränsdragning som likväl bör betraktas som kunskap.

Denna ståndpunkt menar Johannessen att han heller inte är ensam om. Redan Aristoteles talade enligt Johannessen (ibid.) om praktisk kunskap (phronesis) och hantverksmässig kunskap (techne) vid sidan av vetenskaplig kunskap (episteme), där den förstnämnda handlar om "gott" omdöme och den andra handlar om färdigheter<sup>29</sup>. Utöver Aristoteles, skriver Kuhn (1970) om exemplariska lösningar, som fastslår det kognitiva innehållet i teorier och lagar som är nödvändiga för förståelsen och användningen av dessa (d.v.s. teorierna och lagarna). Användningen av teorier och lagar utgör alltså en nödvändig del av förståelsen (ibid.). Och det finns säkerligen fler.

För att förstå Johannessens kunskapssyn är det emellertid viktigt att först förstå hans syn på språk och begrepp. Denna förståelse tar sin utgångspunkt i Wittgensteins (1958, 1967) senfilosofi.

---

<sup>28</sup> En stol kanske just är en stol för till exempel mig (Theis), men för min pappa som är finsnickare är den en möbel. Dessa olika perspektiv gör att begreppet stol får olika dimensioner utifrån olika perspektiv.

<sup>29</sup> Som naturligtvis är mycket svåra att verbalt formulera.

### 3.1.1 *Språk och begrepp*

Som vi har nämnt tidigare framstår det enligt Johannessen som helt oproblematiskt för de logiska positivisterna att artikulera all kunskap.

Även Wittgenstein (1961, enl. Johannessen, 1999) gjorde på 20-talet ett försök i denna riktning, men kom till insikt, som ett resultat av arbetet, att det måste finnas inslag av tyst kunskap i användningen av språket och att relationen mellan språket och användningssituationen är en väsentlig del av språkets betydelseinnehåll.

Enligt Johannessen stod språkets logiska form i fokus för de logiska positivisterna. Genom korrespondensregler, som fastlägger villkoren för användningen, definieras begrepp<sup>30</sup>. Men att ge form åt en korrespondensregel är att utföra en handling i sig, som bygger på att den som definierar redan har en kunskap om språket och det som definieras. Detta innebär att användningen av en definition alltid sker utifrån denna förståelse eftersom förståelsen i sig inte kommer till uttryck i själva definitionen.

Därför ger ofta inte en definition någon ledtråd till hur den skall användas. En definition kan således användas på ett otal olika sätt och är med andra ord regellös. Således finns det inte någon garanti för hur en definition eller regel används. Därför kan inte korrespondensreglerna heller vara de villkor som förbinder begrepp och verklighet, utan snarare har det med handlingar och beteenden att göra. Detta handlings- och beteendemässiga regelföljande är, för Wittgenstein (1967, enl. *ibid.*) en praxis.

Det är denna praxis som ger orden deras mening. Det finns en ”konstitutiv relation mellan språklig begreppsbildning, mänskliga verksamhetsformer och den verklighet som framstår som vår verklighet i kraft av de begrepp som vi har skapat oss för verkligheten.” (Johannessen, 1999, s. 25).

---

<sup>30</sup> ”Tecknet T är definiendum i definition och beskrivningen av begreppskännetecknen utgör det definierade uttrycket (definiens).” (*ibid.*, s. 22). Till exempel stalexemplet i föregående avsnitt.



Att studera och förstå ett begrepp blir därför uddlöst på definitionsnivå och rent språklig nivå. Istället bör man undersöka de situationer (praxisar) där begreppet används, där det förmedlas och förklaras, eftersom det är där villkoren för begreppet kommer till uttryck. Det är därför också i praxis man lär sig att behärska begrepp. Själva utövandet av en verksamhet har ett betydelseskapande inslag i de begrepp som används.

Att behärska ett begrepp blir sålunda en fråga om både färdighet och förtrogenhet med begreppet. En färdighetskunskap i att använda begreppet och en förtrogenhetskunskap med de förhållande som det säger något om.

### *3.1.2 Påstående-, färdighets- och förtrogenhetskunskap*

För att få en klarare förståelse av begreppen påstående-, färdighets- och förtrogenhetskunskap, kan Johannessens analys av verbet veta tjäna som en god utgångspunkt.

Johannessen börjar med hur vi vanligen brukar använda, ”veta att...”, och ställer upp följande påstående:

1. Jag vet att järn utvidgar sig vid uppvärmning;
2. Jag vet att  $2 + 2 = 4$ ;
3. Jag vet att förklaringar tar slut någonstans;
4. Jag vet att folk blir förnärmade när de inte blir tagna på allvar;
5. Jag vet att det är orätt att döda.

(Johannessen, 1999, s. 139).

De första två uttrycker kunskap om naturen och om matematik, som respektive empiriskt och matematiskt kan bevisas. Utsagorna bygger på vetenskapliga teorier och är bortom all tvivel. Jämförelsevis ligger detta enligt Johannessen också närmast de logiska positivisternas kunskapssyn och han kallar detta för påståendekunskap.

Påståendekunskap är alltså sådan kunskap som är möjlig och lättare att artikulera och som kan uppträda i formen av regler, metoder, normer, teorier etc.

I detta avseende framstår emellertid inte de tre sista som lika självklara. Dessa ger ett uttryck för logisk kunskap, människokänedom samt moralisk kunskap och kan till skillnad från de första två, inte utan vidare beläggas empiriskt eller bevisas matematiskt. Det tredje handlar om en insikt i förklaringsrymdens gränser, det fjärde om en erfarenhetsbaserad förståelse för människan och det femte om värden. De två sistnämnda karakteriseras också av att de kommer till uttryck genom handlingar, d.v.s. att handlingar uttrycker vår kunskap om moral och människor. Detta visar på ett utrymme för andra former av kunskap.

Johannessen ger uttryck för detta när han knyter adverbet hur till verbet veta, d.v.s. ”veta hur...”. Med utgångspunkt från detta ställer han upp följande påståenden:

6. Jag vet hur man använder detta program;
7. Jag vet hur en apelsin smakar;
8. Jag vet hur en klarinett låter;
9. Jag vet hur han ser ut.

(Johannessen, 1999, s. 141).

Precis som tidigare, d.v.s. i jämförelse med punkterna 4 och 5, karakteriseras också punkt 6 av att kunskapen kommer till uttryck genom handling. För att avgöra om den är sann eller falsk måste personen ifråga visa prov på hur programmet skall användas. Således får denna form av kunskap ett handlingsmässigt uttryck som berättigas just genom handling. Denna kunskapsform kallar Johannessen för färdighetskunskap.

Det tre sista utsagorna pekar på ytterligare en form av kunskap. Förvisso är det en färdighet att kunna känna igen en klarinett eller person, men själva handling (igenkännandet) bygger på att man har

erfarit ljudet eller ansiktet tidigare. Den bygger på att man är förtrogen med det aktuella sakförhållandet. Detta benämner Johannessen som förtrogenhetskunskap. Förtrogenhetskunskap handlar alltså om att vara förtrogen med en repertoar av situationer som möjliggör handlingar.

### *3.1.3 De tysta inslagen*

Förtrogenhetskunskap, men även till viss del färdighetskunskap framstår, i relation till påståendekunskap, som svåra att artikulera eller notationsmässigt fånga. Hur talar vi till exempel om för någon annan hur klangen av en klarinett eller smaken av en apelsin skall kännas igen? Hur talar vi om hur klangen av en klarinett låter eller hur en apelsin smakar?

Detta hindrar dock inte att denna form av kunskap (med nödvändighet) aktualiseras i en rad olika sammanhang. Det är tveklöst så att man kan peka ut klangen från en klarinett eller lukten från en apelsin. Det är dock svårt att förmedla denna kunskap i ord<sup>31</sup>. Hur gör vi när vi känner igen lukten av den apelsin eller tonen av en klarinett?

Färdighetskunskapen karakteriseras också av att den är svårartikulerad. Kanske inte i lika hög grad som förtrogenhetskunskap, men åtminstone till vissa delar. Till skillnad från förtrogenhetskunskap kommer färdighetskunskap till fullt uttryck i själva utövandet av färdigheten. Därför blir den mer synlig än förtrogenhetskunskapen och därmed mer lättillgänglig.

Att förklara färdigheterna simma eller cykla är emellertid inte heller så lätt. Fysiken och biologin låter oss tala om färdigheterna till en viss gräns, men med antalet delar som skall förklaras (t.ex. hålla

---

<sup>31</sup> Precis som Rolf (1995) påpekar, anser även Johannessen att förmågan att förmedla är beroende av hur välutvecklat språket är om det aktuella fenomenet. Två diregener skulle alltså kunna föra en mer nyanserad och detaljrik diskussion kring olika musikinstrumenters klanger än vad till exempel jag och Odd skulle kunna göra. Därför är gränsen för vad som är tyst beroende av både kontext och individ.

balansen i förhållandet till underlaget samtidigt som tyngd, utväxling och bromsverkan kontrolleras) och sammansätts till en helhet (cykling) blir förklaringarna alltmer svårare.

Dessa två former av kunskap, d.v.s. färdighetskunskap och förtrogenhetskunskap, utvecklas i vår mening ömsesidigt. Varje färdighet vilar på och utvecklas genom förtrogenhetskunskapen. Att känna igen klangen från en klarinett bygger på att man är förtrogen med dess klang, d.v.s. att man har tränat förmågan att känna igen den genom att upprepade gånger ha lyssnat på en klarinett. Omvänt utvecklas också förtrogenheten genom färdighetskunskapen. Varje lyckat igenkänningsförsök blir en del av repertoaren – en del av förtrogenhetskunskapen.

### *3.1.4 Exempels makt*

Enligt Johannessen är alltså färdighetskunskap och förtrogenhetskunskap svåra att direkt formulera verbalt och fånga skriftligt. Det är dock inte allt som är uttryck för oartikulerbart vetande. Enligt Johannessen drar Wittgenstein (1958) en skiljelinje mellan det transitiva och det intransitiva. Det transitiva, det ”överförbara”, fastställer det som är det säregna eller egenartade hos ett visst fenomen. Johannessen exemplifierar detta med meningen: ”Denna tvål har en säregen doft som påminner om den vi använde som barn” (Johannessen, 1999, s. 85).

Jämförelsen karakteriserar här lukten och bidrar till förståelsen hos alla dem som känner till vilken tvål som användes av talaren. Vi skulle till och med få en förståelse, utan kunskap om talarens barn-domstvål, för tvåldoftens karaktär om vi har använt samma tvål (många av oss har som barn till exempel använt Barnängen).

Sägs däremot detta utan en jämförelse, d.v.s. ”Denna tvål har en säregen doft”, är det intransitivt. Detta innebär att mottagaren av uttrycket inte ges någon möjlighet att få en uppfattning om vilken doft det är som avses. I detta avseende handlar det inte om ett försök att föra över något utan snarare om att fästa uppmärksamheten på något, d.v.s. att tvålen doftar säreget.

Önskar vi nyansera uttalandet om tvåldoften, är ovanstående uttryck ganska knapphändig. Men det är just så vi kan uttrycka oss kring förtrogenhetskunskap, genom jämförelser, analogier, metaforer, exempel osv.

För att återgå till ett tidigare exempel; när vi önskar förmedla klangljudet av en klarinett till någon annan, hänvisas vi till att använda en rad exempel som "visar" klarinettens klang. I det här sammanhanget kanske man skulle säga: "Lyssna, det här är en klarinett, hör du klangen? Det där är inte en klarinett, hör du skillnaden?" Likväl innebär detta att vi lär oss på samma sätt. Genom att någon visar, pekar ut och jämför kan vi utveckla förtrogenhets- och färdighetskunskapen.

I den utsträckning vi försöker att förmedla eller utveckla de här formerna av kunskap kommer vi enligt Johannessen (ibid.), med nödvändighet att luta oss mot mer indirekt kommunikation såsom analogier, exempel, metaforer o.s.v.

### *3.1.5 Sammanfattning Johannessen*

I opposition till de logiska positivisterna presenterar Johannessen en annan syn på kunskap. Med utgångspunkt från Wittgenstein anser han att kunskapen inte enbart kan utgöras av välformulerade regler, antingen empiriskt eller matematiskt baserade, som binder ihop språk och verklighet. Vi skulle då frånta oss möjligheten att prata om till exempel estetisk, etisk eller moralisk kunskap.

Med utgångspunkt i Wittgenstein (1958) visar Johannessen (ibid.) också att formuleringen av korrespondensregler är en handling i sig och bygger på en nödvändig förståelse. Denna förståelse kommer inte till uttryck i definitionsakten och det finns därför ingen garanti för hur definitionen skall användas. Att behärska ett begrepp blir sålunda inte enbart en fråga om påståendekunskap, utan även en fråga om färdighet och förtrogenhet. En färdighetskunskap i att använda begreppet och en förtrogenhetskunskap med de förhållande som det säger något om.

Påståendekunskap är alltså sådan kunskap som är möjlig och lättare att artikulera och som kan uppträda i form av regler, metoder, normer, teorier etc. Färdighetskunskap karakteriseras av att den mer är en slags handlingskunskap och förtrogenhetskunskap handlar om att vara förtrogen med en repertoar av exempel.

De två sistnämnda, d.v.s. färdighets- och förtrogenhetskunskap karakteriseras också av de är "tysta". De är alltså svåra att klä i ord och att fånga i skrift. Att kommunicera och utveckla dessa former av kunskap blir därför en fråga om en mer indirekt kommunikation, såsom genom analogier, exempel, metaforer osv.

### **3.2 Kunskap och kompetens enligt Rolf**

Som vi har skrivit i början av det här kapitlet så bekänner sig Rolf till en annan teori om kunskap och kunnande som han baserar på Polanyi, som han dessutom tycker är bättre än den som Johannesen står för. Vi skall därför här kort och summariskt återge polanyisynen, vilken bygger på Bertil Rolfs bok "Profession, tradition och tyst kunskap" (Rolf, 1995). Liksom hos Johannesen handlar det om tyst och praktisk kunskap kopplat till regelföljande och värdering. Synen på tyst kunskap skiljer dock polanyisynen från wittgensteinsynen.

Kärnan i Rolfs (Rolf i Johannesen & Rolf, 1990; Rolf, 1995) kritik är vad han kallar osägbarhetsdoktrinen, d.v.s. att det utifrån wittgensteintraditionen skulle finnas viss kunskap som av princip är omöjlig att formulera språkligt – det ligger i den kunskapens natur att vara osägbar – medan Polanyi enligt Rolf anser att någon del av kunskapen alltid kommer att förbli tyst vid varje givet tillfälle<sup>32</sup>. Den

---

<sup>32</sup> Skiljelinjen verkar alltså gå mellan att en viss *typ* (sort, art, slag) av kunskapen är tyst till sin natur och att någon *del* (alltså inte typ) av kunskapen är tyst. Rolf (1995) har en liknelse om att försöka täcka ett 12m<sup>2</sup> stort golv med en 10m<sup>2</sup> stor matta. Enligt wittgensteinsynen skulle en viss del av golvet aldrig gå att täcka, medan polanyisynen ger att det inte går att täcka hela golvet samtidigt med mattan och att någon del av golvet därmed blir otäckt. Detta enligt Rolf.

tysta kunskapen är alltså tyst till sin funktion snarare än till sitt innehåll.

### 3.2.1 *Tyst kunskap enligt Polanyi*

Att vi har kunnanden som kan vara svåra att berätta om eller uttrycka muntligt, har också Polanyi intresserat sig för. Ett exempel han använder är att vi kan känna igen ett ansikte, men svårligen förklara hur igenkännandet går till och sätta ord på det – ”*we know more than we can tell*” som Polanyi (1966, s. 4) säger. För att känna igen ett ansikte utnyttjar vi ett underliggande kunnande om fysionomiska detaljer som formar en enhet och det är enheten, inte summan av detaljerna, vi känner igen som ett bekant ansikte. Detta underliggande kunnande är det som Polanyi kallar ”tacit knowing/knowledge” vilket fått beteckningen ”tyst kunskap” på svenska. En lämpligare term hade varit ”underförstådd/implicit kunskap”, för som vi skall se nedan är den inte osägbar, utan möjligtvis till delar utsägbar.

Med utgångspunkt från Rolf (1995) är tyst kunskap hos Polanyi att betrakta som ett verktyg som riktar våra tankar och vårt sökande efter ny kunskap. Kunskapen framstår inte som tyst i meningen att den till sin natur är omöjlig att verbalisera, även om någon del av den förmodligen alltid kommer att vara utsägbar, utan det handlar om att kunskapen kan fungera tyst – den har en tyst *funktion*. När kunskapen fungerar tyst gör den det i förhållande till en viss handling vi utför och att den handlingen står i fokus för vår verksamhet. För att vi skall kunna utföra den handling som står i fokus för oss, måste vi dock ha andra kunskaper till stöd för detta och dessa kunskaper fungerar tyst, eftersom vi inte riktar fokus mot dem. Tyst kunskap är alltså situationsbetingad – beroende på vad vi väljer att fokusera på, så framstår vissa kunskaper som tysta till sin funktion.

Som ett exempel på detta kan vi ta mitt (Odds) skrivande av denna text om Polanyis kunskapssyn. I fokus för min handling står att kortfattat försöka förmedla min förståelse för Rolfs tolkning av Polanyis syn på tyst kunskap. Kunskaper som fungerar tyst är då

kunskaper om det svenska språket (grammatik, ordförråd, skrivregler, etc.), kunskap om hur man kan berätta något via skriftspråket, kunskap om att hantera en ordbehandlare, etc. Detta är så att säga underförstådda kunskaper som jag använder för min handling. Vid ett visst tillfälle kommer jag att korrekturläsa denna text och då riktar jag istället fokus på sådana saker som meningsbyggnad, stavfel, syftningsfel med mera – d.v.s. mer grammatiska aspekter av texten. Så min kunskap om svenska språket som i situationen att skriva detta fungerade tyst, är nu istället i fokus för min verksamhet och därmed inte tyst fungerande.

Nu talar inte Polanyi endast om tyst kunskap ("tacit knowledge"), utan också om tyst kunnande ("tacit knowing"), vilket ger kunskapen en mer dynamisk karaktär. Kunskap är alltså inte endast något man har, utan också något man gör – den är handlingsinriktad. Att istället fokusera på kunnande, d.v.s. en aktivitet snarare än en egenskap eller ett innehåll, leder till ett intresse för kunskapsproduktion och kulturella system som gör det möjligt för människor att utveckla och öka sitt kunnande (Blackler, 1995). I Polanyis syn på kunskap och kunnande är den sociala och kulturella aspekten väldigt viktig, vilket inte minst är synligt i hans syn på kompetens som enligt Rolf (1995) är ett slags praktisk kunskap. Innan vi kommer till den skall vi emellertid först ta upp Polanyis begrepp personlig kunskap som är kopplat till tyst kunskap.

### *3.2.2 Personlig kunskap*

Personlig kunskap är enligt Rolfs tolkning av Polanyi en förening av tradition och subjektivitet hos en person – "Polanyi vill genom teorin om personlig kunskap gestalta hur individ, tradition och samhälle förhåller sig till varandra" (ibid., s. 96). Den personliga kunskapen är handlingsinriktad kunskap som personen besitter, men det är även värderingar som gör anspråk på att vara allmängiltiga. Den personliga kunskapen är en integrering av den subjektiva erfarenheten och det som traditionen förmedlar och genom att den har integrerats hos en person kan den anta tyst funktion.



Det kulturen och traditionen kan förmedla är exempelvis teorier, tumregler, värderingar, färdigheter, med mera, som kan användas på ett enligt kulturen giltigt sätt. Detta integreras i den personliga kunskapen och tyst fungerande mönster används för "... att handla och att ordna ett erfarenhetsområde" (ibid., s. 97). Enligt Rolf är kunskapsintegrationen en form av färdighet som följer sin bärare och som inte kan säljas som en vara<sup>33</sup>.

Den personliga kunskapen är kopplad till trädning inom yrkesgrupper, där social identitet enligt Rolf beror på kunskap. Genom att identifiera sig med verksamhetens mål och delta i dess aktiviteter, blir kompetensen personlig – "Färdigheter förutsätter vissa aktiviteter, som förutsätter vissa mål. Mål och värderingar blir självklara delar av verksamheten" (ibid. s. 97).

Rolf (ibid., s. 96) ger följande definition på personlig kunskap:

DEF: Kunskapen K utgör personlig kunskap hos A ifall K uppträder som tyst kunskap hos A i någon situation och K gör anspråk på sanning eller allmängiltighet.

### *3.2.3 Praktisk kunskap*

Enligt Rolf (ibid.) är praktisk kunskap förmågan till goda handlingar och detta innefattar tre villkor: förmåga till handling, förekomst av vissa kvalitetskriterier samt att dessa kriterier utövas i handlandet (ibid., s. 113). Praktisk kunskap är inte heller tur, utan har med skicklighet att göra – att kunna att utföra en handling skickligt är att ha praktisk kunskap i det som handlingen syftar till.

Ordet skickligt förutsätter dock en bedömning av kvaliteten på handlingen och att det finns bedömningskriterier<sup>34</sup> för att skilja

---

<sup>33</sup> Vilket är en parallell till teorier om kunskapsarbetare och kunskapssamhället. Enligt Drucker (1993 enl. Blackler, 1995) så innebär detta att kunskapsarbetare har en unik position i det att de äger produktionsmedlen, d.v.s. kunskapen.

<sup>34</sup> Rolf (1995) använder här ordet kvalitetskriterier, men vi tycker att bedömningskriterier är lämpligare eftersom det leder tankarna till kriterier någon

sämre handlingar från bättre. Vissa handlingar verkar sakna bedömningskriterier, åtminstone i vardagssituationen, som de exempel Rolf (*ibid.*) anför, nämligen att promenera och att skratta. Man talar i allmänhet inte om att promenera eller att skratta skickligt. Bedömningskriterier för skicklighet är alltså en förutsättning för praktisk kunskap.

Den praktiska kunskapen är alltså handlingsinriktad och det är i handling i enlighet med bedömningskriterierna den visas. Den praktiska kunskapen är en förmåga att förverkliga värden, som manifesteras i resultatet av handlingen, i själva handlingen eller i båda. De värden som förverkligas måste inte vara värdefulla eller moraliska för alla eller för flertalet för att handlingen skall vara skickligt utförd och uppvisa praktisk kunskap – i princip kan en mördare vara skicklig, även om mord är en förkastlig handling.

Den praktiska kunskapen vilar också på regler som har tyst funktion, där regler har en väldigt vid betydelse, exempelvis regler för att hålla balansen när man cyklar, regler för att duka ett bord rätt för en bjudning etc. Enligt Rolf (*ibid.*) är ett slags regler för Polanyi maximier som "... är ett slags formuleringar av tumregler, som inte helt kan förstås utan den kunskap de uttrycker" (*ibid.*, s. 99). Maximerna är aldrig fullständiga i förhållande till de regler som används:

Regler för en färdighet (art) kan vara nyttiga, men de avgör inte hur man praktiserar färdigheten. De är maximier, som kan tjäna som ledtrådar för färdigheten endast ifall de kan integreras med praktisk kunskap inom färdighetsområdet. De kan inte ersätta denna kunskap. (Polanyi, 1958, s. 50, enl. Rolf, 1995, s. 100).

Med utgångspunkt i detta definierar Rolf (*ibid.*, s. 114) praktisk kunskap som :

---

använder för att bedöma en handling, snarare än till kriterier som finns i själva handlingen. Kvalitet som ord syftar mer till egenskaper hos själva handlingen, medan bedömning mer syftar till aktiviteten att avgöra kvaliteten.

DEF: Personen A har praktisk kunskap i att utföra handlingen H om och endast om:

- a) Det finns vissa regler eller kriterier för kvalitet som skiljer en väl utförd H från en illa utförd H.
- b) A kan utföra H väl.
- c) När A utför H väl, utgör dessa regler eller kriterier tyst kunskap hos A som gör det möjligt för A:s handling att nå sin kvalitet.

### *3.2.4 Skicklighet, know-how och kompetens*

Rolf (ibid.) går sedan över till att dela in den praktiska kunskapen i tre slag och sätter detta i relation till Polanyis tankar. De tre slagen av praktisk kunskap är skicklighet (skill), know-how och kompetens, där skillnaden dem emellan "... består i en annan dimension av den praktiska kunskapen." (ibid., s. 114).

#### *Skicklighet*

Med termen skicklighet menar Rolf (ibid.) färdigheter som man kan lära sig själv genom sin egen erfarenhet i sin kontakt med verkligheten, exempelvis att cykla, eller också socialt genom t.ex. imitation av en mästare. Skicklighet är kopplat till målinriktat beteende, men där bedömningen av måluppfyllelsen inte vilar på någon social gemenskap. Den skicklige kan själv avgöra om målet är uppfyllt genom sin handling. Skicklighet kan vara mer eller mindre situationsberoende genom att färdigheten kan behöva anpassas till olika situationer där handlingen skall utföras. Rolf konkluderar detta med en definition:

DEF: Skicklighet = Praktisk kunskap i enlighet med regler där regelföljandet i princip kan avgöras utan hänvisning till andra bedömare än den aktör som utför handlingen (ibid., s. 116)

#### *Know-how*

Här utgår Rolf (ibid.) från Ryle (1949) och dennes tankar om know-how och know-that, där know-how är handlingsförmåga och

know-that är vetenskap om verklighetens beskaffenhet<sup>35</sup>. Det som utmärker know-how är att den kunskapen förutsätter intelligens och omdöme – handlandet är förknippat med en bedömning av kvaliteten. Detta skiljer dock inte know-how från skicklighet<sup>36</sup>, utan det som skiljer är att know-how är praktisk kunskap vars kvalitet bedöms i ett socialt sammanhang, alltså i förhållande till någon annan bedömare än den handlande själv. Detta resulterar i en definition:

DEF: Know-how = Praktisk kunskap i enlighet med regler där regelföljandet ytterst kan avgöras endast genom hänvisning till andra bedömare än den aktör som utför handlingen (ibid., s. 118).

Det sociala sammanhanget för bedömning utgörs av omgivande aktörer, grupper, institutioner, organisationer, traditioner eller samhälle enligt Rolf (ibid.), vilka uppställer kriterier som ligger till grund för bedömningen. Dessa regler eller kriterier kan internaliseras så att en person själv kan bedöma kvaliteten på sin handling, men det är alltid så att kriterierna ges av det sociala sammanhanget.

Know-how behöver inte innehålla ett artikulerat språk eller reflektion, även om det kan göra det. De sociala reglerna behöver alltså inte kunna uttryckas i ett språk – de måste inte symboliseras. Det är handlingen som bedöms, inte dess symboliserade uttryck.

Slutligen är också know-how mer eller mindre situationsberoende vilket kan kräva ändring av reglerna. Om ett visst know-how är situationsberoende, krävs ständig inläring och det räcker inte att "... en gång ha tillägnat sig en repertoar att agera i enlighet med en standard" (ibid., s. 120).

---

<sup>35</sup> För Polanyi (1966) representerar Ryles "knowing what" en mer intellektuell kunskap och "knowing how" en mer praktisk. Polanyis syn på "knowing" täcker både teoretisk och praktisk kunskap, eftersom den ena aldrig existerar utan den andra.

<sup>36</sup> Något som inte Ryle gör explicit enligt Rolf (1995).

### *Kompetens*

Kompetens är know-how i förening med reflektion. Kompetens förutsätter reflektion; kompetens förutsätter också know-how, d.v.s. att kunna handla i enlighet med sociala regler för kvalitet. Dessa situationsbundna regler, som kvalificerad praktisk kunskap bygger på, anpassas, tolkas och preciseras. Enligt Polanyi förutsätter detta att reflektion kan påverka handlingsreglerna på ett visst område – att genom en social process, förändra och förnya dem (ibid.).

För att en kunskap skall vara kompetens, skall aktören eller den handlande inte endast kunna anpassa sig själv, utan också genom reflektion kunna påverka reglerna för kvalitet – ”Aktörens reflektion kan utveckla områdets know-how” (ibid., s. 121). Rolf definierar kompetens som:

DEF: En aktör A har kompetens inom området O =

- a) Det finns på området vissa regler eller kriterier för kvalitet som skiljer en väl utförd handling från en sämre eller dåligt utförd handling.
- b) Dessa regler eller kriterier är förankrade bland aktörer på området. Deras bedömningar avgör vad som utgör bättre eller sämre prestationer.
- c) Aktören har förmåga att handla i enlighet med dessa regler genom sin tysta kunskap om dessa regler eller kriterier.
- d) Aktörens förmåga innehåller hantering av symboliserande, teoretiska, reflekterande eller kritiska moment.
- e) De reflekterande momenten kan aktören använda för att på ett förutsebart sätt påverka kvalitetskriterierna inom området.
- f) Denna påverkan gör reglerna bättre – enligt vissa latent regler – än vad de annars skulle ha varit.

Det som gör kompetens speciellt i förhållande till en vardaglig användning av begreppet, är att Rolf kopplar kompetens till en roll och inte ser det som en egenskap hos en person. Om en person i sin

roll inom exempelvis en viss organisation, inte kan uppfylla det som står i definitionen ovan, så är han eller hon i ett inkompetensförhållande till sin organisation. Att kunna och att ha möjlighet att reflektera och därigenom förändra regler eller kriterier, är alltså en förutsättning för kompetens enligt det här synsättet. Att styras av regler man inte själv kan påverka är att vara i ett inkompetensförhållande.

### *3.2.5 Reflektion och kommunikation*

Reflektionen är alltså ett viktigt inslag i Polanyis teori – artikulering i symboler gör att vi kan lära mer. Förvisso kan en intellektuell reflektion över detaljerna eller det partikulära i en handling, störa uppfattningen om handlingsenheten så att detta kunnande kan gå förlorat tillfälligt<sup>37</sup> eller i sällsynta fall för alltid, men reflektionen leder till utökad kunskap som gör att man kan återta kunnandet och fördjupa det (Polanyi, 1966).

Att reflektera innebär att man måste kunna distansera sig från sin tysta kunskap och göra den ”otyst” genom artikulation och symbolisering. Därigenom kan den spridas och kritiserars, vilket leder till att den kan utökas. Eftersom den personliga kunskapen är en integrering av subjektiva erfarenheter och den kulturella omgivningen, är den också underkastad kulturella regler som ger grunden för kunskapens anspråk på allmängiltighet och därigenom ”... kan den också kritiserars i en social kunskapsprocess” (Rolf, 1995, s. 101).

På detta vis kan olika personers erfarenheter spridas till andra – en kognitiv arbetsfördelning blir möjlig som Rolf säger (s. 101) – men erfarenheten kan inte ersättas, eftersom båda parterna i ett sådant här utbyte behöver ”... matcha regelsystemet mot sin erfarenhet” (ibid. s. 101).

---

<sup>37</sup> Kommentatorn till fotbollsspelaren – ”Hur kunde du missa öppet mål???”. Fotbollsspelaren – ”Jag hann tänka för länge. Jag skulle bara ha skjutit”. Man bli dock förmodligen ingen bra fotbollsspelare utan träning och heller inte utan intellektuell reflektion över sitt spel.

Tyst kunskap kan därmed bli reflekterad kunskap, som Rolf (ibid. s. 102) definierar så här:

DEF: Reflekterad kunskap är kunskap som genom artikulation göres fokal och som underkastas logiska operationer i avsikt att tolka, analysera, precisera, jämföra eller kritisera för att kunna dra slutsatser.

På detta sätt kan kunskapen utvidgas enligt Rolfs tolkning av Polanyi och språket tjänar som ett redskap för att artikulera och sätta kunskapen i fokus.

Det språket gör är att förmedla en verklighetsbild och ge oss ett redskap att hantera verkligheten. I huvudsak är det genom språket som individen får kunskap förmedlad till sig – den mesta kunskapsförmedlingen sker språkligt (Rolf i Johannessen & Rolf, 1990). Språket tjänar också syftet att förmedla kulturen och traditionen, och därigenom "... deltar jaget i den bild och de redskap som flätats in i den tradition, det samhälle och den kultur jaget tillhör" (Rolf i Johannessen & Rolf, 1990, s. 28). Det som språket förmedlar är erfarenheter, attityder, värderingar, med mera, som generationer har ackumulerat. Dock sker inte detta överförande som en kopiering, utan individen påverkar och modifierar det traderade kulturarvet och en förnyelse av språket kan äga rum. Språket blir därmed "... ett slags mötesplats mellan tradition och individuell erfarenhet" (ibid., s. 29).

Viktigt är dock att språket inte antar en fasadliknande funktion, såsom professionella språk ibland kan göra. Språkets uppgift är inte att upprätthålla en status eller självbild hos professionen, utan det skall tjäna som ett verktyg i en reflektiv och social kunskapsprocess (ibid.)

### **3.2.6 Sammanfattning Rolf**

Enligt Rolfs tolkning av Polanyis syn har kunskapen alltså en tyst funktion som tjänar som en sorts bakgrundskunskap för det vi fokuserar på. Denna kunskap är personlig såtillvida att den integrerar

den subjektiva erfarenheten med tradition och kultur, vilka förmedlar teorier, värderingar, tumregler, med mera. Den personliga kunskapen är kopplad till sociala identiteter som utmärker olika yrkesgrupper.

Praktisk kunskap är handlingsinriktad kunskap som bygger på tyst och personlig kunskap och innebär att kunna utföra goda handlingar. För att kunna bedöma kvaliteten på handlingar behövs bedömningskriterier och regelföljande som bygger på tyst fungerande regler. Regler är emellertid inte själva tillräckliga för att avgöra hur man praktiserar en färdighet – de kan inte ersätta praktisk kunskap.

Den praktiska kunskapen kan enligt Rolf (ibid.) vara av tre slag, som skiljer sig i hur kvalitetsbedömningen kan ske och i hur regler kan påverkas. Kompetens är praktisk kunskap som bygger på att kvalitetsbedömningen måste ske i ett socialt sammanhang, samt att utövaren av praktisk kunskap genom medveten reflektion kan påverka bedömningskriterier eller regler för att göra dessa bättre.

Reflektion är därmed ett viktigt inslag i praktisk kunskap och den kräver att kunskapen artikuleras och symboliseras så att den ställs i fokus och därigenom kan spridas och kritiseras. Språket tjänar som ett redskap för detta – "... the denotative use of language, as a kind of verbal pointing." (Polanyi, 1966, s. 7). Den tysta kunskapen blir genom språket reflekterad kunskap och därmed kan kunskapen utvidgas. Kunskap är därför inte en statisk egenskap utan snarare en social process eller aktivitet, vilket gör begreppet *kunnande* mycket intressant.

### **3.3 Kunskap och kunnande enligt Johannessen och Rolf**

Det som vi anser framstår som tydligt, är att Johannessen och Rolf tar olika vägar in kunskapsproblematiken, men att de ändå når likartade målsättningar och slutsatser. En avgörande skillnad i utgångspunkterna är att Johannessen utgår från Wittgenstein, som ju var språkfilosof i grunden, och därmed startar med en begrepps-



analys som leder fram till en praxissyn på kunskap. Rolf, som utgår från Polanyi, går snarare in på vad kunnande innebär än på vad begreppet betyder, men slutar även han i en praxissyn på kunskap och kunnande. Skilda angreppssätt leder alltså till likartade slutsatser.

Båda gör ett tydligt avståndstagande från den logiska positivismens korrespondenssyn genom att förlägga stora delar av kunskap och kunnande i utövandet av handlingar och i en social kontext. Därmed kan det inte finnas en objektiv och given verklighet som kunskapen kan spegla, utan det blir snarare en fråga om intersubjektiva tolkningar och insikter. Mycket av det kunnande som vi har, kan inte verbaliseras i form av det som Johannessen kallar påstående-kunskap, d.v.s. verbalspråkliga omsättningar av kunnandet i en begreppsapparat. Det finns kunskaper som är tysta till sitt innehåll.

Det sistnämnda är en skillnad mot Rolf, som anser att vi förvisso inte kan verbalisera all vår kunskap, men att den måste kunna symboliseras för att kunna spridas och förbättras socialt. För Rolf är det inte kunskapens innehåll som är tyst, utan dess funktion. Kunskapen fungerar tyst när den ligger i bakgrunden för vårt handlande.

Här är det alltså öppet för tolkning av vad de båda tänkarna egentligen menar och tolkningssvårigheten kommer mycket från de skilda angreppssätten, där det ena alltså är en mer begreppsmässig analys och det andra närmast en processorienterad. För Johannessen kan inte all kunskap verbaliseras, för Rolf måste kunskap symboliseras för att kunna utvecklas socialt. Frågan är alltså om verbalisering och symbolisering betyder samma sak.

Om man tolkar ordet symbolisering som att på något sätt manifesteras kunskapen och göra den synlig, så är nog skillnaden i synsätt mellan Johannessen och Rolf egentligen inte så stor. För Johannessen har exemplifiering stor betydelse för att kunna kommunicera kring kunskap och då tjänar exemplet som en sorts symbolisering. Å andra sidan så framstår symboliseringen både hos Rolf och Polanyi som beroende av ett språk av mer verbal karaktär, alltså det

som Johannessen betecknar som påståendekunskap. Så, vad är det då som behövs för att kunnandet skall kunna utvecklas i en social kontext och kunna förbättras?

Slutsatsen vi drar är att kunskapen och kunnandet, när det gäller avancerad praxiskunskap, behöver medvetandegöras och synliggöras för att kunna utvecklas. Eftersom den praktiska kunskapen, vare sig denna kallas färdighet eller kompetens, visar sig i skickligt utförda handlingar som i sin tur vilar på egna erfarenheter i kombination med andras förmedlade erfarenheter, är erfarenhetsutbytet centralt. Med grund i Johannessen och Rolf drar vi slutsatsen att språklig verbalisering eller att använda exempel var för sig inte räcker, utan att dessa båda sätt att synliggöra kunnande behövs för att utveckla kunnandet. Exempel utan språkapparat eller en språkapparat utan koppling till kontextuella förhållanden, räcker inte för att nå en reflekterad kunskap.

Den slutsats vi då kan dra är att kunskapens tysta funktion hos Rolf i någon mening motsvarar Johannessens färdighetskunskap och den personliga kunskapen förtrogenhetskunskap. När det gäller professionell kunskap, blir förtrogenhetskunskapen central för know-how och kompetens, eftersom maximer inte ensamma kan förmedla hur en professionell handling skall utföras. För att ta exemplet med ljudet av en klarinett är det alltså en färdighet att känna igen att det är en klarinett som låter. Detta igenkännande har då en tyst funktion och man skulle kunna fokusera på denna tyst fungerande kunskap, kanske genom ett exempel, för att det kunnandet skall bli mer reflekterat. Att spela klarinett professionellt i en symfoniorkester kräver dock mer än att kunna känna igen en klarinett på ljudet och bedömningen om ens insatser i orkestern är goda eller inte, och hur de i såfall kan förbättras, bygger på att bedömningen inte endast utförs av den som spelar instrumentet, utan att det sker socialt. En kommunikation kring detta blir alltså nödvändig.

Till grund för färdighetskunskapen ligger förtrogenhetskunskap, det vill säga en erfarenhet av vad kunskapen är uttryck för. När det gäller ljudet av en klarinett är erfarenheten att en klarinett har en viss klang. Detta kopplat till nödvändigheten att symbolisera

klangen genom ett exempel, att visa att en klarinett låter si men inte så, ligger nära Polanyis personliga kunskap, som är en sammanmältning av egna erfarenheter och traditionen. Att en klarinett har en viss klang hänger ihop med att andra kallar detta instrument för klarinett. Att en klarinett låter på ett visst sätt är alltså också förmedlat av traditionen. Därför kan inte heller en färdighet utvecklas i ett socialt vakuum, särskilt inte när det gäller professionell färdighet eller kompetens

För oss ligger det därmed nära till hands att dra slutsatsen att Johannessens olika kunskapsaspekter och Rolf praktiska kunskap, egentligen handlar om samma sak, nämligen personlig erfarenhet parad med andras erfarenheter. Därmed blir olika former av utbyte av kunskap och erfarenheter intressant, alltså den process som behövs för att nå reflekterad kunskap och därmed möjlighet att förbättra kunskapen.

Vi kommer då in på Johannessens transitiva och intransitiva artikuleringar, där det transitiva behövs för att man skall kunna förhålla sig själv till någon annans erfarenhet och kunskap, eftersom dessa då i någon mening blir överförda. Men för att det överförda skall uppfattas, krävs det att mottagaren också delar erfarenheten, varvid vi är tillbaka i erfarenhetens betydelse. På samma sätt är det med regelföljandet hos Rolf, där manifesta regler aldrig kan ersätta den kunskap de uttrycker. Regler kan endast brukas tillsammans med en praktisk kunskap, vilket kräver erfarenhet av kunskapsområdet. Så regler i form av maximer blir intransitiva för den som inte har den praktiska kunskapen som krävs för att bruka dem; vi skulle också kunna säga att intransitiva regler inte tar sin utgångspunkt i en nödvändig förtrogenhetskunskap.

För att avsluta den här jämförelsen, så är skillnaden att Johannessen diskuterar vilken kunskap som helst, medan Rolf ägnar sig åt professionell kunskap. Resultatet blir ändå att kunskap är socialt betingad och inte möjlig att strikt formalisera och att den bygger på erfarenhet och utbyte med andra. Därför blir inte heller kunskapen något statiskt som kan formuleras i intransitiva regler, utan den är föränderlig och socialt beroende. Detta i sin tur förutsätter att kun-

skapen kan bli reflekterad för att kunna ingå i en social process för att förbättra kunskapen. Därtill krävs någon form av ”språk” som bygger på såväl exemplifiering och liknelser som på begrepp och språkapparat; ibland kanske av en mer verbal karaktär och ibland av en mer icke-verbal.

Frågan är då för vår del hur dessa tankar om kunskap och kompetens kan användas för att fördjupa förståelsen för hur systemutvecklare ser på IT-kvalitet och förmågan att bedöma den samma. Den uppdelning i olika kunskapsformer som Johannessen utvecklar, är för oss ”bekväm” och lämplig för en kategorisering av uppfattningar kring fenomenet IT-kvalitet och de kunskaper som systemutvecklare uttrycker kring det. Vi har därför valt att utgå från dessa kunskapsformer i vår analys av empirin, vilket vi beskriver i nästa kapitel (kapitel 4). Johannessens resonemang lämpar sig också för en diskussion kring begreppet IT-kvalitet som det gestaltar sig i vårt empiriska material, samtidigt som Rolfs syn på kunskap och kompetens ger stöd för att föra in betydelsen av reflektion i syfte att utveckla professionell kunskap i diskussionen. Denna diskussion återfinns i kapitel 5, men först presenteras själva undersökningen.

## 4 Undersökning av systemutvecklares syn på IT-kvalitet

Som vi har visat i kapitel 2 existerar det ett ”hål” i den teoretiska kunskapen om IT-kvalitet, som just har att göra med att praxisdimensionen har bortfallit ur den teoretiska diskussionen i stor utsträckning. Detta till skillnad från övriga designområden, exempelvis arkitektur, där denna del av kunnandet också är väl belyst, vilket vi berörde i kapitel 1. Till viss del finns det en sådan diskussion inom vårt eget ämne (Stolterman, 1991; Hoberg, 1998), men inte med ett direkt fokus på systemutvecklares kvalitetsuppfattning riktad mot produkten och inte heller med det djup som vi eftersträvar. Den enda undersökning vi stött på som berör detta problemområde är som sagt Wilsons & Halls (1998) pilotstudie.

Vad vi vill göra är att undersöka denna aspekt, systemutvecklares kunnande i förhållande till IT-kvalitet, närmare eftersom synen på kvalitet och sätten att hantera detta fenomen, borde ha stor relevans för området. Det är ju trots allt systemutvecklare som, får man förmoda, försöker uppnå kvalitet på denna typ av produkter och som därmed borde syssla med bedömningar av kvaliteten.

För vår del blir undersökningen intressant genom att dels belysa synen på produktkvalitet som ett fenomen och dels att försöka anlägga ett mer praxisbetonat perspektiv som riktar ljuset mot bedömningsförmåga, erfarenhetskunskap, reflektion och kommunikation.

Vårt intresse är därför att försöka fånga vad kvalitet innebär för systemutvecklare, i relation till de kunskapsaspekter eller -former som vi tidigare har diskuterat och som Nordenstam (1983) och Johannessen (1999) har resonerat kring, och i relation till den syn på kunnande och professionalism som Rolf (1995) argumenterar för.

Fokus i den första relationen är därför på kvalitet som ett begrepp och hur det förstås och hanteras av systemutvecklare: Talar sy-

stemutvecklare om kvalitet? Vilka ord använder de för att tala om kvalitet i såfall? När talar de om kvalitet och hur? Vilka betydelser lägger de i begreppet kvalitet?

Fokus i den andra relationen är på kvalitet som ett praktiskt kunskande och därmed systemutvecklares bedömningsförmåga när det gäller kvalitet på IT-artefakter: Hur bedöms kvalitet? Vilken betydelse har erfarenhetskunskap för bedömningsförmågan? Är dialog och erfarenhetsutbyte viktigt när det gäller utvecklingen av bedömningsförmågan?

Innan vi går in på att beskriva den faktiskt utförda undersökningen, skall vi dock först redogöra för var vi stod när vi planerade den, eftersom det har betydelse för att kunna förstå hur vi har gått tillväga och vilka perspektiv som väglett oss.

## **4.1 Utgångspunkter för undersökningen**

När vi planerade vår studie hade vi lite olika idéer som utgångspunkter, bland annat när det gäller hur vi såg på systemutvecklare som antingen konsulter eller utvecklare på en intern IT-avdelning. Vi hade också en syn på systemutvecklare som professionella praktiker som är medvetna om och strävar efter kvalitet i sitt arbete.

### *4.1.1 Vad är systemutvecklare i vår undersökning?*

Systemutvecklare som professionella praktiker ville vi fokusera på, eftersom vi ville försöka nå kunskap om hur de ser på kvalitet och värdering. Att tänka sig systemutvecklare som professionella praktiker är kanske en för idealistisk inställning om man tar sin utgångspunkt i exempelvis läkaryrket som en professionell praktik, som har en oerhört mycket längre historia och som har en gemensam värdegrund och etiskt förhållningssätt, nämligen att bota sjuka och ge omvårdnad. Även om inte alla läkare lever upp till denna bild, så är det en värdegrund i själva yrket som kan spåras tillbaka till åtminstone Hippokrates.

En lika inarbetad grund står nog inte att finna inom systemutveckling, även om exempelvis ACM<sup>38</sup> utarbetat etiska normer för att ge systemutveckling en mer professionell status (Andersen et al., 1993). Däremot anser vi att man måste ha någon sorts yrkesmässig och seriös inställning till sitt arbete för att det skall bli meningsfullt att tala om kvalitetstänkande och värdering. Så även om systemutvecklare inte har några organ som reglerar efterlevnaden av normer såsom advokater eller har en tradition och kultur kring sitt etiska förhållningssätt såsom läkarna, vore det fel om inte utgångspunkten var att systemutvecklare överlag har något sorts professionellt och seriöst förhållande till sitt yrke. Det är den utgångspunkt som vi har i denna avhandling och som gör att det blir intressant att undersöka hur systemutvecklare ser på och bedömer kvalitet på de produkter som de tillverkar.

Bortsett från vår inställning till systemutvecklare som praktiker, svävar vi en aning på målet om vad vi egentligen avser med systemutvecklare över huvud taget. Återigen saknar vi någon sorts bearbetning av vilka roller man kan förvänta sig att finna inom systemutveckling. Det vi kan finna i grundböcker inom ämnet (se exempelvis Lunell, 1985; Capron, 1986; Andersen, 1991) handlar snarare om aktiviteter inom systemutveckling, t.ex. analys, design, realisering och implementering. I vissa projekt eller vissa organisationer går det säkerligen att finna roller som är bestämt knutna till systemutvecklingens olika delar och att dessa roller nästan kan vara som yrken för de personer som utför aktiviteten, exempelvis en programmerare som endast har till uppgift att programmera, men som i framtiden kanske kan bli någon form av analytiker istället. I andra projekt och organisationer kan systemutvecklarna flyta mellan de olika rollerna, utan att någon speciell person är fast knuten till en bestämd roll.

Detta var den utgångsbild av systemutvecklingspraktiken som vi hade när vi inledde vår studie och den bild som vi hade med oss i våra kontakter med olika företag för att finna personer villiga att

---

<sup>38</sup> Association for Computing Machinery

delta i vår studie. Vi förstod att vår bild var ofullständig och att det säkert kunde finnas t.ex. andra roller i systemutveckling än vad vi kände till och därför ville vi vara så öppna som möjligt i våra kontakter med företagen och våra undersökningspersoner, så att vi inte genom någon sorts felaktig eller bristande förförståelse skulle gå miste om intressanta infallsvinklar och data. En bestämd avgränsning var emellertid att vi ville ha kontakt med företag och personer som sysslade med utveckling av administrativa system, eftersom det är den typen av system och utveckling som vi känner bäst till, både genom den utbildning vi själva har genomgått och de kurser som vi har undervisat på.

