

Checklistor i systemförvaltning

Kvalitetssäkring av användbarhet i utvecklingsprocessen

Kandidatuppsats, 15 högskolepoäng, INFK01 i informatik

Framlagd: 2 april, 2009

Författare: Kenny Olofsson
Sasan Zanganeh

Handledare: Umberto Fiaccadori

Examinatorer: Agneta Olerup och Erik Wallin

Abstract

Titel	Checklistor i systemförvaltning- Kvalitetssäkring av användbarhet i utvecklingsprocessen
Författare	Kenny Olofsson Sasan Zanganeh
Utgivare	Institutionen för Informatik
Handledare	Umberto Fiaccadori
Examinator	Agneta Olerup och Erik Wallin
Publiceringsår	2009
Uppsattstyp	Kandidatuppsats
Språk	Svenska
Nyckelord	Systemförvaltning, checklistor, användbarhet, värde, och användarstöd

Abstract

Checklistor för kvalitetssäkring i utvecklingsprocessen inom systemförvaltning är ett utforskat område. Problemet vi belyser är i vilka avseende checklistor fungerar som ett instrument för förbättrad användbarhet och kvalitet i förhållande till det som skall utvecklas. Vidare tittar vi på hur utvecklarna värderar hanteringen av checklistor. Vi har behandlat vilka dimensioner av systemförvaltning som checklistor är verksamma inom, på de olika företagen. Vi har tittat på vilket sätt utvecklarna värdesätter checklistor som verktyg för att säkerställa de krav som finns på användbarhet. Det vi har lyft fram i vår studie är hur utvecklare värderar balansen mellan fördelar ett system för med sig, gentemot den mängd energi en användare måste lägga ner för att förstå systemet till fullo.

Den empiriska studien resulterade i att samtliga informanter var överens om att checklistor ansågs vara ett verktyg för förbättrad kvalitet dock i olika avseende. Vi fann ingen direkt påverkan på att checklistor fungerade som ett instrument för förbättrad användbarhet. Det visade sig snarare att det inträffade indirekt. I vissa avseenden, framför allt prioritering, struktur och lärande, ansåg vi finna kopplingar mellan användandet av checklistor, förenkling och hantering i/av utvecklingsprocessen. Interaktionen mellan checklista och utvecklingsprocessen påverkar i sin tur även användbarhetsaspekterna därmed det indirekta inträffandet.

Förord

Vi vill ta tillfället i akt att tacka vår handledare Umberto Fiaccadori som har ställt upp i vårt och tort under uppsatsens gång i form av feedback och konstruktiv kritik. Vi vill också tacka de organisationer och informanter som har ställt upp på våra intervjuer och spelat en stor roll i uppsatsen. Vi passar också på att tacka alla de vänner och bekanta som ställt upp och korrekturläst uppsatsen samt gett stöd under arbetets gång.

Kenny Olofsson och Sasan Zanganeh

14 april, 2009

Innehåll

1	Problemområde	5
1.1	Frågeställning	7
1.2	Syfte	7
1.3	Avgränsning	7
2	Redogörelse för teoristudie	8
2.1	Vad är en checklista	8
2.2	Systemförvaltning	8
2.3	Användarstöd	10
2.4	Ändringshantering	10
2.4.1	Prioritering	11
2.4.2	Beredning	11
2.4.3	Genomförande och test	12
2.4.4	Införande	13
2.5	Utvecklarnas syn på användbarhet	13
2.5.1	Användbarhet enligt Nielsens definitioner	14
2.6	Värde	15
2.6.1	Syfte med värde	15
2.6.2	Använda sig av värde	15
3	Metod	17
3.1	Tillvägagångssätt	17
3.2	Intervjuer	18
3.2.1	Målgruppen – urval av intervjudeltagare	18
3.2.2	Uppbyggnad av intervjuer	18
3.2.3	Intervjufrågornas utformning	19
3.2.4	Inspelning & Transkribering	19
3.3	Reliabilitet	20
3.4	Validitet	20
3.5	Etik	21
4	Empirisk genomgång	22
4.1	Presentation	22
4.1.1	Företagen	22
4.1.2	Intervjupersoner	23
4.2	Systemförvaltning och utformandet av checklistor	23
4.3	Ändringshantering	24
4.4	Användarstöd	25
4.5	Användande av checklistor	26
4.6	Användbarhet i checklistor samt värde på kvalitet	27
4.7	Prioritering	28
4.8	Beredning och brister i användandet av checklistor	29
5	Diskussion och analys	30
5.1	Ändringshantering	30
5.2	Prioritering	30
5.3	Systemförvaltning och utformandet av checklistor	31
5.4	Användande av checklistor	32
5.5	Användbarhet i checklistor samt värde på kvalitet	33
5.6	Beredning och brister i användandet av checklistor	34
5.7	Användarstöd	34
5.8	Studiens brister	35
6	Slutsatser	36

Bilagor	38
Bilaga 1 - Exempel på checklista	38
Bilaga 2 - Intervjuguide	40
Bilaga 3 - Intervju med Johan Norman, Online i Lund	42
Bilaga 4 - Intervju med Johan, Logica AB	57
Bilaga 5 - Intervju med Thomas Farby, Volvo IT.....	62
Referenser	66

1 Problemområde

Problemet vi vill belysa är i vilket eller vilka avseende checklistor, inom systemförvaltningens utvecklingsprocess, är ett verktyg för ökad kvalitet inom organisationers olika projekt. Kvalitet är oerhört stort begrepp och frågan kring ”vad som definierar” kvalitet är ännu svårare att få grepp om (Macheridis 2005).