#### *4.1.2 Systemutvecklaren ur två perspektiv*

När det gäller systemutvecklarens möjliga roller ansåg vi att det skulle vara intressant med en blandning av konsulter och "in-house", d.v.s. systemutvecklare som arbetar på en intern IT-avdelning eller motsvarande i en organisation. Vår idé var att konsultföretag i större utsträckning måste sälja in sig olika organisationer och därmed har mindre möjlighet att skapa en gemensam, grundläggande syn på kvalitet än vad som är fallet på en intern avdelning inom ett företag. Vi hade en känsla av att det skulle råda en "kunden har alltid rätt"-inställning i ett konsultföretag vilket skulle påverka systemutvecklingskonsulter i stor utsträckning. Om kunden alltid har rätt, så finns ju en risk att spelrummet att ha en viss inställning till kvalitet minskar, att det istället blir kundens kvalitetsuppfattning som kommer att avgöra och att konsulten, som ju är kundstyrd, får mindre inflytande på vad som kan menas med kvalitet.

En annan aspekt är att det är osäkert vilken kontakt en konsult eller en grupp av konsulter kan ha med de tänkta användarna, eftersom det antagligen ofta är kunden/beställaren som avgör hur den kontakten skall se ut. För att utveckla en syn på brukskvalitet är det ju nödvändigt att konfronteras med brukares perspektiv på kvalitet, även om en viss syn på kvalitet-i-bruk naturligtvis också kommer ur konsultens eget bruk av datorprodukter. Säkerligen skiljer sig



denna från den syn de så kallade slutanvändarna har, eftersom konsulten dels är expertanvändare generellt sett, dels förmodligen använder andra typer av program, dels inte är expert på det faktiska, dagliga arbetet inom problemområdet som hon eller han utvecklar program till, vilket gör bruksegenskaperna till en designfråga och inte en användningsfråga.

I förhållande till detta utgick vi från tanken att ett företags interna IT-avdelning har en större chans att skapa en kontinuitet i kontakterna med kunder och att en sådan intern avdelning har ett mindre behov av att "sälja" sig likt ett utomstående konsultföretag. Därmed, ansåg vi, har den en större möjlighet att påverka synen på kvalitet, att ha en pågående dialog med användare, att ha mer tid till reflektion etc.

Så här klara skillnader finns förmodligen inte, utan det varierar från fall till fall och från företag till företag. Dessutom är den interna IT-avdelningen mer lik ett konsultföretag nu än för 10-20 år sedan, genom att vara mer fristående, kanske bolagiserad, och avsedd att konkurrera med andra företag som säljer systemutveckling. Detta är emellertid också spekulativt, eller kanske snarare byggt på allmänkännedom om området, eftersom det med vår erfarenhet endast finns lite litteratur som diskuterar systemutveckling ur ett yrkesperspektiv<sup>39</sup> och som tar upp dessa aspekter.

---

<sup>39</sup> En intressant bok som diskuterar systemutvecklingsyrket ur ett praxisperspektiv är Hoberg (1998) som bygger på ett antal seminarier inom serien Dialoger som drivs av bland andra Bo Göranson. Boken fokuserar på betydelsen av reflektion, språk och exempel som delar i en handlingskunskap inom systemutveckling och är således inspirerad av Wittgensteins senfilosofi och idéer om kunskap och kunnande som uttrycks av Johannessen, Molander och bröderna Dreyfus. Systemutvecklingspraktiken ges här ett delvis annat innehåll än den klassiskt ingenjörsmässiga strukturerade och systematiska processen och för fram betydelsen av erfarenhetskunskap, helhetssyn och kreativitet. En bild av organisationskulturens betydelse för samarbete och identifikation inom systemutveckling i ett konsultföretag, där ett en del av detta sålts eller "outsourcets" till ett kulturellt helt annat företag som det samtidigt skall samarbete med, ges i Dubé & Robey (1999), där de finner att normer, värderingar och arbetssätt är viktiga kulturella inslag även inom systemutveckling.

Dock tror vi ändå att det överlag finns vissa skillnader mellan konsultföretag och vad vi hittills kallat den interna IT-avdelningen, som kanske snarare är ett fristående dataföretag med kulturella och historiska kopplingar till den organisation den tidigare ingick i och som nu är dess största kund. Dessa skillnader tror vi består i ett större åtagande, d.v.s. flera system och projekt för en och samma kund/beställare, tätare samarbete och kontakt mellan systemutvecklare och kunder/användare samt större möjlighet för IT-avdelningen att påverka beställaren med avseende på kvalitet och värderingar.

Genom att försöka få med både "in-house"-utvecklare och konsulter, vill vi skapa en bredd i urvalet – att populationen uppvisar en troligare variation. Vi kommer dock inte att undersöka huruvida dessa eventuella skillnader påverkar synen på kvalitet.

## **4.2 Utgångspunkter för metodval**

Som vi har redovisat ovan är vårt undersökningsintresse riktat mot att nå kunskap om vilken förståelse systemutvecklare har av fenomenet kvalitet på de produkter som systemutveckling resulterar i, d.v.s. datasystem och dataprogram eller, med viss avgränsning, IT-artefakt<sup>40</sup>. Det handlar alltså om att söka och undersöka de innebörder som systemutvecklare lägger i produktkvalitet och de uppfattningar de har om detta fenomen, vilket leder till en tolkande undersökning i Walléns termer (Wallén, 1996). Det kan föra för långt att påstå att det handlar om en undersökning av systemutvecklarnas livsvärld i den bemärkelse som exempelvis Kvale (1997) talar om i samband med kvalitativa intervjuer eller det som Giorgi (1994) kallar för "lived meanings" och som han menar är målet för kvalitativa studier. För oss ger detta en sociologisk vinkel som vi inte tycker att vi undersöker. Emellertid ger också Giorgi uttryck

---

<sup>40</sup> Termen IT-artefakt leder egentligen längre än vad vi avser med exempelvis datasystem, nämligen till vilken konstgjord tingest (alltså artefakt) som helst som har med informationsteknik att göra. En mobiltelefon är också ett exempel på en IT-artefakt, men ligger utanför vad vi är intresserade av.

för att det handlar om att skapa "quality knowledge" och Kvale skriver om kvalitativa intervjuer som ett sätt att försöka förstå världen ur de intervjuades synvinkel och att höra deras uppfattningar, att utveckla innebörden av deras erfarenheter.

Vårt syfte är just att nå "quality knowledge" kring systemutvecklares erfarenheter och förståelse av fenomenet kvalitet, att finna viktiga reflektioner kring kvalitet rika på innebörd och att finna något meningsfullt kring kvalitetsbegreppets betydelser för systemutvecklare. Vi är inte intresserade av att ur ett stickprov dra representativa slutsatser om en population, att göra statistiska generaliseringar eller testa hypoteser mellan grupper, vilket kan vara syftet med en kvantitativ undersökning (Kvale, 1997). Det står därmed klart för oss att vår studie kräver en kvalitativ undersökning av en mer explorativ och tolkande karaktär.

Metodologiskt finns det en mängd olika sätt att genomföra en kvalitativ studie, exempelvis genom olika typer av observationer, intervjuer, dagboksförande, med mera och kombinationer av detta (Patel & Davidson, 1994; Wallén, 1996). En speciell och rätt så omfattande metod av induktiv karaktär är Grounded Theory (Strauss & Corbin, 1990), där teoriuppbyggandet legitimeras genom en djup "rotning" i det empiriska materialet genom en mycket systematisk datainsamling och -analys – att så att säga låta data "tala" så väl att en teori kan syntetiseras utifrån dem. Detta är en intressant men som sagt också omfattande metod eller snarare ansats som kräver mycket av dem som skall använda den, både vad det gäller kunskap och vad det gäller erfarenhet. Syftet är som sagt att induktivt generera en teori kring ett fenomen i verkligheten och inte endast en beskrivning, genom att bygga upp begrepp som grupperas och ges begreppsmässiga beteckningar och som relateras till varandra genom olika samband. Att använda Grounded Theory hade för vår del kunnat vara ett alternativ, men betoningen på teoribyggnad och dess komplexitet gör att den känns för ambitiös i förhållande till både våra förutsättningar och syften. De metoder som vi ansåg vara lämpliga var dels intervjuer och dels någon form av observation, utan den omfattande ansats som Grounded Theory innebär.

#### *4.2.1 Observation som undersökningsmetod*

Enligt Patel & Davidson (1994) är observationer framför allt användbara när man vill studera beteenden och skeenden, t.ex. fysiska handlingar och relationer mellan individer, i naturliga miljöer i samma stund som de inträffar. Man blir därmed inte lika beroende av att de individer man studerar har en tydlig minnesbild, vilken de dessutom skall kunna förmedla så att den uppfattas rätt. Nackdelar finns naturligtvis också, inte minst genom att observationer i allmänhet är både tidsödande och dyra. Det är inte heller så att observationer alltid lämpar sig som metod, eftersom det man studerar kanske faktiskt inte kan observeras, som exempelvis familjebråk, eller att de individer man studerar saknar förmåga till verbal kommunikation. Ytterligare frågetecken när det gäller observationer handlar om observationsmetoden som teknik, eftersom det kan vara svårt att veta t.ex. att en visst observerat beteende är representativt och att också spontana beteenden observeras (Patel & Davidson, 1994).

I princip kan man skilja mellan två olika typer av observationer – strukturerad/systematisk eller ostrukturerad/osystematisk (Patel & Davidson, 1994). I det första fallet, en strukturerad observation, är undersökningsproblemet så väl preciserat att frågan om vilka beteenden som skall observeras och i vilka situationer, i stort sett är besvarad och utredd. Kategorier kan då ställas upp i ett observations-schema som täcker in det man avser att undersöka. En ostrukturerad observation är i allmänhet av en mer utforskande karaktär, genom att man vill kunna inhämta så mycket data som möjligt om ett visst problemområde. Även om uppmärksamheten här är mindre riktad än vid strukturerade observationer, måste även en ostrukturerad observation förberedas genom att teoretiska och empiriska kunskaper styr vad och vilka som skall observeras och hur det skall göras, samt att man måste veta vilken övrig data som är viktig att registrera (Patel & Davidson, 1994).

Den observation som Patel & Davidson (1994) talar om här verkar vara av den sort som vi kallar tyst observation, d.v.s. att man inte interagerar alls i situationen utan just fungerar som observatör.

Man kan också vara mer aktiv och interagerande i observationen och det blir då frågan om så kallad deltagande observation, vilket gränsar till aktionsforskning. Enligt Wallén (1996) är den stora fördelen att man får ”inifrånkunskap”, d.v.s. kännedom om t.ex. sociala samspel, värderingar, erfarenheter och tyst kunskap. Två av studierna inom projektet om paradigmatiske form och funktion (Ehn, 1994), vilket presenteras i avsnitt 7.1, använde deltagande observation framgångsrikt (Eriksson, 1995; Svedemar, 1995b, 1995a). I dessa studier gjordes uppföljande intervjuer, vilket är vanligt vid observationer.

I vår studie av systemutvecklarna hade vi från början ambitionen att genomföra tysta och ostrukturerade observationer som skulle kompletteras med intervjuer, eftersom vi ansåg att många viktiga aspekter på kvalitet ur vårt fokus antagligen framkommer när praktiserande systemutvecklare arbetar med den konkreta situationen att utveckla system i olika stadier. Vi ville därför ha möjligheten att delta vid ett antal projektmöten för systemutvecklingsprojekt, för att som tysta observatörer undersöka hur kvalitet kommer in, uttrycks och hanteras i kontexten av ett faktiskt projekt. På så vis skulle vi kunna se vilka kvalitetsaspekter som kommer upp vid vilket tillfälle i samband med vilken typ av förhållanden eller ”problem” i systemutvecklingsprocessen.

Även om observationen skulle ske i processen att utveckla system, är det produkten av systemutveckling som vi är intresserade av i vår undersökning och fenomenet kvalitet som produktkvalitet. Genom observationer såg vi möjligheten att finna intressanta aspekter som vi skulle kunna belysa genom intervjuer med de observerade systemutvecklarna<sup>41</sup>.

---

<sup>41</sup> Det framstår som naturligt att man kan följa upp en observation med en intervju, men Wallén (1996) redovisar en erfarenhet från ett aktionsforskningsprojekt där detta inte gick så bra, eftersom det inte gav någon ny kunskap och de intervjuade kände sig lätt besvärade av att försöka upprepa det de sagt samtidigt som formuleringar och innehåll föddes i den tidigare diskussionen.

#### 4.2.2 *Intervjuer som undersökningsmetod*

När det gäller intervjuer kan man grovt skilja mellan strukturerade/ostrukturerade och standardiserade/ostandardiserade avseende frågeutrymmet, d.v.s. intervjuarens frihet, och svarsutrymmet, d.v.s. respondentens frihet (Patel & Davidson, 1994). Vid en standardiserad intervju är frågeutformning och frågeordning fastställd och skall följas i varje intervju; vid en ostandardiserad formuleras frågorna av intervjuaren vid intervjutillfället och intervjusituationen styr med andra ord hur dessa skall formuleras. Analogt med detta innebär en strukturerad intervju att de möjliga svaren är formulerade och att deras inbördes ordning är fastställd, respondentens svarsalternativ är med andra ord styrda, medan en ostrukturerad intervju lämnar helt fritt till respondenten att själv formulera sina svar. En enkät med endast slutna svarsalternativ är därmed en såväl standardiserad som strukturerad intervju.

Enligt Kvale (1997) kan huvudsyftet med en intervjuundersökning vara antingen empiriskt eller teoretiskt. Med ett empiriskt huvudsyfte är avsikten att genom undersökningen samla in empirisk information, medan ett teoretiskt huvudsyfte leder till ett intresse för att pröva konsekvenserna av en teori. Vilket syftet är skall klargöras vid tematiseringen där frågan om undersökningens ”vad”, ”varför” och ”hur” skall behandlas:

- *vad*: att skaffa sig förkunskap om det ämne som skall undersökas
- *varför*: att klargöra syftet med undersökningen
- *hur*: att förvärva kunskap om olika intervju- och analystekniker, och att avgöra vilka som är lämpligast i detta sammanhang. (ibid., s. 91)

En intervjus syfte kan vara explorativt eller hypotesprövande. Med ett explorativt syfte introducerar intervjuaren en fråga eller ett problemområde för en intervjuperson och genom att följa upp dennes svar söker intervjuaren ny information och nya infallsvinklar. Med

ett hypotesprövande syfte är intervjuarens intresse riktat mot att finna hur väl en hypotes håller i förhållande till de svar som undersökningspersonerna avger.

Ett annat perspektiv på intervjuundersökningens eller den enskilda intervjuens syfte är om den är till för att undersöka, förändra eller både och. Med ett undersökande syfte är målet att få reda på något väsentligt om det som man undersöker, man vill finna empirisk information som man kan tolka och bygga ny kunskap genom. Att undersökningspersonens självinsikt och förståelse av sin situation förändras är inte ett syfte i sig eller av underordnad betydelse. I ett terapeutiskt samtal, som Kvale (1997) också betraktar som ett form av professionell intervju, är däremot det direkta syftet att genom samtalen få den intervjuade, klienten, att komma till ny insikt om och förståelse för sin egen situation och därmed göra det möjligt för henne eller honom att förändra sin livssituation. Kunskapsbildningen kan här vara av underordnad betydelse, men exempelvis Freuds terapisaftal har lagt grunden för psykoanalysens utveckling, även om dessa har haft det primära syftet att "hjälpa" analysanden.

När tematiseringen är gjord övergår man till planeringen av undersökningen, där man gör en allmän planering och går igenom hur man metodiskt skall gå tillväga i sin undersökning. Viktiga aspekter att tänka på här är tidsdimensionen, intervjuformer och antalet undersökningspersoner som bör delta.

När det gäller tidsdimensionen så skriver Kvale (ibid.) att det är viktigt att skaffa sig överblick över hela undersökningen, eftersom vilka metoder för datainsamling och analys som väljs, påverkar hur undersökningen skall genomföras. Det finns också starka samband mellan metodval som görs på de olika stadierna av en undersökning, där ett beslut på ett stadium påverkar de tillgängliga alternativen på nästa stadium, t.ex. att generalisering till större grupper medför att vissa kriterier om bl. a. storlek och representativitet i urvalet måste ha uppfyllts redan i planeringen. Man skall också från början ha undersökningens ändpunkt i sikte när det gäller undersökningens syfte och slutprodukt, samt den publik man vänder sig

till. Ett annat råd är att skjuta framåt, nämligen att redan från början tänka på de senare stadierna exempelvis genom att ställa kontrollfrågor under intervjun för att underlätta validering, eftersom kvaliteten på intervjuer påverkas av en genomtänkt föreställning om syfte och ämne. Slutligen blir också intervjuaren klokare under undersökningens gång och lär sig mer om de undersökta fenomenen, vilket är en fördel vid en explorativ studie men som kan vara ett problem vid hypotesprövande undersökningar, om det medför att intervjuguiden behöver ändras och därmed kanske medföra ojäm-förbara grupper.

Hur många intervjupersoner som behövs för att göra en undersökning är naturligtvis flytande. Kvales något lakoniska förslag för intervjuundersökningar är ”så många som behövs” men han skriver också att i vanliga intervjustudier ligger antalet intervjuer runt  $15 \pm 10$  (ibid.) I ett par andra avhandlingar inom vårt område där kvalitativa undersökningarna också gjorts med intervjuer, är antalet undersökningsspersoner 20 i ena fallet (Stolterman, 1991a) och 28 i andra fallet (Ingman, 1997). Att ligga mellan 20 och 30 intervjuer av djupare karaktär är väl också ungefär vad som kan klaras av inom rimlig tid och med de små resurser som brukar stå tillbuds<sup>42</sup>. Även detta till synes lilla antal personer kommer att ge upphov till hundratals sidor intervjutranskript som skall analyseras. Ett för stort antal intervjuer leder enligt Kvale (1997) till svårigheter att göra mer ingående tolkningar.

Vilka intervjuformer som finns skriver Kvale (ibid.) helt kort om. Sammanfattningsvis nämner han enskilda intervjuer och gruppintervjuer. Vid gruppintervjuer intervjuas ett antal personer samtidigt, vilket ofta ger upphov till spontana och känsloladdade uttalanden om ämnet för intervjun, men som samtidigt kan leda till en förhållandevis kaotisk datainsamling och svårigheter till systematisk analys. Den enskilda intervjun sker i motsats till detta genom att en person intervjuas av intervjuaren enskilt.

---

<sup>42</sup> Förmodligen är det t.ex. rätt sällan som doktorander har råd att finansiera intervjuutskriften via skrivbyrå och att skriva ut intervjuer är en aktivitet som tar lång tid.



### **4.3 Val av metod och undersökningens genomförande**

Som framgår av diskussionen ovan om olika kvalitativa metoder, är vår undersökning av empirisk och explorativ karaktär fast med vissa hypotesprövande nyanser eftersom undersökningen delvis är teoriladdad<sup>43</sup>. Vårt syfte är att undersöka något väsentligt snarare än att förändra respondentens självinsikt, även om vi förmodligen gjort vissa avtryck genom undersöka vissa aspekter på respondenternas kunskap och kompetens. Undersökningen är också kvalitativ eftersom det är innebörder och betydelser vi vill studera. Valet för vår del föll därför på dels de tidigare diskuterade tysta observationerna och dels på kvalitativa intervjuer. Vad som återstår är då att hitta lämpliga ”objekt” och ”subjekt” att studera, i vårt fall företag i de tidigare diskuterade kategorierna.

#### *4.3.1 Problematiskt att finna företag*

För att genomföra datainsamlingen tog vi under vintern 1998/1999 och våren 1999 kontakt med ett antal företag som vi ansåg kunde vara intressanta ur vår studies synvinkel. I första hand valdes dessa företag ut genom att vi utgick från de företag som institutionen har kontakt med genom praktikperioden på magisterprogrammet, men vi tog också kontakt med företag under de arbetsmarknadsdagar som våra studenter arrangerar<sup>44</sup>. Vi har naturligtvis också använt det vanliga sättet att helt enkelt slå i Gula Sidorna och ringa upp fö-

---

<sup>43</sup> Exempelvis gäller det frågor om systemutvecklarna upplever att de har ett speciellt språk för att tala om kvalitet och uppfattningar i förhållande till de produkter de tillverkar, vilket Stolterman (1991) inte tycker att de har, åtminstone inte när det gäller estetik.

<sup>44</sup> Studenterna på institutionen för Informatik vid Lunds Universitet, anordnar varje år något som kallas Samday – Systemvetarnas arbetsmarknadsdagar – då företag bjuds in att presentera sig själva och rekrytera personal bland studenterna. Det är alltså som någon slags jobbmässa, fast på studenternas initiativ. Den har anordnats under ett antal års tid och har varit både lyckad och trevlig, och för doktorander som vill skapa samarbete med företag, erbjuder den många lämpliga tillfällen att ta kontakt och samtala kring tänkbara projekt.

retag och presentera oss och våra idéer, och därigenom hoppats på att få ”napp” (se bilaga 1)<sup>45</sup>.

För IT-branschen var det gyllene tider under senare hälften av 1990-talet. Bland annat 2000-problemet och Euro-introduktionen skapade mängder av nya arbetstillfällen och branschen växte kraftigt även i övrigt efter en svår period i början av 1990-talet. Alla konsultföretag har haft fullt upp och även IT-avdelningar har varit strängt upptagna. Detta får naturligtvis konsekvenser för doktorander/forskare som vill etablera ett samarbete, åtminstone om undersökningen har en mer grundforskande karaktär. Det visade sig också vara svårt att verkligen få företag att ställa upp, att avsätta tid för att delta i vår undersökning, även om i stort sett alla uttryckte intresse för vår studie. Inget av de företag som vi tog kontakt med och skickade vårt presentationsmaterial till tog någon som helst notis om att vi gärna ville genomföra tysta observationer och när vi frågade framgick det underförstått, att det knappast var aktuellt. Möjligtvis hänger detta samman med att vi inte var tillräckligt påstridiga, men till vårt försvar skall sägas att det var påstridigt nog att bara avkräva dem ett svar om de var villiga att låta oss intervjua ett antal systemutvecklare, inte för att de i allmänhet var ointresserade eller otrevliga, utan för att de var så upptagna<sup>46</sup>.

Vår idé om deltagande observationer som en av våra undersökningsmetoder blev det alltså intet av, utan vi fick koncentrera oss på intervjuer som vår metod att undersöka systemutvecklarnas kvalitetssyn. Det kan tyckas onödigt att, givet den här utvecklingen, diskutera observationer över huvud taget som vi har gjort i avsnitt 4.2.1 ”Observation som undersökningsmetod”, men eftersom vi

---

<sup>45</sup> Det framgår också av bilaga 1 att vi även ville undersöka brukares kvalitetsyn och hur denna förhöll sig till systemutvecklarnas, men det visade sig än mer problematiskt att få kontakt med brukare av de produkter som ”våra” systemutvecklare utvecklat. Detta hade varit en intressant vinkling på undersökningen som vi tyvärr var tvungna att lyfta ut. Vi beslutade oss, vilket framgår, för att koncentrera oss på systemutvecklarna.

<sup>46</sup> Det förhåller sig helt enkelt så, att vi var så glada för att vi över huvud taget fick möjlighet att göra intervjuer att vi inte ville riskera något genom att vara påstridiga.

ansåg att detta skulle ha varit en nyttig metod och det var vår verkliga tanke att genomföra den typen av undersökning, inte minst eftersom det fungerat bra tidigare (Eriksson, 1995; Svedemar, 1995b, 1995a), kan vi inte avfärda det rakt av. Men, tyvärr så blev det alltså inte aktuellt med någon tyst, ostrukturerad observation och vi fick därmed förlita oss helt på intervjun som undersökningsmetod.

Detta kan ha inneburit en i vissa stycken sämre data för oss att arbeta med, då vår tanke med observationen var att det kan finnas intressanta aspekter som visar sig i ett sammanhang av problemlösning för systemutvecklarna, t.ex. under ett projektmöte. Endast intervjuer som instrument innebär att systemutvecklarna talar *om* kvalitet skilt från de situationer där begreppet brukas, snarare än de *talat* kvalitet, vilket leder till att de måste minnas och ta fram kunskaper som troligtvis gör sig bättre bemärkta i praktisk handling. En viss rikhet i data kan alltså ha gått förlorad härigenom. Den data som intervjuerna givit upphov till är dock i sig själv rik.

#### *4.3.2 Datainsamling*

Sammanlagt tog vi kontakt med 19 företag för att göra intervjuer med systemutvecklare, både större konsultföretag, stora företags IT-avdelningar och mindre konsultföretag i Skåne. Slutligen fick vi kontakt med tre företag, ett mycket stort konsultföretag och två före detta interna IT-avdelningar, som numera är två fristående IT-företag men med "moderorganisationen" som största kund. Systemutvecklarna i konsultföretaget arbetade i huvudsak i Göteborg.

På dessa tre företag gjorde vi 19 intervjuer med systemutvecklare i olika roller<sup>47</sup>. Den första intervjun genomfördes i december 1998, åtta intervjuer i slutet av januari 1999 i Göteborg, de övriga i slutet av maj och i början av juni samma år i Malmö och Helsingborg.

---

<sup>47</sup> En intervju genomfördes på engelska. Den transkriberades också på engelska och skickades till respondenten så att denne kunde korrigera den. Innan analysarbetet översatte vi den dock till svenska för att aidentifiera den.

Systemutvecklarna valdes ut genom att våra kontaktpersoner på respektive företag tog kontakt med personer som hade tid och ville delta i studien. Som vi har beskrivit tidigare hade vi inte specificerat vilken typ av roller intervjupersonerna skulle ha i systemutvecklingspraktiken, utan vi lät respektive organisation göra det valet. När våra kontaktpersoner undrade vilka roller vi hade i tankarna, föreslog vi ganska löst roller som vi kunde tänka oss fanns och som var intressanta att studera. Vad vi föreslog var kvalitetsansvarig, projektledare, programmerare och systemerare, men vi sade också att detta var roller som vi kunde föreställa oss, men att det säkert fanns andra roller som kunde vara intressanta som inte vi kände till. Så, fältet lämnades delvis fritt till våra kontaktpersoners egna urval. Urvalet är alltså inte slumpmässigt, utan styrt av respektive organisation och har förmodligen till största delens skett på basis av tillgänglighet. När våra kontaktpersoner hade ordnat intervjupersoner åt oss, tog vi kontakt med dessa och bestämde tidpunkt för intervju.

Av dessa systemutvecklare var 5 kvinnor och 13 män<sup>48</sup>, en övervikt för de manliga systemutvecklarna alltså vilket i viss utsträckning hänger samman med att det på det ena företaget bara var män som intervjuades.<sup>49</sup> De roller som dessa personer innehade var kvalitetsansvarig, projektledare, funktionsdesigner, programmerare, systemerare och delprojektledare.

Intervjuerna har samtliga skett i respondentens egen miljö på arbetsplatsen, vanligtvis i ett mindre konferensrum eller liknande. Den sista omgången intervjuer gjordes emellertid i en ganska bullrig cafémiljö, där folk kom och gick eller satt och fikade. Det var naturligtvis ingen optimal miljö, men det var den enda som stod till buds vid just detta företag. Vi upplever inte att våra frågor har upp-

---

<sup>48</sup> Det summerar ju inte till 19 och förklaringen till det är att en person intervjuades två gånger, eftersom den första intervjun blev för ostandardiserad som vi förklarar längre fram.

<sup>49</sup> Här blev det också lite oklart om alla egentligen sysslade med administrativa datasystem, vilket egentligen spelar mindre roll än vad vi föreställde oss innan.

fattats som särskilt laddade eller känsliga, så att den kollektiva miljön vid dessa intervjuer har försvårat för våra respondenter att reflektera över våra frågor och svara "sanningsenligt" på dessa. Vår inspelningsutrustning var också av så god kvalitet att inte avlyssningen försvårades nämnvärt.

Vid vår första kontakt, där vi egentligen bara skulle lägga grunden till undersökningen på det företaget, gjorde vi en improviserad intervju av det slag som Kvale (1997) skriver mycket om, nämligen samtalet som intervjuform. Vi tyckte intervjun blev för ostrukturerad och svävande, och att vad vi ville fråga om krävde för mycket eftertanke för att respondenten på stående fot skulle kunna reflektera över det på ett meningsfullt sätt. Vi tyckte inte heller att vi var tillräckligt duktiga intervjuare för att vi utan intervjuguide skulle kunna täcka in alla viktiga aspekter. Till stöd för att genomföra de egentliga intervjuerna utarbetade vi därför en intervjuguide (bilaga 2) som inleds med en kort beskrivning av vad vi vill undersöka för att intervjupersonerna skall få en bild av vårt intresse. Själva frågedelen av guiden är indelad i tre delar.

Del 1, uppgifter om respondent och intervju, är av administrativ karaktär för att vi skall kunna hålla reda på vilken intervju det handlar om. Denna del är markerad som "För vår egen del", vilket innebär att inga utomstående skall kunna få tillgång till den informationen, intervjun är alltså konfidentiell.

Del 2, allmänna frågor om respondent, är bakgrundsfrågor som handlar om respondentens utbildning och erfarenhet inom praktiken, samt vilken funktion eller roll hon eller han innehar vid intervjutillfället. Dessa frågor är av social karaktär för att vi skall få en bild av vem det är vi talar med ur ett praktikerperspektiv .

Del 3, teman om bedömningsförmåga och kvalitet, är guidens tyngdpunkt. Det är här som frågorna som berör undersökningssyftet är formulerade. Denna del är indelad i sex teman som alla handlar om kvalitet och bedömningsförmåga. Syftet med denna del är att undersöka både vad begreppet kvalitet och innebörden av begreppet betyder för systemutvecklare. Vi riktar här vårt ljus mot

den enskilde systemutvecklaren och frågor rörande möjligheten att definiera kvalitet, vad hög respektive låg kvalitet på såväl dataprogram som datasystem innebär, huruvida det går att helt förlita sig på mätningar av kvalitet eller om det också krävs någon form av erfarenhetsbaserad, tyst kunskap<sup>50</sup> för att avgöra kvalitet, hur förmågan att bedöma kvalitet påverkas, om och hur systemutvecklare talar med varandra sinsemellan om kvalitet och bedömningar, o.s.v. Det är alltså flera olika aspekter kring kvalitet som vi vill belysa som vi anser har att göra med systemutvecklarnas praktik, vad vi har kallat kvalitet i praxis.

För varje tema tog vi fram ett antal frågor som vi ansåg vara viktiga att belysa och som kunde tjäna som stöd i intervjun. På det viset kan man säga att intervjun fick en rätt tydlig struktur, men vi behöll ändå möjligheterna att ha en mer samtalsliknande intervju genom att frågorna är breda och kan ge upphov till många typer av svar, som i sin tur kan leda till andra frågor kring temat som inte är framtagna från början. Intervjuguiden leder alltså till den vanliga formen av halvstandardiserade och ostrukturerade intervjuer enligt Patels & Davidsons (1994) intervjutypologi.

Inför varje intervjutillfälle var det vår tanke att intervjupersonerna skulle få ut intervjuguiden, fast utan våra delfrågor. De skulle alltså få våra teman att tänka kring på förhand, eftersom vi ansåg att det vi ville undersöka krävde ett visst mått av reflektion, vilket i sin tur kräver tid till eftertanke. Temana är medvetet "luddigt" utformade för att inte för mycket av vår egen förståelse och våra tankar skulle påverka deras reflektion. Vi ville ju också finna spontana tankar kring kvalitet och bedömningsförmåga, men "triggade" av våra teman. Detta hände bara delvis, eftersom inte alla fick våra teman eller hann läsa dem och begrunda<sup>51</sup>. Detta tar vi dock ingen hänsyn

---

<sup>50</sup> Det sätt som vi har använt tyst kunskap på i intervjuerna ligger närmare den betydelse som Johannessen (1999) lägger i begreppet än den som Rolf (1995) lägger. Tyst kunskap har här fått betyda svårartikulerad, erfarenhetsbaserad kunskap.

<sup>51</sup> Fel av oss kanske, men vi överlät ansvaret på att dela ut vårt förberedelsematerial till våra kontaktpersoner, vilket kanske var mindre lyckat med facit i hand.

till i analysen, eftersom vår strävan är att undersöka vad den individuella systemutvecklaren, i sin egen rätt, har för syn på kvalitet och bedömningsförmåga. Vi har ingen ambition att undersöka huruvida en viss typ av förberedelse har samband med vissa svar, vi har inget metaintresse för undersökningen så att säga. Dessutom känns det som om det skulle vara spekulativt att göra den typen kopplingar.

Vid själva intervjutillfället med varje intervjuperson, inledde vi med att prata om syftet med intervjun och vårt avhandlingsintresse. Efter detta vidtog själva intervjun som vi efter godkännande spelade in på band. Varje intervju tog mellan 30 och 90 minuter och följde intervjuguiden i stora drag, ibland som en mer standardiserad intervju och ibland som en mer ostandardiserad.

Vid varje intervju deltog vi båda, den ene som den egentlige intervjuaren och den andre som antecknare av viktiga saker som kom upp under intervjun<sup>52</sup>. I huvudsak sköttes intervjun av den ene, medan den andra kunde inflika frågor eller ställa följdfrågor baserat på de förda anteckningarna. Dessa roller byttes för ungefär varannan intervju så att vi båda skulle vara intervjuledare. Att vi båda deltog under intervjun tror vi upplevdes som laddat av ett par personer, men för de övriga upplevde vi inte att situationen var besvärande, utan att stämningen var avslappnad och obesvärad.

Varje intervju transkriberades sedan tämligen ordagrant<sup>53</sup> och skickades tillbaka till intervjupersonen, som fick möjligheten att göra ändringar för att sedan skicka tillbaka den godkänd och signerad.

Även om själva transkriberingen i sig också är en tolkningsprocess och att mycket försvinner på vägen i övergången från talat språk till skrift på papper (Kvale, 1997), anser vi att en ordagrann transkribering, som alltså också innehåller hummanden, pauser och av-

---

<sup>52</sup> Dessa anteckningar användes endast under själv intervjun och ej som material för analys.

<sup>53</sup> Dock inte i sociolinguistisk mening med mått av pauser i millisekunder.

brott, är önskvärd eftersom man åtminstone i mindre grad ägnar sig åt någon form av dataanalys under själva transkriberingsarbetet. Naturligtvis representerar inte transkriptionerna en ren transformation till skrift och den egentliga empiriska datan finns på banden, men vi bedömer det som mycket svårt att utan lämpliga tekniska hjälpmedel arbeta direkt med ljudbanden. Detta ger även Wallén (1996) uttryck för och anser att det är en fördel att transkribera intervjuer så att utskriften ligger så nära talspråket som möjligt.

### *4.3.3 Analys av intervjuerna*

För att analysera materialet bedömde vi att en systematisk metod med en ganska hög grad av spårbarhet, d.v.s. att man kan härleda tolkningen tillbaka till transkriptionen, förenklar både för oss som skall göra analysen och för den som eventuellt själv vill utreda hur vår analys har gått till och hur vi har kommit fram till de tolkningar vi har gjort. Med ett relativt stort empiriskt material, som 19 transkriberade intervjuer utgör, finns det en fara i att som ganska oerfaren ”gå bort sig” i mängden data och förlora fokus. Vi kan delvis konstatera att tusensidorsfrågan – ”Jag har tusen sidor intervju-transkript, hur skall jag analysera dem?” – som Kvale (1997) skriver om, gäller för oss. Vi säger delvis eftersom vi började med analysarbetet först när alla de cirka 300 sidorna transkript var klara, vilket är en ansats som varken Miles & Huberman (1994) eller Kvale (1997) rekommenderar<sup>54</sup>, men samtidigt har vi haft en ganska utvecklad ram kring de frågor vi har ställt, som en konsekvens av vårt fokus på kunskap och kompetens hos systemutvecklare i förhållande till IT-kvalitet.

Vi valde därför att utgå från de kunskapsaspekter som bland annat Johannessen (1999) diskuterar och som vi har redovisat i avsnitt 3.1. Utifrån begreppen påståendekunskap, färdighetskunskap och förtrogenhetskunskap tog vi fram kategorierna som vi sedan använde som ett ramverk för att koda materialet.

---

<sup>54</sup> Kvale (1997) tycker till och med att den optimala intervjun är tolkad och klar när själva intervjun är avslutad.



### Kodning

Påståendekunskap är sådan kunskap som är möjlig och lättare att artikulera och som kan uppträda i formen av regler, metoder, normer, teorier etc. Således blir det vårt syfte att försöka belysa vilken/vilka teorier, normer och regler som systemutvecklare uttrycker om kvalitet. Vi har valt att dela upp kodningen i två huvudkategorier, begrepp och norm (Figur 4.1).

### Påståendekunskap PÅ

| Huvudkategori | Underkategori    | Kod          | Förklaring  |
|---------------|------------------|--------------|---|
| Begrepp (BEG) | Tekniska (TEK)   | PÅ-BEG-TEK   | Egenskaper vars innebörd är av teknisk karaktär   |
|               | Bruk (BRU)       | PÅ-BEG-BRU   | Egenskaper vars innebörd är orienterad mot bruket |
| Norm (NORM)   | Definition (DEF) | PÅ-NORM-DEF  | Definition som ger begreppet kvalitet en innebörd |
|               | Metod (MET)      | PÅ-NORM-MET  | En metod som ger begreppet kvalitet en innebörd   |
|               | Mått (MÅTT)      | PÅ-NORM-MÅTT | Mått som ger begreppet kvalitet en innebörd       |

Figur 4.1: Kategorier för aspekten påståendekunskap.

Kategorien begrepp syftar tillbaka till att vi vill veta vilka begrepp som används och om de är tekniska eller mer orienterade mot bruket. Kategorien norm syftar här till att fånga vad som är fastlagt. Ett begrepp som används kan alltså också fungera som norm och därför kan det finna överlappningar mellan dessa kategorier. (Vi har här inte skiljt på personliga uppfattningar (norm) och kontextuella normer.)

Färdighetskunskap är handlingsbaserad kunskap, d.v.s. kunskap som kommer till uttryck genom handling. I vår studie handlar färdighetskunskap om kommunikation kring, tilläggning av kunskap om och bedömning av kvalitet (Figur 4.2) Kategorien kommunikation syftar till att vi vill veta hur systemutvecklare kommunicerar kring kvalitet. Är det genom exempel, mått eller något annat? Kategorien tillägna bygger på tanken att vi vill veta hur systemutvecklaren utvecklar känslan för kvalitet. Är det genom utbildning, social interaktion eller genom teorier?

Färdighetskunskap FÄ

| Huvudkategori       | Underkategori | Kod        | Förklaring                           |
|---------------------|---------------|------------|--------------------------------------|
| Kommunikation (KOM) | Exempel (EX)  | FÄ-KOM-EX  | Exempel för att kommunicera kvalitet |
|                     | Språk (SPR)   | FÄ-KOM-SPR | Språk för att kommunicera kvalitet   |
|                     | Begrepp (BEG) | FÄ-KOM-BEG | Begrepp för att kommunicera kvalitet |

*Figur 4.2: Kategorier för aspekten färdighetskunskap.*

| Huvudkategori   | Underkategori     | Kod   | Förklaring  |
|-----------------|-------------------|---|---|
|                 | Mått (MÅTT)       | FÄ-KOM-MÅTT                                 | Mått för att kommunicera kvalitet   |
|                 | Norm (NORM)       | FÄ-KOM-NORM                                 | En norm för att kommunicera kvalitet, t. ex. kravspecifikation, standard eller metod. |
| Tillägna (TILL) | Utbildning (UTBI) | FÄ-TILL-UTBI                                | Kompetens för kvalitet utvecklas genom utbildning                                     |
|                 |                   | FÄ-TILL-UTBI-NEG (tillkommit under kodning) | Kompetens för kvalitet utvecklas inte genom utbildning                                |
|                 | Utbyte (UTBY)     | FÄ-TILL-UTBY                                | Kompetens för kvalitet utvecklas genom social interaktion                             |
|                 | Erfarenhet (ERF)  | FÄ-TILL-ERF                                 | Kompetens för kvalitet utvecklas genom egen erfarenhet                                |
|                 | Teorier (TEO)     | FÄ-TILL-TEO                                 | Kompetens för kvalitet utvecklas genom teorier och böcker                             |

Figur 4.2: Kategorier för aspekten färdighetskunskap (forts.)

| Huvudkategori | Underkategori      | Kod  | Förklaring   |
|---------------|--------------------|--|--|
|               | Teknikutv. (TEK)   | FÄ-TILL-TEK<br>(tillkommit under kodning)    | Kompetens för kvalitet utvecklas genom den tekniska utvecklingen.                |
|               | Exempel (EX)       | FÄ-TILL-EX<br>(tillkommit under kodning)     | Kompetens för kvalitet utvecklas genom exempel.                                  |
| Bedöma (BED)  | Mätning (MÄT)      | FÄ-BED-MÄT                                   | Kvalitet bedöms genom att mäta   |
|               |                    | FÄ-BED-MÄT-NEG<br>(tillkommit under kodning) | Kvalitet bedöms inte genom att mäta  |
|               | Tyst kunskap (TYK) | FÄ-BED-TYK                                   | Kvalitet bedöms genom erfarenhetskunskap   |
|               | Norm (NORM)        | FÄ-BED-NORM                                  | Kvalitet bedöms genom norm, t. ex. metod, kravspecifikation, test eller standard |

*Figur 4.2: Kategorier för aspekten färdighetskunskap (slut)*

Till sist för färdighetskunskap har vi kategorien bedömning som handlar om hur systemutvecklare bedömer "IT-artefakter". Försöker man mäta, har man en norm (t. ex. kravspecifikationen) eller är det den tysta kunskapen (erfarenheten)?

Slutligen har vi förtrogenhetskunskap som förenklat uttryckt handlar om yrkestradition och exempel (Figur 4.3). I vårt analysar-

bete har vi gjort en enklare tolkning av dessa aspekter genom kategorierna kultur och tradition som har fått beteckna företagskultur respektive yrkestradition, dock utan den djupare innebörd som Johannessen (1999) skulle lägga i begreppet förtrogenhetskunskap.

Förtrogenhetskunskap

FÖ

| Huvudkategori    | Underkategori | Kod     | Förklaring  |
|------------------|---------------|---------|---|
| Tradition (TRAD) |               | FÖ-TRAD | Genom att ta del av yrkestraditionen får man ett kvalitetstänkande. Till exempel genom mästerlärning förhållande. |
| Kultur (KULT)    |               | FÖ-KULT | Genom att ta del av en företagskultur får man ett kvalitetstänkande.  |

*Figur 4.3: Kategorier för aspekten förtrogenhetskunskap.*

Som framgår av figurerna tillkom vissa kategorier under kodningens gång och vi lade också till en ny huvudkategori för att täcka in sådant som vi tyckte relaterade till kvalitet på ett sätt som var relevant för vår undersökning, men som inte kan hänföras till någon av de andra kategorierna (Figur 4.4). Som den kategorin är gjord skulle man kunna säga att allt kan passas in under den och det är sant, men vi har använt den för att berika de andra kategorierna, t.ex. att någon anser att användarnas medverkan är mycket viktig för att uppnå god kvalitet. Detta uttalande går inte att passa in i grundschema, men kan vara viktigt för att få större förståelse för om vederbörande definierar kvalitet som att användarna skall vara nöjda.

Varje intervju kodades alltså enligt detta schema och alla som vi tyckte relevanta intervjusvar åsattes en eller flera koder. Eftersom vi fokuserar på produktkvaliteten i vår undersökning, kodades inget som egentligen hade att göra med systemutvecklingsprocessen om det inte samtidigt sade något om produkten explicit. Att sedan ett intervjusvar kan bli kodat i flera kategorier hänger ihop med att delar av svaret berör en kategori medan andra delar berör andra, eller att svaret har en sådan karaktär att det säger något som berör flera kategorier.

Kvalitet

KV

| Huvudkategori      | Underkategori | Kod                                   | Förklaring                                   |
|--------------------|---------------|---------------------------------------|--|
| Uppfattning (UPPF) |               | KV-UPPF<br>(tillkommit under kodning) | Personliga uppfattningar som berör kvalitet. |
| Övrigt (ÖVR)       |               | KV-ÖVR<br>(tillkommit under kodning)  | Övrigt som berör kvalitet.                   |

*Figur 4.4: Tillagda kategorier för övrigt intressant i förhållande till kvalitet.*

Vi kodade alla intervjuer separat och individuellt för att i någon mån minska den bias som kan uppstå när den som kodar också är den som har gjort intervjuerna och som därmed kan ha utvecklat vissa ”skyggglappar” som i för stor utsträckning styr kodandet. Miles & Huberman (1994) rekommenderar att man involverar andra i analysprocessen för att öka reliabiliteten i tolkningsarbetet och göra den mer öppen för andra, t.ex. genom att låta en kollega göra en del av kodningen eller att den kollegan får studera den kodning man själv har gjort och se om den verkar relevant. Samtidigt är mängden möjliga tolkningar enligt Kvale (1997) en av de stora styrkorna med kvalitativ intervjuforskning och en av poängerna är att olika personer kan nå olika tolkningar ur samma material. I sken av detta tyckte vi att ett sätt att öka reliabiliteten var just att vi två gjorde kodningen separat och därmed möjliggjorde flera olika tolkningar av varje intervjusvar.

När alla intervjuer var kodade träffades vi för att jämföra varje intervju som vi kodat för att komma överens om vilken kod som skulle gälla. Vi hade i allmänhet kodat väldigt lika men ibland också helt olika, vilket ledde till intressanta diskussioner om hur vi tolkat intervju svaret och vad vi avsåg med koden som sådan. Ibland var det också så att någon av oss hade missat att koda ett visst intervju svar som den andre ansåg viktigt och genom att vara två olika kodare minskade riskerna att missa viktiga resonemang. Att arbeta på detta sätt var väldigt nyttigt eftersom vi, även om vi skriver den här avhandlingen ihop och har gjort intervjuerna tillsammans, inte alltid tänkte exakt lika och genom arbetssättet ökade tolkningsbredden.

När vi arbetat igenom alla intervjuerna på detta sätt, strukturerade vi upp varje intervju genom att klumpa ihop alla svar under varje kategori för att underlätta nästa steg i analysprocessen. Ett exempel på hur en sådan sammanställning under en kategori kan se ut visas i Figur 4.5 där två intervju svar kodats som färdighetskunskap genom att bedömning av kvalitet är baserad på erfarenhetskunskap.

#### *Meningskoncentrering*

Genom kodningen hade vi alltså fångat det vi tyckte var intressant i intervjuerna och sammanfört alla dessa svar under respektive kategori. Nästa steg i vår analysprocess var att använda det som Kvale (1997) kallar meningskoncentrering, d.v.s. att kondensera innehållet i ett visst intervju svar genom att i få ord omformulera den väsentliga innebörden i det som sagts. Kvale (ibid.) tar upp fem olika analysmetoder, där kategorisering och meningskoncentrering är två olika och separata metoder, som det framstår som att man väljer någon av men inte flera olika<sup>55</sup> beroende på vilket syfte man har med analysen. Den kategorisering som vi har gjort är dock inte så-

---

<sup>55</sup> En vanlig strategi inom kvalitativ forskning för att öka validiteten är att göra så kallad triangulering, vilket är att se om en tolkning eller en utsaga kan stödjas genom att exempelvis jämföra med andra oberoende källor. Enligt Miles & Huberman (1994) kan triangulering ske genom olika datatyper, datakällor, metoder och personer som gör undersökningen. Ett sätt borde alltså kunna vara att använda flera olika analysmetoder på samma data så att exempelvis en viss kvalitativ utsaga också kan stödjas med ett kvantitativt test.

dan som den Kvale (ibid.) talar om, utan den typ av kodning som Miles & Huberman (1994) skriver mycket om.

| <b>FÄ-BED-TYK</b>  |  |
|--|--|
| I7-22: Underhållsbarheten, den mäter vi tyvärr inte utan det är mer...det är mer att man säger till folk att – "Du det här, det här ser konstigt ut. Det här får du ta och göra om" eller "Det här skulle vi kanske göra på ett annat sätt" och så får man ge bra förslag då.                        |  |
| I7-25: Som ett komplement. Att man mäter det...för det...Det beror på vilken nivå man har som utvecklare också. Om man kommer som fräsch från högskolan så kan man inte bedöma en sådan sak, om det är hög underhållsbarhet, eftersom man kanske inte kan språket tillräckligt bra, eller verktygen. |  |

*Figur 4.5: Utdrag från sammanställning under kategorin FÄ-BED-TYK, intervju 7.*

Att bara ha kodat för att sedan gå direkt till tolkning tyckte vi inte att vi kunde göra utan att arbeta mer med själva analysen. Vi har alltså valt att tillämpa meningskoncentrering efter att ha kategoriserat intervjuerna genom att koda dem.

För varje "ruta" i en tabell som ovan, vilket alltså motsvarar det vi kallar intervjusvar eller det Kvale (1997) kallar meningsenhet, gjorde vi alltså en meningskoncentrering så att det som vi uppfattar som väsentligt givet kategorin (koden) skrevs om i färre ord för att skapa en koncisare mening. Ibland var det ganska uppenbart vad den väsentliga meningen var, som då det handlade om enskilda begrepp för att uttrycka kvalitet, men ibland var det betydligt svårare och det kunde krävas att man gick tillbaka till de kodade intervjuerna för att se vad i respektive intervjusvar/enhet som vi egentligen hade kodat och utifrån det ge en koncisare formulering.