Den aspekt av kvalitet vi hanterar är utvecklarnas upplevda värde avseende användning av checklistor i och inför utvecklingsarbeten. I korta ordalag vill vi analysera utvecklarnas syn på användbarhet i den form det (tas upp) i checklistorna. Vår undersökning kommer att baseras på upplevelser kring checklistorna och hur de anses vara ett verktyg för användbarhet. Gulliksen & Göransson (2002) tar även upp att användbarhet är precis som kvalitet, svårdefinierat och kommer därför vara svårhanterligt om det inte sätts i en kontext eller ett avseende. Aspekten vi vill undersöka är hur dessa checklistor upplevs av de olika målgrupperna som använder dem samt vilka ”punkter” eller faktorer de belyser och varför.

Vi anser att också ordet användbarhet ett svårdefinierat uttryck då det kan ses från olika perspektiv. Från vår synvinkel och den tidigare erfarenhet vi har av användbarhet (utifrån de tidigare studierna och litteratur vi hanterat genom vår tid på informatik) anser vi att lättanvändning, effektivitet i arbetet, felhantering (låg risk för inträffande av fel) etc. är faktorer som bör finnas med, på ett eller annat sätt, i utvecklingsprocessen eftersom man i praktiken vill upprätthålla ett snabbt och smidigt arbete. Även om man ställs inför en förändring i form av uppdatering av det redan befintliga systemet eller ny IT-investering anser vi att *komplexitet* är något vi vill reducera. Ett system ska vara lättanvänt och om man gör en korrekt kontextbedömning av den arbetsmiljö systemet skall implementeras i, så kan man reducera komplexitet.

Det perspektiv vi valt att fokusera på är utvecklarnas syn på hur checklistor kan användas som verktyg (i utvecklingsprocessen) för att säkerställa att de kraven som ställs på användbarhet verkligen uppfylls.

Utvecklarnas (då vi talar om utvecklarna som driver projekt och implementerar projekten) upplevda värde av checklistor som verktyg vid utvecklingsprocessen anser vi vara väldigt intressant för denna studie. Vi kommer dock inte fokusera på enbart de aspekter i checklisten som hanterar användbarhet utan kommer även att titta på dess helhet för att senare kunna använda vår empiri till att diskutera användbarhetsperspektivet till vårt avseende. Vi vill, tillsammans med våra kunskaper kring interaktionsdesign och systemutveckling, ta reda på hur organisationer verkligen gör för att säkerställa kvaliteten i sina projekt och hur dessa checklistor utformas och används. Men deras struktur och deras upplevda värde kring de faktorer som checklistorna berör är mer intressant för denna undersökning anser vi.

Området användandet av checklistor inom systemförvaltning inom diverse systemutvecklingsföretag är fortfarande ett relativt utforskat område (vad vi förstått) utifrån vår genomgång av tidigare skrivna uppsatser. Med detta arbete kommer vi kunna bidra till

kunskap då en ny intressant infallsvinkel på fenomenet.

Då vår empiri baseras på företag som använder eller arbetar med systemförvaltning, i det avseende att de till viss del uppdaterar redan befintliga system, vi vill rikta våra frågor till informanter som hanterar dessa checklistor i olika kontexter för att se om vi kan se relevanta samband och/eller olikheter kopplat till deras utvecklingsprocess avseende checklistorna (deras utformning bl.a.), användbarhet samt användarstöd.

Problematiken samt begreppens komplexitet är grunden till att vi vill undersöka en eventuell koppling mellan användandet av checklistor och ökad kvalitet samt användbarhet. Vilket i sin tur kan leda till fortsatt forskning om organisationers problemhantering.

Lawson (2006) tar upp att problem i allmänhet framstår från lösningsförslag och desto mer lösningar det finns, desto mer problem existerar och tvärt om. Dessa två aspekter påverkar kontexten som utvecklingsprocessen är placerad i. Även om just Lawson (2006) betonar deskriptiva designproblemsförutsättningar, så hävdar han ändå att problem (även lösningar) i omvärlden (oavsett vilket yrke/område det rör sig om) leder till konsekvenser som bidrar till ny kunskap inom kontexten. Denna kunskap vill vi (med vår studie) hjälpa till att förstå eller utveckla i de avseende vi undersökt.

Systemförvaltning är ett snabbt sätt att ta ett system som redan är i användning till en ny nivå. Genom att vidareutveckla sitt redan befintliga system kan man spara pengar och tid under utvecklingsprocessen i jämfört med utvecklandet av ett nytt system. Inläringstiden av användarna blir också kortare då dessa redan är insatta i det aktuella systemet. (Nordström 2005)

Flertalet av nuvarande större organisationer som Microsoft, IBM, Hewlett & Packard med flera använder sig av systemförvaltning i form av ett informationsteknologiskt bibliotek. ITIL (IT infrastructure library) är ett ramverk som innehåller olika kärnprocesser som har checklistor för att hantera förvaltning av redan befintliga system (Clacy & Jennings 2007).

Med införandet av ITIL som ramverk möjliggörs en hel del fördelar avseende struktur och processhantering hävdar Brenner (2006) likaså Haverblad (2004). Haverblad (2004) har även manifesterat, i sin teori, fördelar som kan uppnås med införande av ITIL men i vårt problemområde vill vi bara förstora området systemförvaltning och påpeka att det är ett stort och (i vissa avseende) ett fördelaktigt medel för användandet av IT hellre än att dra upp en massa faktorer som senare inte kommer att behandlas.