Det exempel på meningskoncentrering från en undersökning som Giorgi gjort och som Kvale (ibid.) återger, fungerar lite annorlunda än hur vår meningskoncentrering gör. Giorgi (1994) söker på ett annat sätt än vi "lived meanings", Kvales exempel är av en fenomenologisk ansats till meningskoncentrering och vi vår tur har en mer strukturerad och därmed mer systematisk uppläggning innan meningskoncentreringen.

| <b>FÄ-BED-TYK</b>  |  |
|--|--|
| I7-22: Underhållsbarheten, den mäter vi tyvärr inte utan det är mer...det är mer att man säger till folk att – "Du det här, det här ser konstigt ut. Det här får du ta och göra om" eller "Det här skulle vi kanske göra på ett annat sätt" och så får man ge bra förslag då.                        | Underhållsbarhet mäts inte utan uppskattas genom erfarenhetsbaserad kunskap.                         |
| I7-25: Som ett komplement. Att man mäter det...för det...Det beror på vilken nivå man har som utvecklare också. Om man kommer som fräsch från högskolan så kan man inte bedöma en sådan sak, om det är hög underhållsbarhet, eftersom man kanske inte kan språket tillräckligt bra, eller verktygen. | Som ny från högskolan kan man inte bedöma underhållsbarhet, eftersom man saknar praktisk erfarenhet. |

Figur 4.6: Utdrag från meningskoncentrering under kategorin FÄ-BED-TYK, intervju 7.

När detta var gjort hade vi alltså intervjuer som bestod dels av de olika kategoriserade intervjusvaren/enheterna, dels av koncentrerade meningar som återger den väsentliga innebörden givet den åsatta koden (se Figur 4.6).

Nästa steg i processen var att skriva ihop en löpande text av dessa koncentrerade meningar för varje intervju som enligt vår tolkning representerar en sammanfattning. I detta stadium är vi också inne på det som Kvale (1997) menar med egentlig tolkning, även om analysarbetet i sig är en tolkningsprocess. Våra tolkningar är å ena

sidan i stor utsträckning sammanfattningar av meningskoncentrationen med insprängda reflektioner och ligger kanske närmare en analysprocess enligt Kvale, men å andra sidan är meningskoncentrationen i sig för vår del mer tolkningsmässig än analytisk eftersom vi stödjer oss på vår kodning. Sammanfattningarna representerar därmed till stor del tolkningsarbetet. Genom denna process kondenseras en intervju som i transkriptform uppgår till cirka 20 sidor till en sammanfattning på cirka 2 till 4 sidor. Ett exempel på en sådan sammanfattning visas i Figur 4.7.

[...] Värderingarna sker ofta vid granskningstillfällen, vilket innebär att någon annan granskar den kod som har gjorts. Vi dessa tillfällen tittar man på hur saker och ting är lösta och försöker ge vinkar om andra bättre lösningar. På så sätt skapas något av ett exempelbaserat kompetensutbyte. Denna sociala interaktion tycker respondenten är jätteviktigt.

Utanför granskningen finns det inte mycket tid för reflektion och utbyte. Men kommer man på något nytt som kan förbättra saker så måste man försöka förmedla det. Men normalt har man inte riktigt tid för sådana aktiviteter. [...]

*Figur 4.7: Utdrag från sammanfattning/tolkning, intervju 7.*

#### **4.3.4 Kvaliteten på undersökningen**

Eftersom det här är en avhandling om kvalitet, är det på sin plats att även tala om kvaliteten på undersökningen. Inom kvalitativ forskning saknas många av de etablerade former, metoder och tekniker som kan användas inom kvantitativ forskning för att uppnå hög kvalitet, vilket medför större svårigheter att bedöma kvaliteten. Dessutom är diskussionen om kvalitet i forskning och undersökningar av kvalitativ karaktär, tätt kopplad till synen på kunskap och sanning och därmed till vetenskapsteoretiska resonemang<sup>56</sup>.

De kvaliteter som man brukar förknippa med undersökningar är validitet, reliabilitet och generaliserbarhet, vilka alla är viktiga och relevanta begrepp men vars innebörder är nog så svåra att hantera och ännu svårare att relatera till sin egen forskning och undersök-

---

<sup>56</sup> se exempelvis (Enerstvedt, 1989; Kvale, 1989; Salner, 1989; Tschudi, 1989; Kvale, 1997).

ning. På något sätt måste man ändå kunna argumentera för att den forskning och de undersökningar man gör har ett värde och att de resultat man kommer fram till är meningsfulla, annars vore ju arbetet onödigt och förgäves.

Den traditionella synen på validitet innebär att man mäter det som man avser att mäta (Patel & Davidson, 1994; Wallén, 1996), vilket enligt Kvale (1989) kommer från arbetet med psykometriska tester. En variant på detta är att validitet innebär att undersöka det som man har för avsikt att undersöka. Validitet kan indelas i en mängd underkategorier såsom prognosvaliditet (resultaten skall kunna användas för prognosticering) (Wallén, 1996), innehållsvaliditet (undersökningsinstrumentets innehåll är logiskt kopplat till det man vill mäta), samtidig validitet (instrumentets utfall stämmer med andra kriterier för det man vill mäta) (Patel & Davidson, 1994), "construct validity" (mätmetoderna är korrekta i förhållande till det man vill mäta, motsvarar innehållsvaliditet), "internal validity" (kausala samband finns mellan olika villkor) och "external validity" (resultaten skall kunna generaliseras inom ett bestämt område, har koppling till prognosticering) (Yin, 1994) bara för att räkna upp några som man kan stöta på i litteraturen.

Man kan alltså ställa sig en mängd frågor kring validitet, men som exemplen ovan visar är grunderna till begreppet kvantitativa – det handlar om mätning – och skulle man utgå från det så hamnar man i den "moderna" och "positivistiska" synen på kunskap som Kvale kritiserar så kraftigt i förhållande till samhällsvetenskaplig forskning. Den underliggande synen på kunskap och sanning är dock viktig för synen på kvalitet i det här fallet, och vad Kvale (1989) vill göra är att styra mellan en fullständigt relativistisk syn där sanning och kunskap är helt och hållet en individualistisk erfarenhet, ingen mer eller mindre valid än någon annan, och en objektivistisk syn där sanning är en spegling av en objektiv verklighet och där den sanna kunskapen korresponderar med en värld av objektiva och kvantitativa fakta.

Den sistnämnda synen på sanning motsvarar enligt Kvale (ibid.) det filosofiska sanningskriteriet om korrespondens som enligt Kvale

(1989, 1997) varit central i den begränsade positivistiska synen på naturvetenskap som ideal för samhällsvetenskaplig forskning. Med en mer "postmodern"<sup>57</sup> och socialt konstruktiv syn som grund för fenomenologiska och hermeneutiska angreppssätt blir istället koherensriteriet och det pragmatiska kriteriet intressant, genom att de inte anspelar på en objektiv värld av fakta. Istället blir det med koherensriteriet en fråga om logiska samband inom och inre logik i utsagor. Det pragmatiska kriteriet leder i sin tur till att sanningsvärdet för kunskap avgörs av de praktiska konsekvenser kunskapen leder till.

Enerstvedt (1989) komplicerar bilden ytterligare genom en begreppsanalys av sanning och validitet, där han menar att den traditionella uppfattningen om dessa två begrepp är att korrespondens är en definition på sanning (han avviker alltså från Kvale (1989) här) medan koherens och pragmatik båda är former av validitet och kriterier för sanning. Meningen med de distinktioner han gör är att klargöra betydelsen av de två begreppen sanning och validitet, eftersom dessa används synonymt utan eftertanke. Enerstvedts (1989) förslag är en dialektiskt materialistisk uppfattning om relationen mellan sanning och validitet, vilket leder till att den pragmatiska betydelsen av sanning är konstruerad. För Enerstvedt (ibid.) är alltså validitet alltid detsamma som pragmatisk validitet vilken han tycker motsvarar både kommunikativ validitet och pragmatisk validitet hos Kvale (1989).

Som synes är diskussionen om validitet inte en helt enkel fråga att hantera, trots att vi endast ytligt berört den problematik och komplexitet som finns i samband med begreppet. Frågan är var detta lämnar oss? Den grundläggande filosofiska inställningen som vi har i det här arbetet är, vilket borde framgå av övriga resonemang, snarare "postmodern" och socialt konstruktivistisk än "positivistisk". Att vår undersökning skulle förevisa en sanning som en spegling av

---

<sup>57</sup> Begreppet "postmodern" är för övrigt något som Alvesson (1995) diskuterar och kritiserar, eftersom det kan betyda nästan vad som helst och inget genom att en mängd olika men viktiga ämnen trängs in under en hatt.

en objektiv och faktamässig värld tror vi inte på. Alltså är inte korrespondenssynen den vi anlägger.

Möjligtvis är den av Enerstvedt (1989) framförda synen på validitet mer trovärdig för vår del, även om det är svårt att utläsa vad den skulle innebära mer konkret. Att vår undersökning och avhandling skall göra avtryck i den samlade mänskliga kunskapen hoppas vi naturligtvis, men det känns samtidigt svårt att på förhand kunna se vad det pragmatiska resultatet blir. På ett allmänt plan skulle eventuella kunskaper behöva prövas för att se att de leder till pragmatiska konsekvenser. Det andra av Kvaless (1989) begrepp, kommunikativ validitet, som Enerstvedt för in i sitt validitetsbegrepp, ligger närmare till hands för vår del. Det är i kommunikation med en annan part som sanningsvärdet kan beskrivas och i vårt fall blir det det omgivande forskarsamhället som till stor del kommer att fälla avgörandet.

Vi tar också fasta på Kvaless (1997) tanke att valideringen blir beroende på forskarens hantverksskicklighet "... med ständigt kontrollerande, ifrågasättande och teoretiskt tolkande av upptäckterna" (ibid., s. 218). Miles & Huberman (1994) förslår 13 strategier för att göra detta och vi har använt åtminstone två – sökande efter negativa belegg och feedback från informanter. När Miles & Huberman (ibid.) diskuterar detta sker det dock i sammanhanget av olika typer av variabelsättning och abstraherande av data, så att detta lättare låter sig göras. Vår analys är inte så "strukturerad", men vi har inte sökt endast bekräftelse på resultat. När det gäller feedback till informanter har den egentligen varit för svag, men de har åtminstone fått revidera intervjuutskriften så att de återger vad de har sagt och kan stå för. Önskvärt hade naturligtvis varit att även tolkningarna hade varit föremål för samma granskning, men eftersom undersökningen och analysarbetet dragit ut på tiden så har detta inte varit möjligt.

När det gäller hantverksskickligheten tycker vi också att det handlar om att så noggrant som möjligt redovisa och diskutera hur undersökningen och analysarbetet har genomförts, vilket vi har gjort i detta kapitel. Kopplat till detta är också frågan om reliabilitet, som

traditionellt tolkas som det slumpmässiga inflytandet på undersökningsinstrumentet. Detta innebär att en annan forskare skall kunna upprepa undersökningen på exakt samma sätt och komma fram till samma resultat – resultaten skall inte vara beroende på forskaren eller undersökningsinstrumentet.

Som vi tidigare har skrivit är en av styrkorna med kvalitativ forskning och kvalitativa undersökningar, att de ger möjligheter till en mångfald tolkningar vilket kan fördjupa förståelsen snarare än att göra kunskapen mindre värd. Med en konstruktivistisk syn skulle det också vara naivt att tro att det bara kan finnas en rimlig tolkning eller ett rimligt sätt att tolka data och resultat på. Även här tycker vi dock att undersökningsredovisningen svarar mot önskan om reliabilitet, genom att andra får en klar bild över hur vi har nått våra resultat och därmed kan ta ställning till om det verkar troligt att våra resultat faktiskt är relaterade till data och att den kopplingen är stark. När det gäller begreppet reliabilitet kan också noteras att Enerstvedt (1989) föreslår att begreppet validitet skall ersätta eller omfatta också reliabilitet.

Alltså, validitet/reliabilitet ligger för vår del dels i en noggrann genomgång av undersökningens genomförande, d.v.s. hantverkskickligheten, och i en kommunikation med det omgivande forskarsamhället. Som vi har skrivit tidigare har vi också varit två personer som kodat separat vilket ökar den intersubjektiva reliabiliteten. När det gäller reliabiliteten hos intervjuarna anser vi att vi i relativt hög grad har uppfyllt de kvalitetskriterier som Kvale (1997) diskuterar.<sup>58</sup>

Slutligen handlar kvaliteten om generaliserbarhet och i vilken mån våra resultat kan användas för att förstå eller göra förutsägelser om andra situationer och andra personer än våra systemutvecklare. Naturligtvis är också detta kopplat till den kunskaps- och sannings-syn man har, där den ”positivistiska” klassiska samhällsvetenskapen

---

<sup>58</sup> Dessa kvalitetskrav är att intervjuaren skall vara: kunnig, strukturerande, tydlig, vänlig, känslig, öppen, styrande, kritisk, minnesgod och tolkande (s. 138f).

eftersträvat samma möjlighet till generaliserbarhet som naturvetenskaperna. I en "postmodern" samhällsvetenskaplig syn kan inte lagbundenheter för mänskligt beteende och mänsklig kunskap abstraheras, utan det blir en fråga om olika tolkningsakter som kan generera en djupare förståelse och vidgade perspektiv.

Med hänvisning till Schofield (1990) anger Kvale (1997) tre mål för generalisering. Det första målet är att studera det som är, d.v.s. att försöka fastställa det typiska, allmänna, vanliga sett i relation till samhället i övrigt. Det andra målet är att studera det som kanske kommer att finnas – genom att studera "state of the art" i någon verksamhet skulle man kunna generalisera hur framtiden kan komma att te sig i sådana verksamheter. Slutligen det tredje målet, att studera det som kan finnas, vilket kan innebära att intressera sig för ideala och exceptionella situationer och studera vad som pågår där.

Kvale (1997) tar också upp olika former av generaliserbarhet utifrån Stakes (1994) diskussion om generalisering från fallstudier. Den naturalistiska generaliseringen handlar om personlig erfarenhet och tyst kunskap om hur saker förhåller sig. Den leder till förväntningar snarare än formella förutsägelser och den kan verbaliseras och därigenom övergå från tyst kunskap till påståendekunskap. Den statistiska generaliseringen är formell och explicit, samt bygger på slumpmässiga urval. Därigenom kan statistiska analyser appliceras genom exempelvis konfidensprövning av generalisering från urval till population. Slutligen anges analytisk generalisering som innebär en väl övervägd bedömning i vilken mån resultat från en undersökning kan ge vägledning för vad som kommer att hända i en annan situation. Forskaren bygger här sin generalisering på påståendelogik och genom att specificera stödjande belägg och klarlägga argument, kan forskaren möjliggöra för läsaren att bedöma anspråket på generalisering.

När det gäller vår undersökning handlar det om att studera det som finns, d.v.s. det första målet för generalisering, men inte i relation till samhället i stort, utan i relation till teorier och tankar om kunskapsformer och IT-kvalitet (vilket i och för sig handlar om aspekter av

samhället). Den form av generalisering som vi kan göra anspråk på är naturligtvis inte den statistiska, utan det handlar snarare om den naturalistiska, eftersom vi hellre tänker på förväntningar än förutsägelser och förvisso försöker transformera erfarenheter och uppfattningar i vår undersökning till kunskap mer av påståendekarakter. Det finns också inslag av analytisk generalisering i vår argumentation och diskussion, även om påståendelogik kanske är ett för starkt uttryck.

#### *4.3.5 Att presentera empirin*

Att göra en empirisk undersökning och genomföra en analys och tolkning av de data man då samlat in, är ju både ett arbetsamt och genomgripande arbete, samtidigt som det förhoppningsvis leder till den kunskap man är intresserad av. På det viset är en undersökning som den vi har gjort berikande och rolig. Men arbetet kan naturligtvis inte sluta där, även andra, inte minst de som måste eller som är intresserade av att läsa den här avhandlingen, skall få utbyte av arbetet. Det empiriska arbetet måste alltså redovisas på något sätt.

Det traditionella sättet vi har stött på är att citera ur det empiriska materialet i den diskussion som återknyter till empirin och som ger tolkningar perspektiv i förhållande till forsknings- och undersökningsproblemet. Det empiriska arbetet och de tolkningar man har gjort måste naturligtvis diskuteras och reflekteras över om undersökningen skall vara till någon nytta, men det borde kunna göras på ett intressantare och mer engagerande vis än det traditionella att citera friskt från det empiriska materialet, vilket dessutom ofta är mycket svårläsligt för andra än den som har gjort intervjuerna. Detta gäller särskilt om man som vi har transkriberat så att det ligger nära talspråket.

Att presentera empirin på ett sätt som läsaren kan få utbyte och glädje av och som samtidigt förmedlar den analytiska reflektionen är med andra ord ett problem i sig. I och med att vi vänder oss ifrån det sätt vi uppfattar som traditionellt – alltså att citera rakt ur intervjuutskriften – måste vi finna ett annat sätt som vi tycker är bättre



och mer engagerande, eftersom vi verkligen vill förmedla en sammanhängande bild av vad vår analys har resulterat i.

Vi finner därför stöd i Kvaless (ibid.) kritik av det myckna och ofta ointressanta användandet av rent källmaterial, som han relaterar till en önskan om att verkligen visa de empiriska data:

Denna rapporteringsform kan ha påverkats av en kvalitativ hyperempirism, där de många intervjuцитaten får fungera som basfakta. Omfattande ordagranna utskrifter betraktas som en klippfast dokumentation av vad som verkligen sagts under intervjuerna. (Kvale, 1997, s. 230)

Nu handlar inte detta citat endast om själva återberättandet av det funna, utan lika mycket om en diskussion om den metod man använt och tillvägagångssättet som man enligt Kvale (ibid) bör beskriva så exakt som möjligt. Vi tycker att vi tidigare i det här kapitlet har redogjort tämligen omsorgsfullt både för våra utgångspunkter och vårt tillvägagångssätt, så metoden som "black box" (ibid., s. 231) anser vi att vi har undvikit.

Problemet med representationen kvarstår dock eftersom vi än så länge bara argumenterat för hur vi *inte* skall göra. Inspiration till hur det då kan göras måste alltså hittas någonstans. Miles & Huberman (1994) för ett omfattande resonemang om inte endast olika aspekter på analys, utan även om olika sätt att representera undersökningsmaterialet. I huvudsak handlar det om olika former av matriser, tabeller och grafer, vilka också hänger ihop med olika analytiska insatser. Ett spännande undantag är emellertid den referens de har till en undersökning gjord av Richardson (1992, enl. Miles & Huberman, 1994), där författaren skrev om intervjuer till dikter som inte bara kondenserade det intervjuerna faktiskt handlade om utan också förmedlade detta på ett livfullt och känslomässigt sätt; som Miles & Huberman (1994, s. 110) säger: "... displays owe as much to art and craft as they do to 'science'".

Hur kreativt och spännande detta exempel än må vara, kräver det en poetisk ådra som vi inte har och vi skulle inte våga oss på ett så djärvt grepp i vår egen representation. Representation i någon be-

rättelseform tycker vi dock verkar väl tänkt och bör göra det intressantare för läsaren. Vi har själva provat på detta angreppssätt tidigare i våra studier med ett såvitt vi vet gott resultat (Meggerle & Steen, 1994; Ehn et al., 1997a). Vi gjorde då en liten intervjuundersökning kring ett säljstöd hos Volvos återförsäljare och funderade även där på problemet med att representera undersökningsmaterialet. Vi blev då inspirerade av idéer om att kunna berika ett ”tekniskt” berättande med mer journalistiska och litterära impulser (Becker Jensen, 1987) och möjligheten att använda litterära historier och scenarier i systemutvecklingsprocessen (Clausen, 1993, 1994). Vi skrev alltså en novellartad, i och för sig amatörmässig, berättelse utifrån våra intervjuer som försökte förmedla en trovärdig kontextuell känsla och bild av det vi studerat.

Det här sättet att representera anser Kvale (1997) hänger ihop med en postmodern syn på skrivandet där kunskapen som social konstruktion har vunnit mark i förhållande till kunskap som en spegling av en objektiv verklighet och därmed förlorad tilltro till en objektiv värld som kan kopieras och representeras objektivt. Med detta följer då en förskjutning i tonvikten från observerandet av en social värld till en interaktion med denna sociala värld och ett problem med representationen. Ett möjligt sätt att lösa detta är alltså någon form av berättande.

Kvale (ibid.) använder tre olika samhällsvetares sätt att redovisa sina forskningsresultat som exempel på olika berättarstilar<sup>59</sup>. Den ena stilen är att skriva en realistisk berättelse där författaren, som inte syns själv, skapar en autentiskt berättelse genom citat ur verkliga berättelser från de ”infödda”<sup>60</sup>. Bekännelseberättelsen skrivs däremot i första person där de framträdande dragen är små melodramer om vedermödor på fältet och arbetets påverkan på etnografen (ibid.). Där den realistiska berättelsen tar fasta på det som

---

<sup>59</sup> van Maanen, J. (1988): *Tales of the Field*. University of Chicago Press, Chicago.  
Richardson, L. (1990): *Writing Strategies*. Sage, Newbury Park.  
Eisner, E. (1993): Forms of understanding and the future of educational research. *Educational Researcher*, vol. 22 nr. 7, ss. 5-11.

<sup>60</sup> Kvales (1997) beteckning på undersökningspersonerna.

vets så ligger tonvikten i den bekännande berättelsen på den som vet. Den tredje stilen, den impressionistiska berättelsen, försöker sammanföra dessa två perspektiv till en helhet genom att "... uppmärksamma vetandet som verksamhet." (ibid., s. 243). Att Kvale kallar den för impressionistisk hänger samman med att den, liksom en impressionistisk målning, gör ett innovativt bruk av stilar och tekniker, och söker särskilt "... belysa de episodiska, komplexa och ambivalenta realiteter som studeras" (ibid., s. 243).

Miles & Huberman (1994) för också ett kort resonemang om att använda sådan här "fiktions" som ett sätt att redovisa kvalitativ forskning, men varnar samtidigt för att det gör det svårare och svårare att anlägga kvalitetskriterier på resultaten, bl.a. handlar det om frågor om reliabilitet, validitet, användbarhet och bekräftningsbarhet. De avvisar dock inte detta tillvägagångssätt.

Ett exempel som förmodligen (Kvale ger själv inget exempel på en sådan text att jämföra med) är av typen impressionistisk berättelse, är de två så kallade "vignettes", som Wenger (1998) använder, vilka bygger på etnografiska studier från ett sjukförsäkringsföretag i USA och som är berättelser om olika personer som arbetar på en avdelning för ersättningshandläggning. Den första berättelsen illustrerar med huvudpersonen en arbetsdag på avdelningen och den "community of practice" som finns där, medan den andra illustrerar problemet med att använda formaliserade rutiner som användarna inte riktigt förstår. Dessa berättelser liknar faktiskt i sin uppbyggnad den som vi använde i vår magisteruppsats (Meggerle & Steen, 1994) och de berättelser vi skrev till projektets konferensbidrag (Ehn et al., 1995, 1997a). Med Miles & Huberman (1994) och framför allt Kvale (1997) som stöd för att använda berättelser och med oss själva Ehn et al. (1995, 1997a) och Wenger (1998) som exempel på att det är akademiskt accepterat, tycker vi att argumentet att använda berättandet som redovisningsform för undersökningen vilar på en tillräckligt stabil grund.

Vi kommer alltså att använda en berättelse för vår redovisning och med det ovanstående som bakgrund väljer vi att ha en mer impressionistisk stil i vårt berättande och tycker att en form av samtal

mellan två systemutvecklare och någon form av samtalsledare skulle tjäna bra för att redovisa vår undersökning och visa på den komplexitet som kvalitetsbegreppet och bedömningsförmåga uppvisar hos våra undersökningspersoner. Inspiration till just samtalsformen kommer naturligtvis, även om det är att förhäva sig, från den Sokratiska dialogen hos Platon, men även från de radio- och TV-program som filosofen Bryan Magee producerade under 1960- och 1970-talen och som han berättar om i sin bok "En filosofers be-kännelser" (Magee, 1999).

Samtalet måste, för att fungera, kunna innefatta ett flertal perspektiv och åsikter, även om den naturligtvis inte kan sammanfatta allt som framkommer i intervjuerna. Vår uppgift är att välja ut det som vi tycker är viktigt och intressant i förhållande till undersökningsproblemet, men också det som kan vara extra intressant men som kanske avviker från det övriga. Samtalet måste också struktureras kring ett antal områden och följa någon form av linje, så att det uppvisar en meningsfull helhet där frågorna för förståelsen framåt. Ett samtal är ju inte till sin natur en kaotisk mängd röster som blandas på ett slumpartat sätt utan struktur och röd tråd, men naturligtvis inte heller en total systematik och logiskt stringent ordning. Alltså, samtalet måste framställas som en balans mellan dessa ytterligheter.

Den underliggande struktur som då måste finnas, anser vi kan hämtas från de teman som vår undersökning bygger på (se bilaga 2) och som intervjuerna är utformade inom. Dessa teman täcker in det som vi är intresserade av och genom att strukturera samtalet efter dem, gör vi också en direkt återkoppling till undersökningsproblemet. På så vis kan vi säga att vi sluter cirkeln tillbaka till de frågor som vi har ställt i intervjuerna och därmed kommer tematisering, undersökning, analys och redovisning uppvisa en konsistent helhet.

#### *Bakgrund för samtalet*

För att göra ett samtal möjligt måste deltagarna till viss del karakteriseras och ges en "personlighet" – de måste ju ta avstamp någonstans ifrån för att kunna uttrycka åsikter. Men vi sätter alltså citationstecken kring personlighet för att markera att det inte handlar

om litterära gestalter, så deras levnadshistoria är därmed inte intressant i detta sammanhang. Däremot kan deras yrkesmässiga erfarenhet och deras arbetsuppgifter vara viktiga för sammanhanget.

Vi har, som redan nämnts, använt berättelser förut för att dels gestalta vår undersökning om bilsäljare med ett säljstöd (Meggerle & Steen, 1994; Ehn et al., 1997a) i formen av en historia, dels för att gestalta ett tänkbart, framtida Kvalitek i formen av ett scenario (Ehn et al., 1997a)<sup>61</sup>. Där var det viktigare att skapa trovärdiga karaktärer som läsare (i Clausens fall blivande användare) kunde identifiera sig med för att därmed kunna sätta sig in i situationen. Vi hittade därför på två figurer som kunde föra berättelserna framåt och som tjänade syftet att utgöra dessa karaktärer. Vi döpte dem till Eva och Adam.

Dessa två figurer återupplivar vi nu för att använda dem även i detta samtal. Eva var från första början systemutvecklare, medan Adam sedan sist har tröttnat på sina forskarstudier inom arkitektur och bytt karriär till systemutvecklare (vilket passar bra eftersom vissa anser att arkitektur och systemutveckling har många gemensamma beröringspunkter, se exv. Sewell & Sewell (2002) för argument för denna syn). Eva har naturligtvis längst erfarenhet inom branschen av dem båda och har avancerat till kvalitetsansvarig, men har under årens lopp hunnit med många av de olika arbetsuppgifter som systemutvecklare kan sysselsätta sig med, t.ex. programmering, analys och utformning, testning, projektledning, funktionsdesign och databasdesign. Adam är mindre erfaren och har uppenbarligen färre år bakom sig inom branschen, vilket innebär att han mest har sysslat med programmering, databaser och viss analys och funktionsdesign. Moderatoren, som heter Lars, är en rätt karaktärslös figur som har vissa kunskaper om kvalitet och systemutveckling tillräckliga nog för att ”styra” samtalet. Till sin hjälp har han ett antal teman som täcker in det som dialogen skall kretsa kring.

---

<sup>61</sup> Begreppen historia och scenario kommer från Clausen (1993, 1994).

Meningen är att samtalet skall försiggå under trevliga och avslappnade former, varför en lite intimare ”studio” med bekväma fåtöljer och sympatisk inramning är den miljö där det äger rum.

För att återknyta samtalet till empirin så är det annoterat genom fotnoter med illustrativa utdrag ur meningskoncentreringen. Dessa fotnoter står i anslutning till frågor från Lars och föregår därmed den diskussion som frågorna ger upphov till, d.v.s. fotnoterna avslutar inte, som brukligt, förda resonemang, utan att de inleder dem. Vilka intervjuer dessa utdrag är hämtade ur, markeras genom en kod, exempelvis I1-58, som står för intervjunummer och meningsenhet, i detta fall intervju 1, meningsenhet/svar 58. Utdragen kommer alltså inte från själva intervjutranskripten och är därför redigerade och tolkade.

## 5 Systemutvecklarens kvalitetssyn – ett samtal med Eva och Adam

Lars: Välkomna Eva och Adam till det här samtalet som skall handla om hur systemutvecklare ser på kvalitet på de produkter som de utvecklar. Ämnet är ju intressant nu när IT i dess olika former gör sig bemärkt inom nästan alla aspekter av samhället, inte minst i arbetslivet där i stort sett ingen nuförtiden kan arbeta utan att använda dator eller kanske snarare datasystem. Det är det sistnämnda som vi vill att ni skall reflektera över, eftersom ni är här som representanter för den allt större grupp i arbetslivet som har som yrke att tillverka datasystem – ni är ju alltså systemutvecklare. Jag tänkte börja med frågan – Vad är kvalitet?<sup>62</sup>

Eva: Det var ju ingen lätt fråga! Kvalitet är ju väldigt mycket och kan betyda nästan vad som helst egentligen – det beror så mycket på vad man lägger i begreppet, så på det viset är kvalitet något subjektivt och något som ligger i betraktarens öga.

Lars: Vad skulle du lägga i begreppet?

Eva: Ja, ur mitt perspektiv, eftersom jag är kvalitetsansvarig, handlar det väldigt mycket om en god relation till kunden och att alla medarbetare också arbetar med det i siktet. Att det finns ett kvalitetssystem som genomsyrar organisationen och dess arbetssätt.

Adam: Men du sa nåt viktigt där, nämligen att det handlar om perspektiv. Det handlar också om intressenter. För att kunna säga vad kvalitet är måste man också säga ur vems perspektiv kvalitet skall ses, för precis som du säger så kan kvalitet vara så

---

<sup>62</sup> I1-58: Kvalitet är subjektivt och alla har inte samma kvalitetsbegrepp.

I3-17: Kvalitet definieras utifrån olika roller och perspektiv.

I6-16: Det är enligt X väldigt svårt att säga vad som är hög kvalitet inom IT.

väldigt mycket och betyda så många olika saker för olika personer. Du nämnde ju kund och det är en intressent vars perspektiv man kan anta. Man skulle ju också kunna tänka sig att anta användarnas perspektiv, eller utvecklarnas, eller kanske systemförvaltarnas. Man kan nog till och med se olika perspektiv hos utvecklarna – är det en programmerare, en funktionsdesigner eller kanske en databasdesigner?

Eva: Javisst, det är det jag menar och jag tycker att det finns tre perspektiv på kvalitet som i mycket sammanfaller med det du sade, nämligen ett tekniskt, ett användar- och ett organisatoriskt perspektiv. Men ur mitt perspektiv som kvalitetsansvarig så skall jag ha en mer övergripande bild av kvalitet i företaget och då är ju kunden den viktigaste intressenten. Inga kunder, inga system – det är ju så det fungerar. Sedan skulle jag också vilja säga att begreppet kund ibland betyder beställaren, ibland en representant för användarna och ibland själva användarna. Så det är också ett begrepp som kan vara mångtydigt.

Lars: Jag reagerade också på begreppet kund, förutom det du sade om perspektiv. Så när du tänker på kund ur ditt perspektiv som kvalitetsansvarig är det beställaren du menar och att kvalitet i så fall innebär att beställaren skall vara nöjd?<sup>63</sup>

Eva: Jo, slutligen blir det naturligtvis så. Men det är sällan det är beställaren som använder systemet, utan det är ju användarna och då får man det på det viset att om inte användarna är nöjda med vårt jobb och den produkt som vi har ... ja, tillverkat, lär naturligtvis beställaren få reda på detta, och då blir inte heller beställaren nöjd. Så det är inte lätt att bara bena upp de olika rollerna på det här viset.

Lars: Men vad betyder i såfall kvalitet om man utgår från kundens perspektiv?<sup>64</sup>

---

<sup>63</sup> I4-26: Definitionen av kvalitet är väldigt subjektiv och det är den enskilda kunden som avgör vad han tycker är högt eller lågt.

<sup>64</sup> I4-26: Ett system skall ha enkelhet, det skall vara ett verksamhetsanpassat



Eva: Återigen är det en problematisk fråga, för det finns ju inte en standardkund som man kan utgå ifrån när man talar om kvalitet. Kunder är ju väldigt olika och har olika önskemål beroende på den verksamhet de bedriver. Men det är häri det ligger samtidigt – ett system skall vara ett verksamhetsstöd, det skall bidra med något positivt i verksamheten så att den blir effektivare, eller så att man kan hantera sina kunder bättre, eller vad det nu kan vara. Så, ett ord för kvalitet är alltså verksamhetsstöd.

Adam: Men begreppet verksamhetsstöd är ju i sig självt väldigt brett och luddigt och det representerar ju som sagt endast ett perspektiv, om än ett viktigt sådant. För nånstans hamnar man också i att ett program eller system måste vara användbart och fungera – om det hänger sig hela tiden eller räknar fel, eller om databasen blir inkonsistent, fungerar inte systemet eller programmet som det skall och då är det inget stöd i verksamheten. Naturligtvis förutsätter det att beställaren inte vill ha systemet så, vilket den säkert inte vill med den här typen av fel. Samtidigt är kvalitet en resursfråga också i form av tid, pengar och personer, och i det perspektivet får vissa kvaliteter ibland stryka på foten till förmån för att systemet skall vara billigt eller att det skall utvecklas och implementeras snabbt.

Lars: Hur menar du då?

Adam: Ja alltså, det finns ju kvaliteter som kunden eller användaren kan ha svårt att bedöma. Det kan gälla underhållsbarhet eller läsbarhet...

Lars: Läsbarhet?

Adam: Ja, läsbarhet är också ett begrepp som man kan tänka sig att använda när man talar om kvalitet. För mig är det viktigt att program är ordentligt dokumenterade så att man som utvecklare kan förstå vad det gör och vad olika delar i programmet gör. Det handlar också om att programmen är skrivna på ett

---

verktyg. Arbetet skall styra hur verktyget skall fungera. Dålig kvalitet är om verktyget inte passar med användarens vardag.

sådant sätt att man kan förstå hur de fungerar och hur logiken är uppbyggd, och vissa är sämre eller bättre än andra på att åstadkomma detta. Jag tycker att om ett program eller system skall vara underhållsbart måste det också vara läsbart. Annars blir det ju mycket svårare att underhålla, eftersom det blir svårare för mig att sätta mig in i det och göra ändringar. Men som sagt, vissa kvaliteter får ibland stryka på foten till förmån för andra kvaliteter och där kan underhållsbarhet vara en.

Lars: Så underhållsbarhet är en kvalitet som du upplever som viktig men som beställaren inte alltid värderar?

Adam: Ja, det kan vara så eftersom det inte är något som syns hos dem – de ser ju oftast inte koden och arbetar inte med den. Det är ju vårt ansvar att försöka argumentera för att underhållsbarhet är viktigt, men ibland så får man kompromissa med den kvaliteten för det kostar ju naturligtvis mer att göra system underhållsbara, även om det i långa loppet kan bli billigare om systemet är underhållsbart.

Eva: Fast jag tycker inte att det bara handlar om kunden när det gäller underhållsbarhet, även om jag håller med dig om att man ibland kompromissar med den kvaliteten. Jag tycker också att det är så att inte alla utvecklare heller förstår värdet i underhållsbarhet och det tror jag har att göra med att inte alla har erfarenhet av systemförvaltning.

Lars: Jag tycker av diskussionen hittills att ni rör er på flera olika nivåer när det gäller kvalitet. Vi började med en hög nivå som handlade om verksamhetsstöd och nu verkar vi vara på en lägre nivå som handlar om programkvalitet och systemkvalitet. Kan man då säga att kvalitet handlar om egenskaper på flera olika nivåer?<sup>65</sup>

---

<sup>65</sup> I12-10: Kvalitet kan ses ur tre olika perspektiv: systemet skall göra det kunden vill att det skall göra, systemet skall vara bra byggt utifrån de miljöer som finns och koden skall vara underhållsvänlig.

Eva: Ja, det kan man, men det hänger ju också ihop med det här om perspektiv och roller. Med ett visst perspektiv ser man en viss nivå och olika roller inom den nivån ser vissa kvaliteter. Som programmerare blir ju koden intressant i sig och då kan det handla om läsbarhet som vi var inne på tidigare, det kan handla om deklaration av variabler, det kan också handla om sådana saker som att en modul inte skall göra en massa olika saker och att koden skall vara uppdelad i just moduler som har rena gränssnitt till varandra. Vissa av de här sakerna går väl tillbaka till läsbarhet och förstaelighet. Man skulle också kunna se på kvalitet som funktionsdesigner och då är koden inte det man arbetar med, utan då handlar det om funktionalitet – att program och system har de funktioner som de skall ha och att de är utformade på rätt sätt. Ytterligare ett sätt att se på kvalitet är som gränssnittsdesign och då handlar det om logisk uppbyggnad av ett gränssnitt, om dialogstruktur, färgval, med mera.

Lars: Med det här som grund skulle jag vilja säga att kvalitet alltså är många saker, på olika nivåer och ur olika perspektiv. Och om jag har fattat rätt så innebär kvalitet att beställaren/kunden/användaren är nöjd, men att det samtidigt finns kvaliteter som ni som utvecklare kan se och värdesätta, till exempel då läsbar kod.<sup>66</sup>

Adam: Ja, men visst är det å ena sidan så att kunden eller användaren är den som fäller det slutgiltiga avgörandet. Även om vi tycker att det är ett fantastiskt system kanske kunden tycker att det är dåligt och då är det ju det i någon mening. Å andra sidan kan ju de tycka att systemet är jättebra, medan vi tycker att det är sämre av olika anledningar, till exempel underhållbarheten.

---

<sup>66</sup> I11-6: Kvalitet kan ses ur flera olika perspektiv. Sett ur användarnas perspektiv handlar kvalitet om funktionalitet, hur snabbt det går att lära sig en produkt och om produkten uppfyller deras förväntningar. Ur en teknikers perspektiv är kvalitet en fråga om prestanda, stabilitet och att produkten fungerar bra.

Lars: Men går det då att definiera kvalitet och säga att kvalitet är X?<sup>67</sup>

Eva: Ja, på ett sätt går det ju genom att använda ISOs definition och säga att kvalitet är att uppfylla kundens uttalade och underförstådda behov. Fast då har man ju egentligen inte sagt vad kvalitet faktiskt är i själva produkten.

Adam: Nej, jag tycker nog inte att man kan definiera kvalitet på det sätt du frågar efter och det märks ju också på diskussionen hittills. Vad man kanske kan göra är däremot att tala om vilka kvaliteter som man tycker är viktiga och då har vi ju varit inne på några stycken. Vi har talat om underhållsbarhet, modularitet, förståelig dialogstruktur, läsbar och kommenterad kod och dokumenterade program.

Eva: En annan viktig kvalitet tycker jag är prestanda. En fråga mot en databas skall inte ta orimligt lång tid, för då är det något som är fel. För vissa är prestanda väldigt viktigt – programmet skall gå fort. Men jag vill också återvända till mer användarnära kvaliteter och då tycker jag att system och program skall vara stöd och verktyg som fungerar för de uppgifter som användaren använder dem till. Förutom detta tycker jag också att vissa andra ord passar in för att uttrycka en mer bruksmässig kvalitet och det är att program och system skall vara användarvänliga, lättförståeliga och lättlärd.

Adam: Jag vill också flika in att våra produkter skall ge en skön och intuitiv känsla och ha snygga gränssnitt. Men framför allt handlar den bruksmässiga kvaliteten om funktionalitet som vi har varit inne på – alltså att systemen och programmen stödjer de uppgifter som skall utföras med det.

---

<sup>67</sup> I15-70: X tror att man säkert kan hitta någon generell definition på kvalitet i en ordbok eller göra en definition, men rent praktiskt är det något man känner och som kan vara svårt att avgöra.

Eva: Det där med funktionalitet håller jag helt med dig om, men att de skall ge en skön känsla är det nog inte många som tänker på.

Lars: Jag tolkar det som om det är väldigt svårt att ge en definition av kvalitet som inte blir luddig eller allmängiltig.<sup>68</sup> Men vi var ju också inne på kvaliteter/egenskaper och vi kanske skall fortsätta lite på det spåret. Ni har ju nämnt några kvalitetsbegrepp, till exempel då underhållsbarhet och verksamhetsstöd, som ni tycker har med kvalitet att göra. Det verkar som att man, även om man inte kan definiera kvalitet, kan använda olika begrepp för att karakterisera kvalitet.

Eva: Ja, så är det nog. Om man tänker efter så finns det naturligtvis flera olika kvaliteter som man eftersträvar och vi har ju nämnt flera. Det finns fler också än de som vi har nämnt – stabilitet och att exekveringscykeln skall vara effektiv till exempel. Så det är nog så att det är enklare att nämna olika kvalitetsbegrepp av både teknisk och bruksmässig karaktär, än att slå fast en definition av kvalitet. Kvalitetsdefinitioner tenderar precis som du sa att bli luddiga eller allmängiltiga, exempelvis ISOs definition eller att program och system skall fungera och vara ett stöd i verksamheten. En definition kan å andra sidan bli för på för låg nivå också och handla om att program skall ha en klar och logisk struktur, vilket ju inte säger mycket om verksamhetsstöd.

Lars: Ok, då lämnar vi diskussionen om kvalitetsdefinitioner genom att konstatera att sådana är omöjliga eller åtminstone mycket svåra att åstadkomma. Men jag tänker på de här kvalitetsbegreppen som ni använder och på att de ofta innehåller en värdering – exekveringscykeln skall vara effektiv, systemet skall vara ett verksamhetsstöd, med mera. Är de begreppen ni använder uttryck för vad som är hög kvalitet på program och system?

---

<sup>68</sup> I13-21: X tycker inte att man kan finna en konkret och allmängiltig kvalitetsdefinition för varu- och tjänsteproducerande företag som inte är på en hög och "flummig" nivå. Man har inte heller en ISO-standard för alla företag.

Eva: Ja, så är det ju. Ett program av hög kvalitet skall ha flera av de egenskaper som vi har nämnt hittills, där kanske funktionalitet är det viktigaste.

Adam: Jag håller med Eva, men tycker också att hög kvalitet är mer än att uppnå det som kunden har satt upp i kravspecen och att man...

Lars: Ursäkta att jag avbryter, men du nämnde ordet kravspec. Är det den som så att säga styr vilken kvalitet man skall uppnå?

Adam: Kravspecen är ju den specifikation som liksom namnet säger, innehåller de krav på program och system som kunden har satt upp. Fast det är lite fel att säga "kunden satt upp" för det varierar ju också. Ibland är den väldigt noggrant specificerad av kunden i förväg och ibland har vi ett tätt samarbete kring utformandet av den. Men till slut ligger den fast som grund till att bygga systemet.

Eva: Fast det varierar ju också med hur processen skall fungera. Med ett mer evolutionärt eller prototyping-inriktat utvecklings-sätt fungerar ju inte kravspecen på det viset.

Lars: Fast nu börjar vi komma in på själva utvecklingsprocessen och vi skall ju tala om produkten, även om jag förstår att de hänger ihop. Men i alla fall, kravspecen är alltså i allmänhet grunden för vilken kvalitet man skall uppnå?<sup>69</sup>

Adam: Ja, det är den, men det jag tänkte säga var att jag tycker att hög kvalitet är att överträffa kravspecen, det vill säga kundens krav, och ge ett mervärde.

Eva: Jag håller med dig där och tycker att det har att göra med de underförstådda behoven som jag nämnde i samband med ISO-definitionen – hög kvalitet är faktiskt att lämna ett större bidrag än vad kunden egentligen hade förväntat sig. Men samtidigt är detta också lite lurigt, för när det till exempel gäller funk-

---

<sup>69</sup> I14-23: Beställarens krav ligger till grund för bedömning av kvalitet.

tionalitet så skall man inte försöka bygga in mer funktionalitet bara för att det går.

Lars: Så hög kvalitet är förutom det som vi redan tagit upp som begrepp, att man kan ge ett mervärde i förhållande till de önskemål och krav som kunden har, men att man samtidigt inte skall bygga in en massa som kunden eller beställaren egentligen inte har behov av?<sup>70</sup>

Eva: Ja, det skulle jag vilja säga.

Lars: Ok, men låg kvalitet på ett program då, är det motsatsen till det ni precis har sagt?

Adam: Ja, det blir det ju. Ett program av låg kvalitet fungerar inte som det är tänkt att göra, det är inte stabilt och kraschar hela tiden, användaren förstår inte hur det skall användas och det kräver lång inläring, prestanda är dålig så att användaren måste vänta länge på att till exempel få fram en kund på sin skärm eller uppgifter om en vara, väntetider som kan vara väldigt frustrerande om det till exempel handlar om en kassa och det lätt bildas långa köer.

Eva: Och med ett perspektiv som utvecklare så blir ju kvaliteten låg om programmen är illa skrivna och dokumenterade så att de blir svåra att sätta sig in och underhålla och förändra. Om man sedan lyfter blicken till system är många av dessa kvaliteter desamma naturligtvis, men då tillkommer också att program skall kunna samverka på ett bra sätt i systemet och att system skall kunna samverka sinsemellan på ett smidigt sätt. Jag tycker också att användaren skall kunna skapa en modell över hur syste-

---

<sup>70</sup> I14-26: Hög kvalitet på ett system är att det ger ett betydande mervärde i förhållande till specifikationerna, tex. att det blir betydligt snabbare och enklare att använda programmen. Dessutom skall det ha underhållsbarhet så att det inte tar lång tid att nyutveckla.

I5-14: Kvalitet på produkten är att man tillfredsställer kundens behov på kort och lång sikt och att man inte ger dem något jättebra som de egentligen inte vill ha.

men fungerar och samverkar och därigenom kunna förstå konsekvenserna av att använda de olika systemen, alltså hur de påverkar varandra. Men återigen så är väl funktionaliteten viktig här och att systemen stödjer verksamheten.

Adam: Det speciella med datasystem är ju också att de representerar en given bild vid ett givet tillfälle – de så att säga cementerar verksamheten, men verksamheten förändras hela tiden och då är det viktigt att systemen också går att förändra. Ju svårare det är att göra, desto lägre är kvaliteten tycker jag. Sedan tycker jag också att det där du sa om att användaren skall kunna skapa en modell av systemet kanske ligger mer i utbildning egentligen, även om det naturligtvis också är en kvalitet.

Lars: Hur avgör ni i såfall att ni har uppnått den kvalitet som ni skall uppnå? Mäter ni det eller uppskattas det på något annat sätt?<sup>71</sup>

Eva: Mäter kvalitet gör vi ju väldigt lite, för lite egentligen kanske. Men jag tror också att det är oerhört svårt att sätta upp mätbara kriterier för särskilt många kvaliteter och sedan också skapa metoder för att mäta dessa. Hur skall man till exempel kunna mäta kundtillfredsställelse eller enkelhet eller lättlärdhet? De här mjukare kvaliteterna tror jag inte att man kan skapa mått för. Mer tekniska kvaliteter är nog i såfall lättare att sätta upp mått för och mäta och till viss del görs det också – prestanda till exempel kan man ju mäta i antalet sekunder en viss databasfråga tar. När det gäller de mjukare kvaliteterna kan man i och för sig använda en kundenkät där kunden har satt upp viktiga kriterier för systemet och som kunden sedan betygsätter. På det viset kan man få en sorts betygsättning som ger vissa siffervärden. Lite samma sak är det med att se hur många felrapporter

---

<sup>71</sup> I1-114: Kvalitet är subjektivt och svårt att sätta mått på.

I2-45: Det finns saker som är mätbara och de lite luddigare där det är bedömning.

I6-16: Även om företaget har standarder för att mäta kvalitet, tycker X att mätning av kvalitet är mycket svårt.



som kommer in och använda det som något mått. Då kan man se hur många fel som finns och hur lång tid de tar att rätta. Sedan kan man ju diskutera om det är att mäta kvalitet.

Adam: Jag tror inte heller på att mäta kvalitet, annat än väldigt enkla och tekniska kvaliteter precis som Eva sa. Jag känner inte heller till några sådana metoder som man skulle kunna använda till mätning och tror som sagt att det är väldigt svårt att skapa sådana. Det är viktigt också att de metoder och mått man i såfall tar fram är standardiserade och inte skiljer sig mellan olika kunder och olika typer av projekt, för då kan de ju inte jämföras med varandra. Sedan det där du sa om felrapporter, Eva, så skall naturligtvis antalet rapporter vara så litet som möjligt. Att kunna rätta fel snabbt kan i och för sig hänga ihop med olika kvaliteter, men målet är ju ändå att antalet fel skall vara så lågt som möjligt.

Eva: Ja, naturligtvis.