Ett av huvudargumenten för användandet av systemförvaltning är just dess återanvändbarhet (reusability) som är en faktor för att spara/tjäna pengar (Qureshi & Hussain 2007).

Lycett & Giaglis (2000) tar upp varför systemförvaltning avseende ekonomisk vinst är ett intressant område att titta närmre på, då förvaltning finns i stor utsträckning ute hos företag och är ett sätt att förlänga livscykeln på de system som används. Även om Lycett & Giaglis (2000) betonar att de jobbar emot ett utvärderingsramverk för systemförvaltnings aktiviteter, finns en koppling mellan lönsamhet och återanvändning.

1.1 Frågeställning

Huvudfråga: I vilka avseenden fungerar checklistor som instrument för förbättrad användbarhet/kvalitet vid systemförvaltning?

Delfråga: På vilket sätt värderar utvecklarna hanteringen av checklistor avseende användbarhet och kvalitet i förhållande till det som skall utvecklas.

1.2 Syfte

Syftet med denna studie är att bidra till ökad förståelse för checklistors roll avseende utvecklarnas (de som hanterar och implementerar systemen) upplevda värde av användbarhet, deras syn på användarstöd samt hur checklistorna inom systemförvaltning är utformade. Systemförvaltning är av relevans då det som tidigare nämnt är ett sätt att återanvända redan befintliga system och även kostar mindre än att utveckla nya hävdar Nordström (2005), Lycett & Giaglis (2000) och Qureshi & Hussain (2007).

1.3 Avgränsning

Vår avgränsning inom företagen (med de informanter vi använder) kommer ligga på de större utvecklingsprojekten då inte tiden finns för att titta på flertalet små. Vi kommer också att välja bort själva driftbiten av ett IT-system vilket ingår och är en stor del av systemförvaltningsprocessen (Nordström 2005). Då vår fokus ligger på själva utvecklingsbiten.

Vi kommer även att lägga ITIL åtsidan då vi inte funnit empiri som använder sig av det.

Inom användbarhet talar vi även, i problemområdet, om komplexitet men eftersom vi anser att det är väldigt svårt att hantera detta begrepp, väljer vi därför att endast nämna det ytligt.

Syftet är inte att se vilken direkt effekt checklistorna har på användbarhet utifrån kundens perspektiv, utan snarare på vilket sätt (och i vilka avseenden) checklistorna hjälper utvecklarna säkerställa att de krav som ställs på användbarhet faktiskt uppfylls vid vidareutveckling av ett redan befintligt system.

2 Redogörelse för teoristudie

Här följer den teoristudie som vi gjort. Vi kommer sedan använda denna för vår diskussion och analys.

2.1 Vad är en checklista

Checklistor är ett instrument för kontrollering av dokument och programvara. Checklistor används som hjälpmedel i kvalitetsstyrning och fungerar som en sorts minneslapp. Checklistor kan finnas för alla sorters förhållanden längs processlinjen men är viktigast i de tidigare stadierna av utvecklingsprocessen då konsekvenserna av att glömma något är störst här. Andersen (1991)

Andersen (1991) diskuterar användandet av en checklista enligt följande:
”Den ger en översikt över förhållanden som man av erfarenhet vet är särskilt viktigt att undersöka för att gardera sig mot dålig kvalitet” Andersen (1991) s. 453.

Enligt Li, et al., (2008) skall checklistor behandla de risker som finns inom projektet och på ett enkelt sett kunna klassificera dessa.

Checklistor byggs på efterhand med hjälp av nya erfarenheter som utvecklarna tar åt sig från andra projekt. Checklistorna blir sedan en del utav en utbyggd utvecklingsmodell. Det går därför inte att säga att en checklista har en allmän giltighet utan snarare att en checklista är knuten till ett särskilt element i utvecklingsprocessen. Anledningen att den finns där är för att kvalitetssäkra. I bilaga 1 visar vi exempel på en checklista för kravspecifikationer. Den tar upp generella frågor om kravspecifikationen samtidigt som den ställer frågor om kvalitet. (Andersen 1991)

2.2 Systemförvaltning

Systemförvaltning är ett begrepp som härstammar från engelskan. På engelska beskrivs det ofta som ”Software Maintenance” och ofta som ”component-based systems” då det ständigt utvecklas nya komponenter till systemen. Begreppet ”Software Maintenance” kan översättas till flertalet betydelse såsom systemförvaltning och systemunderhåll. Begreppet skall översättas i den kontexten den är i och i vårt fall så handlar det om vidareutveckling av redan befintliga system och kommer därför översättas till

systemförvaltning. (Brandt 2005)

Nordström (2005) har gjort en empirisk studie som visar de olika aktiviteter som sker, inom en organisation, under systemförvaltningsprocessen. Dessa aktiviteter har hon funnit via arbete som konsult under åren 1996-2002. Hon beskriver att IT-drift och systemförvaltning är begrepp som teoretiker ofta försöker skilja åt men i praktiken faktiskt är synonymt med varandra och det enda som skiljer dem åt är vilken roll inom organisationen, som faktiskt utför dem. Nedan följer en tabell som redovisar de olika aktiviteterna som genomförs under systemförvaltningsprocessen vilket i sig kan liknas vid en checklista över vad som bör genomföras.

Tabell 2.1

Förvaltningsverksamhetens huvudaktiviteter och delaktiviteter
Nordström (2005)

Huvudaktivitet	Delaktivitet
Användarstöd	Besvara frågor Informera Utbilda
Ändringshantering	Initiera Prioritera Bereda Genomföra Testa Införa
Förvaltningsstyrning	Strategiskstyrning Operativ styrning
Daglig IT-drift och underhåll	Övervaka Köra Filöverföra Ge preventiv service Ta backuper Problemhantera

Även om Qureshi & Hussain (2007) betonar att de främsta fördelarna med systemförvaltning är dess återanvändning och kostnadseffektivitet, tar de även upp liknande faktorer som stämmer överens med huvudaktiviteterna i tabellen ovan. Deras processmodell för systemförvaltning behandlar användarstöd i form av en kommunikationsfas som handlar om att informera de involverade och besvara frågor i form av en aktiv dialog kring de modulerna som ska förvaltas. Vidare har processmodellens planerings-, implementerings- och driftfaser attribut som korresponderar med Nordström (2005) aktivitetstabell. De olika aktiviteterna samt faserna agerar som en form av kontextbaserad checklista som ska ge stöd för hanteringen av ett utvecklingsarbete. (Qureshi & Hussain (2007)

Givet att varje checklista skall vara kontextbaserad skall det även finnas en förståelse för hur svårdefinierat det är att ta sig an en real checklista för systemförvaltning med avseende för, framförallt, återanvändning. Utvärderingen av en sådan checklista kan nästan aldrig specificeras utan det är för många faktorer som spelar roll innan aktivitetslistan kan sammansättas. Risker som systemförvaltning medför i utvecklingsprocessen (återanvändbarhet av komponenter etc.) kan undvikas genom att jobba utifrån vissa sätt. Att integrera business driven utvärdering (alltså se vad som är mest lönsamt) i en tidig fas är ett sätt att fånga upp så mycket information om varje komponent som möjligt för att, i slutändan, reducera den mängd energi som måste läggas ner för att förstå uppdateringen. Även att se

utvärdering och selektionen (av komponenter och dess lösningsförslag) som en iterativ process i utvecklingsarbetet (Lycett & Giaglis 2000).

2.3 Användarstöd

Användarstöd inom systemförvaltning är något som ofta göms i bakgrunden av allt annat. Att erbjuda användarstöd eller så kallad support inom systemförvaltning är något som ofta är dyrt och kräver mycket resurser men som sällan synliggörs. Användarstöd kan delas upp i två kategorier. Dessa är att besvara frågor och utbilda samt informera. Ofta så finns det en så kallad helpdesk inom företaget som tar hand om enklare frågor och sedan en nivå högre där mer komplexa fall tas upp. Som komplement till helpdesken kan organisationen ge utbildning till användarna. Genom att ge utbildning till användarna kan antalet frågor som ställs till helpdesken minskas radikalt. Utbildning av slutanvändare är ofta något som glöms bort o sker mycket sent förhållande till implementeringen. (Nordström 2005)

Detta i sin tur motsäger Avison & Fitzgerald (2006) där de beskriver RUP. De förespråkar utbildning av personal innan systemet i sig skall implementeras, för att det senare skall kunna användas optimalt av användarna för att öka effektiviteten. Avison & Fitzgerald (2006) rekommenderar att genomföra utbildningar av ens personal till all förvaltning inom organisation dock sker detta oftast bara inför större uppdateringar av det interna systemet.

2.4 Ändringshantering

Enligt Nordström (2005) så utför organisationen ett val av ändringsstrategi när det fastställs att en ändring bör göras inom det befintliga systemet. Det nämns fyra olika strategier för ändringshantering. Den första är helt enkelt att ändringen genomförs, vilket vi kommer närmre in på senare i kapitlet. Den andra strategin är att inte bry sig om att genomföra ändringen. Det är ett val som väljs ganska sällan men väljs då ett system skall avvecklas eller att organisationen helt enkelt känner att ändringen blir för dyr och ändringen inte anses lönsam. Den tredje strategin går ut på att användarna faktiskt anpassar organisationen efter systemet. Detta kan ske både omedvetet och medvetet. Det som oftast ligger till grund för det här är att användarna helt enkelt inte vet var de skall vända sig för att ge förslag på en eventuell ändring i systemet.

Den fjärde strategin leder till att användarna använder systemet till något annat ändamål än vad som från början var tänkt vilket vi kan se från olika studier. Ett exempel på det här är ett företag som Robey & Boudreau (2000) tar upp. Här har användarna börjat använda databasen som ett system att kommunicera med varandra. Nordström (2005) nämner också ett exempel där ett ekonomisystem i en svensk myndighet till slut användes som ett system för tidsrapportering.

Ändringshantering kan ses som en process då flödet är sekventiellt. De aktiviteter som finns benämns olika från organisation till organisation men innebörden är den samma. De aktiviteterna som finns i processen är:

- Prioritera
- Bereda
- Genomföra
- Testa
- Införa

Då en akut ändring behövs kan processen förenklas för att spara tid. (Nordström 2005)

Detta är en process som Nordström (2005) beskriver som typisk enligt hennes empiriska studier. Det är dock viktigt att utvecklarna måste förstå att valet av olika strategier och processer kommer att påverka hur pass bra det slutliga resultatet blir (Rothenberg, et., al. 2003)

2.4.1 Prioritering

Beredning görs och punkter prioriteras inom projektet. Detta görs för att underlätta beslutet om ändringen skall genomföras eller ej. Detta sker iterativt då den ansvarige först gör en grovprioritering redan under initieringen för att sedan gå in på en grundligare nivå.