Lars: Om man inte kan mäta kvalitet, hur avgör man i såfall kvaliteten? För den måste väl bedömas ändå?<sup>72</sup>

Eva: Det är klart att på något sätt måste kvaliteten bedömas, annars har man ju väldigt svårt att se om man har uppfyllt de kvalitetskrav som man vill uppnå. Det finns flera andra sätt än mätning att bedöma kvaliteten och testning är ju väldigt viktigt och mycket använt. Vad man då gör är att testa enskilda program så att de gör vad de skall göra, att de fungerar ihop som tänkt och att man testar funktionaliteten så att den uppfyller de krav som ställts. Så man genomför modultester, integrationstest, systemtest och acceptanstest. Acceptanstesterna skall normalt kunden utföra och då testar man att det som användarna skall

---

<sup>72</sup> I2-38: Man gör inga mätningar.

I13-18: Nya releaser släpps två gånger per år och genomgår olika tester. Varje program modultestas först och integrationstestas sedan med alla andra system på IT-avdelningen. Efter detta acceptanstestar man med test-case från användarna. På detta vis kontrollerar man att programmen är både funktionsmässigt och programmässigt bra.

kunna göra med systemet, givet vissa typer av arbetsuppgifter, verkligen fungerar. Så tester är väldigt vanligt och mycket vanligare nuförtiden än för femton år sedan. Sedan har man ju också metoder och standarder som stöd för hur program skall utformas, hur gränssnitt skall se, hur dokumentationen skall skrivas och så vidare. Kvalitet är också att följa metoder och standarder, att inte avvika från dem.

Adam: Sedan kan man också ha formella kodgenomgångar och andra typer av reviewer, där man jämför med kravspeccen och standarder och på så vis bedömer kvaliteten.

Eva: Men samtidigt måste man se metoder och standarder som stöd, inte regelverk som man absolut inte får avvika ifrån. Om det är så att man har gjort exakt enligt standard men prestandan blir för dålig, måste man ju avvika från standarden för att nå bättre prestanda. Det är samma sak med gränssnitt – om man använder standardiserade gränssnittskomponenter och följer guide lines, men resultatet blir dåligt, måste man återigen avvika från de här riktlinjerna och göra på något annat sätt.

Lars: Så man kan alltså behöva mer eller mindre bryta mot standarden till exempel, för att uppnå hög kvalitet?<sup>73</sup>

Eva: Ja, fast man måste naturligtvis ha goda skäl till det. Man håller sig till standard, guide lines och metoden så länge det fungerar. Men om det inte gör det, måste man naturligtvis avvika från dessa.

Lars: Men att veta när man skall avvika verkar ju förutsätta en sorts erfarenhetskunskap, en fingertoppskänsla?<sup>74</sup>

---

<sup>73</sup> I3-24: Det är viktigt att ha förmågan att se när man måste bryta mot normen (standard, metod, mätal).

<sup>74</sup> I8-32: I bedömningsförmågan ingår att förstå hur långt man kan gå för att nå enkelhet och när man skall stoppa, för man kan driva en förändring hur långt som helst enligt X. Det svåra är att nå en tillräcklig enkelhet och det ligger i bedömningsförmågan.

Adam: Javisst, och det är ju också därför som Eva säger att metoder och standarder skall ses som stöd i systemutvecklingen. Vissa tycker förvisso att metoder skall räcka hela vägen för att uppnå kvalitet, men jag tycker att man endast når så långt och att det som gör kvaliteten hög, som ger spetsen så att säga, är den här fingertoppskänslan. De här strukturerade hjälpmedlen kanske räcker till 80% av kvaliteten, så den klassiska 80/20-regeln gäller även här.

Eva: Ja, och i vissa fall kanske det är enklare att veta när man skall avvika från standard, som med prestanda, men till exempel gränssnitt bedöms ju i mycket utefter erfarenhet och fingertoppskänsla. I erfarenhetskunskapen ligger också att veta när man skall sluta designa något, för det ger sig inte av sig självt utan det kan alltid bli lite bättre eller snyggare. Alla förändringar kan drivas hur långt som helst, men förr eller senare måste man sätta stopp och det hänger ihop med erfarenheten.

Lars: Så om man skulle sammanfatta detta om hur man bedömer kvaliteten, så skulle man kunna säga att mätning inte används, eftersom det dels är svårt att sätta upp vettiga mått och dels därför att det är svårt att skapa bra mätmetoder. För att bedöma kvalitet är det i stor utsträckning tester man använder, men kanske också olika reviewer, och detta sker i förhållande till kravspec och standarder och metoder. Sedan finns det också en viktig erfarenhetskunskap och fingertoppskänsla som krävs för att man skall uppnå hög kvalitet och kunna använda metoderna och standarderna på ett vettigt sätt. Kan man säga så?<sup>75</sup>

Eva: Ja, det tycker jag sammanfattar vad vi har diskuterat och det stämmer med min bild, men jag vill också gärna trycka på

---

<sup>75</sup> I1-91: Den erfarenhetsmässiga, intuitiva och luddiga känslan är den viktiga.

I3-46: Det finns saker i en bedömningsförmåga för en utvecklare som ligger utanför det normerande. Metoder och dylikt är ett ramverk som kompletteras av erfarenhetskunskap kring vad som är bra och gynnsamt, och som är individuellt för X och tillhör det dolda.

den där svårbeskrivbara erfarenhetskunskapen och hur viktig den är.

Lars: Men när ni nu bedömer kvalitet, är det vissa kvaliteter som är viktigare än andra?<sup>76</sup>

Eva: Ja, men det tangerar det som vi var inne på tidigare om perspektiv och nivåer. Men visst, jag tycker ju att funktionalitet är väldigt viktigt och att systemet fungerar som ett stöd i arbetet.

Adam: Jag tycker att dokumentation är viktigt, men naturligtvis är det som Eva säger att funktionaliteten, användbarheten och stödet i arbetet är det viktigaste. Underhållsbarheten är naturligtvis också viktig.

Lars: Nu har vi ju talat en hel del om kvalitet som ett fenomen eller begrepp och hur man bedömer kvalitet. Vi har också varit inne en del på vem som bedömer kvalitet, eller kanske snarare avgör kvaliteten, och där verkar det som om det är kunden eller användaren som gör det.<sup>77</sup>

Eva: Ja, i slutändan är det ju så, men samtidigt vill vi också åstadkomma något bättre än det kunden förväntar sig och vissa tekniska aspekter kan ju kunden ofta inte bedöma, utan det måste vi göra. Så jag tycker nog att kvaliteten avgörs av de involverade parterna tillsammans – kvalitet är den sammanlagda bilden av allas åsikter. Kvaliteten på funktionaliteten måste till exempel avgöras av kunden och användarna, för det är de som skall använda systemet och som vet hur det bör fungera. De mer tekniska bitarna är vi å andra sidan bättre på att bedöma, medan gränssnittets utformning måste bedömas av både användar-

---

<sup>76</sup> I11-26: X tycker att man får kompromissa mellan olika kvaliteter och tex. gäller det i det som X tycker är att cementera företaget och företagsstrukturen i systemet. Systemen blir komplexa och svåra att ändra, fast å andra sidan måste man någon gång ta ett beslut om hur systemet skall se ut och det är att bygga in en begränsning i systemet.

<sup>77</sup> I8-9: Kvalitet är den sammanlagda bilden av alla inblandade parters synpunkter.

na och oss, eftersom de inte har all kompetens för att göra en sådan bedömning.

Lars: Men då är det väl inte bara kunden som avgör kvaliteten?

Eva: Jo, som vi sade tidigare så är det ju slutligen så, men visst är det så att det handlar om bygga rätt system bra, inte bara bygga rätt system, och då hänger det mycket på vår kompetens och vår förmåga att bedöma kvaliteten och kvaliteter. Samtidigt är det naturligtvis också en fråga om att man inte bara kan bygga bra system, man måste även bygga rätt system.

Lars: Hur viktigt är kvalitet då?<sup>78</sup>

Adam: Ja, det är ju en märklig fråga att ställa! Kvalitet är ju naturligtvis oerhört viktigt, har vi ingen kvalitet så har vi heller inget system egentligen. Har vi ingen kvalitet så försvinner våra kunder och vi kan inte få nya. Utan kvalitet går ju företaget under.

Eva: Ja, fast kvalitet har inget egenvärde i sig heller, utan det handlar om att hantera förväntningar och det handlar om image delvis. Det är som med bilar – man förväntar sig högre kvalitet på en Mercedes än på en Lada och vill man ha en Lada så accepterar man lägre image och lägre kvalitet. Att hantera kundens förväntningar på rätt sätt är också en del av kvaliteten.

Lars: Men en Mercedes är ju betydligt dyrare än en Lada, så det är ju en kostnadsfråga också.

Eva: Javisst, men så är det ju med kvalitet på våra produkter också. Mer tid, mer folk och mer pengar resulterar alltid i bättre kvalitet, eller i alla fall i möjligheter till bättre kvalitet. Då kan

---

<sup>78</sup> I4-6: Det finns enligt X inget självändamål med kvalitet, utan den måste bedömas av någon utifrån ett antal kriterier.

I6-36: Kvalitet är inte vad du (leverantörer) tror att du åstadkommer utan vad kunden tror att du åstadkommit och om det uppfyller deras förväntningar.

man testa mer, rätta fler fel och vara noggrannare helt enkelt. Så kvalitet är alltid en kompromiss mellan olika kvaliteter.

Lars: Så man får alltså göra avkall på vissa kvaliteter till förmån för andra?

Adam: Ja, det var ju som jag sade innan, att ofta får dokumentationen stryka på foten för att det inte hinns med. Ibland är det så bråttom med utvecklingen av systemet att kunden är beredd att offra vissa kvaliteter för att få en snabb implementation. Då kanske sådana saker som bra och logisk uppbyggnad, underhållsbarhet och kommenterad kod kommer i andra hand.

Lars: Vad tycker ni om det då?

Adam: Nej, det är ju inte roligt och man vill ju göra ett bra arbete, men ibland får man som sagt kompromissa.

Lars: Men då kanske kvaliteten blir låg.

Adam: Ja, det kan den ju bli då.

Lars: Men vad händer då?

Eva: I ett sådant fall som Adam talar om, får ju kunden den kvalitet kunden kan vänta sig, men annars får man naturligtvis höra det. Då ringer användare och är missnöjda, och i absolut värsta fall kan det bli juridiska problem också.

Lars: Men det sätt ni får reda på om kvaliteten upplevs som för dålig, så är det genom att användarna ringer till er?

Adam: Ja, de ringer ju när det är en massa problem som måste åtgärdas eller så får vi det via helpdesk och felrapporter.

Eva: En del av kvaliteten ser vi ju också som sagt var genom våra egna bedömningar via tester och så, så all kritik kommer inte via användarna. Men visst, upplever de att kvaliteten är för låg får vi ju höra det och den feedback vi får handlar nästan alltid om problem och brister.

Lars: Är inte det lite tråkigt?<sup>79</sup>

Adam: Jo, det är det ju, men samtidigt kan man lära sig en hel del av det och få erfarenheter av vad som är dåligt och vad som inte fungerar så bra. Man får ju också erfarenhet av vad användare och kunder anser vara kvalitet, och den erfarenheten är väldigt bra att ha när man går vidare i andra projekt.

Eva: Jag vill också lägga till där, att kunskapen om och erfarenheten av användarnas kvalitetssyn får man genom att tala med dem och det gör man ju när de ringer och påtalar olika problem och brister.

Lars: Så den negativa feedbacken är bra ur erfarenhetssynpunkt. Den positiva feedbacken, är den lika viktig för erfarenheten?<sup>80</sup>

Eva: Ja, det borde den egentligen vara, men det är sällan som vi får någon sådan. Hälsan tiger still i stor utsträckning.

Lars: Så ni får alltså bara höra om problem och inte om fördelar eller sådant som är bra eller har blivit bättre?

Eva: Ja, delvis är det så, men visst får vi ibland höra att ett system har blivit bra eller att det fungerar bra, och man kanske till och med skålar i champagne någon gång. Till viss del kanske också en sådan kundenkät som jag nämnde i början kan ge positiv feedback. Men den positiva kritiken är så sporadisk och ytlig, den ger inte så stora möjligheter till erfarenhetsuppbyggnad.

Lars: Hur menar du då?<sup>81</sup>

---

<sup>79</sup> I10-39: Om kvaliteten upplevs som låg av användarna, får X feedback och den lär sig X mycket av och får erfarenhet av vad som ställer till problem hos användarna.

<sup>80</sup> I12-46: Feedback på att kvaliteten upplevs som hög är bra för att bygga upp kunskap som kan användas i andra system eller uppdrag, om tid ges till att reflektera över vad det är i lösningen och situationerna som gör att det upplevs som positivt och hög kvalitet.

Eva: Den negativa kritiken är naturligtvis mer detaljerad, eftersom det handlar om problem och brister på en mer detaljerad nivå – att en viss funktion inte fungerar helt korrekt eller någonting räknar fel eller något sådant. Den negativa kritiken ger helt enkelt fler detaljer. Den positiva däremot är mer av karaktären att det fungerar bra eller att det går mycket snabbare att arbeta med det, det är inte några detaljer i den kritiken och då är det svårare att lära sig någonting konkret att ha med sig i sin erfarenhet.

Lars: Man lär sig alltså inget av den?

Adam: Det är ju inte det som Eva säger, utan snarare att den kritiken är odetaljerad och därför lär man sig på ett mer subtilt plan. Man minns ju trots allt själv vad det är man har gjort och om det leder till hög kvalitet kan man kanske återanvända det en annan gång. Sedan är det naturligtvis också så att det känns bra att höra att man har gjort ett bra arbete, det är positivt för självkänslan.

Lars: Men då är det något jag inte riktigt förstår och det är varför ni inte själva undersöker vad användarna tycker om kvaliteten när det är tyst. Varför tar ni inte reda på det själva?<sup>82</sup>

Adam: Tid helt enkelt, det finns ingen tid till det. När någonting fungerar bra och det inte kommer in så många klagomål så är det dags att ta itu med nästa projekt.

Eva: Jag håller helt med Adam, det ges ingen tid till att undersöka varför kvaliteten upplevs som hög. Det finns heller ingen tid

---

<sup>81</sup> I16-34, 35, 36, 37: Om kvaliteten upplevs som hög får X positiv feedback, exempelvis att programmet är trevligt att använda, vilket indikerar att X är på rätt håll men endast omedvetet leder till ny erfarenhetskunskap.

<sup>82</sup> I12-81: X är helt säker på att man skulle kunna bli en bättre praktiker om man fick tid till att reflektera över kvalitet i ens produkter, men uppföljning och reflektion ges aldrig den tid och resurser som behövs och blir därmed bortprioriterat.



till att reflektera över vad det kan vara som är bra och utan den reflektionen blir det ju svårt att förstå vad som kan vara bra.

Lars: Varför ges inte den tiden?

Adam: Vi är problemfokuserade helt enkelt. Problem och brister måste åtgärdas, och helst snabbt, medan vi är glada om det är tyst, för då måste det fungera och användarna klara sig själva, vilket vi också vill uppnå. Sedan som sagt, när allt verkar fungera är det dags att kasta sig över nästa projekt.

Eva: Tyvärr är det så som Adam säger, men när den positiva kritiken faktiskt når oss kan den alltså bidra till en bättre bedömningsförmåga och en viktig erfarenhet.

Lars: Nu kom vi naturligt in på bedömningsförmåga. Vi har ju talat lite om hur man bedömer kvalitet och att det är svårt att sätta upp mått och mäta till exempel. Tycker ni metoder och mätning räcker för att bedöma kvalitet, förutsatt att man då mäter?<sup>83</sup>

Eva: Nej, det tycker inte jag att det gör. Att skapa mätmetoder och mått verkar väldigt svårt, särskilt när det gäller de mjukare kvaliteterna. Jag har också svårt att se att mätning skulle kunna ge allt det man behöver veta för att avgöra om kvaliteten är tillräckligt bra. Samma sak gäller med metoder och standarder av olika slag – de räcker helt enkelt inte hela vägen. Metoder och standarder fungerar ju som ramverk och en grund för kvalitet, men de måste användas med förnuft och då krävs det erfarenhet. Vissa saker, till exempel gränssnittet, bedöms egentligen helt och hållet subjektivt och då finns det inga metoder som garanterar att man uppnår hög kvalitet, även om det finns guide lines som stöd för utformningen. Du sade något om 80/20-regeln Adam och den tror jag stämmer i stor utsträckning – att mer formaliserade angreppssätt som metoder och så räcker till 80% av

---

<sup>83</sup> I7-24: X skulle vilja att man mätte kvaliteten (tex. underhållsbarhet) men endast som ett komplement till den erfarenhetsbaserade bedömningsförmågan.

kvaliteten, men för de resterande 20% behövs erfarenhet och bedömningsförmåga.

Adam: Ja, man måste ha en fingertoppskänsla också för att kunna bedöma kvaliteten. Dessutom är ju kvalitet något föränderligt och målen måste flyttas framåt hela tiden – man kan inte vila på lagrarna så att säga. Jag hade i och för sig gärna sett att man kunde mäta lite bättre, till exempel underhållsbarhet, men samtidigt tar det tid att utforma sådana mätmetoder och det är frågan om det är värt det.

Lars: Hur uppskattar man i såfall underhållsbarheten nu?<sup>84</sup>

Adam: Ja, det sker ju genom erfarenhet, att man har lärt sig att vissa sätt att göra saker och ting på är sämre än andra när det gäller underhållsbarheten.

Lars: Å andra sidan finns mått och mätmetoder inom exempelvis Software Engineering för att till exempel mäta komplexiteten. Hur kommer det sig att ni inte använder dem?

Adam: Några sådana känner inte jag till eller har hört talas om.

Eva: Inte jag heller.

Lars: Ok, men ni tror i alla fall inte att metoder och så vidare räcker för att bedöma kvalitet. Kan man då säga att det också krävs en tyst, erfarenhetsbaserad kunskap?

Eva: Ja, definitivt och det har ju också framkommit i diskussionen flera gånger att erfarenhetskunskapen är väldigt viktig och utan den skulle det för övrigt bli svårt att utnyttja metoder på ett bra och vettigt sätt.

Lars: Vad består då den tysta kunskapen av?<sup>85</sup>

---

<sup>84</sup> I7-22: När det gäller underhållsbarhet mäter man inte, utan man utbyter muntliga bedömningar mellan kollegor och ger råd.

I7-22: Underhållsbarhet mäts inte utan uppskattas genom erfarenhetsbaserad kunskap.

Adam: Ja, det blir ju svårt att säga om den är tyst, för då är den ju tyst så att säga. Men det handlar ju om erfarenheter man samlar på sig under åren i olika projekt – saker man har gjort som fungerat bra eller mindre bra, sådant som man har hört av andra med större erfarenhet. Man utvecklar med tiden en känsla för kvalitet och den är väl en del av den tysta kunskapen.

Eva: Ja, för kvalitetsbedömning är i mycket en personlig sak – jag har min syn på kvalitet och andra har en annan. Så man har personliga ramar för kvalitetsbedömning som formas av ens erfarenheter totalt sett, alltså inte bara inom yrket, och dessa ramar påverkar mycket hur du bedömer kvalitet. Sedan måste naturligtvis de ramarna anpassas efter din nuvarande arbetsituation så att de stämmer med det arbete och den arbetsplats du är involverad i nu. Det är väl där kvalitetssystemet kommer in också som jag nämnde i början.

Lars: Så i bedömningsförmågan är den tysta, erfarenhetsuppbyggda kunskapen viktig. Vad mer ingår i en bedömningsförmåga tycker ni?

Adam: Jag tycker att för att kunna bedöma kvaliteten, måste man känna till programmet eller systemet väldigt bra, samtidigt som man har kunskap om det användningsområde det är till för.

Lars: En sorts domänkunskap alltså?

Adam: Ja, det tror jag är viktigt.

---

<sup>85</sup> I1-29: Det finns aspekter på kvalitet som man inte kan uttala.

I6-43: Varje människa har en personlig referensram för att bedöma kvalitet, ett filter som påverkas av den sociala interaktionen. Så den sociala interaktionen är mycket viktig för den anpassar personens filter och syn på sanningen. Den här referensramen slår inte igenom i standarder och statistik, men har väldigt stort inflytande på den tysta kunskapen.

I8-51: Erfarenheter från arbetet och olika projekt är en stor del av bedömningsförmågan och bidrar med det avgörande för om kvaliteten blir hög eller låg och som går utöver det som kan täckas av utbildning och mallar.

Eva: Sedan tror jag också – fast det kanske är mer en egenskap än en kunskap – att man måste kunna vara lyhörd i förhållande till kunder och användare. Lyhördhet är alltså också en del av bedömningsförmågan.

Lars: Hur utvecklas och formas den här bedömningsförmågan?<sup>86</sup>

Eva: En del kommer genom utbildning, att man får grunden. Till exempel så har jag som kvalitetsansvarig gått olika kurser som handlar om kvalitetssystem och processtyrning, där en CMM-kurs är en av dem. Sedan har man naturligtvis med sig lite i bagaget från den utbildning som man gick innan man blev systemutvecklare – systemvetenskap kanske, eller kanske någon mer teknisk inriktning. Så till viss del har man en grund i utbildning och kurser, men det är som sagt bara en grund.

Adam: Fast jag måste säga att jag inte tycker av kvalitet togs upp särskilt bra i den utbildning jag gick, alltså systemvetenskap. Vi hade ingen kurs i kvalitet explicit, utan det smög sig nog mer in i andra kurser och hanterades inte som ett begrepp i sig, vilket jag tycker är synd.

Eva: Nej, det kan nog stämma.

Lars: Den fortbildning som ni har fått genom ert yrke, har den handlat om kvalitet?

Eva: I någon mening handlar det mesta om kvalitet, men om man inte är kvalitetsansvarig så handlar kurser oftast om olika utvecklingsverktyg, programmeringsspråk och databaser. När man har kommit en bit på vägen blir det aktuellt med kurser inom projektledning och teamledning och sådana saker.

---

<sup>86</sup> I1-91: Man bygger upp en erfarenhetsbank som är viktig för den viktiga luddiga och intuitiva känslan när det gäller kvalitet, det som inte går att mäta.

I5-51: Genom utbyte av erfarenheter, som är viktigt, och diskussioner lär man sig och formar en bedömningsförmåga.

Lars: Så kurser och utbildning ger en grund, men fokuserar sällan på kvalitet som sådant?<sup>87</sup>

Eva: Ja, det kan man säga.

Lars: Om det inte är utbildning som utvecklar bedömningsförmågan, vad är det då? För den utvecklas väl?<sup>88</sup>

Adam: Javisst utvecklas den, det gör den ju hela tiden. Som jag sade tidigare så är ju kvalitet något föränderligt, så bedömningsförmågan måste också ändras eller utvecklas.

Eva: Ja, så är det och det är framför allt genom erfarenhet den utvecklas. Genom att arbeta i olika projekt, samlar man under årens lopp på sig en massa erfarenheter som läggs till ens kunskapsbas och används i andra projekt. Den här referensramen jag var inne på tidigare ändras ju genom erfarenheter man får genom att arbeta med systemutveckling och har man då jobbat på lite olika företag, får man en lite bredare referensram och den har man med sig till nästa företag.

Adam: Man kan också få erfarenheter från att ha sett andra system på andra ställen, även om det är mer ovanligt. Sedan ger den här feedbacken vi var inne på tidigare också nyttiga erfarenheter.

Eva: Ja, och man kan också ha en sorts praktik ute hos användarna för att se hur deras vardag fungerar och där kan man lära sig en hel del nyttigt, för då får man på plats se hur de system man utvecklar fungerar. 10 sekunders svarstid kanske inte verkar vara så mycket, men ute hos användarna kan det vara en väldigt lång tid, och det kan vara svårt att förstå och tänka på

---

<sup>87</sup> I9-57: En del av bedömningsförmågan utvecklas till en början genom att läsa eller gå kurser.

I6-42: Regler och standarder för att bedöma kvalitet kan läras in genom utbildning

<sup>88</sup> I3-46: Genom att arbeta i projekt får utvecklarna en kunskaps- och erfarenhetsbas kring vad som fungerar bra och är gynnsamt.

innan man själv har upplevt konsekvenserna. Så erfarenhet från användarnas situation kan också vara bra.

Lars: Så det finns många källor till erfarenhet och det är främst genom erfarenhet som bedömningsförmågan utvecklas?

Eva: Ja, helt klart. Vi var ju också tidigare inne på hur viktig erfarenhet var för bedömningsförmågan och att mer normativa angreppssätt endast räcker till kanske 80% av kvaliteten och att metoder och andra formella sätt måste användas med förnuft, att det krävs fingertoppskänsla för att verkligen nå bra kvalitet. Så det finns en klar koppling mellan kvalitet och erfarenhet och därför bli ju erfarenheten den viktigaste biten i utvecklandet av bedömningsförmågan. Med erfarenhet skaffar man sig också checklistor på vad som är viktigt att tänka på så att man inte missar något.

Lars: Erfarenheten är viktig säger ni och då borde väl även utbytet av erfarenheter vara viktigt?<sup>89</sup>

Adam: Du menar att man genom att samarbeta tar del av varandras erfarenheter?

Lars: Ja, att samarbete och utbyte med kollegor är viktigt för erfarenhetsuppbyggnaden.<sup>90</sup>

Adam: Ja, naturligtvis är det utbytet väldigt viktigt. Det är ju bland annat genom att ta till sig av mer erfarna som man bygger upp sin egen erfarenhet.

---

<sup>89</sup> I4-54: Dialogen mellan utvecklare är mycket viktig och kan handla om exv. användarvänlighet.

I15-54: Man utbyter erfarenheter genom att fråga varandra om hur man löst saker och ting och tittar på varandras lösningar, samt att man har en liknande struktur så att man känner igen programmet.

<sup>90</sup> I15-72: För X är det viktigt att man inte låser in sig i ett hörn utan hela tiden deltar i en löpande dialog kring bra lösningar som kan användas igen, hur saker och ting skall lösas i program och vad som redan finns.

Eva: Jag håller med Adam – utbytet med kollegor är mycket viktigt.

Lars: Hur går det till, hur sker det här utbytet?<sup>91</sup>

Eva: Det sker på många sätt. Till exempel när man utvecklar något specifikt, någon gränssnittskomponent kanske, så kan man ta kontakt med kollegor som har utvecklat något liknande tidigare för att se hur de gjorde och vilka tips de har.

Adam: Det kan också vara att man ser något som någon annan gör, men som man inte tycker är riktigt bra ur underhållssynpunkt till exempel, och då kan man prata om det och ge tips på hur det kan göras bättre. Eller att man själv är osäker och vill ha synpunkter från sina kollegor.

Lars: Det sker så att säga i det löpande arbetet?

Eva: Ja, men det kan också vara mer systematiskt genom att man har olika kompetensgrupper som bevakar olika områden och sedan sprider den kunskapen till andra. Man kan också samla projekterfarenheter och försöka sprida dem internt, till exempel via ett intranät. Folk åker ibland på konferens, alltså någon utvecklarkonferens, och då får de berätta om den när de kommer hem och dela med sig av erfarenheten. Andra sätt kan vara att ha en projektdagbok för sin egen del och i den skriva ned sina erfarenheter. Så det finns lite olika sätt och det här med att samla och sprida projekterfarenheter tycker jag är viktigt.

Adam: Fast jag tycker att den systematiken du nämnde är rätt sällsynt och att det oftast handlar om att utbyta erfarenheter inom projektet och då kopplat till en viss produkt, det vill säga den man håller på att utveckla. Just det att man undrar om någon gjort en si eller sådan lösning och hur de gjorde den. Jag

---

<sup>91</sup> I1-91: Genom att ha coaching-mentalitet kan man dela med sig av sin erfarenhetsbank och den luddiga, intuitiva känslan kring kvalitet som är den viktiga.

tycker ofta att erfarenheterna har en tendens att stanna inom de olika projekten.

Eva: Jag vill också återvända till de personliga referensramarna, för de ändras ju genom att samarbeta med andra och höra deras synpunkter. Dessutom är det så att de olika personerna i en projektgrupp har lite olika syn på kvalitet och det gäller att försöka jämka ihop de olika synsätten så att alla i projektet har en liknande syn på kvalitet och vad det är man skall utveckla. Sedan har man ofta personer som fungerar som coacher i verksamheten som tar sig an de nyanställda och förmedlar arbetssätt och synsätt, exempelvis vad det innebär att vara en god konsult.

Lars: Sker det här utbytet med andra systemutvecklare, som då kanske inte är era kollegor?

Adam: Du menar om man träffar en konsult från ett annat företag eller någon på en konferens och diskuterar kvalitet med den personen?

Lars: Ja, exempelvis.

Adam: Det kan ju naturligtvis hända, men jag tror att det är ovanligt. Jag tycker som sagt att utbytet kring kvalitet och det här med bedömningsförmåga sker inom projekten och med de personer som ingår i projektet.

Eva: Man kan ju också väga in användarna i detta och säga att man har ett utbyte med dem, och det utbytet påverkar bedömningsförmågan också. Ibland har man med användare eller kanske så kallade superusers i projektgruppen och då sker ett naturligt utbyte med dem och genom deras större kunskap om verksamheten som skall stödjas kan man lära sig mycket.

Lars: Så bedömningsförmågan påverkas mycket av samarbete och utbyte med andra, främst kollegor, men det sker sällan särskilt strukturerat eller systematiskt?

Eva: Till viss del finns en systematik, vilket naturligtvis är olika för olika företag, men det mesta är nog ostrukturerat och osys-



tematiskt ja. Men helt klart är att det här utbytet är väldigt viktigt hur det än görs.

Lars: Spelar andra faktorer in i utvecklandet av bedömningsförmågan, till exempel teorier om kvalitet, teknikutveckling, med mera?<sup>92</sup>

Adam: För de allra flesta tror jag inte att kvalitetsteorier kommer in i bilden – för vissa kanske, men inte för de flesta. Som jag sade tidigare har i alla fall jag inte stött på det under min utbildning heller. Teknikutvecklingen påverkar såtillvida att nya verktyg leder till nya möjligheter som kan vara bättre. Exempelvis objektorientering gör det möjligt att bygga vackrare program...

Lars: Vackrare program?

Adam: Ja, att de är modulära och skiktade, vilket jag tycker är vackert. Men det är ju vad jag tycker och det betyder inget i förhållande till exempelvis prestanda, så det jag tycker är vackert får ibland stå tillbaka för andra kvaliteter. Men det har vi ju redan varit inne på. Men teknikutvecklingen kan också påverka sådant som gränssnittsutformning, så att man kan göra bättre dialoger och så vidare. Teknikutvecklingen leder dock inte automatiskt till detta, utan den ger möjligheter som objektorientering då.

Lars: Men hur påverkar det bedömningsförmågan?

Adam: Nya möjligheter leder till nya lösningar och det i sin tur till nya sätt att bedöma kvalitet på sådana lösningar. Men det måste inte göra det och flaschiga gränssnitt behöver ju inte vara bra, utan kan bli kladdiga och svårbegripliga istället.

Lars: Så en terminallösning kan vara bättre än en Windows-lösning?

Adam: Ja, är den enkel, effektiv och snabb så tycker jag det.

---

<sup>92</sup> I7-51: Det är de egna erfarenheterna som utvecklar bedömningsförmågan, inte teorier om kvalitet.

Lars: Som jag förstår det så har kvalitetsteorier inget stort genomslag i utvecklandet av bedömningsförmåga, medan teknikutvecklingen öppnar för nya möjligheter och därmed sätt att bedöma kvalitet?

Eva: Ja, så kan man nog sammanfatta det.

Lars: Jag tänkte också på det ni sade om hur viktigt det är med samarbete och utbyte med andra kollegor för formandet av bedömningsförmågan. Det borde ju innebära att ni talar med varandra om kvalitet och då undrar jag hur det går till.

Eva: Jag tycker väl egentligen inte att vi talar om kvalitet i den bemärkelsen.

Lars: Hur menar du då?<sup>93</sup>

Eva: Jag menar att vi inte talar om kvalitet med det ordet eller om kvalitet som ett begrepp och vad det innebär. När man talar om kvalitet gör man det i ett projekt i samband med en viss produkt, inte om kvalitet generellt. Det är ju inte så att vi står på fikarasten och pratar om kvalitet med varandra.

Lars: Hur kommer det sig?

Eva: Man gör inte det helt enkelt, men det kan ju hänga ihop med att man inte reflekterar tillräckligt över vad kvalitet är och inte heller har tid till det.

Lars: Håller du med om det Adam?

Adam: Ja, det gör jag nog. Jag tror inte att man är medveten om kvalitet på det viset, utan det kommer in på ett annat sätt. Men jag tror ändå att man visst reflekterar över kvalitet, men inte som att man sitter och tänker på vad det ordet kan betyda.

---

<sup>93</sup> I5-54: Diskussioner kring erfarenhet sker inte spontant på exempelvis fikaraster enligt X, utan sker genom att man aktivt söker upp någon som bollplank.

Lars: Jag tyckte att ni tidigare sa att man inte fick tid till att reflektera över vad som upplevdes som bra kvalitet och att användarna var tysta och du Eva verkar vara inne på samma linje här också, men du håller inte riktigt med då Adam?

Adam: Jo, det gör jag väl, för visst är det så att man inte har tid till reflektion, men jag tror som sagt att det ändå finns ett visst mått av reflektion.

Eva: Jag tycker man borde reflektera mer över kvalitet och ta upp den diskussionen, för den är viktig. Hur gör andra företag till exempel med kvalitet, har de speciella mätmetoder och mått eller vissa kriterier de skall uppfylla? Sådana saker skulle man kunna diskutera.

Lars: Vad skulle behövas tycker du då för att få en mer aktiv diskussion?<sup>94</sup>

Eva: Ja, att man hade seminarier och diskussioner och kanske bjöd in gästföreläsare som talar om kvalitet och annat runt omkring det. Att man reflekterar mera naturligtvis.

Lars: Så, man talar inte om kvalitet på det viset riktigt, men om ni skall förmedla en kvalitetsuppfattning till någon annan, vilka ord och begrepp använder ni då?<sup>95</sup>

Adam: Ja, man kan säga att detta är inte stabilt eller användarvänligt eller något sådant, prestandan kanske är för dålig. Men jag tycker inte att vi har speciella ord och begrepp som kan användas för att förmedla en kvalitetsuppfattning.

---

<sup>94</sup> I12-58: X anser att reflektion och diskussion kring kvalitet kan stödjas dels genom att företaget skaffar en övergripande systemutvecklingsstrategi och dels kurser och seminarier som ger en mer teoretisk kunskap kring kvalitetsbegreppet och olika sätt att mäta kvalitet.

<sup>95</sup> I2-71: Få ord för att beskriva gränssnittet och svårt att beskriva det utan att visa det.

I8-84: Har inga allmänna begrepp för att diskutera kvalitet, exempelvis underhållsbarhet, men aspekten kan diskuteras fast med andra ord och varierande beteckningar.

Lars: Men det tycker jag är lite märkligt, för i början av samtalet använde ni en hel del ord för att tala om kvalitet, så det känns lite paradoxalt att det inte skulle finnas ord och begrepp.

Eva: Ja, fast vi tänker nog inte på dem som speciella ord och begrepp för just kvalitet. När vi talar om kvalitet med varandra så sker det nog i ett mer vardagligt språk. Dessutom tycker jag att många ord och begrepp är för mångtydiga och otydliga för att man skall kunna vara precis i det man säger och veta att den man talar till också uppfattar det man menar.

Adam: Ja, jag tror också att vårt sätt att förmedla kvalitetsintryck sker genom ett mer vardagligt språk. Men visst kan man prata om ett visst system och vilken funktionalitet det har, men det blir ändå otydligt.

Lars: Du sade språk där och jag undrar om ni tycker att det finns ett speciellt språk för att förmedla kvalitetsuppfattningar?<sup>96</sup>

Eva: Det hänger väl i och för sig ihop med om vi har begrepp och ord, och jag tycker inte att vi har ett speciellt bra språk, om vi nu har något. Som jag sade nyss så är det svårt att säga vad man menar och det är svårt att veta att andra förstår vad man menar. Så det är nog samma problem här.

Adam: Jag tycker nog att det finns ett språk och det kan innehålla sådana saker som att om man följt standard, guidelines med mera. Men visst, det är kanske inte så specialiserat som du frågar efter. Förresten kan man faktiskt också se att de här mer formaliserade sakerna som standard och metod också förmedlar en viss kvalitetssyn, och då blir de ju också delar i ett språk.

---

<sup>96</sup> I2-60: Man talar om aspekter på kvalitet, men kanske inte i formen av ett språk.

I15-56: X tycker inte att det finns något speciellt språk med specialtermer för bra eller dåligt för att tala om kvalitet och uppfattningar, utan man använder vanliga ord som bra/dåligt, snyggt/dåligt, krångligt/lätt och logiskt/rörigt.

Eva: Nej, jag håller inte med Adam här. Jag tycker inte att vi har ett speciellt språk, i alla fall inget som jag känner till.

Lars: Använder ni exempel som utgångspunkt för att förmedla en kvalitetssyn, exempelvis ett väldigt bra system ni har sett någonstans?<sup>97</sup>

Eva: Till viss del använder vi exempel, men nog inte på något strukturerat och systematiskt sätt. Men om man skall tala om ett bra gränssnitt till exempel, måste man ju visa det samtidigt. Och över huvud taget utgår våra diskussioner om kvalitet från exempel, nämligen det system man just arbetar med, för att återknyta till diskussionen innan.

Lars: Så, att förmedla intryck av ett system är svårt?

Adam: Ja, det blir ju det, särskilt om man inte kan visa och provköra det.

Lars: Skulle ni vilja ha haft ett bättre språk med begrepp och exempel i såfall?<sup>98</sup>

Eva: Ja, det hade ju varit bra och det hade kunnat skärpa diskussionen om kvalitet och göra det lättare att få fram vad man menar.

Lars: Tiden går mot sitt slut och för att sammanfatta det hela är kvalitet svårt eller omöjligt att definiera, men samtidigt kan man namnge vissa egenskaper som har med kvalitet att göra. Att bedöma kvalitet sker på många sätt och tar sin utgångspunkt i kravspecar och formaliserade saker som metoder och standarder.

---

<sup>97</sup> I4-58: Man utnyttjar inte goda exempel strukturerat, men de finns där ändå.

<sup>98</sup> I3-64: Ett kvalitetsspråk med gemensamma begrepp kan vara bra och underlätta när man pratar om kvalitet, så att man pratar om samma sak.

I14-68, 69: X skulle gärna vilja ha ett språk för att tala om kvalitet och uppfattningar, för det hade gjort det enklare i den meningen att det verkligen hade uttryckt något om kvalitet. Exempelvis säger inte ordet objektorientering att det är kvalitet på en produkt som är utvecklad objektorienterat – det är inte säkert att det leder till kvalitet.

Testning och subjektiv bedömning är de sätt man bedömer på, medan mätning av kvalitet är mycket svårt eftersom det är svårt att sätta upp bra mått och skapa mätmetoder. Eftersom en ganska stor del av bedömningen är subjektiv – 80/20-regeln – är erfarenhet mycket viktigt och det är också främst via erfarenhet som bedömningsförmågan utvecklas, även om grunden till den kommer genom utbildning. Erfarenhet får man också genom den feedback som man får från användarna, men det handlar nästan bara om den negativa, eftersom den positiva sällan kommer fram. Man ges heller ingen tid till att reflektera över vad som i såfall är bra om det är tyst, vilket det är när det fungerar. En annan mycket viktig källa till utvecklandet av bedömningsförmågan, är samarbete och utbyte med andra, främst kollegor. Andras erfarenheter av olika slag är också en källa till den egna erfarenheten, dock sker utbytet mest inom projekt och inte på ett mer allmänt plan. Trots att utbytet med andra är så viktigt, diskuteras inte kvalitet som sådant i någon större omfattning, utan nästan alltid inom projekt i förhållande till det system som projektet just håller på med. Det verkar inte heller finnas vare sig speciella begrepp eller ett speciellt språk att använda i det utbytet, liksom man inte använder exempel i så stor utsträckning. Ett mer utvecklat språk och ett mer strukturerat användande av exempel är önskvärt. Att reflektera över kvalitet gör att bedömningsförmågan kan utvecklas mera och det kräver tid. Håller ni med om denna sammanfattning?

Eva : Ja, det stämmer väl.

Adam: Javisst.

Lars: Ja, då tackar jag er för att ni har tagit er tid att dela era tankar och erfarenheter kring kvalitet med oss.

## 6 IT-kvalitet i praxis – en diskussion

I detta kapitel kommer vi att föra en diskussion utifrån teorierna om kunnande, kompetens och kvalitet relaterade till resultaten av vår undersökning, vilka alltså återfinns i samtalet ovan.

Samtalet, som det är skrivet, är självständigt och besvarar en del av de frågor som undersökningen är byggd på. Därför återfinns en del svar av mer explorativ natur i själva samtalet, t.ex. hur viktigt kvalitet är, som vi inte kommer att föra en fördjupad diskussion om här. Däremot kommer vi att integrera våra teoretiska delar med undersökningens resultat, vilket alltså innebär att fokusera på IT-kvalitet, kunnande och kunskap i relation till empirin. När det gäller teorin om kvalitet handlar den mycket om begrepp och definitioner, d.v.s. det som Johannessen skulle kalla påståendekunskap, varför den passar i en diskussion kring begreppet IT-kvalitet. Men som läsaren kommer att märka är relationerna mellan IT-kvalitet, teorierna om kunskap och kunnande, och empirin få. Detta förhållande är än synligare i senare delar av denna diskussion, men det är också en av poängerna. Svårigheten att skapa relationer mellan IT-kvalitet och övriga delar ligger i hela avhandlingens perspektiv, som bottnar just i att relationerna inte behandlas i teorien.

När det gäller teorin om kunnande och kunskap, anser vi att Johannessen för ett begreppsanalytiskt resonemang medan Rolf går in på kunskapens funktion. Som vi har skrivit tidigare anser vi att de når ett likartat mål, nämligen praktisk kunskap, men alltså med två olika angreppssätt. Detta medför att Johannessen tjänar som en bra grund för en diskussion om kvalitetsbegreppet och dess betydelse, medan Rolf blir intressantare när det gäller reflektion för att utveckla kunskap. Vi har också visat att det finns vissa likheter i Johannessens och Rolfs resonemang (se avsnitt 3.3) som vi kommer att använda i diskussionen där detta är lämpligt. Dock kommer olika delar i diskussionen ha sin tyngdpunkt förskjuten åt endera tänkarens idéer beroende på deras olika utgångspunkter och angreppssätt, vilka passar olika bra till diskussionens innehåll. Sam-

mantaget är det vår förhoppning att diskussionens helhet berikas genom detta angreppssätt.

Kapitlet börjar alltså med en begreppsanalytisk diskussion om IT-kvalitet, som sedan följs av en mer färdighetsbetonad del om hur kvaliteten bedöms. Därefter följer en diskussion om vad som ingår i bedömningsförmågan och vad som ligger till grund för den och hur den utvecklas. Vi avslutar kapitlet med en reflektion över huruvida IT-kvalitet och bedömning kommuniceras mellan systemutvecklare, för att tillsist avsluta med en sammanfattning.

## **6.1 Begreppet kvalitet och dess betydelse**

Synen på kvalitet varierar mellan forskningsområden och över tiden. Inom Software Engineering verkar synen domineras av att kvalitet och dess egenskaper skall vara objektiva och absoluta. Intresset präglas av ett sökande efter generaliserbara och mätbara kvaliteter samt ett intresse för att finna fungerande tillhörande tematiska uttryck. Kvaliteten utgörs således ofta av ett mått.

Den tidiga forskningen inom MDI-området präglades av ett liknande intresse. Den är dominerad av ett mekanistiskt, biologiskt och behavioristiskt synsätt där människan ses som en informationsprocessor eller biologisk maskin. Denna syn har emellertid starkt kritiserats. Mätbara aspekter är inte tillräckliga för att förstå den komplexitet som användandet av en IT-artefakt utgör. Detta har givit upphov till andra synsätt inom MDI-området, såsom de flexibla, subjektiva och sociala perspektiven. Icke desto mindre har varje perspektiv förespråkare respektive antagonister (se kapitel 2).

Dessa båda områden har också myntat en hel mängd olika kvalitetsbegrepp och en del definitioner, d.v.s. IT-kvalitet ur ett påståendekunskapsperspektiv. Frågan är med utgångspunkt i detta hur våra systemutvecklare ser på IT-kvalitet som en påståendekunskap och hur deras förståelse ter sig ur ett begreppsperspektiv.



I vår empiriska studie har vi visat att systemutvecklarnas förståelse för kvalitet tar sin utgångspunkt både i kvaliteter som traditionellt anses som mätbara och kvaliteter som lämpligen inte går att mäta. Detta innebär att förståelsen till exempel inbegriper kvalitetsbegrepp som underhållsbarhet, prestanda, modifierbarhet, koppling, användbarhet och användarvänlighet, lättförståelighet och lättlärdhet, dock utan att särskilt många av de relativt väl etablerade begrepp som vi har tagit upp i teoridelen används.

Den bakomliggande synen på begreppen skiljer sig däremot från det som framkommer i vissa delar av teorin. Den eftersökta objektiviteten anses inte av våra systemutvecklare som fungerande eller hållbar i praktiken. Den bryter också mot systemutvecklarnas grundläggande syn på kvalitet. Systemutvecklarna anser att det är mycket svårt att definiera kvalitet och enligt vissa av dem är det omöjligt<sup>99</sup>. Kvalitet anses dessutom alltid vara en fråga om värdering. Systemutvecklarna syn stämmer bättre överens med den senare utveckling inom MDI-området som utgörs av ett bakomliggande humanistiskt perspektiv. De flesta systemutvecklarna anser att kvalitet är en fråga om en subjektiv värdering vilket liknar de antagande som görs genom att betrakta användaren av datorer som ”mänskliga aktörer.

Skälen till systemutvecklarnas syn är många, men kan sammanfattas så här: kvalitet är beroende av kontext, perspektiv och yrkesroll, tid och till viss del tekniska förutsättningar<sup>100</sup>.

---

<sup>99</sup> De olika typer av definitioner som framkommer i empirin är alla dem som vi nämner i kapitlet om IT-kvalitet (se avsnitt 2), utom den transcendentala synen, samt kombinationer av dessa. Det finns med andra ord ingen definition som helt och hållet omfattar allt det som kvalitet kan vara.

<sup>100</sup> I relation till hela den empiriska studien kan vi också dra slutsatsen att kvalitet verkar vara beroende av praxis. I studien kommer detta till uttryck genom att respondenternas syn på kvalitet generellt sett liknar varandra inom samma praxis, men skiljer sig generellt sett åt mellan olika praxisar (företag). Det bör dock påpekas att det inte har varit vårt syfte att undersöka denna typ av skillnader och därför måste dessa antaganden anses vara aningen spekulativa. Men anledningen till att vi nämner det, är för att påpeka att det även här kan finnas skillnader (kanske kulturella sådana) och att det kan ge uppslag till nya empiriska studier.

Kontext eller sammanhang anses påverka kvalitetsbegreppets betydelseinnehåll. Beroende på vilken kontext man befinner sig i får kvalitet olika innebörder och således lyfts olika egenskaper fram. Befinner man sig i ett projekt präglas innebörden och diskussionen starkt av projektets förutsättningar och av den produkt som skall utvecklas. Befinner man sig i en diskussion om hur ett projekt kan säkerställa produktkvaliteten, blir innebörden och de prioriterade egenskaperna andra. Detta tydliggörs, med undantag från vissa överlappningar, också i jämförelsen mellan olika projekt, där uppgiftskaraktären och kravspecifikationen kan variera kraftigt.

I studien kommer det också till uttryck att kvalitet är starkt relaterad till vilket perspektiv som anläggs och vilken yrkesroll man har. Perspektiv och yrkesroller styr så att säga vilka kvaliteter som ses och uppmärksammas. För en programmerare är programkoden det som utgör huvudintresset och därför blir kvalitetsbegrepp såsom läsbarhet, underhållsbarhet och sammanhållning samt koppling<sup>101</sup> mellan moduler intressanta. För en gränssnittsdesigner, som mer handhar gränssnittets form, funktion och struktur, blir det däremot intressant med kvalitetsbegrepp som användbarhet, dialogstruktur o.s.v. Perspektiv och yrkesroll styr och avgränsar således kvalitetsbegreppets innehåll.

Kvalitet betraktas inte heller som ett statistiskt eller absolut fenomen, utan det förändras över tiden. Förändrade arbetsuppgifter, omvärldsförändring och den tekniska utvecklingen kan till exempel göra det som tidigare betraktades som hög kvalitet till något som betraktas som låg kvalitet. Dock är inte allting som inte bygger på dessa nya premisser dåligt – ett äldre system kan inneha en väldigt hög kvalitet.

Kvalitet är också internt ett komplext fenomen på så sätt att det inrymmer en variation av olika egenskaper. Kvalitet är alltså sam-

---

<sup>101</sup> Sammanhållning och koppling är två etablerade begrepp som ingen av våra systemutvecklare själva använder, utan detta beskrivs i andra ordalag såsom att moduler skall ha rena gränssnitt mot varandra och att en modul inte skall göra en massa olika saker.

mansatt till karaktären och beskriver en helhet som består av olika delar. God eller hög kvalitet ligger i den komplexa helheten. Varje del bidrar till helheten, men för att beskriva detta värde måste varje förklaring referera till de enskilda delarnas bidrag till helheten.