Organisationen prioriterar in änderingsarbetet i olika kategorier och dessa är

- Akut ändring
- Önskemål
- Ingen ändring
- Normal ändring

De ändringar som går under akuta ändringar är när någonting i systemet slutat fungera. Detta leder oftast till intäktstappande och får därför högsta prioritet. Ibland så fungerar systemet och alla delprocesser fortlöper som de skall dock blir det fortfarande fel någonstans. Detta klassas oftast också som akuta fall och för högsta prioritet. Vid flertalet fall så tas beslutet att ändringen inte skall genomföras. Detta grundas på att förarbetet under initieringen har gjorts av en som inte har haft tillräcklig kunskap inom området och ändringen faktiskt inte behöver genomföras. Den vanligaste ändringen är den som faller under den normala kategorin. Denna får inte den högsta prioritering men har ofta en deadline som måste följas. (Nordström 2005)

Qureshi & Hussain (2007) ser på det här från en extern vinkel alltså hur det här ser ut från ett konsultföretags perspektiv. De påstår att först så kontaktar företaget som har systemet en konsult där sedan en prioriteringen av en ändring görs. Detta följs sedan upp av en planeringsfas där det görs en så kallat CBA (Cost benefit analysis) som innebär att de som prioriterar helt enkelt tar reda på om ändringen är värd att genomföra eller ej. De säger också att det är väldigt viktigt att de som planerar under planeringsfasen ser till att så många komponenter som möjligt verkligen återanvänds då det är onödigt att uppfinna hjulet en gång till. (Qureshi & Hussain 2007)

2.4.2 Beredning

Arbetet fortlöper med mer specifik planering av änderingsarbetet. De som planerar börjar med att ta reda på vilka andra program och moduler inom systemet som blir berörda av ändringen. Sedan följer en så kallad konsekvensanalys. Denna innehåller information i form av en kostnadsplan för ändringen samt vilka konsekvenser ändringen kan ha för närliggande system.

Det som oftast uppfattas som svårast är att beräkna tid och kostnad för projektet då det görs efter intuition och erfarenhet. Det är svårt att göra en resursuppskattning och se alla konsekvenser, ändringen kommer att genomföras. Det är därför viktigt att viss iteration tillåts då prioriteringen kan behöva göras om. Det som gör att resursuppskattningen är svårt och ofta är bristfällig är att beredningen ofta varit bristfällig på grund av dåligt dokumenterad kod och det är svårt att veta var i koden det behövs ändras och var det kommer följa konsekvenser utifrån ändrande i koden. (Nordström 2005)

2.4.3 Genomförande och test

Genomförande och test utförs ofta simultant och processen är iterativ. Utvecklarna och testarna utför ändringar och testar tills de vet att det fungerar så som det är tänkt i kravspecifikationen. De arbetsuppgifter som utförs i samband med genomförande är:

- Ändring i kod och ändring i process/produkt
- Ändring av dokumentation
- Beskrivning av utfallet

Det som utförs i form av testning är.

- Upprättande av testdokumentation
- Genomförande av program- Integrations- och systemtest
- Genomförande av acceptanstest
- Ändring av användardokumentation

Det kommer ofta klagomål från ledningen att testerna drar ut på tiden. Detta i sig är inte konstigt då testerna ofta är den mer tidskrävande delen i processen. Det är även så att testen måste genomföras under en längre tid då majoriteten av de som blir berörda av ändringen måste vara delaktiga. Oftast så är det svårt att få anställda att ställa upp på tester och projektgruppen får ofta efterfråga anställda som vill ställa upp. Ibland så klagas det även på att det är svårt att genomföra flertalet ändringarna parallellt. Detta är vanligt förekommande efter som omprioriteringar görs. Kommer det in ett akutfall så måste det de just då arbetar med släppas för att åtgärda det akuta. (Nordström 2005)

Som ovan nämns så kan testning ofta dra ut på tiden samt kosta en stor summa pengar. Detta är något som Mariani & Pezze (2005) behandlar där de har utvecklat ett ramverk för testning som skall kunna appliceras på i stort sett alla vidareutvecklade komponenter oavsett vilket system de tillhör just för att öka hastigheten. De anser också att ofta när utvecklarna vidareutvecklar en komponent så måste de återigen testa komponenten samtidigt som de testat de andra komponenterna som är acossierade till komponenten i fråga. Detta är också något som Qureshi & Hussain (2007) går in på och vilka hävdar att testningen bör innefatta alla de komponenter som berörts av ändringen.

Mariani & Pezze (2005) testsekvens ser något annorlunda ut jämfört med den Nordström (2005) nämner. De behåller ungefär samma punkter som ovan nämns, men ställer sig annorlunda, vad det gäller acceptanstest. Detta är något de inte behandlar över huvudtaget.

2.4.4 Införande

Nordström (2005) beskriver gällande införandet av en ändring så finns det tre olika strategier man använder sig av:

- Kontinuerlig hantering
- Releasehantering
- Kombination av release- och kontinuerlig hantering

Med kontinuerlig ändring hävdar Nordström (2005) att ändringarna sker i den takt problem uppstår. Detta har nästan all systemförvaltning inslag av. Med releasehantering så släpper man en stor samling av ändringar på en i förväg planerad tidspunkt. Här arbetar de ofta efter prioriteringar där man genomför de viktigaste först och sedan arbetar sig ner.

Detta påminner om RAD där de som prioriterar använder sig av regler kallade MoSCoW rules för att prioritera vad som bör göras. (Avison & Fitzgerald, G. 2006)

- M = "must" Utan de här kraven så blir systemet inte lönt att implementera.
- S = "should" Detta är krav som systemet borde ha. Dessa kan vara måsten som blivit bortprioriterade för att tillmötesgå deadline. Det är också krav för att organisationen ska få ut full effekt av sitt nya system
- C = "could" Detta är krav som systemet kan ha. Ges det tid och resurser att färdigställa de här kraven så görs det.
- W = "would" Detta är krav som inte kommer levereras. Dessa kan lämnas ute för de inte gagnar organisationen på något större sett. Dessa kan också göras vid en uppdatering av systemet.

MoSCoW-reglerna säkerställer att de mest vitala delarna av kravspecifikationen uppfylls. De säkerställer också att inga önskelistor av vad användarna vill ha med i systemet finns med i kravspecifikationen. Alla krav måste bli kategoriserade. Oftast så sätts MSC som högsta prioritet för att sedan ge tillfälle för att utelämnas alla Cn och ibland även ett antal S. (Avison & Fitzgerald, G. 2006).

På senare tid har majoriteten av företagen börjat använda sig av en kombination av de båda ovan nämnda. Där företagen väljer att släppa de mindre ändringarna separata för att sedan göra större releaser av de ändringar som innefattar större delar av systemet. (Nordström 2005)

2.5 Utvecklarnas syn på användbarhet

I vårt avseende har användbarhetsaspekten varit vinklad åt utvecklare och deras syn på användbarhet, kopplat till sin utvecklingsmetod. Det vill säga vilket upplevt *värde* anser utvecklare att checklistor har för att säkerställa att kraven på användbarhet verkligen uppfylls. Vi anser därför att det är viktigt (för er som läsare) att förstå användbarhet, i detta avseende, inte har att göra med "slutanvändare" i utvecklingsarbetet utan mer åt hur olika aspekter i checklistor hanterar användbarhetsaspekter. Det är åter utvecklarnas *upplevda värde* på dessa aspekter vi är ute efter att lyfta fram avseende användbarhet.

Huvudsyftet med användbarhet är att få användare att uppnå specifika mål eller avseende vid interaktion. Detta medför ju såklart att systemet måste gå att använda. Som utvecklare av diverse applikationer, tjänster eller produkter så ställs utvecklarna inför 2 frågor:

- Hur säkerställer man som utvecklare, användbarheten i ett utvecklat system?
- Hur demonstrerar/mäter vi användbarhet?

Dessa två frågor, som Dix, et al., (1997) hävdar, är för det första väldigt svåra att sätta ramar för, men vi kan med hjälp av diverse litteratur försöka sätta dessa i perspektiv och i det avseende som hjälper en svara på frågställningen Dix, et al., (1997). Vidare nämner även Preece, et al., (2002) att desto komplexare kontext, desto komplexare checklistor (eller metoder som Preece, et al., (2002) faktiskt säger) appliceras på, desto svårare är det att sätta ramar för användbarhetsaspekter där bl.a. frågor som hur *mätningen* av användbarhet hanteras.

Definitioner av användbarhet existerar i mycket av den litteratur vi granskat och vi har sett att mycket av den informationen är sig lik från studie till studie. Vi beskriver ett par definitioner för att underlätta förståelsen kring användbarhet då vi anser att dessa är relevanta för att dem hjälper oss att förstå vilka aspekter utvecklare tar hänsyn till när de värdesätter användbarhet i en produkt eller tjänst.

Vi kommer att hålla oss till Nielsen (1993) då de flesta skrifterna mer eller mindre överensstämmer med varandra. Ett exempel är ISO 9241-11 definitioner som består av i princip identiska attribut nästan identiska med de Nielsen beskriver nedan. Även författare som Preece, et al., (1997) hävdar att faktorer som *learnability*, *error* osv. ska betraktas när vid definitioner av användbarhet. Dock vill vi även nämna att många av dessa definitioner realiserar med att det finns en form av slutanvändare med i beslutsfattandet. Nielsen (1993) tar mer upp allmänna definitioner, eller snarare faktorer, som berör användbarhet överlag och det är därför som vi valt att beskriva just dennes definitioner.

2.5.1 Användbarhet enligt Nielsens definitioner

Enligt Nielsen (1993, p.29), är användbarhet en sammansättning komponenter som är associerade med 5 attribut:

- *Learnability*: Bygger på att systemet skall vara enkelt att förstå och adoptera så att användarna snabbt kan sätta igång med sitt arbete.
- *Efficiency*: Handlar om att öka effektivitet. När användaren väl fått grepp om systemet så skall högre nivåer av produktivitet åstadkommas.
- *Memorability*: Systemet skall vara lätt att överskåda och memorera så att användaren, vid t.ex. frånvaro en längre tid, kan snabbt återuppta integreringen utan att behöva ”lära sig” allt från början
- *Error*: Låg felprocent i systemet samt en hög grad på felhanteringen så att katastrofala fel eller ”errors” elimineras eller helt enkelt INTE får ske.

- *Satisfaction*: Behaglighet vid integrering av systemet är viktigt och en aspekt många oftast glömmet. Detta attribut bygger på att få användaren att "tycka om" systemet eller "sitt arbete" i helhet.