Som vi tidigare har redogjort för representerar Johannessen (1999), i förhållande till den logiska positivismen, en annan syn på kunskap. Johannessens (ibid.) bryter mot synen att kunskap endast utgörs av empiriska eller matematiska välutformade regler som binder ihop språk och verklighet. I den meningen stämmer systemutvecklarnas förståelse av kvalitet väl överens med Johannessens kunskapssyn. Kunskap i allmänhet respektive kunskap om kvalitet skulle därmed inte vara möjlig att göra objektiv och kontextoavhängig, eller i termer av Rolf (1995), omsättas i maximer eller regler som i sig själva uttrycker all nödvändig kunskap.

Förståelsen för att kvalitet inte går eller är mycket svårt att definiera förstärks också av Johannessens kunskapssyn. Johannessen (ibid.) menar att själva definitionsakten kräver en förståelse och kunskap om begreppet ifråga. Denna förståelse syns inte i själva definitionen och därför kan den inte följas eller operationaliseras. Att behärska begreppet kvalitet kan därför inte enbart vara en fråga om vad ett begrepp språkligt innebär utan det måste även finnas ett inslag av färdighet och förtrogenhet i behärskandet av begreppet<sup>102</sup>. Därtill har vår empiri visat att förståelsen för kvalitet anses vara beroende av kontext och perspektiv samt att den förändras över tiden. Att definiera kvalitet i en logisk positivistisk anda blir därför inte meningsfullt. Däremot kan en definition användas som en precisering av begreppet, d.v.s. att den låter användare av begreppet uppmärksamma vissa aspekter, och som en utgångspunkt

---

<sup>102</sup> Som ett undantag från de definitioner som vi har råkat på, framstår Crosbys definition av kvalitet, "Quality is conformance to requirements" (1979) som möjlig att operationalisera. Behärskandet av denna definition kräver dock på samma sätt som andra definitioner både färdighet och förtrogenhet. I den meningen är definitionen inte meningsfull. Därutöver är definitionen möjlig att kritisera från antagandet att bilen Yugo är lika bra som en Mercedes eftersom båda överensstämmer med de krav som är ställda på respektive bil. Som konsument och bilägare tycker i alla fall vi att detta inte är sant.

för en närmare precisering i praxis – att använda ett begrepp som ett sorts verbalt pekdon (Polanyi, 1966). Låt oss illustrera med ett exempel. Betrakta följande generella definition:

Kvalitet är teknisk, organisatorisk och användningsexcellens.

Definitionen ger möjligheten att fästa uppmärksamheten på att kvalitet består av tre delar, nämligen teknik, organisation och användning. Dessa tre delar kan i sin tur användas som en utgångspunkt för att skapa en förståelse för kvalitet. Men märk väl att kunskapen om och erfarenheten av att kvalitet består av just dessa tre delar och vad dessa tre delar kan innebära, inte är synlig. Att behärska denna definition blir alltså en fråga om både färdighet och förtrogenhet.

Det är kanske på grund av detta som det finns så många definitioner av kvalitet. Försöken att operationalisera definitioner misslyckas just på grund av att kunskapen och kunnandet bakom definitioner inte är synlig. Därmed föds kontinuerligt nya definitioner.

I termer av Johannessen kan man också säga att definitionen är intransitiv. Den låter läsaren uppmärksamma vissa saker, men den talar inte om vad dessa saker innebär. Generella definitioner är också till sin karaktär intransitiva. Man skulle då kunna tänka sig att definiera varje del. Men man skulle då upptäcka att innehållet i de nya definitionerna behöver definieras. Man skulle då befinna sig i en oändlig regress och sammanhanget skulle snart gå förlorat.

Då behärskandet av begreppet kvalitet även blir en fråga om färdighet och förtrogenhet, blir kvalitet alltid en fråga om värdering. Med risk att bli tjatig kan Johannessens analys av verbet ”att veta...” återigen tjäna som exempel. Fast den här gången gäller det att veta vad och följande utsaga är formulerad:

Jag vet vad kvalitet är!

Man skulle här direkt kunna nämna den tidigare definitionen och sluta sig till att man vet vad kvalitet är. Men rimligen är det så att vi trots vår verbaliserade ordförståelse kan utveckla tekniskt och or-

ganisatoriskt dåliga system; och är det inte så att vi kan utveckla system med mycket hög kvalitet utan att för den sakens skull kunna definiera kvalitet? Det finns alltså inget motsatsförhållande, även om vi anser att ett kunnande och en repertoar av begrepp är en nödvändig del i utvecklandet av färdighet och förtrogenhet<sup>103</sup>. Därför kommer kvaliteten till uttryck genom praxis och de system som utvecklas och används. Det är här vi kan se vad till exempel teknisk excellens kan vara. Men för att avgöra om exemplet är ett uttryck för teknisk excellens måste man kunna jämföra, antingen implicit genom erfarenheten eller explicit genom ett annat exempel.

Att kunna ge exakta definitioner åt begrepp som kvalitet och fastställa deras betydelser och innebörder, innebär att skapa regler som skall styra begreppsanvändningen.

Sammanfattningsvis har vi kommit fram till följande:

- *Kvalitetsbegreppets innebörd är värdering. Kvalitetsbegreppet kan alltså inte helt och hållet regelstyras.*
- *Kvalitet är inte ett statiskt och absolut fenomen, utan det är i ständig förändring beroende på att arbetsuppgifter, omvärldsfaktorer och tekniska möjligheter ständigt förändras. Kvalitet är därmed ett kontextuellt fenomen.*
- *Kvalitet är ett komplext fenomen som inrymmer en stor variation av olika egenskaper och är därmed en helhet bestående av många delar.*
- *Kvalitetsbegreppet kan ges en generell definition som kan fungera som ett instrument för att fästa uppmärksamheten, men definitionen blir verkningslös när den operationaliseras eftersom kunskapen och kunnandet bakom definitionsakten inte är synlig.*

---

<sup>103</sup> Som vi påpekade tidigare fäster den uppmärksamheten kring vissa delar.

## 6.2 Hur kvalitet bedöms

I vår undersökning har vi visat att kvalitetsbedömning nästan uteslutande är baserad på erfarenhet och inte i mätning eller förment objektiva metoder, d.v.s. bedömningen av kvaliteten tog sin utgångspunkt i förtrogenhetskunskapen. Då förtrogenhetskunskap svårligen låter sig verbaliseras blir bedömningen här alltid en fråga om värdering. Rent principiellt utesluter detta dock inte att låta ett explicit exempel tjäna som referenspunkt. Det framstår också som möjligt att tillämpa mätning och komma fram till ett numeriskt värde för detta exempel. Det vi inte får glömma är, att även om vi har riktvärden att jämföra med, som ger vid handen att något är bättre eller sämre än ett visst värde, räcker det inte för att avgöra om det är bra eller dåligt. Låt oss säga att vi har 7 som ett mått på underhållbarhet. Riktvärdet är 6 på en skala från 1 till 10. De slutsatser som kan dras är att det första värdet är bättre än riktvärdet, men vi vet fortfarande inte om det är bra. Precis som empirin visar beror detta på den kontext inom vilken mätningen sker. Värdet 7 skulle kanske i en lagerhanteringssystemkontext betraktas som bra, men i en annan kontext, till exempel reaktorstyrning, kanske värdet 7 för underhållbarhet betraktas som dåligt<sup>104</sup>. Bedömningen av kvalitet blir, oavsett att den mäts, alltså en fråga om värdering och jämförelse som tar sin utgångspunkt i den aktuella kontexten.

En annan aspekt som kan förklara att kvalitetsbedömning nästan alltid tar sin utgångspunkt i förtrogenhetskunskapen, kan sökas hos Polanyi, som enligt Rolf (1995) identifierar olika typer av uppgifter och handlingar med olika komplexitet i förhållande till kvalitetsregler och den grad av koordination som krävs mellan deltagare som utför handlingen. En polycentrisk uppgift, d.v.s. en uppgift vars dellösningar är ömsesidigt beroende av varandra, kräver koordination mellan deltagarna i en social lösningsprocess – det som en medlem gör i ett projekt är beroende av och påverkar vad andra kan göra. Den sociala ordning som krävs för att lösa en polycent-

---

<sup>104</sup> Jämför till exempel med den önskvärda robustheten i ett system som används i ett kärnkraftverk respektive på skrivbordet.

risk uppgift kan inte styras centralt – den kräver spontan koordination.

En komplex verksamhet med flera inblandade aktörer i en form av praxis kan svårligen centralstyras, utan den kräver stor frihet för de inblandade parterna att på olika sätt påverka sitt handlingsutrymme och sitt arbetssätt. Att försöka slå fast vad kvalitet är och sätt att bedöma detta, vilket framstår som en komplex uppgift både sett ur teorin och vår empiri, skulle vara ett sätt att centralstyra och reglera bedömningsförmågan, som då skulle övergå till att vara ett system av maximer i Rolfs terminologi. Maximer kan dock inte ersätta erfarenheten eller förtroendet, utan dessa krävs för att maximerna skall kunna brukas varvid vi är tillbaka i att kvalitet ändå är en förtroendebaserad värdering<sup>105</sup>.

Om vi då bortser från en strävan efter att reglera själva kvalitetsbegreppets betydelse och sätt att objektivt försöka bedöma kvalitet genom mätning och ser på hur bedömning görs av våra systemutvecklare, så framstår även andra inslag i kvalitetsbedömningen som ansatser till reglering. I systemutvecklingsprocessen används bl.a. guidelines, metoder och standarder som sätt att uppnå kvalitet och på samma sätt som med mätning kan man fråga sig om dessa kan användas som en centralstyrning som inte behöver vila på en förtroendeskunskap.

Om man utgår från Stolterman (1991) och Hoberg (1998), så är det högst troligt att systemutveckling är en polycentrisk uppgift som inte kan centralstyras. Det borde innebära att metoder, guidelines och standarder aldrig kan vara tillräckliga för att kvaliteten skall bli hög på datasystem och -program – ett mekaniskt följande av metoder och uppfyllande av standarder garanterar inte kvalitet. Inte heller kan dessa normativa inslag räcka för att bedöma kvalitet. Detta skulle då innebära att det krävs ytterligare kunskaper utöver att kunna följa metoder och standarder för att dels kunna

---

<sup>105</sup> Som vi har skrivit tidigare så anser Wilson & Hall (1998) att skillnader i tolkning och förståelse för IT-kvalitet är skälet till att kvalitetssystem tenderar att misslyckas.

uppnå hög kvalitet, dels för att kunna bedöma kvalitet – det krävs också det som Johannessen kallar förtrogenhetskunskap.

Detta stämmer helt överens med vår undersökning eftersom endast en person anser att metoder och standarder borde räcka för att nå hög kvalitet och att det sålunda inte borde finnas något utanför normer som krävs ytterligare. Alla andra anser att normativa angreppssätt inte räcker hela vägen, man talar bland annat om 80/20-regeln, d.v.s. att metoder och standarder räcker till 80% av kvaliteten, medan 20% ligger utanför. Metoder och standarder skall också tjäna som stöd, inte som styrning.

Ytterligare sätt att bedöma kvaliteten är oftast tester av olika slag, ibland genom någon form av granskning, kritik från kollegor och inte minst genom att man själv som systemutvecklare kan uppfatta eller upptäcka att kvaliteten i någon form brister. Alla dessa sätt bygger på en värdering byggd på erfarenhet, snarare än värderingsfria kriterier och tillvägagångssätt. Detta sammantaget med det vi skrivit ovan, leder till att kvalitetsbedömning måste vila på erfarenhet och förtrogenhet och att bedömningsförmågan därmed har mer innehåll än färdigheten att följa metoder och göra beräkningar för att uppnå värderingsfria mål.

Sammanfattningsvis har vi kommit fram till följande:

- *Bedömning av kvalitet sker genom värdering och jämförelse, vilket därmed omöjliggör ickekontextuella och objektiva kvalitetskriterier.*
- *Kvalitet uppskattas alltså på en mängd olika sätt, dock sällan genom mätning eller andra förment objektiva och normativa sätt. Metoder och beräkningssätt räcker inte för att kunna bedöma kvalitet.*
- *En viktig del av bedömningen av kvalitet vilar därför på erfarenhet och förtrogenhet, och därmed är kvalitetsbedömningen i väsentliga avseenden en personlig värdering.*
- *Kvalitetsbedömning kan därmed endast till delar regelstyras och dessa regler kan endast tjäna som stöd för bedömningsförmågan.*



### **6.3 Bedömningsförmågans innehåll**

Som vi har sett hos både Rolf och Johannessen, är regelföljande ett viktigt inslag i praktisk kunskap och praktiskt kunnande. En viss typ av regler eller ett visst innehåll i regler, är de metoder och tumregler som förmedlas via kultur och tradition. I systemutvecklingspraktiken finns det en lång tradition av systematisering, standardisering och metodanvändning. Det finns naturligtvis en uppsjö av olika systemutvecklingsmetoder med olika fokus och underliggande uppfattningar om systemutveckling som process och datasystem som fenomen. Det finns också gott om olika former av checklistor, guidelines och andra "rättesnören" för att styra eller stötta systemutvecklingsprocessen. Till detta kommer olika former av standarder och standardförfaringssätt.

Det intressanta i detta är om de systemutvecklare som vi har intervjuat, ser denna typ av normerande regler som stöd eller som en form av absolut styrning, d.v.s. att frågan är om det finns inslag i kunnandet om kvalitet och i bedömningsförmågan som går utöver det som normer föreskriver.

När det gäller processen att utveckla system, har redan Stolterman (1991) besvarat denna fråga genom sin undersökning av systemutvecklarens praxis och där kunnat visa dels att systemutvecklare, i princip mot intentionerna i åtminstone den tidens systemutvecklingsmetoder, tidigt eller omedelbart börjar att tänka på en vision av det färdiga systemet, dels att de använder systemutvecklingsmetoder som ramverk för att utforma privata metoder – man tar delar av dem som man tycker fungerar bra i en viss situation och anpassar detta till en egen metod. I Hoberg (1998) finns en berättelse om ett projekt som hamnade helt fel genom att helhjärtat följa en bestämd metod, som gav fel fokus i förhållande till uppgiften. Så att systemutveckling skulle kunna ske som ett mekaniskt följande av normativa regler, finns det inga empiriska belägg för som vi har stött på och det skulle, i förhållande till teorier om praktisk kunskap och praktiskt kunnande som vi har presenterat, vara förvånande om man skulle stöta på detta.

Om vi återvänder till Rolfs tolkning av Polanyi och teorin om personlig, praktisk kunskap, så är den egna erfarenheten oerhört viktig för att kunna förstå de av en tradition förmedlade reglerna för en färdighet. Den personliga kunskapen är alltså en sammansmältning av subjektiv erfarenhet och det som traditionen förmedlar i form av tumregler och värderingar etc. Som vi såg tidigare i avsnitt 3.2.3 kan inte regler ensam avgöra hur man praktiserar en färdighet och de kan vara stöd endast om de integreras med praktisk kunskap inom färdighetsområdet, eller i Johannessens terminologi, med förtrogenhetskunskap.

Den ytterligare kunskap som då krävs har vi kallat för tyst kunskap i vår undersökning. Vi har då, som vi har redovisat i undersökningskapitlet (kapitel 4), använt denna term för erfarenhetsbaserad kunskap som är svår att verbalisera. Denna typ av kunskap verkar vara väldigt viktig för våra systemutvecklare, men är naturligtvis också svår att förmedla innehållet i. Att den är erfarenhetsgrundad ligger i själva vår frågeställning, men likväl framstår just erfarenheten som väldigt viktig och den framförs som den mest betydelsefulla grunden för bedömningsförmågan.

Om vi återvänder till Rolfs polanyitolkning skulle det innebära att systemutvecklarna har tyst fungerande kunskap som de använder för att åstadkomma och bedöma kvalitet på det som är i fokus för deras handling, nämligen produkten. Den här tysta kunskapen utmärks också av en sorts fingertoppskänsla, som gör att kvaliteten kan bli hög. Att nå hög kvalitet innebär för flera av våra systemutvecklare, att överträffa kundens förväntningar och krav, att förstå även outtalade önskemål och behov hos kunder och användare. Kvalitet må vara att uppfylla kravspeccen – hög kvalitet är att överträffa den.

Bedömningsförmågan kräver alltså tyst, erfarenhetsbaserad kunskap samtidigt som själva utvecklingsprocessen framstår som polycentrisk givet den tidigare diskussionen. Om systemutveckling betraktas som en polycentrisk uppgift kan den inte planstyras i en stor omfattning. Det kräver i sin tur utrymme för mer spontan koordination och kompetens, d.v.s. att reglerna kan påverkas för att upp-

nå en förbättring av dem. Att helt planstyra uppgiften via metoder och andra normer, det som ligger inom Johannessens begrepp påståendekunskap, är inte möjligt, utan inom färdighetsområdet behövs kunnande som går utanför dessa – det behövs en bedömningsförmåga som också bygger på erfarenhetskunskap. Detta kunnande uttrycks av våra systemutvecklare som tyst kunskap. Frågan är då vilken roll metoder, standarder, guidelines med mera kan spela för bedömningsförmågans innehåll.

Rolf har, utifrån sin tolkning av Polanyi, en diskussion om förhållandet mellan en karta och kunskap, där han menar att kartan i sig inte utgör kunskap, men att den kan ge kunskap; att kunskap kan utvinnas ur den beroende på sociala regelsystem och mänskliga tillämpningar av detta system. Kartläsningen vilar också på personlig kunskap som går utanför själva regelsystemet. Därför innehåller kartan, enligt Rolf (1995), information som genom att behandlas i enlighet med sociala meningsregler och genom att tolkas med hjälp av personlig kunskap, kan resultera i kunskap.

Så om vi jämför med Rolfs kartmetafor, innebär det att metoder, standarder och guidelines i sig inte utgör kunskap, men man kan utvinna kunskaper ur dem och man kan ha dem som stöd för sitt kunnande, men inte ersätta det. Normerande angreppssätt måste användas på ett meningsfullt sätt, vilket förutsätter kunskaper som ligger utanför normerna själva. När normen inte håller måttet måste man som systemutvecklare välja en annan väg genom att bryta mot normen för att uppnå hög kvalitet, vilket också sker.

Sammanfattningsvis har vi kommit fram till följande:

- *Att nå hög kvalitet innebär mer än att uppnå målen i kravspecifikationen, hög kvalitet är att åstadkomma ett mervärde för kunden.*
- *För att kunna göra detta måste systemutvecklare ha erfarenhetskunskap, i detta sammanhang tyst kunskap, om och en fingertoppskänsla för vad som leder till hög kvalitet.*
- *Det behövs därmed kunskaper som ligger utanför det som kan täckas av regler och normer. De kan tjäna som en grund för kvalitetsbedömning, men*

*kan behöva brytas mot när den praktiska erfarenhetsbaserade kunskapen så kräver.*

## **6.4 Bedömningsförmågans utveckling och grund**

Den tysta kunskapen eller det tysta kunnandet är alltså i det här perspektivet, en viktig ingrediens i våra systemutvecklarens yrkesutövning. Den hämtar sin näring från flera olika håll. Den kommer via erfarenheter från olika projekt man har varit med i, olika typer av lösningar man har gjort, andra arbetsplatser man har varit på tidigare, olika roller man har haft i systemutvecklingsprojekt etc. Någon talar om personliga ramar för kvalitetsbedömning, ramar som påverkas i den sociala situation som man befinner sig i på just den arbetsplatsen eller det projektet och som ligger till grund för ens egen uppfattning om och bedömning av kvalitet. Dessa ramar måste jämkas med andras uppfattningar – det måste ske en koordination av kvalitetsuppfattningar inom såväl projekt som företag.

Det som alltså framstår som den viktigaste källan för utvecklingen och formandet av bedömningsförmågan är erfarenheten. Kurser, teorier om kvalitet och den tekniska utvecklingen har betydligt mindre betydelse. Domänkunskap, alltså kunskap om brukskontexten, är också betydelsefull men även den vilar till största delen på personlig kunskap och erfarenhet, t.ex. genom någon form av praktik där systemet används eller via kontakter med brukarna.

En viktig bit i erfarenhetsuppbyggnaden är också den erfarenhetsöverföring som kan ske mellan olika systemutvecklare – man frågar varandra till råds eller kommenterar andras lösningar. Att ta till sig andras råd och kunskaper upplevs som viktigt för att bedömningsförmågan skall fortsätta att utvecklas.

En annan källa till erfarenhetsuppbyggnad och utveckling av bedömningsförmågan, borde vara den feedback som kommer från brukare och kunder. Anser då våra systemutvecklare att feedbacken är betydelsefull för att bygga upp sitt kunnande och sin bedömningsförmåga?

När kvaliteten på produkten upplevs som låg av användarna eller avnämarna, resulterar det i att dessa ringer eller skickar felrapporter. Den negativa kritiken, d.v.s. felen och problemen, märks alltid och måste hanteras. Den upplevs genomgående som viktig och betydelsefull, inte minst eftersom problem helt enkelt måste åtgärdas, men också genom att den ger erfarenhet av vad som kan ställa till problem och vad som uppfattas som dåligt av användarna. Eftersom fel måste åtgärdas ges också automatiskt tid till detta. Den negativa kritiken är också mer detaljerad och ger därför mer konkret återföring, vilket därmed leder till mer påtagliga erfarenheter.

När det gäller den positiva kritiken är bilden mer splittrad. Vissa anser att den definitivt kommer fram, medan andra tycker att det är tyst om kvaliteten upplevs som bra eller hög – hälsan tiger still. ”No news is good news” är en annan paroll som passar in, även om alla gärna vill ha positiv kritik. Den är alltså mer sporadisk, kanske till och med tyst, och i allmänhet betydligt mindre detaljerad, och det är därigenom svårare att bedöma på vilket sätt den leder till en erfarenhetsuppbyggnad och en utveckling av bedömningsförmågan.

Den positiva kritiken upplevs dock som viktig och anses av flera kunna leda till erfarenhetskunskaper, som i sin tur är väldigt viktiga för bedömningsförmågan och kvalitetsuppfattningen, även om de alltså har svårt att uttrycka hur den påverkar. Det kan ibland vara så att den positiva kritiken faktiskt leder till en mer konkret lärdom, även om feedbacken oftast verkar vara av en mer ”klapp-på-axeln”-karaktär, än detaljerad, positiv feedback. Dock är det så att systemutvecklarna själva kan reflektera över kvaliteten genom att göra kopplingen mellan att något upplevdes som bra och den kunskap de har om hur de utformat det som gav upplevelsen, och det är en kunskap och en erfarenhet de kan ta med sig.

Med detta som bakgrund framstår det som att den allra viktigaste källan till bedömningsförmågans utveckling är erfarenhetskunskap, som får sitt innehåll från flera olika håll. Den rent personliga erfarenheten av olika projekt och arbetsplatser paras med erfarenhet från brukskontexten i större eller lägre grad, med den kontakt man

har med brukarna och kunderna och inte minst med den feedback, främst negativ, som kommer från brukare och kunder. Allra viktigast för bedömningsförmågan utveckling och grund är alltså förtrogenhetskunskapen, den som är erfarenhetsbaserad och tyst fungerande.

Sammanfattningsvis har vi kommit fram till följande:

- *Den viktigaste grunden för utvecklingen av bedömningsförmågan är den personliga erfarenheten av olika projekt, kunder, brukare etc. d.v.s. den tyst fungerande förtrogenhetskunskapen.*
- *Ett viktigt inslag i denna erfarenhet är den feedback systemutvecklare får från kunder och brukare. I första hand handlar det om negativ feedback när den positiva är förhållandevis sparsam. Den negativa feedbacken innehåller också fler detaljer för erfarenhetsuppbyggnad.*
- *Ett annat viktigt inslag är det utbyte som sker mellan systemutvecklare och därigenom möjligheten att dra nytta av andras artikulerade erfarenheter och åsikter.*

## **6.5 Reflektionens roll i utvecklingen av bedömningsförmågan**

När nu erfarenhet framstår som så viktigt, blir det intressant att diskutera hur reflektion påverkar erfarenhetsuppbyggnaden. Att kunna och få möjlighet att reflektera framstår som nödvändigt för att erfarenheten skall kunna bli kunskap, vare sig denna är av färdighets- eller förtrogenhetskaraktär. I Rolfs tolkning av Polanyis kunskapssyn är möjligheten till reflektion för att förbättra know-how en nödvändighet – en person är kompetent om hon eller han genom reflektion ges möjlighet att påverka de sociala reglerna och bedömningskriterierna för kvaliteten på en handling, vilket i sin tur bygger på att underförstådd kunskap genom symbolisering och verbalisering, transformeras till reflekterad kunskap. Denna kunskap kan genom en social process av kritik, värdering och analys utökas och ge underlag för en förbättrad underförstådd kunskap –

kunnande kan bli kunskap som kan leda till bättre kunnande. Praxisens know-how, d.v.s. praktisk skicklighet vars kvalitet bedöms socialt, kan därmed utvecklas. Att inte kunna påverka regler och bedömningskriterier leder alltså till ett inkompetensförhållande.

Att reflektion är en viktig komponent i såväl praktisk kunskap som för kunskapsutveckling, är naturligtvis ingen nyhet och heller inte unikt för Polanyi. Schön (1983, 1987) och Molander (1996) har också utvecklat de tankarna. För Schön är den praktiska kunskapen en integrerad helhet av handling och reflektion – man agerar, ser resultatet och utifrån det agerar man igen. På det viset lyssnar man på situationens ”svar” och man har en konversation med situationens ”material” – man har ett kunnande-i-handling och en reflektion-i-handling, som uppvisar en integrerad och omedelbar helhet. Denna reflektion kan ske på flera plan, genom att man kan ta ett eller flera steg tillbaka och exempelvis reflektera över reflektionen-i-handling. Kunnande-i-handling kan också bli kunskap-i-handling, genom en ”teoretisering” av kunnandet (Schön, 1983), vilket rimligen borde vara besläktat med Rolfs verbalisering och symbolisering.

Hur förhåller sig i såfall detta till våra systemutvecklare och deras praxis? Denna fråga har naturligtvis många bottnar och vi har inte haft något speciellt frågetema kring detta, utan låtit det komma in på olika sätt och under olika teman i intervjuerna. Termen reflektion är också problematisk, eftersom man kan se reflektion som en biprodukt av att försöka lösa problem, eller se reflektion som en poäng i sig – man kan alltså fokusera på olika funktioner hos reflektionen.<sup>106</sup>

Hos våra systemutvecklare verkar reflektionen ske inom projektgrupper i anslutning till den uppgift man precis har för handen. Det sker alltså en social koordination inom projektgrupper, där kvalitetsuppfattningar måste jämkas så att de blir mer gemensamma. En

---

<sup>106</sup> Vilket med en variant på polanyisyn skulle innebära att reflektion kan ha en tyst funktion, eller att reflektionen kan vara i fokus och därmed inte ha tyst funktion.

mer övergripande reflektion verkar alltså saknas, vilket borde leda till att en koordination på en mer generell nivå endast sker i liten grad, åtminstone på ett spontant plan. Kvalitetssystem syftar till just denna typ av koordination, men blir samtidigt en form av planstyrning, beroende, naturligtvis, på vilken form kvalitetssystemet antar. Vad man använder som tillägg till de mer planstyrande kvalitetssystemen, är olika typer av mentorer som förmedlare av synsätt och arbetsmetoder – vilket är av en mer spontan/icke-planstyrd karaktär, dock för förmedling snarare än för reflektion.

När det gäller reflektion-i-handling och kunnande-i-handling fungerar även systemutvecklingen på det viset, eftersom man gör någonting, testar det och ser vad som händer. Utifrån det kan man dra slutsatser om huruvida den sak man har gjort, en dialogruta kanske, fungerar bra och som det är tänkt, eller ännu hellre bättre än tänkt. Man kan också ta ett steg tillbaka från situationen genom större tester som sker enligt vissa planer och som genererar vissa protokoll, eller genom mer systematiserade granskningar av exempelvis kod, granskningar som inte var så vanligt i våra intervjuer, eller av funktionalitet i förhållande till kravspecen, vilket var vanligare. Vi får dock inte intrycket att reflektionen är i fokus här, snarare att problemlösningen är det, vilket naturligtvis kräver reflektion för att kunna finna fel och åtgärda dessa.

Vid den andra typen av reflektion, d.v.s. att den är i fokus och ett viktigt moment i sig, är ett genomgående tema i våra intervjuer att systemutvecklare sällan ges möjlighet att reflektera över vad kvalitet kan betyda, vare sig som ett generellt begrepp eller mer specifikt kopplat till de system och program som de utvecklar. När ett projekt är avslutat, är det omedelbart dags att ta itu med nästa – det gives ingen eller väldigt lite tid till eftertanke. Det finns sällan någon form av systematik eller organisatorisk struktur som kan ge utrymme för reflektion – det verkar inte vara en del av yrket eller av arbetet att ha någon distanserad reflektion. Vad vi menar här är att den intellektuella processen att reflektera över begreppet kvalitet, dess betydelse och innehåll på distans från den direkta vardagliga situationen, i syfte att lära sig mer, i stor utsträckning verkar saknas. Det förs inga generella diskussioner om kvalitet, t.ex. vid en fika



eller lunch, och man har inte heller någon form av seminarier kring detta, utan diskussioner om kvalitet och reflektioner över kvalitet verkar vara knutna till projekten och projektgrupperna, d.v.s. den pågående handlingen att utforma system och program. Detta torde innebära en form av reflektion-i-handling, vilket inte är en brist, men förhållandet gör också att man missar ytterligare nivåer av reflektion som kan vara nyttiga.

Genom att reflektera över kvalitetssyn och bedömningsförmåga, d.v.s. den tyst fungerande kunskapen, kan denna komma i fokus för uppmärksamheten och undersökas, på ett liknande sätt som de övre nivåer av reflektion som Schön (1983) talar om. En god grund för reflektion över kvalitet på de produkter man tillverkar, borde vara den negativa och positiva kritik som kommer från användare och kunder. När kvaliteten upplevs som hög respektive låg borde det finnas underlag för reflektion och lärande.

Som vi skrev tidigare uppfattar våra systemutvecklare att det sällan ges tid till reflektion, särskilt inte om kvaliteten uppfattas som bra eller hög och ett genomgående förhållande verkar vara att man som systemutvecklare är problemorienterad – fokus är på det som inte fungerar. När kvaliteten upplevs som låg av brukare och kunder hörs alltså alltid den kritiken och det leder till handling. Det är ju också en typ av reflektion, men syftet är inte primärt att genom reflektionen lära, utan syftet är att kontrollera och upptäcka fel och felaktigheter som behöver åtgärdas – reflektionen som sådan är inte i fokus. Rimligtvis borde detta ändå leda till en reflektiv process som kan leda till erfarenhetsuppbyggnad, om den typen av reflektion är önskvärd.

Att man är problemfokuserad och att man inte ges tid, verkar vara orsaken till att man inte reflekterar på ett övergripande och kunskapsmässigt plan över kvalitet som fenomen och vad det betyder, innebär och innehåller. Problemfokusering och tidsbrist verkar också ligga bakom varför man inte systematiskt undersöker den av användare och kunder upplevda kvaliteten när denna är hög.

För att kunnandet och bedömningsförmågan skall kunna utvecklas, eller åtminstone utvecklas bättre, framstår reflektion och en möjlighet att fördjupa förtrogenhetskunskapen som viktigt. Det är också intressantare att se kunnandet som en social aktivitet än att fokusera på kunskap som en statisk egenskap, vilket förutsätter att den sociala kunskapsprocessen tillåts äga rum. Till viss del gör den det hos våra systemutvecklare, men sällan i akt och mening att fokusera på bedömningsförmågan och fenomenet kvalitet i sig själv. Detta märks också genom att de systemutvecklare som vi intervjuade sällan reflekterade över dessa frågor även om de upplevdes som viktiga.

Om kunnandet skall ses som en social process, där alltså reflektion och olika former av symbolisering av kunnandet är nödvändiga inslag, blir kommunikationen mellan systemutvecklare intressant när det gäller IT-kvalitet och kvalitetsbedömningar. En social kunskapsprocess utan kommunikativa inslag verkar mycket svår och frågan är då hur kommunikationen mellan systemutvecklare i vår undersökning ter sig kring kvalitet och bedömningar.

Sammanfattningsvis har vi kommit fram till följande:

- *Reflektion över såväl hög som låg kvalitet sker i relativt liten utsträckning och sällan på ett systematiskt sätt.*
- *Inte heller sker någon reflektion över bedömningsförmåga och fenomenet kvalitet på ett övergripande plan, i syfte att förbättra och stärka praxis know-how.*
- *Utrymmet för reflektion inom praxis begränsas av ett fokus på tid och problem i samband med specifika projekt och en, i våra ögon, omedvetenhet om vikten av reflektion för att förbättra kunnandet kring kvalitet.*

## **6.6 Sker en kommunikation kring IT-kvalitet och bedömning?**

Vi har tidigare redogjort för reflektionens betydelse. För Rolf (1995) är reflektion en fokusering på kunnandet och för att vidareutveckla kunnandet behövs en verbaliserad reflektion – ett utbyte.

I vår studie har vi delvis funnit stöd för denna tanke. Utbytet mellan systemutvecklare ansågs som en nödvändig del av kompetensutvecklingen. Det är bland annat genom utbyte man bygger upp erfarenheten. Även om det sker ett mer formaliserat utbyte, till exempel genom kompetensgrupper som bevakar ett område och sedan sprider det vidare, sker utbytet oftast i det löpande arbetet, inom ett visst projekt. Utbytet är därför starkt relaterat till det aktuella projektet och de problem och lösningar som finns inom detta.

Utbytet kan ta sin utgångspunkt i att man till exempel behöver tips för hur en gränssnittskomponent skall utformas eller att man ger synpunkter på någon annans programkod. Systematiken för att sprida erfarenheter gjorda inom ett projekt saknas dock som sagt och därför har dessa erfarenheter en tendens att stanna inom projektet och hos deltagande personer.

Utbytet sker alltså i stor utsträckning mellan projektdeltagande personer som generellt sett har liknande referensramar. Ibland använder man sig emellertid av möjligheten att inkludera användare i projektet. Utbytet får därmed en annan karaktär och sker mellan olika roller.

Trots att det poängteras att kvalitet är otroligt viktigt för systemutvecklare, pratas det alltså inte generellt om kvalitet. Diskussionen lyfts varken upp (utanför projekten) eller fram (synliggörs).

Därtill anser systemutvecklarna generellt sätt att de saknar ett språk för att kunna kommunicera kvalitet. Paradoxalt nog, trots att systemutvecklarna kan nämna en rad olika kvaliteter, används inte dessa ord i en dialog kring kvalitet. Förvisso används till exempel standarder och riktlinjer som stöd för att förmedla vissa kvaliteter. Dessa anses dock vara för grova instrument och är inte tillräckligt

specialiserade eller förfinade för att vara goda utgångspunkter för att föra en djupare dialog om kvalitet. Undantagsvis används även exempel på andra system som utgångspunkt för att förmedla kvalitet. Detta tillhör dock ovanligheten.

Som vi påpekade tidigare använder de systemutvecklare vi intervjuat endast ett fåtal av de relativt väl etablerade begrepp som vi tog upp i teoridelen. Istället verkar kommunikationen och utbytet kring kvalitet och bedömningar ske genom vardagsspråket. Man tycker dock att möjligheterna att diskutera som ges genom vardagsspråket är grova och knapphändiga och önskar därför ett ”bättre språk”.

Om vi knyter an till Johannessens förståelse av kunskap, där färdighets- och förtrogenhetskunskap utgör viktiga inslag, är det möjligt att peka på vissa skillnader. De tysta inslagen, i både färdighets- och förtrogenhetskunskap, resulterar i att vi med nödvändighet får använda oss av indirekt kommunikation, såsom metaforer, analogier, exempel o.s.v. I vår studie visade det sig att man enbart hade tillgång till, som de själv uttryckte det, ett knapphändigt vardagsspråk. Utifrån både Johannessen och Rolf skulle alltså systemutvecklarna som ingick i vår studie, ha sämre möjligheter att utveckla kompetensen kring kvalitet, just på grund av bristen av uttrycksformer. Denna brist kan dock ha många orsaker.

En anledning kan vara den situation där reflektionen ägde rum (d.v.s. intervjusituationen). Systemutvecklarna reflekterade utanför den praxis som normalt utgör deras arbete, vilket egentligen går emot var Johannessen menar att färdighets- och förtrogenhetskunskap kommer till uttryck.

Men det kan också vara så, som vi också mer sluter oss till, att systemutvecklare varken är tränade i eller förfogar över den repertoar av begrepp och medel som behövs för att uttrycka den här formen av kunskap. Detta kommer till uttryck genom att de faktiskt rudimentärt använder vissa begrepp såsom prestanda och underhållsbarhet samt standarder och riktlinjer för att uttrycka vissa kvaliteter. De verkar också medvetna om vilka brister som finns i

deras språk för att uttrycka kvalitet, vilket enligt oss styrker detta argument.

Sammanfattningsvis har vi kommit fram till följande:

- *Systemutvecklare kommunicerar sinsemellan om kvalitet, men nästan alltid i samband med ett specifikt projekt och sällan generellt för att utveckla sin kunskap och sitt kunnande. Detta trots att kvalitet framhålls som mycket viktigt.*
- *Möjligheten att kommunicera lider av att systemutvecklarna inte upplever att det finns precisa och utvecklade begrepp för kvalitet; de använder inte heller flertalet av dem som vi har funnit i vår teoretiska genomgång.*
- *De upplever inte att det finns något speciellt språk med precisa begrepp, utan utbytet sker genom ett knapphändigt vardagsspråk.*
- *Användandet av exempel som stöd för utbyte och förmedling av kvalitetsuppfattningar är ovanligt och sällan systematiskt. Istället är det specifika och aktuella projekt som tjänar som exempel i ett utbyte, men inte med syftet att förmedla kvalitetsuppfattningar i en generell ansats för att förbättra bedömningsförmågan.*

## 6.7 Sammanfattning

Vi har nu fokuserat på vad vi anser vara viktiga aspekter på empirin och teorin, och sambanden dem emellan, och funnit att kvalitet som begrepp och fenomen inte, eller endast i liten utsträckning, hanteras som ett kunnande på ett mer medvetandegjort plan hos våra systemutvecklare. Vi har inte haft för avsikt att värdera deras kunnande om kvalitet och fälla några omdömen om den, utan vår avsikt har varit att belysa hur de uppfattar kvalitet och ställa det i relation till teorier om kunskap, kunnande och kompetens som de kommer till uttryck hos Johannessen och Rolf.

Vår argumentation har utgått ifrån att kvalitet som fenomen måste vara synligt som ett inslag i systemutvecklarens professionalism och att det borde ha en vikt som gör att det kan hanteras på ett själv-

ständigt sätt för att utveckla bedömningsförmågan. Olika sätt att utveckla kunnandet och kompetensen när det gäller IT-kvalitet på produkten, kan vara att mer systematiskt arbeta med exempel och begrepp som en sorts symbolisering/ett synliggörande samt reflektion över detta, i syfte att utveckla sin bedömningsförmåga. Kvalitetssystem i all ära, men att styra processen för att uppnå kvalitet utan att diskutera, visa och reflektera över vad kvalitet på produkten kan vara, verkar för oss en smula märkligt och ensidigt.

Den syn på kvalitet som kommer fram hos våra systemutvecklare är att IT-kvalitet är ett komplext fenomen, vars bedömning utgår från såväl perspektiv som intressent. Att definiera kvalitet och att finna objektiva mått framstår därför som mycket svårt och mycket av bedömningsförmågan och förståelsen för kvalitet tar sin utgångspunkt i erfarenheten eller förtrogenhetskunskapen. Bedömningsförmågan låter sig därmed inte regleras eller fångas i maximer som inte behöver vila på erfarenhet. Sett ur detta perspektiv framstår det som en aning märkligt att olika sätt att förenkla för erfarenhetsuppbyggnad, exempelvis genom verbaliserad reflektion och användning av exempel, inte används. Skälet till detta är i allmänhet brist på tid men också brist på stöd för denna process, samt att fokus inte riktas på betydelsen och innehållet i kvalitet och bedömningsförmåga.

Fokus och tid är dock en fråga om prioritering och intresse. Om det finns ett intresse för aktiv reflektion på olika plan i syfte att öka sitt kunnande och förbättra sin bedömningsförmåga, kan också dessa aspekter ges större betydelse och utrymme. Kunniga och kompetenta praktiker behöver utifrån de teorier vi har tagit upp, möjlighet att utöka sitt kunnande genom att analysera, värdera och kritisera kunnandet. För detta behövs reflekterad kunskap och utrymme för att förbättra praxisens regler och kriterier genom en social kunskapsprocess, vilket innebär att ibland lyfta blicken från de konkreta vardagliga problemen och situationerna.

Reflektion och kommunikation i syfte att lära sig mer om IT-kvalitet och därigenom förbättra sin förmåga att bedöma och åstadkomma kvalitet, är också något som förenade tankarna i två

projekt som bedrevs vid institutionen för Informatik. Dessa projekt syftade till att skapa möjligheter för att dels undersöka IT-kvalitet med betoning på bruk, dels förbättra bedömningsförmågan.

## 7 De två projekten

Som vi berörde i kapitel 1 tog våra funderingar kring IT-kvalitet och systemutvecklarnas praxis, sin utgångspunkt i bland annat de projekt som vi deltog i under de första åren av våra forskarstudier. Dessa projekt hade en liknande tankegång som vår egen studie, men också vissa viktiga skillnader. Flera av de slutsatser som vi har dragit i kapitlet ovan överensstämmer i mycket med de antaganden som projekten byggde på, men dessa antaganden byggde inte på någon omfattande empirisk forskning. Projektens syften var också annorlunda eftersom strävan i dessa var att hitta sätt och verktyg för att avhjälpa bristen på designkompetens, en brist som var projektens utgångspunkt.

Vi skall längre fram i avhandlingen reflektera över hur vår undersökning förhåller sig till projekten, men först skall vi presentera dessa.

### 7.1 Projektet IT-designkvalitet – paradigmatiske form och funktion

Projektet ”IT-designkvalitet – paradigmatiske form och funktion” startade hösten 1993 med Pelle Ehn som projektledare med syftet att förbättra systemutvecklarnas designkompetens. Grunden för projektet utgjordes av uppfattningen att brukskvaliteten, d.v.s. den kvalitet som brukarna upplever i brukssituationen, på IT-artefakter<sup>107</sup> är dålig<sup>108</sup>. I sitt välkända ”designmanifest”, som ur-

---

<sup>107</sup> Begreppet IT-artefakt är en sammankoppling av informationsteknik (-teknologi) och artefakt. Artefakt innebär ett konstgjort föremål, d.v.s. något som är gjort av människor, till skillnad från naturliga föremål. Begreppet gör det möjligt att inte ”fastna” i de betydelser som nedläggs i t.ex. datasystem och dataprogram. En IT-artefakt kan ju också vara en mobiltelefon eller en ”intelligent” hushållsmaskin, eller kanske ett datorspel. Vi kommer dock att fokusera på datasystem och -program.

<sup>108</sup> Att vi skriver uppfattning här hänger ihop med de kritiska tankar som vi haft kring projektet och som redovisas längre fram i avhandlingen och i en artikel på IRIS-21 (Meggerle & Steen, 1998). Bl.a. har vi haft svårt för att



sprungligen presenterades 1990 och publicerades 1991 (Winograd i Winograd et al., 1996, s. 2), ger Mitchel Kapor, grundaren av Lotus Corporation och en av männen bakom Lotus 1-2-3, uttryck för en massiv kritik av den bransch han själv verkar i:

Despite the enormous outward success of personal computers, the daily experience of using computers far too often is still fraught with difficulty, pain, and barriers for most people ... The lack of usability of software and the poor design of programs are the secret shame of the industry ... (Kapor, 1996, s. 3).

Detta citat ger uttryck för att kvaliteten på de produkter som systemutveckling resulterar i brister, eftersom bruket av dem ger upphov till flera problematiska upplevelser. Felet som Kapor ser är att fokus inom systemutvecklingsområdet är på programmering:

Computing professionals themselves should take responsibility for creating a positive user experience. Perhaps the most important conceptual move to be taken is to recognize the critical role of design, as a counterpart to programming, in the creation of computer artifacts ... By training and inclination, people who develops programs haven't been oriented to design issues. This is not to fault the vital work of programmers. It is simply to say that the perspectives and skills that are critical to good design are typically absent from the development process, or, if present, exists only in an underground fashion. (ibid., s. 3)

Kapor går sedan vidare med att föreslå en "software design viewpoint" som påminner om det perspektiv på design som kan finnas inom t.ex. arkitektur. För Kapor innebär design mer än att bara sätta samman koddelar till program, design är en fråga om att ha en helhetssyn som tar sin utgångspunkt i brukarnas behov och de omständigheter bruket sker inom. Han vill att "software designer" skall

---

finna empiriska belägg för att brukskvaliteten skulle vara dålig. Oftast rör det sig om uttalanden från ett fåtal kända personer och någon sorts allmänkunskap, som ger vid handen att brukskvaliteten skulle lämna en del övrigt att önska.

vara en egen profession vars utbildning påminner om andra designutbildningar. Mjukvarudesignern skall vara för dataprogram, vad arkitekten är för hus och byggnader.

Dessa problem som Kapor ger uttryck för, tog projektet som utgångspunkt och som utmaning. Att förbättra brukskvalitet är i såfall inte en fråga om att utveckla ännu en systemutvecklingsmetod, utan en fråga om att förbättra de praktiserande systemutvecklarnas och programmerarnas förmåga att bedöma kvaliteten på de produkter som de framställer.

Projektets syfte är att bidra till professionella IT designers kompetens att utveckla IT artefakter av god brukskvalitet. Projektet är inspirerat av kvalitetstänkande inom arkitektur och industriell design, inte minst olika stilhistoriska och stilteoretiska angreppssätt (Ehn, 1994, s. 2).

En bärande tanke inom projektet var, med utgångspunkt i exempelvis Schön (1987), att designers har behov av en repertoar av exempel och förebilder, som kan användas för att gestalta och namnge ett designproblem (Molander, 1996)<sup>109</sup>. För projektet utsträcktes detta genom en koppling till Kuhns paradigmbegrepp, det vill säga att exemplen skulle vara paradigmatiska genom att baseras på erkända och skolbildande förebilder, och inte endast på förebilder som designern själv genom praktiserande uppbyggt som sina privata. Detta leder till att det uppfattades att det fanns<sup>110</sup> ett behov av en sorts stilteori som kan inbegripa dessa paradigmatiska exempel:

Mer specifikt utgår projektet från att det är av stor betydelse att utveckla begrepp och en repertoar av exempel för att inom det informationsteknologiska området kunna reflektera över

---

<sup>109</sup> Molander har här översatt Schön (1987) som kallar det för "naming and framing".

<sup>110</sup> Vi använder här preferitum eftersom projektet är avslutat och eftersom detta är en retrospektiv presentation av projektet. Detta innebär inte att dessa idéer inte längre skulle vara giltiga.

förebilder och stilarter på liknande sätt som inom de mer etablerade designdisciplinerna (Ehn, 1994, s. 2).

Vad projektet såg var att det i en arkitekts eller industriell designers kompetens, ingår att vara förtrogen med de förebilder, traditioner och stilarter som finns i disciplinen. För arkitekturens del kan dessa vara olika -ismer, såsom funktionalism, modernism, o.s.v. Stilarter är det som utgör det paradigmatiska i olika designdiscipliner, eftersom dessa utgörs av de förebilder, erfarenheter och goda exempel som bildar en övergripande referensram under en *viss period*, det som i efterhand ofta benämns med just olika -ismer. Om man kan utsträcka det så långt som att inbegripa även Kuhns idéer om anomalier och kriser i detta resonemang låter vi vara osagt<sup>111</sup>, men det finns skäl att poängtera att referensramarna är aktuella under vissa tidsperioder och att nya förhållanden i andra perioder skapar nya referensramar. Det är alltså en dynamisk komponent i designkompetens som stilarter utgör. Problemet är att skall man följa Kuhns tankegång, dominerar ett synsätt, paradigm, varje tidsperiod; det finns inte utrymme för flera olika samtidiga paradigm<sup>112</sup>.

I vilket fall, var projektets synpunkt att det är en viktig del i designkompetensen att vara förtrogen med paradigmatiske form och funktion, vilket innebär att "... konstruktivt kunna relatera sin egen design såväl till en bestämd tradition som till olika stilarters grundläggande principer, viktigaste artefakter, ledande designers, o.s.v." (ibid., s. 3).

Stilarter kan också existera inom IT-området, om man väljer att betrakta exempelvis objektorienterad och funktionsorienterad programmering som olika paradigm, eller skrivbordsmetaforen och dialogpartnermetaforen som paradigmatiske exempel på hur inter-

---

<sup>111</sup> Kuhns teorier handlar om vetenskap och dess utveckling och inte design. Design är inte ett vetenskapsteoretiskt område där teorier kan visa sig vara ohållbara i samma anda som inom (natur)vetenskaper.