2.6 Värde

Nedan följer en genomgång av begreppet värde

2.6.1 Syfte med värde

Syftet med att lyfta fram värde som delkapitel i teorin är för att öka förståelsen kring utvecklarens upplevda värde avseende systemförvaltning (och de aspekter checklistorna inom systemförvaltning medför). Konceptet värde bygger på att förstå kundens behov samt reducera kostnad för användarmedverkan. Att använda termen värde i olika avseenden kan ske på olika sätt och beroende på vad som skall åstadkommas. Det ska beaktas att värde är ett väldigt komplext begrepp. Om leverantören ägnar sig åt att förstå vad värde för kunder handlar om, så minskar de risken för negativa konsekvenser. I vårt fall talar vi om att effektivisera i termer om att reducera "customer-perceived sacrifice", som handlar om ett antagande kring den mängd energi och tid kunden måste lägga ner för att förstå en artefakt (i detta fall system eller dylikt som checklistan bidrar till), genom att minimera relationen mellan kund och artefakt (minska tiden det tar att lära sig, köpa, sätta igång etc., mer om detta kommer nedan). (Ravald & Grönroos 1996)

2.6.2 Använda sig av värde

Till att börja med vill vi hantera en intressant aspekt inom konceptet värde som bygger på någonting som i teorin kallas "customer-perceived value". Mellan 80–90-talet så började konceptet värde introduceras allt mer på marknaden. Då talade man om värdet av konsumenter dock fanns det ingen klar definition av termen värde i de tidiga studierna av bl.a. Engel et al., (1990), Zaltman & Wallendorf (1983) och Ravald & Grönroos (1996). Däremot så vinklade Peter & Olson (1993) diskussionen kring värde avseende användbarhet (utility) då en kund köper en produkt/tjänst. Monroe (1991) bygger vidare på denna diskussion genom att fastställa förhållandet mellan förutfattade fördelar (perceived benefits) och förutfattad mängd energi/tid man uppoffrar (perceived sacrifice). Perceived sacrifice inkluderar alla de kostnader en kund ställs inför vid inköp som priset, tillgänglighet, leverans, installation, order, handling, reparation, förvaltning etc. Perceived benefits är en kombination av fysiska attribut, service, support, mental tillfredsställelse (svår att mäta men betraktas ändå) etc. Detta förhållande avser hjälpa utvecklare av produkter/tjänster att betrakta olika dimensioner och infallsvinklar för att värdesätta sin artefakt.

För att rikta lite kritik mot termen värde ska komplexiteten värde medför beaktas. Inte nog med att det är svårdefinierat och bygger på mycket normer och värderingar. Förhållandet mellan förutfattade fördelar och ”egen uppoffring” är väldigt objektivt hävdar Zeithaml (1988). Trots att de delar samma vision kring värde (i det avseende då en kund ställs inför en produkt eller tjänst) som Monroe (1991), så belyser de problematiken kring förhållandet eftersom Zeithaml (1988) anser att det är subjektivt och förändras individ till individ.

3 Metod

Ett kapitel där vi beskriver hur vi genomfört vår studie.

3.1 Tillvägagångssätt

I denna undersökning har vi använt oss av en kvalitativ forskningsmetodik för att forma vår forskningsstrategi. Vi utgick från den kvalitativa ansatsen eftersom våra kunskaper, om användning av checklistor, är relativt liten. Vidare har detta gett oss en djupare och mer detaljerad förståelse för hur vi ska bemöta forskningsfrågan. (Jacobsen 2002)

Studiens kvalitet ökas och bidrar till en förhöjd trovärdighet, samt stiger möjligheten för generaliserbarhet (dock är vi medvetna om att det blir en form av generaliserbarhet inom vår avgränsning) (Holme, et al., 1997). Vidare tar Seale (1999) upp att om forskaren hanterar sin resurser rätt, så ökar givetvis studiens kvalitet som i sin tur bidrar till en förbättrad validitet och större möjligheter till reala resultat.

Den kvalitativa förhållningen har vi valt då vi, i denna studie, har en betoning på individers perspektiv och kunskap (Holme, et al., 1997). Enligt Jacobsen (1993) är intervjuer den mest erkända och konfirmerade metoden för datainsamling vilket är upphovet till varför vi vill göra en kvalitativ undersökning. Vi har givetvis inte samma kompetens inom kontexten som de kommande informanterna har och därför ska intervjuerna hjälpa oss belysa fenomenet. Den har givetvis formats utifrån vår teori så att vi kunde vinkla undersökningen åt det håll som gjorde att vi kunde besvara vår frågeställning men det är viktigt, enligt Bryman (2002), att ge utrymme till individerna så att de får chansen att öppna och/eller vidare fördjupa en viss fråga. Intervjuguiden finns i bilaga 2.

Studien vi utför har, till stor grad, blivit subjektiv då den bygger på vad vi har uppfattat som intressant och relevant för just denna undersökning. I ansats att försöka minska subjektiviteten, som i sin tur medför en minskad risk för partiskhet och snedvridna synsätt skapade av våra värderingar och normer, har vi valt att sträva efter att få empiriskt material bifogat så som transkriberingar. Detta med ett klart godkännande från våra informanter och respondenter såklart i ett informerat samtycke.

Vi har strävat efter att möjliggöra replikerbarhet i metodiken för denna studie. Detta kan i sin tur vara upphov till vidare studier eller till hjälp för framtida läsare. Resultatet behöver däremot inte bli det samma då Fitzgerald, B. et al., (2002), lika så Avison & Fitzgerald, G. (2006) tar upp att en utvecklingskontext är unik för varje organisation vilket medför att replikerbarhet från organisation till organisation inte bör anses som lämpligt sätt att förhålla

sig. Anledningen till att vi beskriver i detta kapitel, så gott vi kan, hur vår undersökning har utförts (samt ger beskrivande bakgrund om informanterna) är eftersom IT-användning kontinuerligt förändras i dagens samhälle vilket har en direkt påverkan på *vad* organisation måste kontinuerligt ta hänsyn till, samt *på vilket sätt* (och vilka avseende) då IT-användning skall hanteras. (Alvesson & Svenningsson 2007).