<sup>112</sup> Enligt Wallén (1996) uppfattade Kuhn en vetenskap som omogen om den arbetade med flera tankemönster parallellt, vilket skulle innebära att en vetenskap inte kan ha flera parallella paradigm. Denna inställning kan naturligtvis kritiseras som för begränsande och görs så av Holmqvist (1994).

aktionen mellan människa och dataprogram skall utformas. Olika systemutvecklingstraditioner kan också sägas bilda olika former av skolor, som skiljer sig åt väsentligt i synen på vilka aspekter hos informations- eller datasystem som skall dominera och hur processen att utveckla systemen och programmen skall se ut. Här framstår det t.ex. som möjligt att finna en skillnad mellan den systemteoretiska och den sociotekniska traditionen (Nurminen, 1988; Bansler, 1989), med sina respektive förgrundsfigurer och tillskyndare.

Även om det kan finnas möjligheter att finna olika skolbildningar inom IT-området, så skiljer sig detta åt på två viktiga punkter från andra, mer mogna designområden. För det första förefaller inte de skolbildningar som går att urskilja, betraktas som stilarter i den mening som är märkbar inom exempelvis arkitektur och industriell design; det finns inte samma systematiska bearbetning av IT-områdets olika "stilar", inga -ismer som inbegriper alla de delar i en stilart som nämnts ovan. För det andra finns det brister i förmågan att bedöma stilarter utifrån etiska och estetiska designideal, samt att kunskapen om "... relationer mellan designprocess och faktisk kontextuell social och organisatorisk användning av processens produkter" (Ehn, 1994, s. 3) är begränsad.

Dessa brister i designkompetensen var det projektets mål att i någon mening avhjälpa. IT-designprocessen skulle förbättras, men inte genom att föreslå eller utarbeta nya systemutvecklingsmetoder. Förbättringen skulle ligga i förmågan att bedöma den kontextuella kvaliteten på de produkter som systemutveckling resulterar i. Att kunna göra bra bedömningar, att ha ett omdöme, förutsätter enligt projektet att det finns en repertoar av exempel och förebilder som kan användas till att reflektera över de produkter som den egna designen resulterar i. Syftet med projektet var alltså att utveckla denna förmåga och att utveckla en praktiskt användbar stilteori för en sådan repertoar.

Vad hos IT-artefakter är det då som skall bedömas, vad är det som en IT-designer skall reflektera över? Här var projektet återigen inspirerat av designteori och arkitektur, där man ofta förstår och betraktar artefakter utifrån aspekterna form, funktion och struktur,

samt hur dessa är relaterade till varandra (Lundequist, 1992; Lundequist, 1995). Det är viktigt att påpeka och förtydliga att det inte är frågan om olika delar av eller olika egenskaper hos en artefakt, utan att det är fråga om perspektiv eller synvinklar som man *anlägger* på en artefakt.

Formaspekten har att göra med den enskilda upplevelsen och erfarenheten av en artefakt. Funktionsaspekten avser artefaktens ändamål, syfte och användbarhet. Dessa aspekter är knutna till den kontext vari artefakten befinner sig. Upplevelsen och användbarheten/ändamålet kan endast meningsfullt tolkas utifrån vilket sammanhang artefakten används i. Till skillnad från detta är den sista aspekten, struktur, kontextoavhängig eftersom den har att göra med artefaktens materialegenskaper, som är som de är oberoende av tolkning.

I projektets mening var det meningsfullt att anlägga dessa perspektiv även på datasystem- och program, som trots allt inte är fysiska artefakter såsom hus. Om vi tillämpar dessa aspekter på IT-området, kan detta exemplifieras med ett datasystem som utvecklats för att automatisera (eller snarare semiautomatisera) ett lager. Ur ett funktionsperspektiv kan systemet användas för att belägga lagerhyllor effektivt, kontrollera lagersaldon, beställa rätt material för plockning, avrapportera materialåtgång, materialsatsning för produktionen, etc. Ur ett formperspektiv blir det en fråga om hur de som använder systemet för dessa ändamål, upplever hur det är att bruka det.

När det gäller dessa två aspekter, skiljer det sig inte så mycket åt från hur de kan tolkas inom exempelvis arkitekturen. Även en byggnad är avsedd för ett visst ändamål och de som brukar det har en upplevelse och erfarenhet. Den sista aspekten, struktur, skiljer sig däremot markant från betydelsen inom andra designområden som hanterar fysiska artefakter. Det material som materialhanteringssystemet är uppbyggt av är endast till delar fysiskt, nämligen hårdvaran. Den andra delen är det icke-fysiska material som datasystem är uppbyggda av, vilket då är mjukvaran. Struktur är här då ett datatekniskt och datalogiskt material, med den speciella egen-

skapen att det används för symbolmanipulering. I fallet med materialhanteringssystemet består materialet av datorer, nätverkskopplingar, kablar, olika former av bildskärmar, vilka utgör de fysiska delarna av systemet. Till detta kommer databaser, systemprogram, applikationsprogram, m.m., som utgör de icke-fysiska, abstrakta delarna av materialet. Utvecklingen av detta material är till stor del en uppgift för ingenjörsinriktade discipliner som datateknik och datalogi, d.v.s. discipliner som har ett intresse för materialets egenskaper i sig. Form och funktion däremot är knutna till användning och därmed ”... av kontextuell designvetenskaplig karaktär” (Ehn, 1994, s. 6).

Den kompetens som krävs för att hantera dessa kontextuella form- och funktionsaspekter är den som projektet avsåg att förbättra. Dessa aspekter låter sig inte lätt fångas inom ett datatekniskt eller datalogiskt ingenjörsperspektiv, utan snarare kräver det en designforskning med en inriktning mot samhälls-, beteende- eller humanvetenskap som sätter de kontextuella aspekterna i centrum. Härigenom kan god form och funktion hos IT-artefakter uppnås. (ibid.)

Med denna teoretiska bakgrund och dessa målsättningar, skulle alltså projektet skapa en praktiskt användbar stilteori, med paradigmatiska exempel och begrepp för att kunna reflektera över IT-artefaktens form, funktion och struktur. En utgångspunkt var att denna kunskap finns att söka ute i praktiken, dock inte systematiskt bearbetad, och att detta kan ske genom studier av praktisk användning av existerande IT-artefakter. Det var alltså ett produktperspektiv som anlades i projektet.

Att fånga existerande förebilder på god och brukbar design, förutsätter dock begrepp som till stora delar saknas inom IT-området, begrepp som kan uttrycka något meningsfullt om IT-artefaktens form, funktion och struktur. Att uttrycka något meningsfullt förutsätter en värdering, vilket innebär att varje aspekt måste kopplas till en värderingsgrund eller ett kvalitetsperspektiv. Projektet utvecklade därmed en metamodell för att inledningsvis undersöka kvaliteten på IT-artefakter, som provades i tre magisteruppsatser

(Ganjabadi & Henriksson, 1994; Meggerle & Steen, 1994; Eriksson, 1995).

## **7.2 Den första artefaktmodellen**

Den aspekt av IT-artefakter som framstår som den bäst kända och genomarbetade inom de olika forskningsområden som behandlar informationsteknologi, är den som har att göra med IT som material, d.v.s. strukturaspekten. Det finns en väl utvecklad begreppsapparat och väl utvecklade kunskapsområden, där en djupgående kännedom om hårdvara och mjukvara utgör kärnan. I huvudsak är det detta som datateknik och datalogi har som studieobjekt. Mjukvarans inneboende eller inbyggda egenskaper är också det som främst sysselsatt Software Engineering, som har en lång tradition av både forskning och praxis (se exempelvis Pressman, 1987; Sommerville, 1996).

Om strukturaspekten kan sägas vara väl känd och utvecklad, förhåller det sig annorlunda med funktion och form. Funktionsaspekten har successivt tilldragit sig större intresse genom insikten att nytta och användbarhet inte endast är en fråga om hur väl datasystem och -program är tekniskt realiserade. Användbarhet kan till stora delar bara avgöras om kontexten och de tänkta brukarna ses som en viktig del i ansatsen att skapa datasystem. Detta kommer till uttryck i exempelvis socioteknik, där ett fruktbart samspel mellan teknik och socialt system är målet med systemutveckling. Tanken att de blivande användarna har specifik kunskap att bidra med under systemutvecklingen, är särskilt tydlig inom området participativ design där vissa strävanden även går mot att användarna skall styra systemutvecklingen och systemutvecklarna skall vara resurspersoner med sin större kännedom och kunskap om de tekniska möjligheterna och begränsningarna (se ex. Friis, 1991; Bødker et al., 1993; Ehn, 1993). En annan linje som varit tydlig i Skandinavien är den kritiska (Bansler, 1989) där systemutvecklarna, i dessa fall forskare, tagit aktivt parti för de blivande användarna och satt deras intressen i fokus.

Formaspekten, vilken syftar till upplevelsen och erfarenheten av att bruka en viss artefakt, har enligt vår kunskap knappt alls behandlats, även om vi kan skönja en del ansatser inom MDI-området (Mullet & Sano, 1995)). Datasystem har ju inte primärt utvecklats för att användare skall ha någon positiv upplevelse av att använda dem. Syftet har i allmänhet varit att effektivisera och rationalisera arbetet för att därigenom ge ett större ekonomiskt utbyte med mindre resurser. Detta är åtminstone den gängse bilden av datasystem. I takt med att IT blivit en vardagsteknologi och de tekniska möjligheterna att skapa potentiellt ”trevligare” system och program har ökat, bör de upplevelsemässiga aspekterna av tekniken och dess användning har kommit mer i fokus. Inte minst gäller detta datorspel där upplevelsen är det konkreta syftet och ändamålet, men förmodligen gäller detta också standardsystem som Microsoft Office, Claris Works och liknande produkter. Vi har emellertid inte kunnat finna en analys eller diskussion om detta i någon större utsträckning, fränsett Gelernters (1997) diskussion om skönhets betydelse hos maskiner och inte minst Apples Macintosh-datorer.

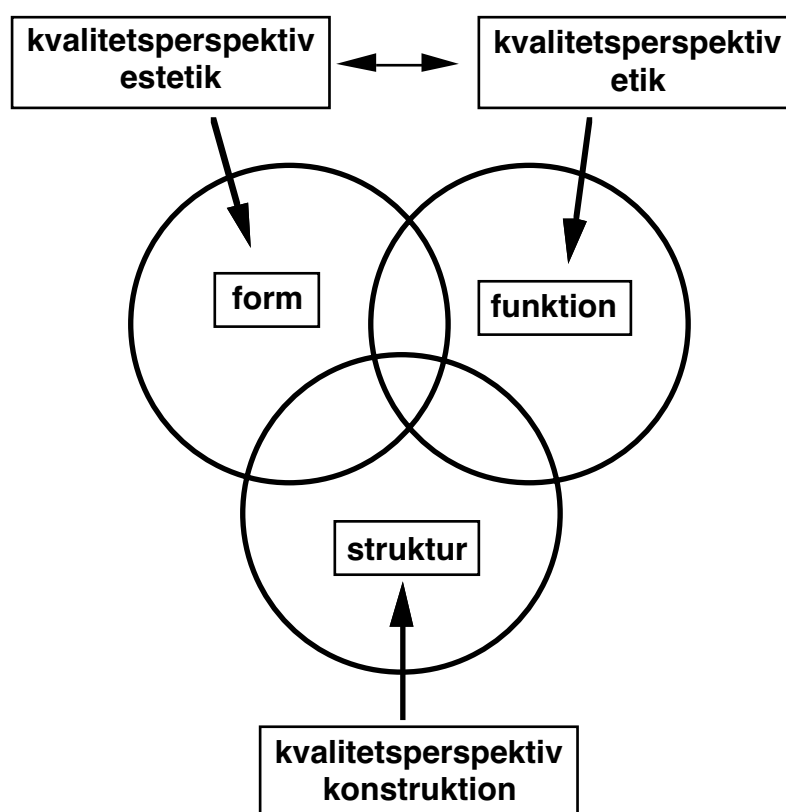
### *7.2.1 Värderingsperspektiv som kan kopplas till modellen*

Eftersom projektets utgångspunkt var att kompetensen att bedöma funktions- och formaspekter kan utvecklas genom en designforskning inriktad mot samhälls-, beteende och humanvetenskap, blir det meningsfullt att knyta värderingen av dessa aspekter till etik och estetik. Etik är en klassisk, filosofisk fråga, som också behandlats av Churchman i sin ”systems approach”. Estetik är en annan klassisk fråga som exempelvis Lundequist (1992), med utgångspunkt i Wittgenstein, behandlar som ”design som språkspel”. Strukturaspecten är som nämnts väl behandlad redan inom IT-området och värderingsgrunder för detta finns inte minst inom Software Engineering och MDI.

Modellen (Figur 7.1) utgör ett system av de tre artefaktaspekterna och tillhörande värderingsperspektiv, där struktur sammankopplas med ett konstruktionsperspektiv, funktion med ett etiskt perspektiv och form med ett estetiskt perspektiv. Detta skapar då en helhet



som visar att alla dessa kvaliteter och aspekter är viktiga och möjliga att se inom varje enskild artefakt i bruk. Detta innebär att exempelvis den tekniska uppbyggnaden av ett datasystem har en koppling till hur brukarna upplever systemet. Det finns naturligtvis också då en koppling mellan syftet och ändamålet med användningen av systemet och de upplevelser som brukarna har av det, liksom att tekniska möjligheter och begränsningar påverkar vilken användning som är möjlig.



Figur 7.1: projektets metamodel för artefaktaspekter och kvalitetsperspektiv relaterade till dessa

Till projektet var kopplat ett så kallat reflekterande team, eller "idébollplank" om man så vill, av namnkunniga och skickliga forskare från olika discipliner som skulle hjälpa oss att åstadkomma bra resultat. När den här modellen visades för dem, uppkom ett problem som vi själva inte reflekterat över. Modellen har formen av ett Wenn-diagram vilket för tankarna till unioner och snitt, bl.a. undrade någon om inte snittet mellan form-funktion-struktur uttryckte den optimala kvaliteten. Det var inte så det var tänkt att modellen

skulle fungera, även om tolkningen att det är ett samband mellan de olika aspekterna tydligt kommer fram.

### **7.3 Den andra artefaktmodellen**

För att presentera idéerna internationellt, både för en kritisk granskning och för att sprida våra tankar, skrev projektet en artikel till "Third Decennial Conference Computers in Context: Joining Forces in Design" (Ehn et al., 1995) där vi utvecklade våra idéer samt föreslog embryot till vad som senare blev projektet "The Qualitheque", vilket vi diskuterar längre fram i avhandlingen.

Artikeln blev accepterad men med viss reservation, nämligen att artefaktmodellen saknade en djupare teoretisk förankring. Som nämnts hade vi tidigare också fått kritik för att modellen förde tankarna till ett Wenn-diagram. Detta ledde till ett omarbetande av modellen, där projektet tog sitt stöd i Habermas teorier om kunskapsintressen och kommunikativt handlande för att diskutera brukskvalitet (Ehn et al., 1996a; Ehn et al., 1997a).

Enligt (ibid.) kan man förstå samspelet oss emellan som individer utifrån tre "världar": den objektiva, den sociala och den subjektiva. Dessa former av samspel är enligt Habermas (1985, enl. ibid.) språkligt inbäddade i vår kultur och när vårt språkliga samspel är inriktat mot gemensam förståelse, kallar Habermas det för kommunikativt handlande (ibid.). Det kommunikativa handlandet sker genom tre sammanhängande former av rationalitet, som kan kopplas till de olika "världarna".

I den objektiva "världen" sker det kommunikativa handlandet genom en kognitiv-instrumentell rationalitet riktad mot värdering av objektiva fakta, d.v.s. ett fokus på korrekthet. I ett kvalitetsperspektiv handlar detta om att ha kontroll.

I den sociala "världen" är det en fråga om en moralisk-praktisk rationalitet riktad mot bedömning av sociala handlingar och huruvida

dessa följer normer och praxis för socialt samspel. Kvalitet blir då en fråga om sanningsenlighet och etisk värdering.

I den subjektiva "världen" handlar det om en estetisk-praktisk rationalitet riktad mot bedömning av känslomässiga och konstnärliga uttryck. Kvalitet blir en fråga om uppriktighet och estetisk bedömning.

Ehn et al. (1996a; 1997a) kopplar sedan detta till Habermas tankar om kunskapsintressen (Habermas, 1968a, 1968b, enl. *ibid.*), där den kognitiva-instrumentella rationaliteten leder till ett intresse för teknisk eller instrumentell reglering med ett fokus på observation, empirisk analys och instrumentell kontroll av systems tekniska funktionalitet. Med ett sådant intresse ser vi aspekter i vår omvärld som ting, händelser och villkor som i princip kan hanteras på ett teknik- eller naturvetenskapligt sätt.

Det sociala och praktiska intresset riktar sig mot uttolkning samt mänskligt förstående och kommunikation. Riktas detta intresse mot mellanmänsklig kommunikation, leder det till ett fokus på dialog, relationer och förståelse så som inom social- och humanvetenskaper.

Slutligen ser Ehn et al. (1996a; 1997a) också ett estetiskt-expressivt intresse, som leder till ett fokus på den typ av känslomässiga upplevelser och kreativitet som finns inom konstens värld<sup>113</sup>.

Om man nu överför detta till IT-området, finns det en koppling mellan de tidigare presenterade aspekterna struktur, funktion och form, samt de tillhörande kvalitetsperspektiven. Strukturaspekten kan då sägas tillhöra den objektiva "världen" där rationaliteten är riktad mot objektiva fakta, kunskapsintresset mot instrumentell kontroll, rationell design och teknisk funktionalitet. Med denna utgångspunkt blir kvalitet en fråga om kontroll.

---

<sup>113</sup> Ehn et al. (1996a; 1997a) konstaterar här att Habermas inte hade detta perspektiv i sin första analys.

Funktionsaspekten placeras sålunda inom den sociala "världen" med en norm- och praxisinriktad rationalitet, där kunskapsintresset handlar om förståelse, uttolkande och kommunikation. Kvalitet blir därmed en fråga om etik.

Slutligen kopplas formaspekten till den subjektiva "världen" med dess rationalitet av känslomässiga och konstnärliga uttryck, samt dess kunskapsintresse riktat mot känslomässiga upplevelser och kreativitet. Kvalitet handlar sålunda om estetik.

| "Värld"             | objektiv   | social  | subjektiv  |
|---------------------|--|---|--|
| Artefaktaspekt      | STRUKTUR <ul style="list-style-type: none"> <li>• hård- och mjukvara</li> <li>• material eller medium</li> </ul> | FUNKTION <ul style="list-style-type: none"> <li>• praktisk användning</li> <li>• symbolisk användning</li> </ul>                            | FORM <ul style="list-style-type: none"> <li>• upplevelse av bruk</li> </ul>  |
| Kvalitetsperspektiv | KONTROLL <ul style="list-style-type: none"> <li>• mjukvarumetriker</li> <li>• kvalitetsstandarder</li> </ul>     | ETIK <ul style="list-style-type: none"> <li>• användbarhet</li> <li>• nytta</li> <li>• makt</li> <li>• anspråk</li> <li>• värden</li> </ul> | ESTETIK <ul style="list-style-type: none"> <li>• lämplighet</li> <li>• stil</li> <li>• balans</li> <li>• likhet</li> </ul> |
| "Usability Style"   | ÄNDAMÅLSENLIGHET: En "lämplig balans" mellan struktur, funktion och form   |   |  |

Figur 7.2: Den andra artefaktmodellen (modifierad efter Ehn et al., 1996a, s. 22)<sup>114</sup>

<sup>114</sup> Vi har i figuren och presentationen i övrigt, lyft bort kopplingen till den romerska arkitekten Vitruvius och hans ansats att tänka på byggnader i termer av firmitas, utilitas och venustas, vilket påminner en hel del om struk-

Detta blev alltså den modell (se Figur 7.2) som utvecklades för att kunna beskriva och bedöma brukskvalitet hos IT-artefakter, som skulle användas för att skapa användbara exempel på kvalitet-i-bruk som skulle infogas i en stilteori. Det är också denna modell som skulle ge vägledning vid skapandet av exempel för den virtuella "studion" Kvaliteket som beskrivs längre fram i avhandlingen. Innan vi kommer dit skall vi emellertid gå in på projektets andra viktiga del, nämligen idén om en praktiskt användbar stilapparat.

Den paradigmatiske repertoaren av exempel var tänkt att inordnas i en praktiskt användbar stilteori bestående av olika stilarter som representerar förebilderna i form av principer, viktigaste artefakter och ledande designers. Frågan är då vad en stilart och en stilteori är för något, och hur man bör gå till väga för att skapa en sådan.

## **7.4 Stilar och stilteori**

I projektgruppen fanns det en outtalad tro på att det dels var relativt rättframt att formulera en stilteori, dels att de som hade denna kunskap var arkitekter. Vad som behövde göras var "bara" att sätta sig in i deras användning av stil, ta en lämplig definition och anpassa den till projektets syften. Efterhand som diskussioner uppstod om stil inom projektet, ökade dock förvirringen kring vad stil egentligen är. Jag och Theis tog på oss uppgiften att reda ut begreppen. Vi vände oss därmed, i linje med projektets allmänna föreställningsram, till arkitektur och arkitekturteori. Vi trodde att detta skulle vara en relativt lätt avklarad uppgift, eftersom arkitekterna naturligtvis arbetade medvetet med stil.

Så var emellertid inte fallet visade det sig<sup>115,116</sup>. Tvärtom var begreppet stil ytterligt svårgripbart och hade varit ett ämne för diskussion

---

tur, funktion och form.

<sup>115</sup> Vi tog kontakt med några personer på arkitekturutbildningen på LTH i Lund för att skaffa oss en klarare bild av hur stil användes i bl.a. utbildningen. Vi var inte klokare när vi gick därifrån än vad vi var när vi kom dit.

<sup>116</sup> Som Enkvist (1973) konstaterar måste den som vill förstå det grundläggande

sedan antikens dagar, särskilt inom litteraturen. Vi gjorde därför ett försök att göra en genomgång av hur stilbegreppet hanterats inom främst arkitektur och lingvistik, och vilka möjliga stildefinitioner som de hade att erbjuda. Vår förhoppning och tro var att vi ändå skulle nå en kunskap som skulle göra det möjligt att formulera en vettig stilteori för IT-området, i enlighet med projektets målsättning.

I början var stil ett förhållandevis enkelt begrepp som användes för att karakterisera författare och vilken tänkbar grupp av författare denne tillhörde, baserat på olika "drag" i texterna som kan vara gemensamma med andra författare<sup>117</sup>. Detta har fortsatt även in i vår tid och i vardagstal pratar man om deckare, science fiction, kriminalromaner, etc. Säkerligen finns det skäl till en sådan karakterisering, men vid en närmare studie av de enskilda verken skulle man lika säkert kunna finna så stora skillnader att karakteriseringen antagligen endast blir en form av ytlig indelning som egentligen inte tillför något värde av bestående djup.

Mycket kritik har inom lingvistik och litteraturforskningen enligt Olsson (1982) riktats mot användningen av stilbegreppet. Bland annat har Gray (1969) diskuterat och kritiserat ett antal olika stilbestämningar inom litteraturforskning och nått den slutsatsen att begreppet stil är en vilseledande tanke på samma sätt som den antika fysikens begrepp eter. Grays slutsats är att stilbegreppet borde avskaffas. Olsson konstaterar, med utgångspunkt i ytterligare två kritiska personers arbeten<sup>118</sup>, att stilstilforskningen befinner sig i en kris.

---

de och essentiella i stilstilforskning vara en allvetare.

<sup>117</sup> Enligt Zeitler (1982) var också stil ett prestigeord under 1800-talet – en riktig konstnär hade stil.

<sup>118</sup> Nämligen:

Louis T. Milic: *Against the Typology of Style*, i *Essays in the Language of Literature*, ed. Chatman & Levin, Boston 1967, ss. 442–450. Olsson använder omtrycket *i Style and Criticism*, ss. 283–292.

Stanley Fish: *Is There a Text in This Class?*, Camb. Mass. & London 1980. Olsson använder de två uppsatserna "What is Stylistics and Why are They Saying Such Terrible Things About It?", vilka återfinns på ss. 69–96, resp. 246–267.

I samma skrift som Olssons bidrag återfinns i har Göran Hermerén (1982) utvecklat en diskussion om stilbegreppens logik och anatomi<sup>119</sup>, där han på en övergripande nivå, utan att gå in djupt på något enskilt område där stil används, presenterar vilken typ av frågor som blir aktuella vid en studie av stilbegreppens logik och stilstilforskningens teoretiska fundament. De frågor<sup>120</sup> han där tar upp, visar alla på att det finns såväl filosofiska som andra problem kopplade till en användning av stilbegrepp och stilbestämning. Några av dessa är till exempel de problem som har att göra med de (dolda) antaganden som enligt Hermerén görs vid resonemang om stilar.

Hermerén tar också upp exempel på vid vilka problem som stilbegrepp används i forskningen och vilka begreppens användningsområden är. Utan anspråk på fullständighet, anger Hermerén nio olika typer av problem och användningsområden:

1. *Datering.* När tillkom X?
2. *Lokalisering.* Var skapades X?
3. *Attribution.* Vem gjorde X?
4. *Beskrivning.* Hur karakteriseras X?
5. *Komparation.* Vilka likheter/skillnader finns det mellan X och Y?
6. *Klassifikation.* Till vilken grupp/typ hör X?
7. *Tolkning.* Vad betyder olika egenskaper hos X?
8. *Förklaring.* Varför har X vissa egenskaper?
9. *Kvalitetsbedömning.* Hur bör X värderas?

De sex första punkterna förefaller hänföra sig mest till kategorisering och om man nöjer sig med enkla och ytliga stilbestämningar kanske detta skulle fungera<sup>121</sup>. Svårare verkar det bli med de reste-

---

<sup>119</sup> Hermerén (1982, s. 200) använder följande för en diskussion om denna: ”ett komplex av konstant eller ofta återkommande egenskaper X som karakteriserar (eller används för att karakterisera) Y”.

<sup>120</sup> Det rör sig bl. a. frågor om för vilken typ av problem som stilbegreppet används inom forskningen, vilka (dolda) förutsättningar, resonemang om stilar bygger på och hur man skulle kunna avgöra om två verk tillhör samma stil. Inalles tar Hermerén (ibid.) upp sju frågor av många möjliga till diskussion.

<sup>121</sup> På detta sätt bör väl mycket av stilbestämningar inom olika områden funge-

rande tre punkterna som förutsätter en kvalitativ värdering av verket i fråga och som rör sig mot omständigheter runt tillkomsten av verket, den kontext som verket ingår i, vem som bedömer verket, etc. För att göra denna sorts bedömningar, krävs det (som det framstår för oss) att man tar in även sådant som inte är synligt i själva artefakten. Det verkar som om det blir nödvändigt att känna till skaparens intentioner, idéer, etc. som kanske söker sitt uttryck i verket.

Enligt Hermerén finns det två olika sorters problem vid användningen av stilbegrepp som bör hållas isär. Dels kan man vara ute efter att beskriva och dels att förklara ett verk, vilket medför skilda vetenskapsteoretiska problem. När det gäller förklaringsproblemet är det viktigt att fixera förklaringsobjektet, d.v.s. dels det som skall förklaras och dels det som förklarar. Hermerén (*ibid.*, s. 193) skiljer på två typer av detta problem:

1. Vad beror stilmförändringar (stilmväxlingar etc.) på?  
Schematiskt, där X är den sökta variabeln:  $X \rightarrow$  stilmförändring
2. Vad beror på stilmförändringar (stilmväxlingar)?  
Schematiskt: stilmförändring  $\rightarrow X$

Frågan är då vad X står för i detta schema och om det är rimligt att förvänta sig att orsakerna till stilmväxlingar är desamma inom olika konstarter. Skulle det förhålla sig på det viset, att det finns bakomliggande orsaker av samma typ till stilmförändringar "...gäller då principen 'samma orsak, samma verkan'? Det skulle innebära att det inte bara finns ytliga likheter mellan t. ex. renässanslitteratur, renässansmusik och renässansmåleri utan att man här borde kunna

---

ra, exempelvis inom konsten, där olika -ismer är vanliga och används för att kategorisera såväl verk som konstnärer, och föra dessa till stilar och grupperingar av konstnärer. Det är i alla fall så vi som lekmän uppfattar saken. En konsthistoriker eller konstkännare skulle säkert kunna rikta kritik mot ett så förenklat synsätt.



finna djupgående analogier, som beror på att orsakerna till förändringarna på dessa områden är av samma slag” (ibid., s. 194)<sup>122</sup>.

Hermerén fortsätter sedan med att visa olika möjliga ”värden” på X i schemat, där han grovt skiljer mellan materialister och idealister, även om han tycker motsättningen mellan dessa i sammanhanget ger en onyanserad bild. Materialisterna utgår från att stilförändringar beror på materiella (sociala och ekonomiska) förändringar, medan idealisterna hänför stilförändringar till idémässiga förhållanden. Med utgångspunkt från dessa olika inställningar, kan det vara möjligt att låta X stå för förändringar i världsbild och tänkesätt, normförändringar, politiska eller ekonomiska förändringar, tekniska eller vetenskapliga innovationer etc. D.v.s. faktorer som inte lätt låter sig klassificeras som materiella eller ideella, och som dessutom inte behöver vara oberoende av varandra.

I vår jakt på en lämplig stildefinition kom vi relativt tidigt i kontakt med en avhandling av den finska arkitekturhistorikern Anna-Majja Ylimaula, där hon utifrån en fenomenologisk ansats, inspirerad av Heidegger, diskuterar stilen i Antoni Gaudis, C. R. Mackintosh’ och Otto Wagners arkitektur (Ylimaula, 1992). Vi är vare sig filosofer eller arkitekter så vi avstår från försök att tränga in på djupet i hennes avhandling, men hennes utgångspunkt framstår för oss att vara, att det sätt som stilar och stilbegreppet vanligtvis används på inom arkitekturen, endast tillför ett moment av kategorisering och klassificering som bygger på detaljer i byggnaders fasader och liknande ytligheter. För henne är stil en kvalitet som går djupare än så, som skiljer arkitektur från endast en byggnad. Stil är ett uttryck för arkitektens underliggande filosofi, idé eller kanske ”äkthet” som kommer till uttryck i en byggnad som då kan betraktas som arkitektur; stil är ”...a quality of greater substance [än bara perioder av fasad-detaljer]” (ibid., förordet).

---

<sup>122</sup> Olsson (1982) finner visst stöd för en sådan tanke i en undersökning av svensk litteratur från 1600 till 1760, som han jämför med en undersökning inom en annan konstform gjord av Wölfflin, varvid båda kommer fram till att renässansen karakteriseras av yta, medan barocken karakteriseras av djup.

Style presupposes authenticity, the life and society around the building. Architecture expresses values which transcend above mere time and place. Dividing architecture into periods of style is only a matter of categorising, which helps the classification but it takes style away from its real context, from the philosophy behind the building. Thus style presupposes artistic truth and brings it forth (ibid., abstract)

Uppenbarligen är i såfall stil ett ganska komplext begrepp som har ett större djup än en kategorisering uppbyggd på visuella element och deras likheter och skillnader under olika perioder. Ylimaulas förståelse av arkitektur är sådan att den påverkas av en mängd faktorer som omger den: "Architecture is in various ways dependent on the economic, political, industrial and technological conditions of a society. It is equally dependent on the development of contemporary culture, especially the visual arts, music and literature" (ibid., förordet)<sup>123</sup>. Detta ger naturligtvis en ganska komplex bild av arkitekturen, men det avgörande för Ylimaula är alltså att stilen uttrycker arkitektens filosofi.

Vi skall som sagt inte gå in på djupet i hennes avhandling, eftersom vi känner att vi inte behärskar ämnet tillräckligt väl. Men citaten ovan speglar en speciell inställning till vad det är som bör ligga till grund för en stilförståelse inom arkitekturen.

Under seklens lopp har flera olika uppfattningar presenterats om vad stil är uttryck för och betyder. Dem vi fann i vårt sökande efter en lämplig stilteori sammanfattas kort i figur Figur 7.3 och uttrycker en mångfald som har påverkat våra egna upplevda möjligheter att komma vidare med begreppet.

#### *7.4.1 Och vad blev det av det...?*

Vår undersökning av stilbegreppet och den mångfald olika tolkningar av stilbegreppets innebörd och betydelse har visat, åtmin-

---

<sup>123</sup> En syn på vad som påverkar stilutvecklingen som både Grenander Nyberg (1993) och Ginzburg (1982) förmodligen skulle instämma i.

stone för oss, att det finns så många tankar och riktningar inom studiet av stilbegreppet som verkar kloka, även om de ibland verkar stå emot varandra.

| <b>Evolutionär – livscykel</b>             |   |
|--|---|
| Vasari, 1500-talet                         | Konst som faser. En stil föds, mognar och dör ut.                 |
| Winckelmann, 1700-talet                    | Klimatet, naturen och den sociala ordningen.                      |
| Ginzburg, 1900-talet                       | Epoken och dess tekniska, samhälleliga och naturliga möjligheter  |
| <b>Känslor och värderingar i samhället</b> |   |
| Wölfflin, 1800-talet                       | Populära och kollektiva känslor i samhället.                      |
| Lindgren, 1900-talet                       | Värderingar i samhälle och område.                                |
| Grenander Nyberg, 1900-talet.              | Design är påverkad av kulturen.                                   |
| Semper, 1800-talet                         | Tradition, behov och nödvändighet i samhället.                    |
| <b>Personliga värderingar och filosofi</b> |   |
| de Buffon, 1700-talet                      | Konstnärens grundläggande värderingar uttrycks i stilen.          |
| Ylimaula, 1900-talet                       | Stil är uttrycket för konstnärens underliggande filosofi.         |
| <b>Den ”genetiska koden”</b>               |   |
| Riegl, 1800-talet                          | Stil är relativ genom vissa ”transformationskoder”.               |
| <b>Den semiotiska synen</b>                |   |
| Blonsky, 1900-talet                        | Stil är ett system av värden, fristående från orsak och ändamål.  |
| <b>Oanvändbarhetssynen</b>                 |   |
| Gray, 1900-talet                           | Begreppet stil erbjuder ingen praktisk nytta och borde avskaffas. |

Figur 7.3: Sammanställning av några stildefinitioner (bearbetad och översatt från Ehn et al., 1997a)

Att göra ett sunt och rationellt val utifrån dessa, framstod för oss som en för stor svårighet – vi hade helt enkelt inte möjlighet att göra detta val. Det enda val som projektet hade gjort, var att inte gå på den traditionella konsthistoriska linjen i vår stilförståelse. Detta baserat på Scrutons (1979) bok om vad som är estetik i arkitekturen. Där kritiserar han bland annat idén att arkitekturen är ett uttryck för tidsanden (Zeigeist), vilket då också är en kritik mot den idén som en förståelse för stilbegreppet. Om alla byggnader i en viss period uttrycker tidsanden, hur kan man då skilja de goda exemplen från det dåliga? Tidsanden är därmed inte något bra sätt att försöka förstå arkitekturens estetik.

Vad vi i projektet kunde få ut av vår studie av stilbegreppet, var emellertid att det uppenbarligen skulle komma att krävas många exempel på IT-artefakter i bruk, och att dessa exempel genom sin mängd skulle kunna ge oss möjligheter att finna stilarter gångbara inom vårt eget område. Grays (1969) förkastande av stilbegreppet kunde vi förstå, men vi tyckte samtidigt att en stilteori och stilarter ändå skulle vara viktiga redskap inom IT-området.

Att samla in alla dessa exempel, upplevde vi emellertid som en både för stor och för viktig uppgift för att kunna genomföras på ett bra sätt med bara vår lilla forskargrupp. Projektgruppen lanserade därför ett nytt projekt som skulle göra detta möjligt. Det projektet blev döpt till Kvaliteket och presenteras nedan.

## **7.5 Projektet Kvaliteket**

Idén till Kvaliteket<sup>124</sup> uppstod när projektet skrev en artikel till den tioårliga konferensen ”Computers in Context 1995” (Ehn et al., 1995). Artikeln beskrev projektet, dess idé, erfarenheter och avslutades med ett scenario, där två uppduktade personer, Adam och Eva, fann ett ställe på Internet som innehöll representationer av kvalitet-i-bruk på IT-artefakter.

---

<sup>124</sup> Kvaliteket fanns på adressen <http://qualitheque.ics.lu.se>, men tyvärr har hårdvaran kraschat och därför finns inte Kvaliteket längre på Internet.

Tillsammans med projektets teoretiska erfarenheter av stil, d.v.s. att det behövdes många exempel för att om möjligt kunna prata om och urskilja stilar, skapades Kvaliteket. Kärnidén var att skapa plats för representationer av brukskvaliteten på IT-artefakter och var därigenom ett tillvägagångssätt för att skapa en större mängd exempel som i ett senare skede skulle kunna ligga till grund för teorbildning.

### *7.5.1 Ett medium för reflektion kring kvalitet i bruk*

Med fokus på kvalitet-i-bruk skulle Kvaliteket vara en mötesplats – ett forum för diskussion om brukskvalitet, en repertoar av exempel, en inspirationskälla för praktiker, en kunskapsbas för forskare och en möjlig utbildningsplats för lärare och studenter. Kvaliteket skulle vara den första internationella och virtuella designstudion på Webben. Med hjälp av den senaste tekniken skulle det vara möjligt för alla att beskåda, begrunda, debattera och interagera med olika ”utställningsföremål”.

Kärnan i Kvaliteket var utställningsföremålen, d.v.s. representationerna på kvalitet-i-bruk på IT-artefakter – de som också kallas exempel eller bidrag. Dessa var en del av projektets bidrag för att avhjälpa den brist på kompetens som Kapor skisserade (Kapor, 1996, se s. 190). Genom exemplen skulle besökaren ha tillgång till fakta, information och analyser av IT-artefakter som dagligen brukas. Materialet skulle både gå på djupet och bredden, och på så sätt tillhandahålla besökare ett ur projektets synvinkel unikt material.

Besökaren skulle också ges möjlighet att interagera med bidragen. Därmed skulle besökaren, genom så kallad ”handpåläggning” (”hands-on experience”) kunna bilda sig en bättre uppfattning och djupare förståelse för både beskrivningen och utställningsföremålet.

För att Kvaliteket skulle uppfattas som en aktuell och relevant kunskapsbas, skulle det finnas möjlighet till reflektion. De enskilda exemplen skulle kunna kommenteras och debatteras. Men även övergripande debatter om till exempel kvalitet skulle möjliggöras. Innehållet skulle kunna variera från kommentarer kring detaljer i

en IT-artefakts gränssnitt, till debattinlägg rörande grundläggande förhållningssätt gentemot begreppet kvalitet. På så sätt skulle debatten vara ett aktivt utbyte av idéer och reflektioner kring kvalitet i allmänhet, kvalitet-i-bruk och informationsteknologi. Detta skulle ge en dynamisk och organisk karaktär och göra det meningsfullt för praktiker, studenter och forskare att använda och bidra till Kvaliteket.

För projektet var denna reflektion, utbytet av idéer och eventuella kritik också forskningsmässigt viktiga. Det var här den kritiska massan – samfundet – skulle skapas. Det var bland annat i debatten nya kvalitetsbegrepp, eventuella stilar och paradigmer skulle komma fram. Det skulle precis som i vår förebild arkitektur, skapas ett forum för diskussion om designers professionalism, kompetens och kunskap, samt deras uppfattning om kvalitet – speciellt brukskvalitet.

I och med detta var det angeläget att Kvaliteket betraktades som ett seriöst försök. Det skulle därför vara en merit att få ett exempel accepterat som ett utställningsföremål. För att upprätthålla en sådan karaktär var det tanken att det skulle finnas två filter i form av en redaktion och en tävling med jury. Redaktionens uppgift var att utifrån projektets teoretiska perspektiv recensera bidragen. När bidragen var godtagbara ur projektets ögon blev de exempel i Kvaliteket och därmed deltagare i tävlingen. Tävlingen var både ett lockbete och ett andra filter för att få fram de exempel som var de bästa. I tävlingen skulle det finnas priser för de bidrag som ansågs ge det bästa intrycket, de som var besökarnas val, de som uppvisade den bästa grafiska formen och användningen av presentationsteknik och till slut den representerade artefakt som bedömdes ha den bästa brukskvaliteten. Det var vår tro att dessa tillvägagångssätt skulle bevara en kontinuerligt hög kvalitet på forumet och att Kvaliteket skulle tas på allvar.

Då Kvaliteket som sagt kretsade kring kvalitet-i-bruk, framstod det som viktigt att Kvaliteket i sig skulle utgöra ett gott exempel på IT-design. Det skulle utstråla god design i alla avseenden och bygga på den skandinaviska designtraditionen. Målet var att flytta fram ut-

vecklingen inom Webb-tillämpningar så pass att Kvaliteket kunde bli stilbildande för design av applikationer på Webben. Med de rådande stilarna på Webben tillsammans med vår förkärlek för svensk form, blev det naturligt att välja den skandinaviska formen som koncept för Kvaliteket<sup>125</sup>.

För att ytterligare skapa en förståelse för Kvaliteket och dess syfte kan en reflektion över ett bidrag vara på sin plats. Det bidrag som vi har valt är en representation av ett system som heter MHS (MöbelHusSystemet) vilket används på IKEA. Bidraget är författat av Mikael Svedemar (Svedemar, 1995b, 1995a)<sup>126</sup>.

### *7.5.2 Ett exempel på en representation i Kvaliteket*

Exemplet eller bidraget är uppbyggt kring historier. Historierna tar sin utgångspunkt i olika personers upplevelser av systemet (se Bild 7.1).

Historierna förtäljer oss att IKEA-MHS är ett system med många syften. Den mer formella övergripande beskrivningen anger att det syftar till att hantera gods och information. Men historierna berättar också för oss att MHS mer specifikt också används som leveranssystem, informationssystem, blankettgenerator och säljstöd.

IKEA-killen som arbetar i varuutlämningen får genom MHS orderlistor som innehåller de varor som är på utgång. Personal i kundtjänst använder systemet för att svara på förfrågningar från kund om varor. Personal i IKEA-butiken använder det både som säljstöd, varubokningssystem och blankettgenerator.

---

<sup>125</sup> Vår förståelse för den skandinaviska formen är i all korthet att den kännetecknas av funktionalitet mer än yta. Även om den grafiska formen framstår som viktig tillåts den aldrig dominera över artefaktens funktionalitet. Formgivningen är mer präglad av en nyttyosyn, där man eftersträvar en ren, förnuftig, sund och robust form. Detta utesluter dock inte lekfullhet och värme. Ergonomi har också en stark ställning inom den skandinaviska traditionen. Alltså, ingen design för designens skull (Meggerle & Steen, 1997).

<sup>126</sup> Bidraget går att återfinna på följande adress:  
<http://qualitheque.ics.lu.se/home/qualhi/sh/aprmay96/index.htm>

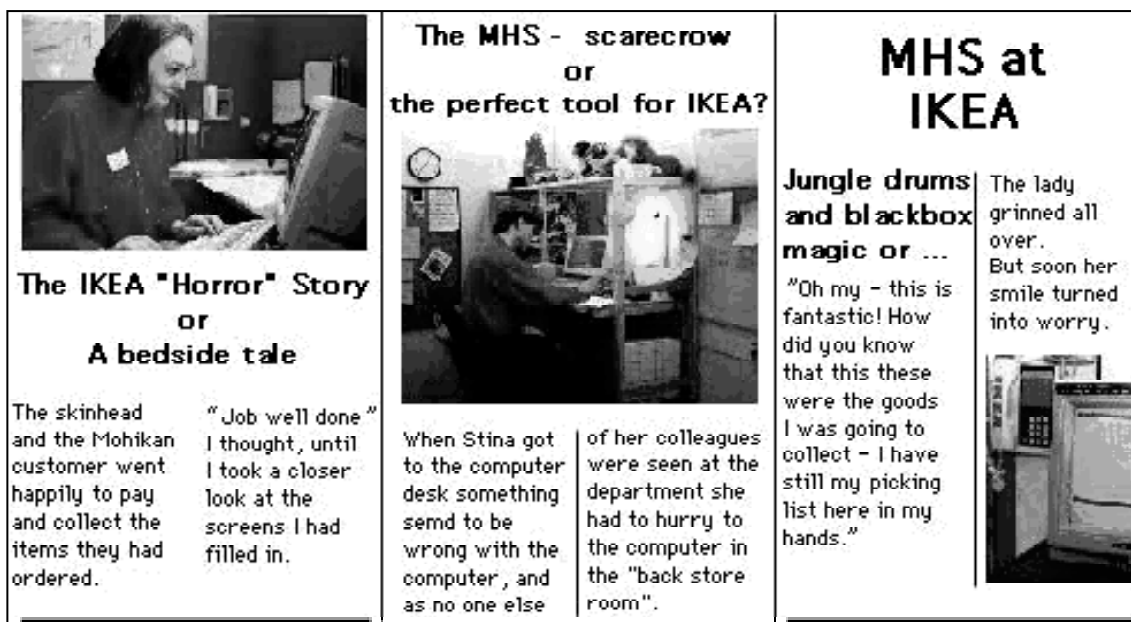


Bild 7.1: Startbilden till bidraget om Ikea-MHS.

MHS uppvisar genom historierna en variationsrikedom i användning och ser man närmare till hur det upplevs kan samma variation återfinnas där.

IKEA-killen som använder systemet i varuutlämningen känner en viss stolthet då en av kunderna berömmar honom för den snabba hanteringen av en beställd vara. Likadant känner lagerkillen när en annan anställd behöver hjälp med hanteringen av MHS. Systemet upplevs av den andra både som komplext och svårgripbart eftersom hon inte är speciellt förtrogen med systemet, vilket leder till att hon känner sig lite rädd för det.

Om den mer tekniska sidan av MHS finns det ganska mycket material. Av denna beskrivning kan vi lära oss att MHS består av administrativ del (BOS) och en kassadel ("cashier"). Den administrativa delen körs på VAX/VMS. Klienterna är terminaler som är uppkopplade via ett lokalt nätverk. Systemet är sedan kopplat till ett centralt system via ett "Wide Area Network". Systemet är till huvuddelen skrivet i Pascal och använder sig av en relationsdatabas (Oracle). Gränssnittet är kommandobaserat.

Sammanfattningsvis kan man säga att det är en ganska rik bild som ges av IKEA-MHS. Den rika bilden känns dock något ytlig. Det



finns ingen analys av det som är berättat vilket innebär att man som besökare själv får analysera exemplet. Med andra ord är det upp till besökaren att bilda sig en uppfattning om brukskvaliteten på IKEA-MHS. Beskrivningen är också lättsmält, men ibland kan man som besökare bli lite förvirrad av alla hypertextlänkar och tappa orienteringen. Därtill blir man ibland något förundrad över vem det är som berättar. Är det forskaren eller är det personal?

Så här kan alltså ett bidrag/exempel/representation vara utformat och fungera, och sammanlagt blev det till slut fem bidrag av olika karaktär och utifrån olika datasystem som kom in i Kvaliteket. Emellertid verkar det som om inte allt gick som det var tänkt med Kvaliteket.

### *7.5.3 Några reflektioner om varför det inte lyckades*

Utifrån alla tidigare omnämnda krav lyckades projektet bygga upp Kvaliteket som en produkt. En produkt för att möjliggöra en gemenskap, en kritisk massa och ett tillvägagångssätt för att få in så många exempel som möjligt. Men det gick inte riktigt som det var tänkt.

För att etablera Kvaliteket och förmå i första hand andra forskare att bidra, presenterades och diskuterades både idéerna och Kvaliteket i en rad olika sammanhang. Ofta bemöttes det som mycket positivt och intressant, och ibland även med en viss hängivenhet för att delta. Men sedan hände inte mycket mer och vi kan här endast spekulera om olika anledningar till detta.

Först av allt finns det nog fog för att kritisera det redaktionsarbete som utfördes av deltagarna i projektet. Ett bygge av denna kaliber, såsom vi föreställde oss det, kräver oerhört mycket engagemang och hängivenhet. Som redaktion borde vi på ett bättre sätt, med envishet, försökt förmå forskarkollegor och andra att bidra med exempel. Till viss del lyckades vi, men inte fullt ut. Redaktionen borde möjligen också gjort bättre ”reklam” för de exempel som faktiskt fanns, bidragit med analyser av bidragen, skapat eller provoce-

rat fram debatter och således på ett bättre sätt försökt skapa början till ett samfund.

Uppgiften att framställa ett bidrag framstår också i efterhand som stor och svår. Den av projektet formulerade artefaktmodellen framstår inte som helt enkel att tillämpa (modellen diskuteras i avsnitt 7.3) och det krävs mycket engagemang och tid att framställa ett bidrag. Som en fingervisning kan författarnas egen studie av ett säljstöd tjäna som exempel.

I vår magisteruppsats gjorde vi en studie av ett säljstöd på Volvo och utförde för denna sex semi-strukturerade intervjuer på vardera två timmar (Meggerle & Steen, 1995; Ehn et al., 1997a). Intervjuerna transkriberades och därefter analyserades materialet utifrån artefaktmodellen (bortsett från struktur som vi valde att lämna utanför studien). Till sist presenterades analysen. Alla som har erfarenheter av den här typen av arbete vet att det tar tid. Om det dessutom skulle ha blivit ett bidrag till Kvaliteket (vilket det aldrig blev) skulle underlaget omformas för webben<sup>127</sup> och kanske kompletteras med en skalad prototyp som illustrerar den representerade IT-artefakten<sup>128</sup>. Att förmå andra forskare och praktiker, att parallellt med andra intressen och arbetsuppgifter skapa ett utställningsföremål till Kvaliteket var kanske för mycket begärt.

Spekulativt skulle ett frågetecken också kunna vara den nytta, det erkännande och den prestige som det var tänkt att ett bidrag skulle ge en bidragsgivare. Projektets mål var att det jämförelsevis skulle kännas som att få en artikel publicerad i en välkänd tidning eller accepterad till en konferens. Trots att Kvaliteket blev omnämnt i *interactions*<sup>129</sup> (Ehn & Löwgren, 1996), i en välkänd forskares antologi (Winograd et al., 1996), på en workshop på PDC-96 (Ehn et al.,

---

<sup>127</sup> D.v.s. ett format som passar webben, till exempel HTML, GIF-bilder och Av-filmer. Dessutom anser vi att det är stor skillnad mellan hur ett material presenteras i löpande text på papper och på webben. Utan att precisera vad denna skillnad utgörs av, anser vi att det är ännu ett moment som tar tid.

<sup>128</sup> Kanske i Java, Shockwave eller med hjälp av någon annan teknik.

<sup>129</sup> En tidsskrift från ACM.

1996b) och på en designkonferens i Stockholm (Ehn et al., 1997b) och att en välrenommerad jury sattes samman, lyckades vi nog inte med att skapa en tillräcklig känsla av prestige, nytta och erkännande.

Om dessa spekulationer är de rätta vet vi inte. Kanske svaren ligger djupare än så här eller så ligger det en smula sanning i varje skäl till varför Kvaliteket inte blev en framgång. Den uteblivna framgången resulterade i ett kritiskt ställningstagande och en ny inriktning för vårt avhandlingsarbete. Vi skall i nästa kapitel reflektera över detta kritiska ställningstagande och IT-kvalitet i praxis, samt relationen till projekten.

## 8 Projektet och IT-kvalitet i praxis – en reflektion

Vi har nu presenterat vår egen undersökning och våra forskningsproblem för den här avhandlingen, och även presenterat de två projekt vi deltog i tidigt i vår forskarutbildning. Som vi skrev tidigare finns det stora likheter mellan utgångspunkterna för det arbete vi har gjort och dessa projekt. Till viss del så bekräftar våra undersökningsresultat de antaganden som gjordes i projekten. Det finns också viktiga skillnader mellan vårt arbete och projekten och dessa föreligger i synen på hur en förbättring av bedömningsförmåga och kvalitetskunnande skall kunna åstadkommas.

Som framgår av inledningskapitlet (kapitel 1) så lyckades inte projekten helt och Kvaliteket blev aldrig den källa till empiri som vi ursprungligen behövde för vårt avhandlingsarbete. Detta ledde till en mindre kris och ett ifrågasättande av flera idéer som projekten vilade på. I det följande skall vi redogöra för dessa problem och relatera de tankar som projekten utgick ifrån med vår egen undersökning och slutsatser från denna.

### 8.1 Att definiera kvalitet

I projektet "IT-designkvalitet – paradigmatiske form och funktion" utvecklades en modell, som sedan vidareutvecklades i projektet "Kvaliteket", vars syfte var att fånga fenomenet brukskvalitet som ett system av kvalitetsegenskaper och utgångspunkter för värdeomdömen. Målet var att karakterisera innehållet och omfånget i begreppet genom att definiera olika perspektiv, där brukskvalitet definierades som balansen mellan de olika perspektiven<sup>130</sup>. Denna balans kan exemplifieras genom antagandet att upplevelsen blir sämre om en IT-artefakt inte uppfyller användares syften eller om

---

<sup>130</sup> Det vill säga aspekterna form, funktion och struktur med de sammanhörande kvalitetsperspektiven estetik, etik och kontroll.

den tekniska kvaliteten brister. Ett mindre tydligt men ändå tänkbart samband är att maktbetonade kvaliteter, såsom att ett datorsystem används för att kontrollera brukares nedlagda arbetstid, kan medföra en sämre upplevelse.

Men hur upptäcks och görs dessa samband? Vid en närmare betraktelse framstår det som att det saknas "teorier", mer än att säga att aspekterna är relaterade tillvarandra, som kan göra användaren av modellen uppmärksam på olika samband. Vad och var skall man söka? Låt oss säga att en dator har en klockfrekvens på 150 MHz och att brukaren i fråga har en god upplevelse av systemet. Kommer upplevelsen att bli bättre om klockfrekvensen ökar? Gäller detsamma för det motsatta, d.v.s. ger en sänkning av klockfrekvensen en sämre upplevelse? Exemplet kan tyckas vara något naivt och överförenklat, men det pekar på att vi inte vet hur ett sådant samband ser ut, kommer till uttryck och hur det bör tolkas.

För att ytterligare illustrera problematiken kan vår egen studie som gjordes på Volvo tjäna som exempel (Meggerle & Steen, 1994, 1995; Ehn et al., 1997a). En av slutsatserna vi drog var att säljstödet symboliska kvalitet var av märkbart värde. Med säljstödet betraktade säljarna sig inte längre som "hästhandlare", utan såg sig som mer professionella. Detta kan innebära att nyttan med säljstödet kan vara att få dess brukare att framstå och känna sig på ett speciellt sätt genom att datorsystemet signalerar ett "symboliskt" värde. Men hur relateras detta till struktur? Rent ut sagt: Vad är det i "burken" som bär upp detta värde? Är det den goda underhållsbarheten? Eller den höga tillförlitligheten? Är det gränssnittet eller ett antal gränssnittelement? Är det de laserskrivna snygga och redliga offerterna som kunden får i handen som uppskattades så mycket? Eller, till sist, är det allt detta sammantaget? Modellen och den teori den bygger på ger här ingen vägledning. På så sätt kändes de möjliga sambanden spekulativa och modellen kan därför sägas präglas av en viss teorilöshet<sup>131</sup>. Vi saknade helt enkelt påstående-

---

<sup>131</sup> Det är ingen determinism vi efterlyser, d.v.s. givet upplevelse U så har vi funktionen F och strukturen S. Utan snarare "teorier" som kan användas dels för vägledning och dels som utgångspunkt för tolkning.

kunskap eller med andra ord formaliserad förtrogenhet- och färdighetskunskap. Likaså saknade vi själva i projektet, i alla fall författarna, den nödvändiga praktiska kunskapen, d.v.s. precis det vi nämnde, förtrogenhets- och färdighetskunskap. Kunskapen bakom definitionsakten saknades och därför medförde operationaliseringen svårigheter i tillämpningen av modellen.

Därtill kan även projektets inramning av begreppet brukskvalitet diskuteras i relation till modellen och föreliggande studie. Inom projektet har det från början varit klart att brukskvalitet handlar om de kvaliteter som uppstår i bruket och att detta bruk har varit den kontext som omger en slutanvändare<sup>132</sup>. Men vad och vem är en brukare? Är det en sekreterare som använder ett ordbehandlingsprogram eller är det, som i vår studie på Volvo, en säljare som använder ett säljstöd? Eller kan säljchefen som använder säljstödet för försäljningsstatistik och kontroll betraktas som användare? Kan de programmerare som underhåller systemet betraktas som användare? Ramen som projektet gav begreppet brukskvalitet framstod därför inte som speciellt tydlig. Detta förstärks genom att modellen även bjuder in andra typer av användare.<sup>133</sup>

Tar man exempelvis strukturperspektivet och tillhörande kvalitetsperspektiv i närmare betraktelse kan man till vissa delar ifrågasätta om en del av innehållet är relevant för en representation av brukskvalitet. Detta perspektiv tar sin utgångspunkt i vad vi skulle kalla det klassiska sättet att betrakta kvalitet – d.v.s. med fokus på mjukvaran (Dahlbom & Mathiassen, 1993). Några begrepp som kan anses intressanta ur detta perspektiv är underhållsbarhet, flyttbarhet, testbarhet, tillförlitlighet (struktur) samt ett antal mätmetoder eller praktiska tillvägagångssätt för att kontrollera dessa (kontroll). Om vi till exempel finner att en specifik mjukvara har hög sammanhållning, låg koppling, är modulärt uppbyggd och väl dokumenterad

---

<sup>132</sup> Vi borde nog ha använt ordet slutbrukare här, men eftersom det inte är etablerat och låter krystat används begreppet slutanvändare.

<sup>133</sup> För ett litet exempel på hur olika kvalitet kan uppfattas av olika typer av intressenter eller i vissa fall ”användare”, se (Walsh, 1994).

kan vi påstå att underhållsbarheten är god (Pressman, 1987; Sommerville, 1996; Bass et al., 1998). Detta kan leda till att felaktigheter och andra skavanker lättare kan korrigeras. Dessutom kan det innebära att mjukvaran på ett bättre sätt kan förändras och utvecklas i nya riktningar för att möta omgivningens krav. Detta kan ha en långsiktig betydelse för till exempel en säljare, men vid det specifika brukstillfället är det vår tro och erfarenhet att det är av ringa betydelse. Däremot är dessa kvaliteter mycket viktiga för en annan yrkesgrupp, nämligen för dem som underhåller och utvecklar datasy-stemet.

I vår studie av säljstödet visade det sig också att det fanns möjlighet att från ”ovan” kontrollera antalet sålda bilar. Denna funktion var inte tillägnad säljarna. I studien upplevdes detta inte som ett problem utan snarare som säljkultur. Betraktas detta som en del av bruket är det fråga om inte ytterligare en typ av användare bjuds in (av systemet), d.v.s. i vårt fall en säljchef. Projektets ursprungliga fokus på någon slags ”slutanvändare” framstod därför inte som genomtänkt.

Projektets inramning av begreppet brukskvalitet i relation till modellen kändes därför som tveksam. Det bjuds in nya typer av brukare och därmed blandas omedvetet olika perspektiv, kontexter och roller.

### *8.1.1 Att använda exempel*

Ett av de grundläggande målen inom projekten var att skapa eller att finna paradigmatiska exempel. Ser man på den kontext begreppet är hämtat ifrån, d.v.s. arkitektur, innehar dessa exempel vissa kvaliteter. Dessa kvaliteter är ofta relaterade till en viss stil eller tradition och det paradigmatiska i det hela är att det är ett ypperligt exempel på det som exemplifieras, d.v.s. stil. En äkta stil reflekterar moral och estetiska värderingar i den kultur där den är skapad. Den är inte baserad på försköning eller eklekticism och är inte en trend driven av behov och variation. Den har en lång överlevnad, till skillnad från trend, med djupa rötter i samhället, där stilen i sig är en

del av meddelandet. En igenkännbar stil reflekterar en historisk, kulturell och artistisk kontext som i vissa drag är familjär (Ehn, 1994; Mullet & Sano, 1995; Ehn et al., 1997a). Det paradigmatiska exemplet reflekterar en stil och stilen reflekterar "samhället" där den uppträder. Tillsammans utgör stil och paradigmatiska exempel en utgångspunkt för dialog och diskussion. Två arkitekter som diskuterar en byggnad har förmågan att hänvisa till byggnadens stil (helhet), jämföra den med andra stilar, diskutera dess olika delar och relatera detta till helheten. Stil och paradigmatiska exempel är två viktiga element i arkitektens språk. Ett språk för att beskriva deras yrke, att systematisera och strukturera deras värld och historia. Stil skapar struktur i överflödet.

Med denna förebild som tro var målet med projekten att på ett liknande sätt skapa samma möjligheter till kommunikation och tillgängning av bedömningsförmåga gällande brukskvalitet. Projektet tog dock avstånd från arkitektens något ensidiga fokus på byggnaders yttre struktur och förde in ett bruksperspektiv<sup>134</sup>. Det var brukaren och brukskvalitet som var i centrum.

Men begrundar vi begreppen stil, paradigmatiska exempel och brukskvalitet framstår en övergripande skillnad i fokus – en skillnad ifråga om vem som värderar. Låt oss säga att en användare utvecklar ett eget litet system för registrering av personuppgifter. Användaren tycker att det är en klar förbättring mot den tidigare rörliga manuella bearbetningen. Funktionsmässigt fungerar systemet som vilket registersystem som helst och upplevs som riktigt bra. Användaren ser sig själv som en vinnare och visar stolt upp sitt system. Men som forskare och praktiker ser man att det inte finns några ledtexter vid inmatningsfält. Vissa kommandoalternativ finns inte i menyerna utan aktiveras genom kryptiska flertangentskombinationer. Gränssnittet bryter mot en mängd kända principer. Naturligtvis är koden ganska dåligt skriven och kan närmast betraktas som spaghettkod. Nu kan man fråga sig utifrån vårt uttalade bruksfokus om det verkligen är vi eller någon annan än brukaren själv

---

<sup>134</sup> Överför vi detta till arkitektur skulle man till exempel fråga sig hur boendet upplevs i en viss byggnad.



som skall bedöma och värdera strukturen. Men kan användare göra det? Enligt vår mening, är svaret nej. Alltså får vi här två motstridiga värderingar.

För att återkoppla till exempel, uttrycker paradigmatiska exempel stilen. I vårt fall skulle det bli: exemplet uttrycker brukskvalitet som är en stil. Men vilken stil uttrycker detta exempel? Kan det vara ett paradigmatiskt exempel? Svaret är enligt oss nej. Stilen ligger enligt modellen i balansen mellan perspektiven och i ovan skisserade exempel, även om det är påhittat, finns det ingen balans.

Dessa sistnämnda frågeställningar för oss vidare till en ny problematik. För att avgöra om ett exempel är paradigmatiskt eller inte, krävs en referenspunkt och möjligheten att jämföra. Det var ju projektgruppens uppgift att ur mängden exempel skapa generella stilar. Men precis som Hermerén (1982) påpekar är detta en hönan-eller-ägget problematik. Vilket kommer först? Är den representerade IT-artefakten ett resultat av en viss stil eller har den givit upphov till en viss stil? Båda implikationerna, IT-artefakt  $\rightarrow$  stil eller stil  $\rightarrow$  IT-artefakt, bygger på en jämförelse och en kunskap om den rådande stilen. I den första implikationen leder IT-artefakten till en ny stil, d.v.s. en stilbrytning – ett nytt paradigm, och i den andra uppvisar IT-artefakten stora likheter med den rådande stilen, som därför kan betraktas som upphovet till IT-artefaktens karaktär. Således behövs en kunskap om den rådande stilen innan en stilbrytning eller stilekvivalens kan bekräftas. Denna kunskap tyckte vi inte att vi hade.

En lösning på denna kunskapsbrist skulle kunna vara att låta en representation företräda den rådande stilen. Vi väljer helt enkelt en IT-artefakt som riktmärke. Detta låter sig dock inte göras eftersom de bruksegenskaper (ev. stilmarkörer) som representeras är kontextuella och därför inte kan generaliseras. Samma egenskaper borde alltså inte uppstå i annan kontext, vilket skulle ge upphov till ständiga stilbrytningar.

Gör vi däremot oss av med den eftersökta generaliteten skulle kontextuella stilbegrepp som är oberoende av varandra möjligen kunna

skapas. Man skulle kunna låta en grupp av exempel med liknande egenskaper uttrycka en stil. Stilkaraktistika för denna mängd skulle till exempel kunna vara att IT-artefakten stärker säljarnas professionalism – kanske professionalismstärkande IT-artefakter. Alla exempel som utmärks av denna egenskap skulle kunna ingå i denna mängd. Vidare skulle också ett exempel kunna ingå i flera olika mängder eftersom exemplets andra egenskaper kan ha likheter med andra mängder. Precis som ovan bygger metoden på jämförelse för att avgöra om en representation av kvalitet i bruk tillhör mängden eller inte. Men istället för att jämföra med en rådande generell stil, grupperas exemplen efter kontextuella likheter och skillnader. Men då frikopplas egenskaper från den övriga delen i representation och vi bortser från den balans som modellen skisserar. Borde inte stilen ligga i balansen? Att använda stil eller att skapa en stilteori inom IT-området och i projektet, framstår med andra ord som problematiskt.

## **8.2 IT-kvalitet i praxis**

Vi har nu kritiserat och skisserat det som vi har upplevt som problem i projekten. Det är bland annat av dessa anledningar våra egna utgångspunkter är formulerade som de är i föreliggande avhandling. Frågan är då hur vår undersökning relaterar till de projekt vi deltagit i och de utgångspunkter som formulerades i dem – hur förhåller sig våra resultat till projektens antaganden och målsättningar? För att diskutera detta skall vi inleda med att lyfta fram resultaten från vår egen studie.

Vår diskussion i kapitel 6 lyfter, som sagts tidigare, fram vissa viktiga resultat av vår undersökning kopplade till teorier om IT-kvalitet och kunskap/kunnande. Vad vi har gjort är att anlägga ett praxisperspektiv som innebär att dels låta praktiker komma till tals i frågan om IT-kvalitet och dels att koppla detta till praktiskt kunnande. Detta är ett synsätt som vi anser har saknats i diskussionen om fenomenet IT-kvalitet. De aspekter vi speciellt har belyst handlar om kvalitetsegenskaper och kvalitetsdefinitioner som en kunskapsfrå-

ga, bedömningsförmågans betydelse, innehåll och utveckling, samt kommunikationens och reflektionens betydelse för det praktiska kunnandet kring IT-kvalitet. Det handlar alltså om IT-kvalitet som ett fenomen och bedömningsförmåga kopplad till detta.

När det gäller fenomenet IT-kvalitet, så framstår det som egenskaper etiketterade med begrepp och olika typer av definitioner. I förhållande till detta har vi funnit två saker. Den ena är att relativt få av de begrepp som framstår som väl befästa och inarbetade inom Software Engineering och MDI används, vilket hänger ihop med att de inte förfaller vara lika väl inarbetade hos de systemutvecklare som vi har intervjuat. Dessa begrepp framstår inte heller som delar i en språkapparat och som kommunikativa instrument.

Det andra som vi har funnit i samband med fenomenet IT-kvalitet, är att definitioner antingen är mycket svåra att fastställa och konkretisera, eller helt enkelt omöjliga att åstadkomma på ett sådant sätt att de går att operationalisera. Definitioner verkar vara av den karaktären att de antingen uttalar sig väldigt generellt om kvalitet och därmed inte ger någon djupare information om hur kvaliteten kommer till uttryck, eller att de är för specifika för att fånga hela vidden av kvalitet. De leder alltså egentligen inte till konkreta svar på vad IT-kvalitet är och hur den skall uppnås.

Att det förhåller sig på detta vis har vi kopplat till främst Johannesens (1999) teorier om praktiskt kunnande som leder till att påståendekunskap, i detta fall kvalitetsdefinitioner, inte är uttömmande och självständiga aspekter på kunskap. Definitionerna kan aldrig ensamma rymma all den kunskap som behövs för att uppnå kvalitet, samtidigt som definitionsakten i sig utgår från kunskaper som inte själva är av påståendekaraktär. Vad definitioner kan göra är att rikta uppmärksamheten mot vissa aspekter som måste beaktas för att uppnå kvalitet, men de kan inte ersätta det erfarenhetsbaserade kunnandet av färdighets- och förtrogenhetskaraktär. För att en definition skall vara meningsfull måste den kopplas till färdighet och förtrogenhet, annars kommer den att bli intransitiv i det att den förvisso uttalar sig om något men samtidigt inte förmedlar någon förståelse. I Rolfs (1995) termer är kvalitetsdefinitioner att likna vid

maximer som inte själva innehåller all nödvändig kunskap, utan endast kan tjäna som ledtrådar om de kopplas till en praktisk kunskap inom färdighetsområdet.

Svårigheten att definiera kvalitet hänger också ihop med att kvalitet kan ses ur olika perspektiv och ur olika intressenters synvinklar. Kvalitet är t.ex. inte endast att kunden är nöjd med produkten, den skall helst också kunna underhållas för att ta ett exempel. Därför framstår även systemutvecklarna själva som intressenter som i sin tur kan anlägga olika perspektiv och fokusera på olika egenskaper och tillmäta dem olika vikt. IT-kvalitet kan därför inte bli något enkelt, utan är ett sammansatt och komplext fenomen.

Detta leder till att färdigheten och förtrogenheten, eller den personliga kunskapen, är av central betydelse för de systemutvecklare som vi har intervjuat. Eftersom mycket av det som kan betraktas som IT-kvalitet inte låter sig operationaliseras och kvantifieras, blir det också mycket svårt att sätta upp mått för och att mäta kvalitet, annat än om det skulle gälla väldigt tekniska och enkla egenskaper. Inte heller när det gäller att bedöma kvaliteten kan alltså normativa angreppssätt av påståendekaraktär vara tillräckliga, vilket leder till att det viktigaste sättet att bedöma kvalitet för våra systemutvecklare bygger på färdighet kopplad till förtrogenhet.

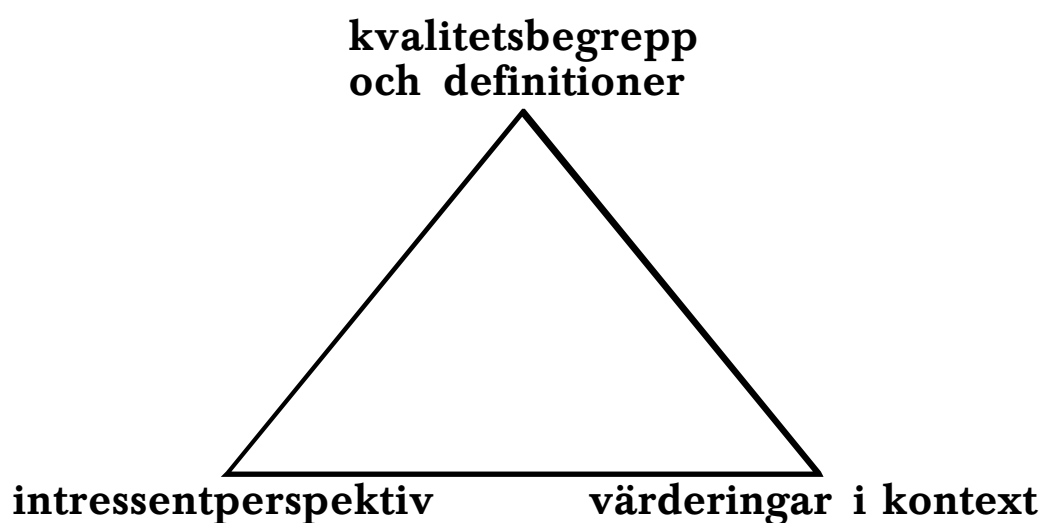
Den mesta kvalitetsbedömningen är alltså erfarenhetsbaserad och det är också erfarenheten som är den viktigaste källan till bedömningsförmågans innehåll och utveckling. Även om cirka 80% av kvaliteten upplevs kunna nås genom normativa och formella angreppssätt, exempelvis metoder, så bygger de resterande och mycket viktiga 20% på vad vi har kallat tyst kunskap i undersökningen. Att erfarenheten är så viktig är dock inte något som framkommer i den litteratur om IT-kvalitet som vi har använt i denna avhandling. Denna praxisdimension förefaller att vara en glömd aspekt inom området.

Förvånande är också att reflektion och kommunikation inte tillmäts så stor betydelse i praxisen, när erfarenheten upplevs som så viktig. Kommunikationen hindras av en brist på en lämplig språkapparat

och en frånvaro av indirekta kommunikativa inslag som exempel och metaforer. Reflektionen hindras av bristen på tid, vilket innebär att aktivt reflekterande över innebörden i IT-kvalitet och kvaliteten på de produkter man framställer inte äger rum i någon större omfattning. Som Rolf (1995) påpekar är reflekterad kunskap nödvändig för att praxisens know-how skall kunna utvecklas.

Vi avslutar denna återgivning av våra resultat med en bild (Figur 8.1) som vi tycker stämmer bra med hur fenomenet IT-kvalitet gestaltar sig för oss. Bilden skall inte tolkas som en strikt geometrisk figur, där storlek och yta har relevans, utan den tjänar till att visa relationer mellan viktiga aspekter som vi anser utgör fenomenet IT-kvalitet.

IT-kvalitet i praxis är ett komplex av tre aspekter, där begrepp och definitioner kan peka på viktiga aspekter att tänka på, men där också kvalitet alltid är en värdering som kopplas till vissa perspektiv. Vi kan alltså inte definiera IT-kvalitet, men vi kan definiera fenomenet IT-kvalitet som ett antal begrepp och värden sedda ur intressenternas perspektiv och bedömda i den kontext där datasystemet/IT-artefakten utvecklas, underhålls och används. Bilden innebär att kvalitetsegenskaper, som delvis kan uttryckas genom begrepp och definitioner, väljs av intressenter för vissa kontexter och att intressenternas bedömningar av kvaliteten sker i relation till dessa kontexter och alltid är värdering. Relationerna mellan aspekterna skall dock inte tolkas som absoluta och enkla, vilket vi har visat i den här avhandlingen. Begreppen och definitionerna är inte entydiga och uttömmande och måste inte ingå som element i en bedömning av kvalitet. De värden de representerar varierar över tiden genom att nya tillkommer och att gamla kanske försvinner beroende på att intressenter och kontexter utvecklas, förändras och tillkommer. Därmed är också IT-kvalitet som fenomen föränderligt och komplext, och längre än så här vill vi inte sträcka oss i den här avhandlingen.



*Figur 8.1:Fenomenet IT-kvalitet som tre aspekter.*

Vi hoppas att dessa synpunkter och vår avhandling har lett till en fördjupad förståelse av IT-kvalitet i praxis och att de leder till viktiga reflektioner, och kanske aktioner, hos våra läsare och att de tycker att det har varit en intressant och tankeväckande läsning. Därmed tackar vi för visat intresse och sätter punkt.

## 9 Avslutande kommentarer

Vi avslutade i föregående kapitel själva avhandlingen med att visa hur IT-kvalitet gestaltar sig som ett komplex av tre aspekter och konstaterade att vi inte ville gå längre. Detta för att inte inbjuda till spekulation och nya vinklingar på problematiken. Vi kan dock inte avstå helt från det och tillåter oss i dessa avslutande kommentarer att ta ut svängarna och måla med lite bredare pensel.

Både de projekt vi har deltagit i och vår egen studie utgår ifrån att kvalitet är viktigt och att den naturligtvis kan bli bättre. Utgångspunkten i projekten är förvisso att den brister, särskilt när det gäller bruksegenskaper, och att dessa brister hänger ihop med att professionella praktikers bedömningsförmåga måste förbättras. Vår egen studie utgår snarare från att denna bedömningsförmåga är intressant att studera i sin egen rätt, utan att utgå från att den skulle vara bristfällig.

En förenande utgångspunkt är alltså att kvalitet är viktigt för att IT-artefakter skall kunna brukas på ett meningsfullt och värdefullt sätt. Problemet är samtidigt att kvalitet i sig är ett positivt värdeladdat ord och det gör det svårt att argumentera för att kvalitet är viktigt, för vem skulle säga att kvalitet inte är viktigt? Dock är inte kvalitet en objektiv egenskap med ett absolut värde, utan som bilden ovan visar handlar kvalitet alltid om värdering utifrån perspektiv och att en maximering av alla egenskaper utifrån alla perspektiv är en omöjlighet, eller åtminstone mycket kostsamt. Som vi har visat i vår studie är kvalitet alltid en kompromiss mellan de olika perspektiven och att det handlar om vilken nivå av kvalitet man vill uppnå – kvalitet är också en fråga om att hantera förväntningar.

För att återvända till kvalitetens betydelse, så uppträder den något märkliga situationen att alla säger att kvalitet är viktigt, utan att kunna tala om vad kvalitet egentligen är. Det finns alltså en paradox inbakad i strävan efter kvalitet – hur skall man kunna sträva efter något som man inte riktigt vet vad det är? Svaret på denna fråga är en önskan om externalisering av kvalitetens betydelse och kvali-

tetsbedömning; att kunna förlita sig på standarder, metoder, mått med mera. Men eftersom kvalitet är så komplext så fungerar detta endast delvis, beroende på att formaliserad kunskap i formen av exempelvis metoder inte kan användas utan erfarenhetsbaserad kunskap om tillämpningen av dessa och de anpassningar och kompromisser man måste göra. Den erfarenhetsbaserade, praktiska kunskapen är alltså av central betydelse och i det perspektivet var utgångspunkterna i projekten riktiga – bedömningsförmågan, den praktiska, personliga kunskapen, är av avgörande betydelse.

Den stora skillnaden mellan vårt tänkande och den strävan som fanns i projekten, är synen på hur bedömningsförmågan kan förbättras. Kommunikation och reflektion framstår som centrala inslag för att bedömningsförmågan skall kunna utvecklas. Som vi har visat i vår studie är kommunikation och reflektion inga självklara och "medvetna" inslag i systemutvecklingspraxisen, åtminstone inte där vi har studerat den. Att förbättra möjligheterna till reflektion och kommunikation framstår därför som viktigt, men det finns flera sätt att göra detta.

Tankarna i projekten var att denna möjlighet skulle bestå i reflektion över och kommunikation kring paradigmatiska exempel, ungefär som vi inom projekten föreställde oss att arkitekturpraxisen fungerade. På detta sätt skulle systemutvecklare kunna ta erkända, goda förebilder som en utgångspunkt för att stödja reflektionen för att förbättra sin bedömningsförmåga och genom Kvaliteket ville vi inom projekten skapa ett "samfund" med dessa paradigmatiska exempel och kommunikation kring dem, i syfte att systemutvecklare skulle kunna reflektera över sin egen design i förhållande till erkända förebilder.

Det kan säkert finnas goda skäl till en sådan ansats, men förutom alla de problem med den ansatsen som vi har diskuterat ovan, för det också med sig ett utifrånbetonat expertperspektiv på problematiken. I en praxis där reflektion och kommunikation inte är självklara och medvetna processer, blir en sådan ansats en kontradiktion. För hur skall man reflektera och kommunicera om det inte finns en kultur eller tradition av att diskutera IT-kvalitet? Det fram-



står också som nödvändigt att man kan definiera kvalitet för att nå erkända förebilder, en definitionsakt som vi har funnit omöjlig att genomföra – kvalitet är ett för komplext fenomen för att låta sig fångas i definitioner som inte blir intransitiva. Så det ena problemet är kvalitetsbegreppets komplexitet, som kräver praktisk kunskap för att kunna hanteras, och det andra är den manifesta form de goda förebilderna skulle få, genom att de påläggs praxisen utifrån.

Ett annat sätt att skapa möjligheter för förbättring av bedömningsförmågan, är att anlägga ett "gräsrotsperspektiv" på denna målsättning och utgå från praxisen själv – att så att säga låta grunden för kommunikation och reflektion komma underifrån. Som vi har visat i studien används få av de begrepp som myntats inom viktiga teoretiska områden när det gäller systemutveckling. Dessa begrepp skulle kunna användas i större utsträckning inom systemutvecklingspraxisen som en utgångspunkt för att forma en språkapparat, för att användas för att nå det Rolf (1995) kallar för reflekterad kunskap. Den stora skillnaden mellan denna tänkta ansats och den som projekten baserades på, är att betydelseinnehållet i och tolkningen av språkapparatens och dess begrepp skulle formas *inom* praxisen och bygga på de färdigheter och den förtrogenhet som uppstår där. Det handlar i såfall om att medvetandegöra behovet för systemutvecklare och få dem att erövra en teoretiskt väl etablerad språkapparat som de kan anpassa för sina syften. Vår syn på hur bedömningsförmågan kan utvecklas är alltså att det måste ske inom praxisen med utgångspunkt i de tankar och begrepp som finns inom teorin, men inte genom att vi lägger på praxisen färdiga definitioner och betydelser som skall vara förebildliga, eftersom vi inte utgör den praxis som vi studerar.

Med utgångspunkt hos Johannessen (1999) räcker det dock inte med verbal artikulation för att uttrycka kunskap, eftersom viss kunskap är av en sådan karaktär att den inte lätt låter sig fångas i ett verbaliserat språk. I sådana situationer krävs indirekta "språk" i form av metaforer, exemplifieringar med mera, precis så som utgångspunkterna var i projekten. Den tanken är fortfarande bärkraftig eftersom kvalitet har visat sig vara ett så komplext begrepp och eftersom bedömningsförmågan har ett mycket starkt inslag av tyst

eller tyst fungerande kunskap. Olika typer av exempel på IT-artefakter är därför en mycket god idé för indirekt kommunikation kring kvalitet och viktiga kvalitetsegenskaper. Frågan är dock återigen om dessa exempel skall komma in i praxisen utifrån eller om de skall uppstå i praxisen själv.

Vår tro är att det är mer meningsfullt om de uppstår eller framställs i den kontext där färdigheterna och förtroendet finns, eftersom de då kan få det innehåll som lämpar sig inom systemutveckling så som systemutvecklare själva skulle anse sig ha nytta av dem.

Vad vi alltså efterlyser är att systemutvecklare i mycket större utsträckning än idag, reflekterar över och kommunicerar kring IT-kvalitet, dess betydelser och tolkningar, samt viktiga egenskaper som ger begreppet kvalitet ett innehåll. Att eftersträva objektiva och generella definitioner av IT-kvalitet verkar lönlöst, eftersom de inte innehåller all den kunskap som krävs för att uppnå kvalitet. Inte heller formaliserade angreppssätt såsom metoder, riktlinjer och mättekniker räcker, eftersom de tenderar att bli maximer som inte kan brukas på ett vettigt sätt utan praktisk kunskap om det de uttrycker. Bedömningsförmågan låter sig inte helt regelstyras och därmed krävs andra ansatser för att förbättra bedömningsförmågan, vilket måste vara en pågående process eftersom såväl teknik som tillämpningsområden ständigt utvecklas och förändras, och därmed förändras också vad som kan vara hög kvalitet.

Om vi skulle betrakta detta som ett nytt projekt, skulle det vara att samarbeta med systemutvecklare för att föra ut väl befästa kvalitetsbegrepp som en första grund för ett verbalt språk att kommunicera med. Vi skulle också behöva finna och/eller arbeta fram goda exempel, goda i den meningen att de visar på något betydelsefullt när det gäller IT-kvalitet, som kan användas för reflektion. Goda exempel och goda begrepp ihop bör vara en kombination som kan leda långt. Samtidigt måste naturligtvis såväl reflektion som kommunikation i syfte att stärka bedömningsförmågan ges utrymme för att kunna ske aktivt och betraktas som ett viktigt inslag i praxisen systemutveckling. Till detta kommer också att aktivt undersöka hur kvaliteten på produkterna uppfattas av brukarna och detta inte

endast när den upplevs som bristfällig. Även uppfattningar av god kvalitet måste bearbetas och analyseras eftersom det finns mycket god erfarenhet att ta till vara kring detta.

Slutligen hade det varit nödvändigt att på något sätt utvärdera projektets resultat, vilket endast, som det framstår för oss just nu, hade kunnat ske genom att systemutvecklarna själva uttrycker huruvida deras bedömningsförmåga hade förbättrats eller utvecklats.

För att avsluta denna avhandling är våra resultat i mycket överensstämmande med utgångspunkterna i de projekt vi deltog i begynnelsen av våra forskarstudier. Det som skiljer våra tankar från tankarna i dessa projekt, är att vi mer tror på en "gräsrotsnivå" när det gäller hur exempel och språk skall utvecklas för att de skall kunna fungera som stöd för att kunna förbättra bedömningsförmågan. Denna bedömningsförmågas vikt framgår inte av den teoretiska behandlingen av IT-kvalitet som ett begrepp, men framstår i vår undersökning som mycket viktig.

## Populärvetenskaplig sammanfattning

Alla har vi väl någon gång varit med om att datorer, dataprogram och datasystem inte fungerar som vi har tänkt oss. Inte sällan inträffar det fel och konstigheter som kan vara frustrerande och, i värsta fall, direkt farliga. Många gånger undrar man hur ett visst program eller system skall användas och hur de som har designat och byggt det har tänkt och resonerat.

I stort handlar detta om kvalitet eller ännu hellre olika kvaliteter, d.v.s. egenskaper hos produkter som påverkar upplevelsen av att använda dem, hur lätta de är att förstå och lära sig att använda, vilken nytta man kan ha av dem, möjligheten att förändra dem till nya krav som omgivningen ställer och möjligheten för dem som har utvecklat dem att själva förstå hur de fungerar och kan anpassas.

Att ITs genomslag i samhället dessutom ökar och berör allt fler människor och verksamheter och därmed har en stor påverkan på många människors vardag, gör det relevant att studera kvalitet på den alltmer omfattande och närvarande informationstekniken (IT) som datasystem och dataprogram är en del av.

Kvalitet har också blivit ett sorts slagord och närmast ett krav för olika verksamheter för att kunna konkurrera på marknaden. I en allt större utsträckning talas det om och efterfrågas olika satsningar för kvalitetsarbete, exempelvis ISO-9000 certifieringar, för att garantera kvalitet och trovärdighet. Detta gäller också inom den bransch och de verksamheter som tillverkar mjukvara, d.v.s. datasystem och dataprogram. Det handlar då om en del av det totala området kvalitet som kallas IT-kvalitet, d.v.s. kvalitet på informationsteknik.

Det finns alltså goda skäl att intressera sig för och studera systemutvecklares kunskap om och syn på IT-kvalitet.

Anledningen till att vi, Odd och Theis, intresserar oss för systemutvecklarens perspektiv, hänger ihop med att bland andra Mitchell Kapor (1996), som var en av grundarna till Lotus Corporation och en av huvudmännen bakom det en gång mycket framgångsrika kalkylprogrammet Lotus 1-2-3, har riktat kritik mot systemutvecklarens förmåga att designa goda och användbara produkter. Mitchell Kapor efterlyser nya arbetssätt och en bättre förmåga att skapa god kvalitet för användarna.

Frågor om arbetssätt och förmåga leder också till ett intresse för systemutvecklare som yrkesmänniskor. Hur en systemutvecklare tänker och arbetar är, bortsett från studier av Erik Stolterman (1991) och Christer Hoberg (1998), dock dåligt belyst.

Kopplas detta till att kvalitet blir allt mer viktigt, samtidigt som kritik riktas mot systemutvecklarens förmåga att uppnå god kvalitet, uppstår ett intressant område att studera. Detta område är IT-kvalitet som begrepp och systemutvecklarens kunnande om och syn på kvalitet. Kvalitet som begrepp är väl bearbetat inom delar av vårt område, däremot finns det mycket lite skrivet om systemutvecklare som yrkesutövare.

Kvaliteten på system och program blev redan tidigt i ITs historia ett problem, eftersom den tidens teknik krävde att man ägnade all uppmärksamhet åt att utnyttja datorernas mycket låga prestanda på bästa sätt. System och program skrevs med effektivitet som ledstjärna, vilket ledde till att snabba och resurssnåla men kryptiska program skapades.

När de som använde dessa enkla system och program, i huvudsak tekniker och vetenskapsmän inom USA försvarsmakt, var de samma som byggt dem, var det egentligen inga större problem. I takt med teknikens utveckling och spridning till andra verksamheter i samhället, och andra typer av användare, började emellertid mängden programvara och komplexiteten i dem att öka och därmed ökade också behoven att kunna förstå dem för att rätta de fel som alltid finns i program. En marknad för programvara uppstod också och den översvämmades med dåligt byggda program. Så

småningom var situationen så besvärlig att en mjukvarukris uppstod under 1960-talet och ett eget ämne, Software Engineering, skapades för att forska fram arbetssätt och teorier för att åstadkomma mjukvara som lättare kunde rättas och anpassas. Målet var att skapa objektiva och matematiskt grundade utvecklingsmodeller, som skulle anammas av den växande mjukvaruindustrin. Dessa ansatser lyckades inte helt, men gav ändå upphov till en bättre förståelse för god teknisk uppbyggnad av mjukvara.

Den tekniska kvaliteten på mjukvara förbättrades så sakteliga, men nya bekymmer uppstod. Även om mjukvaran är tekniskt perfekt hjälper det inte mycket om användarna av systemen och programmen inte förstår hur de kan användas och har svårt att lära sig dem. Inte heller hjälper teknisk perfektion mycket om mjukvaran inte stödjer de arbetsuppgifter som användarna utför. Man insåg därför efterhand att kvalitet på mjukvara inte bara handlar om rent tekniska egenskaper, utan att kvalitet lika mycket är en fråga om användbarhet. Därmed uppstod ett nytt område, Människa-Dator Interaktion (MDI), för att studera detta vetenskapligt och praktiskt.

Även inom MDI var man till en början intresserad av objektiva metoder och teorier som skulle vara allmängiltiga för alla människor. Människan sågs som ett sorts biologiskt maskineri som mjukvaran kunde anpassas till. Detta begränsade synsätt på människan kritiserades dock dels för att det utesluter den verkliga arbetssituation som de tänkta användarna befinner sig i, dels för att det inte tar med sociala och mänskliga perspektiv på arbete. MDI-området utvecklades därför i en mängd olika riktningar och är numera ett tämligen omfattande ämnesområde.

Software Engineering och MDI har på var sitt sätt bidragit till mycket av förståelsen för IT-kvalitet och hur man kan uppnå god kvalitet. De har också bidragit med en mängd definitioner och kvalitetsbegrepp som kan fästa uppmärksamheten på viktiga egenskaper.

Systemutvecklares kunskap om och syn på IT-kvalitet behandlas däremot inte i den teori om kvalitet som vi har studerat. De enda vi

har funnit som har undersökt det här är David Wilson och Tracy Hall (1998), som visar att synen på kvalitet varierar kraftigt mellan olika systemutvecklare och deras chefer.

Vi väljer att se IT-kvalitet som en kunskapsfråga och för att göra detta har vi utgått från, förutom teorierna om kvalitet, Kjell S. Johannessens (1999) tolkning av Wittgensteins senfilosofi och Bertil Rolfs (1995) tolkning av Polanyis teorier om praktisk kunskap. I korthet handlar detta om att det finns kunskaper som visar sig när man utför en handling. Under lång tid i det västerländska samhället har äkta och sann kunskap endast varit sådan som kan formuleras i logiskt klara satsar och regler. Viss kunskap låter sig dock inte fångas på detta sätt, exempelvis moralisk kunskap. Det är också så att det krävs kunskap för att formulera en logisk regel, men den kunskapen framgår inte av själva regeln. Det måste alltså finnas andra kunskaper än de som klart och koncist kan formuleras.

Kjell S. Johannessens använder begreppen påstående-, färdighets- och förtrogenhetskunskap för att tala om olika sorters kunskaper. Påståendekunskap är kunskap som kan formuleras och uttryckas verbalt. Färdighetskunskap däremot visar sig i handling; det är i handlandet man kan se kunskapen. För att kunna handla, och därmed uppvisa färdighetskunskap, måste man vara bekant med en mängd situationer som möjliggör handlingen. Man måste alltså också ha förtrogenhetskunskap och även den är mycket svår att uttrycka verbalt. Det begrepp som har kommit att beteckna denna typ av kunskaper är tyst kunskap. För att ändå kunna kommunicera den tysta kunskapen måste man förlita sig på indirekta medel som exempel och liknelser.

Även Bertil Rolf behandlar praktisk och tyst kunskap. Skillnaden är utgångspunkten att den tysta kunskapen fungerar tyst, snarare än att den inte är möjlig att uttrycka verbalt. Det handlar alltså mera om en sorts kunskap som vi utnyttjar i bakgrunden när vi utför våra handlingar. Viss kunskap kanske aldrig kan uttryckas verbalt, men vilken kunskap det är kan vi inte peka ut.

Enligt Bertil Rolf har man praktisk kunskap om man kan utföra en handling väl enligt vissa regler eller kriterier som skiljer en illa utförd handling från en väl utförd och om reglerna eller kriterierna fungerar tyst. Att ha praktisk kunskap handlar alltså om att kunna utföra handlingar med god kvalitet i förhållande till olika kriterier. Om den som har praktisk kunskap skall vara kompetent innebär det dessutom i korthet att kriterierna är förankrade hos andra utövare som avgör kvaliteten på ens handling, att kriterierna på något sätt kan uttryckas och reflekteras över, samt att utövaren kan påverka kvalitetskriterierna för att förbättra dem. Endast då kan man tala om kompetens enligt detta synsätt.

Kompetens kan med andra ord aldrig endast handla om att följa regler. Dels krävs det erfarenhet för att förstå reglerna, dels måste reglerna kunna förändras till det bättre. Därför är reflektion och olika sätt att uttrycka kunskap avgörande för att den skall kunna utvecklas tillsammans med andra. Det gäller då att fokusera på tysta kunskap och uttrycka den så att olika personers erfarenheter kan spridas och diskuteras. Den egna erfarenheten kan dock inte ersättas, eftersom den behövs för att förstå regelsystemet som man försöker förbättra.

Detta är våra teoretiska utgångspunkter – IT-kvalitet samt kunskap, kunnande och kompetens. I vår empiriska undersökning av systemutvecklarens kunskap om och syn på kvalitet ställer vi bland annat följande frågor: Vilka begrepp använder de för att tala om kvalitet? Vilka betydelser lägger de i begreppet kvalitet? När talar de om kvalitet och hur? Hur bedömer de kvalitet? Är erfarenhet viktigt för att kunna bedöma kvalitet? Är erfarenhetsutbyte viktigt för att utveckla den praktiska kunskapen om kvalitet? Dessa frågor hänger ihop med vår vilja att se systemutveckling som en praxis, d.v.s. ett yrkesmässigt sammanhang med regler och kriterier för bedömning och praktisk kunskap. Att tala om begrepp och definitioner innebär att tala om kunskap som kan uttryckas verbalt, d.v.s. påståendekunskap. Vi vill även försöka förstå den praktiska kunskapen i form av färdighet och förtrogenhet, eller personlig praktisk kunskap för att använda Bertil Rolfs begrepp. Förmågan att bedöma kvalitet, erfarenhetens och reflektionens betydelse, samt erfa-



renhetsutbyte är då intressanta att studera, eftersom de är uttryck för praktisk kunskap och kompetens.

För att få svar på dessa frågor genomförde vi 19 djupintervjuer med systemutvecklare, programmerare, projektledare och funktionsdesigners. Dessa intervjuer genomfördes under perioden december 1998 till juni 1999 på tre olika företag, varav ett var ett konsultföretag och de två andra interna IT-avdelningar. Alla intervjuerna spelades in på band och skrevs ut ordagrant.

När det var gjort analyserade vi materialet med ett kodningsramverk byggt på Johannessens tre kunskapsformer och gjorde sedan tolkande sammanfattningar av intervjuerna. För att presentera resultaten skrev vi ett fiktivt samtal mellan tre personer. Projektledaren och systemutvecklaren Eva, hennes man Adam, som är programmerare och systemutvecklare, och Lars, som fungerar som samtalsledare, för ett samtal om vad kvalitet betyder, hur viktig den är, hur den bedöms, hur förmågan att bedöma den utvecklas och hur man kan tala om en bedömning av kvalitet med andra systemutvecklare.

Det vi kommer fram till när vi analyserar systemutvecklarnas förståelse för och kunskap om begreppet kvalitet, är att kvalitet alltid är en fråga om någons uppfattning och att kvalitet inte helt och hållet kan regelstyras, t.ex. genom standarder, mätmetoder och utvecklingsmetoder.

Kvalitetsbegreppet är också komplext och inrymmer många olika egenskaper som tillsammans utgör innebörden av kvalitet. Eftersom det är ett svårt och sammansatt begrepp, kan systemutvecklarna inte ge en allmän definition som slår fast vad kvalitet faktiskt är.

Bedömningen av kvaliteten på system och program sker genom att någon gör en värdering och jämför detta med vad man anser vara bra eller tillräckligt. Sällan handlar värderingen om att mäta eller beräkna. Därför vilar förmågan att bedöma kvalitet i viktiga avseenden på den personliga erfarenheten och därmed kan värdering av kvalitet endast till viss del regelstyras genom exempelvis olika

sätt att mäta eller beräkna. Metoder och beräkningssätt räcker alltså inte för att värdera kvalitet.

För att uppnå hög kvalitet räcker det enligt våra systemutvecklare inte att uppnå det som kunden har satt upp som krav på systemet eller programmet. Hög kvalitet blir det om kunden får något som är bättre än han eller hon har förväntat sig. För att nå hög kvalitet måste systemutvecklare ha en erfarenhet av och fingertoppskänsla för vad som kan ge hög kvalitet. För att kunna göra bedömningar kan regler och normer fungera som en grund, men när erfarenheten kräver det måste man bryta mot dem.

Förmågan att bedöma kvalitet utvecklas främst genom den personliga erfarenheten från olika projekt, kunder, användare o.s.v. Viktiga för att bygga upp denna erfarenhet är de åsikter som kommer från kunder och användare. Främst handlar det om negativa åsikter om saker och ting som inte fungerar som det är tänkt, mer sällan handlar det om beröm. Att systemutvecklarna talar med varandra om erfarenheter och åsikter är också viktigt för att bygga upp erfarenhet.

Om systemutvecklarna anser att erfarenheten är det viktigaste när det gäller kvalitet, kan man tänka sig att det skulle vara viktigt för dem att reflektera över vad kvalitet är, vad ordet betyder, vad som är hög eller låg kvalitet o.s.v. för att bli bättre på kvalitet och öka kunskapen. En sådan reflektion förekom dock i liten utsträckning, dels på grund av att systemutvecklarna inte hade tid avsatt för detta i arbetet, dels för att de var fokuserade på att lösa problem och rätta fel, dels för att de i allmänhet var omedvetna om att detta skulle kunna vara viktigt.

För att utveckla sin kunskap och sitt kunnande om kvalitet är det viktigt, utgår vi ifrån, att systemutvecklare talar med varandra om kvalitet och uppfattningar. Systemutvecklarna för samtal med varandra och anser att dem vara viktiga, men de mesta av samtalen handlar om aktuella projekt som de deltar i och sällan om att öka varandras kunskap i allmänhet. Detta kan bero på att systemutvecklarna upplever att de inte har ett speciellt bra språk eller bra

begrepp för att tala om kvalitet och värdering och att de därför använder ett ganska knapphändigt vardagsspråk som det är svårt att vara precis med. Användandet av exempel, liknelser m.m. var ovanligt hos våra systemutvecklare, i alla fall med syftet att mer allmänt diskutera kvalitet, värdering o.s.v. för att förbättra kunskapen om kvalitet och förmågan att bedöma kvalitet.

Systemutvecklarens bedömningsförmåga var också intresset i projektet "IT-designkvalitet – paradigmatiske form och funktion" och "Kvaliteket", som vi deltog i när vi blev forskarstuderande. Dessa projekt leddes av professor Pelle Ehn och pågick mellan 1993 och 1997. Grundtanken i båda projekten var att brukskvaliteten var låg på grund av systemutvecklarens bristande bedömningsförmåga och att de för att förbättra den behöver goda förebilder i form av stilar för system och utvecklare att jämföra sig med snarare än ytterligare utvecklingsmetoder. Vi i projekten fann därför att det var mer fruktbart att tänka på brukskvaliteten på produkterna, alltså systemen och programmen, än på sättet de utvecklas. Brukskvalitet uppfattades förenklat som att system skall vara bra byggda, vara väl anpassade till användares behov och ge en god upplevelse och känsla vid användning, d.v.s. bruk. En annan utgångspunkt var att systemutvecklare till viss del redan har förebilder och kunskaper som inte ryms i utvecklingsmetoder.

I projektet Kvaliteket utvecklade projektgruppen en speciell och allmänt tillgänglig webbplats, där man kunde finna beskrivningar av system ur de tre perspektiven på brukskvalitet, d.v.s. byggkvalitet, behovsrelevans och upplevelsekvalitet. En debattdel ingick också i Kvaliteket och där skulle enskilda beskrivningar och kvalitet i allmänhet debatteras. Det fanns även en namnkunnig jury som skulle bedöma kvaliteten på beskrivningarna och ge pris till de bästa, för att åstadkomma goda förebilder. På så vis ville vi i projektet skapa en mötesplats där goda förebilder skulle uppstå och användas för jämförelse och diskussion. Framgången uteblev dock och det är ett av skälen till att vi har skrivit den här avhandlingen.

Ett annat skäl är att vi, Odd och Theis, såg ett problem med projektens kvalitetsmodeller – begreppet brukare var dåligt utrett. En viss

aspekt av brukskvalitet, nämligen byggkvalitet, betraktades i modellerna som värderingsfri och fri från användningens sammanhang. Detta håller vi inte med om, eftersom även t.ex. de som bygger om och bygger ut system kan ses som någon form av användare. Vi har funnit att för dem är inte byggkvalitet något objektivt eller värderingsfritt. Även projektens modeller och begreppet stil var besvärliga precis som kvalitetsbegrepp och kvalitetsdefinitioner är det. Modellerna uttrycker inte den praktiska kunskap som krävts för att utforma dem och ger därför dålig vägledning om man inte själv har rätt kunskap för att tolka dem.

Arbetet med stilarter blev besvärligt i och med att detta begrepp har många betydelser. Projektgruppen hade svårt att finna ett bra sätt att ordna en stilapparat, eftersom det var svårt att veta om ett visst system eller program gav upphov till en stil eller om det var resultatet av en befintlig stil – en hönan-eller-ägget situation. Att stilarna dessutom skulle vara allmängiltiga, samtidigt som modellerna betonade sammanhanget som systemen och programmen användes i, gjorde det än mer besvärligt.

Vi har i denna avhandling haft liknande utgångspunkter som de ovan diskuterade projektens. Men vi har istället för att konstatera att systemutvecklare har bristande kunskap om och förmåga att bedöma kvalitet, ställt oss frågan hur deras kunskap och kunnande ser ut. Det vi kommer fram till är att kvalitet långt ifrån är någon värderingsfri egenskap och att det i och för sig går att definiera kvalitet, men att definitionerna endast kan användas för att rikta uppmärksamheten. Den kunskap som krävs för att åstadkomma en definition finns inte med i själva definitionen och därför kan en definition inte ersätta kunskap och kunnande om kvalitet. Innebörden i kvalitet är också beroende av ett sammanhang av arbetsuppgifter, omvärldens utseende och teknikens möjligheter. Dessa faktorer förändras ständigt och därför förändras också kvalitetsbegreppets innebörd kontinuerligt.

Det är alltid så att värdering av kvalitet sker av någon intressent, t.ex. en systemutvecklare, utifrån ett perspektiv. Eftersom det är en fråga om värdering, fungerar inte kriterier som är oberoende av

någons uppfattning eller som inte tar hänsyn till det sammanhang som systemet eller programmet ingår i. Därmed kan olika kvaliteter, eller egenskaper, vara olika viktiga för olika personer och projekt, och kvalitet blir därför alltid en kompromiss mellan intressenter med olika perspektiv.

Att kvalitet fungerar på detta sätt leder till att bedömning av kvalitet till avgörande grad bygger på personlig erfarenhet. Metoder, sätt att mäta, standarder m.fl. reglerande inslag är endast stöd för erfarenheten, inte ersättningar. Det mesta av systemutvecklarens kunskap om kvalitet bygger också på erfarenhet, d.v.s. tyst fungerande färdighet och förtrogenhet, som är den viktigaste delen i bedömningsförmågan. Detta kommer inte fram i teorierna om IT-kvalitet.

Eftersom erfarenhet är så viktigt är det förvånande att systemutvecklare resonerar och talar så lite om kvalitet med varandra, i alla fall med det direkta syftet att öka sin kunskap och förbättra sitt kunnande om kvalitet. Dels beror detta på att de inte får den tid de behöver för att reflektera, dels beror det på att de saknar ett bra direkt och indirekt språk med begrepp, termer, exempel och metaforer. Som framgår av teorin om kunskap och kunnande behöver man använda indirekta medel, såsom exempel och liknelser, för att kunna uttrycka den så kallade tysta kunskapen, den som är svår att fånga i ord. Det framgår också av teorin att kunskapen behöver reflekteras över och uttryckas för att kunna utvecklas.

De slutsatser vi drar om IT-kvalitet i praxis är att begrepp och definitioner kan peka på viktiga aspekter att tänka på, men att kvalitet alltid är någons värdering ur vissa perspektiv. Vi kan alltså inte göra en bra definition av IT-kvalitet, men vi kan definiera fenomenet IT-kvalitet. Det är ett antal begrepp och värden sedda ur intressenters perspektiv och bedömda i det sammanhang som systemet eller programmet utvecklas, underhålls och används i. Det innebär att kvalitetsegenskaper, som delvis kan uttryckas genom begrepp och definitioner, väljs av intressenter för vissa sammanhang och att intressenternas bedömningar av kvaliteten sker i relation till dessa sammanhang och alltid innebär värdering. Begreppen och definitionerna är inte entydiga och uttömmande och måste inte ingå som

element i en bedömning av kvalitet. De värden de representerar varierar över tiden genom att nya tillkommer och att gamla kanske försvinner, beroende på att intressenter och sammanhang utvecklas, förändras och tillkommer. Relationerna mellan aspekterna skall inte tolkas som absoluta och enkla, vilket vi har visat i den här avhandlingen.

## English summary

This work is about IT quality and systems developers' knowledge and view of it. IT quality is considered as the characteristics of a system or program that influence e.g. the experience of using it, the easiness to learn and understand it, how it could be customised according to new demands, and the possibility for those who have developed it to understand how it works and can be enhanced. There are at least three good reasons for studying the concept of IT quality and systems developers' knowledge and view of it. Firstly, everybody has sometimes experienced that computer systems and programs do not work, as we would like them to do. They have frequently faults and peculiarities that can be frustrating and, in worst case, fatal. Secondly, the use of information technology is continuously increasing and it is becoming more and more a part of the ordinary day. Consequently, information technology itself and the quality of information technology affect a lot of people. And thirdly, systems developers' knowledge about and view of IT quality appears to us to be an unexplored issue.

However, there are lots of ways of studying quality. In this thesis we study IT quality as a concept and as a question of knowledge and knowing. Our perspective of knowledge is not only knowledge that can be verbalised and expressed as clear logical statements and rules, but also experience-based knowledge that is expressed in actions in practical situations. This implies both a theoretical and practical focus on systems developers' knowledge and a focus on professional systems developers in practice.

Accordingly, one of the main concepts in this thesis is IT quality, which is presented partly from a historical viewpoint describing different thoughts and points of departure. The views of IT quality that we discuss are slightly different, but they share similar cores and also some of the further developments of the views. In the beginning of the strive for quality improvements, the main focus was

on control. Behind this was a mathematical and logical view and measurement was thus regarded as the best way to ensure quality. This view is still present, but further research has led to other more pragmatic, subjective and social views.

The concept of IT quality is also represented within a few different fields that designate qualities. Two such fields are Software Engineering and Human Computer Interaction. In the early days of Software Engineering, quality was equal to efficiency, i.e. the most efficient use of scarce machine resources, but since then Software Engineering has invented a large number of quality concepts. Concepts like maintainability, flexibility, portability are some that are used to express IT quality. A similar development is recognisable within the field of Human Computer Interaction. It started with a few IT quality concepts which has proliferated to a large number, but with usability as the main focus.

Surprisingly enough and despite all research and all concepts, there seems to be a lack of discussion about the relation between systems developers and IT quality. What we mean is that the discourse on IT quality excludes the systems developers and the systems development practice. It is also for this reason that we in this thesis study systems developers' knowledge in relation to IT quality.

Knowledge and competence are furthermore two important concepts that this thesis is built around. In opposition to the logical positivism, which only accepted knowledge as valid if it empirically or logically corresponded with reality, knowledge is in this thesis expressed as socially dependent acts and experiences. Knowledge can in this sense not reflect an objective and obvious reality. Rather it is a question about inter-subjective interpretations and personal experiences. There is therefore knowledge that cannot be verbalised, so called "tacit" knowledge.

Knowledge is therefore socially dependent and cannot be formalised. The foundation of knowledge rests on experience and communication with others and in that sense it cannot be static. Thus, participating in a social process of developing knowledge demands



the ability to reflect upon one's knowledge, which furthermore demands a language - sometimes verbal and sometimes non-verbal.

With this point of departure in the concept of IT quality, theoretical and practical knowledge and practice we explore systems developers knowledge about IT quality. We have asked questions like; how do systems developers apprehend quality? Which concepts do they use when talking about quality? How do they judge quality? How can they act for better quality? How do they develop their skills?

Our empirical study, comprising 19 in-depth interviews, shows that the concept of IT quality is inherently complex and that it in itself contains a variety of qualities that together constitute the meaning of IT quality. Therefore, a common definition of IT quality as a point of departure for actual use is meaningless.

Moreover, systems developers' judgements of IT quality are often based on values, comparisons, and personal experience of high and low quality. It is rarely a question of regulation by rules such as measurements, calculations and standards. Methods, standards, different ways of measuring etc., i.e. rules for assessment, can support judgements, but are not enough for assuring and assessing quality.

Furthermore, in order to reach high quality it is not enough to conform to the customer's requirements; high quality is to exceed beyond these requirements. Therefore, the ability to reach high quality rests on personal experiences from practice and "fingerspitzgeföhle", i.e. practical knowledge and the ability to judge. Rules and standards can serve as starting points for this ability, but when the situation and experience raise other demands the rules have to be broken.

The ability to judge is mainly developed by the personal experiences from different kinds of projects, customers, users and so on. The feedback from customers and users is one part of this experience used in this development. However, the feedback is often negative, such as reports about faults (bugs) and things that do not work. It is rarely a question about commendation. The fact that

systems developers within projects share experiences is also important for the development of the ability to assess IT quality.

As experience is a key ingredient in judging and creating high quality it seems plausible to believe that a general reflection upon quality, what quality is and what high and low quality could be, would take place. This is surprisingly not the case. This kind of reflection is quite unusual since firstly, the systems developers are given no time for these kinds of activities, secondly, because they are focused on problems and problem solving, and, thirdly, because they in general are unaware that this could be important.

However, they talk about qualities within the systems development projects and they consider this type of conversation as important. But the purpose is not very often to enhance the knowledge and competence, it is more related to the current project and the problems within it.

The conclusion we draw about IT quality in practice is that concepts and definitions can pinpoint important aspects, but that quality always is somebody's evaluation from a perspective. In that sense quality cannot be defined with the purpose of operational use. However it is possible to define quality as a phenomenon, which is constituted by concepts, values, perspectives and contexts. This means that interested parties choose qualities, that can in part be expressed by concepts and definitions, in context and that these parties' judgements of quality always are carried out in context and always are valuations. The relation between these four aspects should however not be regarded as absolute and simple. The concepts and definitions are ambiguous and inexhaustible and do not have to be an integral part of judgements of quality. Moreover, the values that these represent change over time. Some disappear and new one is created as the context evolves and changes and as interested parties change.



## **IT kvalitet i praxis - En empirisk undersökning**

**Theis Meggerle, [theis.meggerle@ics.lu.se](mailto:theis.meggerle@ics.lu.se)**

**Odd Steen, [odd.steen@ics.lu.se](mailto:odd.steen@ics.lu.se)**

LUND UNIVERSITET

Institutionen för Informatik

Ole Römers väg 6

223 64 Lunds

046 – 2223376

### **Bakgrund**

Kritiska röster har riktats mot de produkter som mjukvaruindustrin tillverkar. Kvaliteten är inte vad den borde vara, utan snarare upplevs det som svårt och problematiskt att använda IT-system.

Det finns anledning att tro att en del av denna problematik kan relateras till den förmåga som professionella systemutvecklare har för att bedöma kvalitet och att arbeta med de betydelser som begreppet kvalitet kan ges.

Mycket av det arbete som gjorts när det gäller IT-kvalitet<sup>135</sup>, har strävat i en riktning att operationalisera kvalitetsbegreppet och därigenom göra det

---

<sup>135</sup> Denna utsaga tar sin utgångspunkt i de teorier som beskriver olika sätt för att uppnå

möjligt att styra arbetet för att uppnå en hög kvalitet, t. ex.. genom standardisering och metodanvändning. Trots detta verkar det som kvaliteten, särskilt brukskvaliteten, fortfarande lämnar en del övrigt att önska.

## Undersökning

Mot denna bakgrund tror vi att det finns viktiga insikter att nå genom att studera hur begreppet kvalitet uppfattas och behandlas i den praktiska situationen att utveckla IT-system och i den situation där det används. Undersökningens syfte kan därför formuleras som: *vilken förståelse finns för kvalitet i praxis?* I praktiken innebär detta två studier. En studie av systemutvecklare och en av brukare. Studierna har samma syfte, men kommer med utgångspunkt från våra intresse att utföras med olika tonvikter. Skillnaden kommer bland annat till uttryck genom den tid vi ägnar åt respektive studie och antalet respondenter vi önskar att intervjua.

### En studie av systemutvecklarens och brukares bedömningsförmåga

Förståelsen för kvalitet framstår som nära relaterad till den bedömningsförmåga som systemutvecklare innehar. På förhand kan man spekulativt påstå att den ingår som en viktig del av systemutvecklarens professionalitet. Således avser undersökningen såväl att undersöka den del av systemutvecklarens bedömningsförmåga som är relaterad till kvalitet som den existerande förståelsen för kvalitet.

Detta medför att undersökningen försöker belysa vilka delar som ingår i systemutvecklarens bedömningsförmåga och hur denna formas. Mer specifikt innebär detta att det är av intresse att fånga vilka kvalitetsbegrepp som används, innebörden av dessa begrepp, vilka värderingar, normer och regler som ligger till grund för dessa, vilka situationer som dessa används i och hur dessa formas (t. ex. genom standarder, företagskultur, utbildning, socialt genom deltagande i projekt, exempel, etc.).

De tilltänkta intervjuerna är till sin karaktär närmast ostrukturerade, med undantag för ett antal punkter som stöd. En anledning till detta är att vi önskar en rejäl fördjupning i de aspekter som vi avser att undersöka. Således kommer intervjuerna innehålla ett fåtal frågor där respondenterna har möjlighet till längre reflektion än vad som framstår som hanterbart när det

---

hög kvalitet på IT-system.

är många frågor. Intervjuerna blir kanske därför mer ett samtal än en traditionell intervju.

### **Studiens genomförande**

I studien önskar vi att preliminärt använda två tekniker för att uppnå undersökningens syfte, nämligen intervju och observation. Syftet med att använda dessa är att försöka genom observation fånga vilka kvalitetsbegrepp som används i processen för att senare skapa en bredare och djupare reflektion intervju. Detta bygger på tanken att det är inte alltid så lätt att formulera och reflektera över begrepp utanför och isolerat från den situationen där det används. Genom möjligheten att referera till den situation där det används är det vår tro att det blir lättare för respondenter att reflektera över på vilket sätt det för tillfället faktiskt användes.

Studien kommer till en början att delas upp i tre omgångar. Den första är studien av brukarens bedömningsförmåga och förståelse för kvalitet. Anledningen till att vi vill göra denna studie först är att det möjligen skapas ett bättre underlag till studien av systemutvecklarna som ingår i den andra omgången. Den tredje och sista omgången är eventuell till sin karaktär med tanke på om vi skulle behöva komplettera vissa uppgifter. Således ser hela studien ut som följande:

Studie 1: En studie av brukarnas bedömningsförmåga och förståelse för kvalitet.

Förfaringsätt: Intervju

Studie 2: En studie av systemutvecklarens bedömningsförmåga och förståelse för kvalitet.

Förfaringsätt: Intervju och observation

Studie 3: Eventuellt en uppföljning.

Förfaringsätt: Intervju

Vilken tid studierna kommer att ta i anspråk är svårt på förhand att precisera. Men uppskattningsvis tar respektive intervju två timmar.

Vi vill poängtera att denna skissering är ett förslag och är öppen för förändring. Dock är någon form av intervju med systemutvecklare nödvändig i vårt arbete.

## Frågemall för empirisk undersökning

*Syftet med intervjun är att försöka ”tränga in i” systemutvecklarens bedömningsförmåga och kvalitetssyner. Vilka begrepp för olika kvaliteter på datasystem och dataprogram är det systemutvecklare använder? När används de? Vad betyder de? Vad är det som gör att ett program eller system har låg eller hög kvalitet?*

*Vi vill också försöka nå en bättre kunskap och förståelse för hur systemutvecklare talar om kvalitet med varandra; vilket ”språk” har de för kvalitetsaspekter? Strävar systemutvecklare efter att ha ett gemensamt språk och en gemensam syn på kvalitet när de arbetar tillsammans? Hur, i så fall, förmedlar systemutvecklare kvalitetssyner sinsemellan? På vilka sätt formas en bedömningsförmåga? Sker det genom kurser, erfarenhet, reflektion och diskussion med andra systemutvecklare, eller på andra sätt? Hur berättar en systemutvecklare för en annan om ett bra system och vilka begrepp och ord används då?*

*Vi har valt att inte ställa upp en mängd frågor, utan vill hellre att intervjun skall ta formen av ett samtal, som täcker in de olika tematiska frågor vi har angivit. Vi tror att vi härigenom har större möjlighet att nå längre än vad vår egen förståelse för kvalitetsaspekter och bedömningsförmåga sträcker sig.*

---

### Del 1: Uppgifter om respondent och intervju

1. Intervju (band): (För vår egen del)
2. Datum: (För vår egen del)
3. Företag: (För vår egen del)
4. Namn: (För vår egen del)

---

### Del 2: Allmänna frågor om respondent

- Vilken utbildning har du?

- Antal års erfarenhet av området?
  - Vilka områden har du erfarenhet ifrån? (t.ex. programmering, systemering, projektledning, etc.)
  - Fortbildning? Vilka?
- 

### **Del 3: Teman om bedömningsförmåga och kvalitet**

#### 1. Definition av kvalitet

- hur definieras kvalitet?
- vad innehåller definitionen?
- vad grundar sig definitionen på?
- hur använder man en definition av kvalitet?
- går det att definiera kvalitet?
- hur tillämpar ni detta och hur kontrolleras det?

#### 2. Karakteristika för hög respektive låg kvalitet

- vad är låg/hög kvalitet på ett program?
- på ett system?
- hur avgör man om kvaliteten är hög eller låg?
- vilka kvaliteter/egenskaper är viktigast för att avgöra kvaliteten?
- vem avgör om kvaliteten är låg eller hög?

#### 3. Graden av kvalitets betydelse?

- hur viktigt är kvalitet?
- får man ge avkall på vissa kvaliteter till förmån för andra?
- vad händer om kvaliteten upplevs som för låg?
- vad händer om kvaliteten upplevs som hög?

- några exempel?

#### 4. Bedömningsförmåga

- ger metoder och mätning tillräcklig uppfattning om kvalitet?
- eller krävs det också ”tyst kunskap”?
- vad ingår i en bedömningsförmåga?

#### 5. Utveckling och formandet av bedömningsförmåga

- hur utvecklas bedömningsförmågan?; kurser, erfarenhet, etc.
- påverkas bedömningsförmågan av samarbete och utbyte med andra?
- vilka yttre faktorer spelar in?; ex. kvalitetsteorier
- påverkar teknikutvecklingen bedömningsförmågan?

#### 6. Förmedling av bedömning

- hur talar man med andra praktiker om kvalitet?
- vilka begrepp och ord används för att förmedla kvalitetsuppfattningar?
- hur förmedlas t.ex. en bild av ett bra resp. dåligt program eller system?
- finns det ett ”språk” för att tala om kvaliteter och uppfattningar?



# Referenser

- Abbott, L. (1955): *Quality and competition: an essay in economic theory*. Columbia University Press, New York. (enl. Garvin, D. A., 1988)
- Adelakun, O. (1998): A Process Model for Achieving Information Systems Quality. Ur Buch, N. J., Damsgaard, J., Eriksen, L. B., Iversen, J. H. & Nielsen, P. A., red. (1998): *IRIS 21 - Information Systems Research in Collaboration with Industry*.
- Alvesson, M. (1995): The Meaning and Meaninglessness of Postmodernism: Some Ironic Remarks. *Organization Studies*, vol. 16, nr. 6, ss. 1047-1075.
- Alvesson, M. & Sköldberg, K. (1994): *Tolkning och reflektion : vetenskapsfilosofi och kvalitativ metod*. Studentlitteratur, Lund.
- Andersen, E. S. (1991): *Systemutveckling principer, metoder och tekniker*. Studentlitteratur, Lund.
- Andersen, R., Johnson, D., Gotterbarn, D. & Perolle, J. (1993): Using The New ACM Code of Ethics in Decision Making. *Communications of the ACM*, vol. 36, nr. 2, ss. 98-107.
- Anonymous, red.(1983): *Om stilforskning : föredrag och diskussionsinlägg vid Vitterhetsakademiens symposium 16-18 november 1982*. Almquist & Wiksell Int., Stockholm.
- Baber, R. L. (1997): Comparison of Electrical "Engineering" of Heaviside's Times and Software "Engineering" of Our Times. *IEEE Annals of the History of Computing*, vol. 19, nr. 4.
- Bannon, L. (1991): From human factors to human actors: the role of psychology and human-computer interaction studies in systems design. Ur Greenbaum, J. & Kyng, M., red. (1991): *Design at work: cooperative design of computer systems*.
- Bansler, J. (1989): Systems Development Research in Scandinavia: Three Theoretical Schools. *Scandinavian Journal of Information Systems*, vol. 1, ss. 3-20.
- Bass, L., Clements, P. & Kazman, R. (1998): *Software Architecture in Practice*. Addison-Wesley, Reading, Mass.
- Becker Jensen, L. (1987): *Ud af elfenbenstårnet - fortælleteknik for fagfolk der vil skrive en god historie*. Teknisk Forlag, København.
- Bjerknes, G., Bratteteig, T. & Kautz, K., red.(1992): *Preceedings of the 15th IRIS : Information Systems Research Seminar in Scandinavia, August 1992, Larkollen, Norway*. Universitetet, Oslo.

- Blackler, F. (1995): Knowledge, Knowledge Work and Organizations: An Overview and Interpretation. *Organization Studies*, vol. 16, nr. 6, ss. 1021-1046.
- Blomberg, J., Kensing, F. & Dykstra-Erickson, E., red.(1996): *Proceedings of PDC '96*. Computer Professionals for Social Responsibility, Palo Alto.
- Blonsky, M. (1991): The New Aesthetic Order. *Connoisseur*. September. (enl. Ylimaula, A-M., 1992)
- Brunsson, N. & Jacobsson, B. (1998): *Standarisering*. Nerenius & Santérus förlag, Stockholm.
- Buch, N. J., Damsgaard, J., Eriksen, L. B., Iversen, J. H. & Nielsen, P. A., red.(1998): *IRIS 21 : information systems research in collaboration with industry : proceedings of the 21th Information Systems Research Seminar in Scandinavia, 8-11 August 1998 at Sæby Søbad, Denmark*. Department of Computer Science, Aalborg University, Aalborg.
- Budgen, D. (1994): *Software design*. Addison-Wesley, Wokingham.
- Buffon, G. L. d. (1986): Discours sur le style, ed. Henri Guyot, Classiques pour tous, Libraire Hatier, Paris 1920, Finnish translation Anto Leikola. *Parnasso* nr. 4, ss. 193-194. (enl. Ylimaula, A-M., 1992)
- Burr, A. & Owen, M. (1996): *Statistical methods for software quality : using metrics to control process and product quality*. International Thomson Computer Press, London.
- Bødker, S., Grønbaek, K. & Kyng, M. (1993): Cooperative Design: Techniques and Experiences From the Scandinavian Scene. Ur Schuler, D. & Namioka, A., red. (1993): *Participatory design : principles and practices*.
- Capron, H. L. (1986): *Systems analysis and design*. Benjamin/Cummings Publ. cop., Menlo Park, Calif.
- Card, S. K., Moran, T. P. & Newell, A. (1990): The Keystroke-Level Model for User Performance Time with Interactive Systems. Ur Preece, J. & Keller, L., red. (1990): *Human-Computer Interaction*.
- Carroll, J. M., red.(1987): *Interfacing thought : cognitive aspects of human-computer interaction*. MIT Press, Cambridge, Mass.
- Clausen, H. (1993): Narration as a Tool for the Systems Designer. *Design Studies*, vol. 14, nr. 3.
- Clausen, H. (1994): Designing Computer Systems from a Human Perspective – The Use of Narratives. *Scandinavian Journal of Information Systems*, vol. 6, nr. 2, ss. 43-57.
- Crosby, P. B. (1979): *Quality is Free : the Art of Making Quality Certain*. McGraw-Hill, New York.

- Dahlbom, B., Kämmerer, F., Ljungberg, F., Stage, J. & Sørensen, C., red.(1995): *Design in context : proceedings of the 18th Information Systems Research Seminar in Scandinavia, IRIS 18, 11-13 August, 1995 in Gjern, Denmark.* Göteborg Department of Informatics, Göteborg University.
- Dahlbom, B. & Mathiassen, L. (1993): *Computers in Context – The Philosophy and Practice of Systems Design.* NCC Blackwell, Cambridge.
- DeMarco, T. (1979): *Structured analysis and system specification.* Prentice-Hall, Englewood Cliffs, N.J.
- Denzin, N. K. & Lincoln, Y. S., red.(1994): *Handbook of qualitative research.* Sage Publications, Thousand Oaks.
- Dijkstra, E. W. (1979a): Go To statement considered harmful. Ur Yourdon, E., red. (1979): *Classics in software engineering.*
- Dijkstra, E. W. (1979b): The humble programmer. Ur Yourdon, E., red. (1979): *Classics in software engineering.*
- Dijkstra, E. W. (1979c): Programming considered as a human activity. Ur Yourdon, E., red. (1979): *Classics in software engineering.*
- Dix, A., Finlay, J., Abowd, G. & Beale, R. (1993): *Human-Computer Interaction.* Prentice-Hall, London.
- Drucker, P. F. (1993): *Post capitalist society.* HarperBusiness, New York.
- Dubé, L. & Robey, D. (1999): Software stories: three cultural perspectives on the organizational practices of software development. *Accounting Management and Information Technologies*, vol. 9, ss. 223-259.
- Ehn, P. (1988): *Work-oriented design of computer artifacts.* Almqvist & Wiksell International, Stockholm.
- Ehn, P. (1993): Scandinavian Design: On Participation and Skill. Ur Schuler, D. & Namioka, A., red. (1993): *Participatory design : principles and practices.*
- Ehn, P. (1994): *Informationsteknologisk designkvalitet – ett projekt om paradigmatiske form och funktion.*, Department of Informatics, Lund University, Lund.
- Ehn, P., Gustafsson, N., Meggerle, T., Steen, O. & Svedemar, M. (1996a): *The Qualitheque – communicating quality.*, Informatik, Lund.
- Ehn, P. & Löwgren, J. (1996): The Qualitheque systems at an exhibition. *interactions*, vol. 3, nr. 3, ss. 53-55.
- Ehn, P., Meggerle, T., Steen, O. & Svedemar, M. (1995): What Kind of Car is this Sales Support System? On Styles, Artifacts and Quality-in-use. Ur Kyng, M. & Mathiassen, L., red. (1995): *Computers in Context – Joining Forces in Design.*

- Ehn, P., Meggerle, T., Steen, O. & Svedemar, M. (1996b): Communicating Quality. Ur Blomberg, J., Kensing, F. & Dykstra-Erickson, E., red. (1996): *Proceedings of PDC '96*.
- Ehn, P., Meggerle, T., Steen, O. & Svedemar, M. (1997a): What Kind of Car is this Sales Support System? On Styles, Artifacts and Quality-in-use. Ur Kyng, M. & Mathiassen, L., red. (1997): *Computers and Design in Context*.
- Ehn, P., Meggerle, T., Steen, O. & Svedemar, M. (1997b): Workshop: Communicating Quality. *Proceedings of the 2nd. European Academy of Design Conference*, Stockholm.
- Ellis, C. & Flaherty, M. C., red.(1992): *Investigating subjectivity : research on lived experience*. Sage Publications, Newbury Park.
- Enerstvedt, R. T. (1989): The Problem Of Validity In Social Science. Ur Kvale, S., red. (1989): *Issues of Validity in Qualitative Research*.
- Enkvist, N. E. (1973): *Stilforskning och stilteori*. Gleerup, Lund.
- Eriksson, E. (1995): *Ett informationssystemets stil – en studie av IKEAs MöbelHusSystem med fokus på kvalitetsaspekterna form och funktion*. Magisteruppsats, Institutionen för informatik, Lunds universitet, Lund.
- Eriksson, I. & Törn, A. (1997): Introduction to IST issue on Information Systems Quality. *Information and Software Technology*, vol. 39, nr. 12.
- Feigenbaum, A. V. (1961): *Total quality control: engineering and management: the technical and managerial field for improving product quality, including its reliability, and for reducing operating costs and losses*. McGraw-Hill, New York,.
- Friis, S. (1991): Dependencies in Research on User Controlled Information Systems Development. Ur Nurminen, M. I. & Weir, G. R. S., red. (1991): *Human jobs and computer interfaces : proceedings of the IFIP WG 9.1 Working Conference on Human Jobs and Computer Interfaces, Tampere, Finland, 26-28 June, 1991*.
- Ganjabadi, S. & Henriksson, A. (1994): *Informationsteknologisk Designkvalitet – firmitas-commoditas-venustas*. Magisteruppsats, Institutionen för informatik, Lunds universitet, Lund.
- Garvin, D. A. (1988): *Managing quality : the strategic and competitive edge*. Free Press, New York.
- Gelernter, D. H. (1997): *Machine beauty : elegance and the heart of technology*. 1st uppl., Basic Books, New York.
- Ginzburg, M. I. (1982): *Style and epoch*. Published for the Graham Foundation for Advanced Studies in the Fine Arts and the Institute for Architecture and Urban Studies by MIT Press, Cambridge, Mass.

- Giorgi, A. (1994): What kind of knowledge does qualitative research yield? Ur Ugglå, H., red. (1994): *Varför kvalitativ forskning är nödvändig – Kunskap, tillvägagångssätt, nytta. Seminarium 25-26 augusti 1994.*
- Gray, B. (1969): *Style – The Problem and Its Solution.* Mouton, Paris.
- Greenbaum, J. & Kyng, M., red.(1991): *Design at work: cooperative design of computer systems.* Lawrence Erlbaum, Hillsdale, N. J.
- Grenander Nyberg, G. (1993): *Stilhistoria.* Medborgarskolan, Stockholm.
- Habermas, J. (1985): *Theorie des Kommunikativen Handelns.* Suhrkamp, Frankfurt. (enl. Ehn, P. et al., 1997a)
- Habermas, J. r. (1968a): *Erkenntnis und Interesse.* Suhrkamp, Frankfurt am Main. (enl. Ehn, P. et al., 1997a)
- Habermas, J. r. (1968b): *Technik und Wissenschaft als "Ideologie".* Suhrkamp, Frankfurt a.M. (enl. Ehn, P. et al., 1997a)
- Halstead, M. H. (1977): *Elements of software science.* Elsevier, New York. (enl. Shapiro, S., 1997)
- Hermerén, G. (1982): Stilbegreppens logik. Ur Anonymous, red. (1983): *Om stilforskning : föredrag och diskussionsinlägg vid Vitterhetsakademiens symposium 16-18 november 1982.*
- Hoberg, C. (1998): *Precision och improvisation – om systemutvecklares yrkeskunnande.* Dialoger, Stockholm.
- Holmqvist, B. (1994): Om relativism. Ur Olausson, L., red. (1994): *Idéhistoriens egenart : teori- och metodproblem inom idéhistorien.*
- Ince, D. (1995): *Software Quality Assurance – a student introduction.* McGraw-Hill Book Company, London.
- Ingman, S. (1997): *Förtroende och datorbruk.* Doktorsavhandling, Institutionen för Informatik, Lunds universitet, Lund.
- Johannessen, K. S. (1999): *Praxis och tyst kunskande.* Dialoger cop., Stockholm.
- Johannessen, K. S. & Rolf, B. (1990): *Om tyst kunskap: två artiklar.* 2. uppl., Uppsala Universitet, centrum för didaktik, Uppsala.
- Jones, J. C. (1981): *Design methods : seeds of human futures.* 1980 with a review of new topics. uppl., J. Wiley, New York.
- Juran, J. M. (1974): *Quality Control Handbook.* 3. uppl., McGraw-hill, New York. (enl Garvin, D. A., 1988)
- Kan, S. H. (1995): *Metrics and Models in Software Quality Engineering.* Addison-Wesley, Reading.

- Kapor, M. (1996): A Software Design Manifesto. Ur Winograd, T., Bennet, J., De Young, L. & Hartfield, B., red. (1996): *Bringing Design to Software*.
- Knuth, D. (1979): Structured programming with go to statements. Ur Yourdon, E., red. (1979): *Classics in software engineering*.
- Kuhn, T. S. (1970): *The structure of scientific revolutions*. 2. uppl., University of Chicago Press, Chicago.
- Kvale, S., red.(1989a): *Issues of Validity in Qualitative Research*. Studentlitteratur, Lund.
- Kvale, S. (1989b): To Validate Is to Question. Ur Kvale, S., red. (1989): *Issues of Validity in Qualitative Research*.
- Kvale, S. (1997): *Den kvalitativa forskningsintervjun*. Studentlitteratur, Lund.
- Kyng, M. & Mathiassen, L., red.(1995): *Computers in context : joining forces in design : proceedings*. Department of Computer Science, Aarhus University, Aarhus.
- Kyng, M. & Mathiassen, L., red.(1997): *Computers and Design in Context*. MIT Press, Cambridge.
- Langefors, B. (1979): Behovet och framväxten av universitetsämnet Administrativ Databehandling. *SSIaren* nr. 2.
- Lawson, B. (1990): *How Designers Think - The Design Procesee Demystified*. Butterworth-Architecture, Oxford.
- Lindgren, A. (1914): *Rakennus – ja koristetyylit*.Helsinki. (enl. Ylimaula, A-M., 1992)
- Lundequist, J. (1992): *Designeteorins kunskapsteoretiska och estetiska utgångspunkter*. Projekteringsmetodik, KTH, Stockholm.
- Lundequist, J. (1995): *Design och produktutveckling - Metoder och begrepp*. Studentlitteratur, Lund.
- Lunell, H. (1985): *Datalogi – en inledande översikt*. 2. uppl., Studentlitteratur, Lund.
- Löwgren, J. (1995): *Perspectives on Usability*. Lith-IDA-R-95-23, Department of Computer and Information Science, Linköping University, Linköping.
- Magee, B. (1999): *En filosofers bekännelser*. Wahlström & Widstrand, Stockholm.
- Meggerle, T. & Steen, O. (1994): *Ett säljstöd i bruk – en studie utifrån kvalitetsbegreppen estetik och etik*. Magisteruppsats, Institutionen för Informatik, Lunds universitet, Lund.

- Meggerle, T. & Steen, O. (1995): A Sales Support System in Use – The Volvo Case. Ur Dahlbom, B., Kämmerer, F., Ljungberg, F., Stage, J. & Sørensen, C., red. (1995): *IRIS 18 – Design in Context*.
- Meggerle, T. & Steen, O. (1997): Är estetiken och kulturen viktig för gränssnittsdesign? *designjournalen*, vol. 4, nr. 2, ss. 19-32.
- Meggerle, T. & Steen, O. (1998): A Critical Evaluation of a Project. Ur Buch, N. J., Damsgaard, J., Eriksen, L. B., Iversen, J. H. & Nielsen, P. A., red. (1998): *IRIS 21 – Information Systems Research in Collaboration with Industry*.
- Messeter, J. (2000): *Operatörens blick : om inplacering av IT-stöd i erfarenhetsöverföring inom en lokal praxis*. Institutionen för informatik Lunds universitet, Lund.
- Miles, M. B. & Huberman, A. M. (1994): *Qualitative data analysis : an expanded sourcebook*. 2. uppl., Sage Publications, Thousand Oaks.
- Mills, H. D. (1975): The New Math of Computer Programming. *Communications of the ACM*, vol. 18, s. 44.
- Molander, B. (1996): *Kunskap i handling*. 2. uppl., Daidalos, Göteborg.
- Mullet, K. & Sano, D. (1995): *Designing visual interfaces : communication oriented techniques*. SunSoft Press, Englewood Cliffs, NJ.
- Norberg-Schultz, C. (1966): *Intentions in Architecture*. 2. uppl., Universitetsforlag, Oslo.
- Nordenstam, T. (1983): Ett pragmatiskt perspektiv på datautvecklingen. Ur Göranson, B., red. (1983): *Datautvecklingens filosofi*.
- Norman, D. A. (1990): *The design of everyday things*. 1. uppl., Doubleday, New York.
- Nurminen, M. I. (1988): *People or computers : three ways of looking at information systems*. Studentlitteratur, Lund.
- Nurminen, M. I. & Weir, G. R. S., red.(1991): *Human jobs and computer interfaces : proceedings of the IFIP WG 9.1 Working Conference on Human Jobs and Computer Interfaces, Tampere, Finland, 26-28 June, 1991*. North-Holland, Amsterdam.
- Olausson, L., red.(1994): *Idéhistoriens egenart : teori- och metodproblem inom idéhistorien*. Symposium, Stockholm.
- Olsson, B. (1982): Stil och stiltypologier – presentation av några problem och ett forskningsprojekt. Ur Anonymous, red. (1983): *Om stilforskning : föredrag och diskussionsinlägg vid Vitterhetsakademiens symposium 16-18 november 1982*.
- Oskarsson, Ö. & von Schantz, C. (1987): *Kvalitetssäkring av programvara*. 2. uppl., Mekanförbundets förlag, Stockholm.
- Page-Jones, M. (1988): *The practical guide to structured systems design*. 2. uppl., Prentice Hall, Englewood Cliffs, N.J.

- Parnas, D. L. (1979): On the criteria to be used in decomposing systems into modules. Ur Yourdon, E., red. (1979): *Classics in software engineering*.
- Patel, R. & Davidson, B. (1994): *Forskningsmetodikens grunder – Att planera, genomföra och rapportera en undersökning*. 2. uppl., Studentlitteratur, Lund.
- Paulk, M. C. (1994): *The Capability maturity model : guidelines for improving the software process*. Addison-Wesley Pub. Co., Reading, Mass.
- Peshkin, A. & Eisner, E. W., red.(1990): *Qualitative inquiry in education : the continuing debate*. Teachers College Columbia University, New York.
- Pirsig, R. M. (1974): *Zen and the Art of Motorcycle Maintenance : an Inquiry into Values*. Morrow, New York.
- Polanyi, M. (1958): *Personal knowledge*. Routledge and Kegan Paul, London. (enl. Rolf, B., 1995)
- Polanyi, M. (1966): *The tacit dimension*. 1. uppl., Doubleday, Garden City, N.Y.
- Preece, J. & Keller, L. S., red.(1990): *Human-Computer Interaction : Selected Readings : a Reader*. Prentice Hall, Hemel Hempstead.
- Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. & Carey, T. (1994): *Human-computer interaction*. Addison-Wesley, Wokingham.
- Pressman, R. S. (1987): *Software engineering : a practitioner's approach*. 2. uppl., McGraw-Hill, New York.
- Richardson, L. (1992): The consequences of poetic representation – Writing the other, rewriting the self. Ur Ellis, C. & Flaherty, M. C., red. (1992): *Investigating subjectivity : research on lived experience*. (enl. Miles, M. B. & Huberman, A. M., 1994)
- Riegel, A. (1923): *Stilfragen: Grundlegen zu einer Geschichte der Ornamentik*. 2. uppl., Berlin. (enl. Ylimaula, A-M., 1992)
- Rolf, B. (1995): *Profession, tradition och tyst kunskap*. 2. uppl., Nya Doxa, Nora.
- Ryle, G. (1949): *The concept of mind*. Hutchinson, London. (enl. Rolf, B., 1995)
- Salner, M. (1989): Validity in Human Science Research. Ur Kvale, S., red. (1989): *Issues of Validity in Qualitative Research*.
- Sanders, J. & Curran, E. (1994): *Software quality : a framework for success in software development and support*. Addison-Wesley Pub. Co., Reading, Mass.
- Saussure, F. d. (1916): *Cours de Linguistique Generale*. 2. uppl., Lausanne-Paris. (enl. Ylimaula, A-M., 1992)
- Schofield, J. W. (1990): Increasing the Generalizability of Qualitative Research. Ur Eisner, I. E. W. & Peshkin, A., red. (1990): *Qualitative Inquiry in Education*.(enl. Miles, M. B. & Huberman, A. M., 1994)



- Schuler, D. & Namioka, A., red.(1993): *Participatory design : principles and practices*. L. Erlbaum Associates, Hillsdale, N.J.
- Schön, D. A. (1983): *The reflective practitioner : how professionals think in action*. Basic Books, New York.
- Schön, D. A. (1987): *Educating the Reflective Practitioner*. Jossey-Bass Inc., San Francisco.
- Scruton, R. (1979): *The Aesthetics of Architecture*. Princeton University Press, Princeton.
- Semper, G. (1989): *The Four Elements of Architecture and other writings 1851-1869, English translation H. F. Mallgrave & Wolfgang Herrman*. Cambridge University Press, Cambridge. (enl. Zeitler, R., 1983)
- Sewell, M. T. & Sewell, L. M. (2002): *The software architect's profession : an introduction*. Prentice Hall, Upper Saddle River, NJ.
- Shapiro, S. (1997): Splitting the Difference: The Historical Necessity of Synthesis in Software Engineering. *IEEE Annals of the History of Computing*, vol. 19, nr. 1, ss. 20-54.
- Sommerville, I. (1996): *Software Engineering*. 5. uppl., Addison-Wesley, Wokingham.
- Stake, R. E. (1994): Case Studies. Ur Denzin, N. K. & Lincoln, Y. S., red. (1994): *Handbook of Qualitative Research*. (enl. Miles, M. B. & Huberman, A. M., 1994)
- Stevens, W., Myers, G. & Constantine, L. (1979): Structured design. Ur Yourdon, E., red. (1979): *Classics in software engineering*.
- Stolterman, E. (1991a): *Designarbetets dolda Rationalitet – en studie av metodik och praktik inom systemutveckling*. Doktorsavhandling, Institutionen för Informationsbehandling, Umeå Universitet, Umeå.
- Stolterman, E. (1991b): *Utskrifter av 20 intervjuer med systemutvecklare : delredovisning av projektet "Konstruktiv systemutveckling"*. Institutionen för Informationsbehandling, Umeå Universitet, Umeå.
- Strauss, A. L. & Corbin, J. M. (1990): *Basics of qualitative research : grounded theory procedures and techniques*. Sage Publications, Newbury Park, Calif.
- Svedemar, M. (1995a): Is The Ikea Furniture Store System Art? On quality-in-use and how to describe a system in use. Ur Dahlbom, B., Kämmerer, F., Ljungberg, F., Stage, J. & Sørensen, C., red. (1995): *IRIS 18 – Design in Context*.
- Svedemar, M. (1995b): *MHS, ett möbelhussystem i bruk – Sysstmet som stödjer IKEAs medarbetare till sin fulla potential?*, Department of Informatics, Lund University, Lund.

Tschudi, F. (1989): Do Qualitative and Quantitative Methods Require Different Approaches to Validity? Ur Kvale, S., red. (1989): *Issues of Validity in Qualitative Research*.

Uggla, H., red.(1994): *Varför kvalitativ forskning är nödvändig – Kunskap, tillvägagångssätt, nytta. Seminarium 25-26 augusti 1994*. Forum för Humanvetenskaplig Forskning, Örebro läns landsting/ Högskolan Örebro, Örebro.

Vasari, G. (1912-1914): *Lives of the most eminent painters, sculptors and architects. Le vite de più eccellenti pittori scultori ede architettori, Firenze 1568, English translation Gaston du C. de Vere*. London. (enl. Ylimaula, A-M., 1992)

Wallén, G. (1996): *Vetenskapsteori och forskningsmetodik*. Studentlitteratur, Lund.

Wenger, E. (1998): *Communities of practice : learning, meaning, and identity*. Cambridge University Press, Cambridge, U.K.

Whiteside, J. & Wixon, D. (1987): Discussion: improving human-computer interaction – a quest for cognitive science. Ur Carroll, J. M., red. (1987): *Interfacing thought : cognitive aspects of human-computer interaction*. (enl. Löwgren J., 1995)

Wilson, D. N. & Hall, T. (1998): Perceptions of software quality: a pilot study. *Software Quality Journal* nr. 7, ss. 67-75.

Winckelmann, J. J. (1974): *Geschichte der Kunst des Alterthums*. Dresden. (enl. Ylimaula, A-M., 1992)

Winograd, T., Bennet, J., De Young, L. & Hartfield, B., red.(1996): *Bringing Design to Software*. Addison-Wesley Publishing Company, New York.

Wittgenstein, L. (1958): *Preliminary studies for the "Philosophical investigations" generally known as The blue and brown books*. Blackwell, Oxford.

Wittgenstein, L. (1961): *Tractatus logico-philosophicus : the German text of Ludwig Wittgenstein's Logisch-philosophische Abhandlung*. Routledge & Kegan Paul, London. (enl. Johannessen, K. S., 1999)

Wittgenstein, L. (1967): *Philosophische Untersuchungen*. 2. uppl., B. Blackwell, Oxford. (enl. Johannessen, K. S., 1999)

Wölfflin, H. (1946): *Prolegomena zur einer Psychologie der Arcitektur, Kleine Schriften (originally 1886)*. Benno Schwabe, Basel. (enl. Ylimaula, A-M., 1992)

Yin, R. K. (1994): *Case Study Research – Design and Methods*. 2. uppl., SAGE Publications, Thousands Oaks.

Ylimaula, A.-M. (1992): *Origins of Style –phenomenological approach to the essence of style in the architecture of Antoni Gaudi, C. R. Mackintosh and Otto Wagner*. Dissertation, Acta Universitatis Ouluensis series C, Department of Architecture, Oulu University, Oulu.

Yourdon, E., red.(1979): *Classics in software engineering*. Yourdon Press, New York.

Zeitler, R. (1982): Om stilforskningen i konsthistorien – Särskilt om utvecklingstankens inflytande på stilteorierna. Ur Anonymous, red. (1983): *Om stilforskning : föredrag och diskussionsinlägg vid Vitterhetsakademiens symposium 16-18 november 1982*.