3.2 Intervjuer

Nedan följer en genomgång av den datainsamlingsmetod vi har valt

3.2.1 Målgruppen – urval av intervjudeltagare

Vi hade för avsikt att rikta våra intervjuer till personer med högre kompetens och erfarenhet inom branschen. Detta resulterade i en ansats att försöka införskaffa informanter med en högre erfarenhet inom systemförvaltning. I vår urvalsprocess utgick vi från ett antal krav på både företagen samt informanterna. Kravet på företag var att de använder sig av systemförvaltning till den grad att det existerar uppdateringen av redan befintliga system. Informanterna ville vi ha god erfarenhet kring systemförvaltning och gärna kunna svara på frågor kring checklistor.

Vi valde att ta vidare kontakt med företagen Online i Lund, Logica AB samt Volvo IT då de uppfyllde det första kravet. Två av dessa var oss geografiskt nära men Volvo IT som var placerat längre bort gjorde att vi inte hade möjlighet att genomföra intervjun ansikte mot ansikte.

Vidare så ansåg vi att de informanter vi fått tag på inom företagen till signifikant grad uppfyller vårt andra krav. Detta eftersom två av tre informanter är representanter för en större grupp. Johan Norman, som är IT-chef på Online i Lund samt Thomas Farby som är projektledare och business-solution manager inom Volvo IT. Johan på Logica är en systemutvecklare som arbetat inom organisationen i två år. Även om första intrycket inte nödvändigtvis medför lika mycket ansvarsområden som de andra två informanterna vill vi hävda att i samtal med Johan på Logica fick vi intrycket av att han är kunnig och mycket väl kan bidra med intressanta aspekter till vår intervjustudie. Vidare är även Johan på Logica relativt ny i arbetslivet i jämförelse med de andra två vilket kan medföra ett mer uppdaterat svar på våra intervjufrågor.

3.2.2 Uppbyggnad av intervjuer

Vi har riktat vår intervjustil åt det semistrukturerade hållet. Vi hade som mål att utföra intervjuerna ansikte mot ansikte och försöka ha en kombination av öppen och strukturerad stil dock fick en av intervjuerna göras som e-intervju då personen befinner sig långt iväg och på grund utav geografiska skäl valdes detta intervjusättet. Vi ansåg det viktigt att ställa en viss

fråga med en stor bredd till att börja med. Detta gav respondenten en chans att uttrycka sig på sitt sätt och kunde därmed hjälpa oss vinkla våra delfrågor (till föregående fråga) åt det hållet som vår frågeställning behöver, för att kunna besvaras. Även delfrågorna skall ha en viss öppenhet så att våra respondenter får uttrycka sig fritt i sitt svar och inte ”ja & nej” struktur. Detta gör att semistrukturen gav oss specifika svar på vår frågeställning. Men vi undviker att styra intervjun utan låter informanterna tala fritt. (Kvale 1996)

Både Kvale (1996) och Bryman (2002) nämner att kontentan av en semistrukturell användning blir att informanternas svar bidrar till spontana frågor som mycket väl kan bli en ny intressant infallsvinkel vilket gör att vi har kunnat ställa följd frågor för att gräva ännu djupare.

Vi är medvetna om att den semistrukturella intervjuemetoden har medfört en bredare uppfattning kring en viss fråga vilket öppnar upp nya infallsvinklar etc. Vi har även haft för avsikt att gräva djupare i varje intervjufråga för att komma in på våra informanternas upplevda värde kring checklistor, kopplat till ett visst avseende. Vi gjorde vår ansats till detta genom att tillåta informanten att uttrycka sitt uppfattande och det värde dem anser ha kring användandet av checklistor i utvecklingsprocessen avseende systemförvaltning. Kvale (1996) hävdar att djupgående analyser bidrar till ett djup som kan kompensera det innehållet i data forskaren får ut eftersom han/hon vidareutvecklar det kvalitativa spektrumet.

3.2.3 Intervjufrågornas utformning

Frågorna är utformade efter den teoretiska studien då teori kring förvaltning, styr våra frågor åt ett håll som hjälper oss fastställa syftet. Vi har med hjälp av intervjufrågorna lyft fram information som berör checklistorna på så sätt att det (i olika avseenden) fungerar som ett sätt för utvecklarna att säkerställa att kraven på användbarhet verkligen uppfylls. Med dessa frågor lyfter vi fram en intervjuguide (se bilaga 2) som har agerat som underlag för intervjuerna.

Som Kvale (1996) nämner så vill vi undvika ”varför” frågor då det kan ställa informanten i ett otryggt/obehagligt svars läge. Intervjun inriktar sig i dagsläget till människor med IT-vana samt kunskap kring systemförvaltning. Vi lyckades även få kontakt med IT-chefer (2/3 informanter = chefer) vilka antas innefatta högre validitet och reliabilitet avseende roller inom organisationen då vi antar att en IT-chef har högre kompetensområde än en enskild utvecklare (även om det nödvändigtvis inte är så i verkligheten).

3.2.4 Inspelning & Transkribering

En intervju kan ske på ett flertal sätt. Vi har använt oss av den mest vanliga dokumenteringen av intervjuer via ljudinspelning som sedan transkriberas. (Jacobsen, D 2002) Mest förekommande är användandet av en digital inspelningsmaskin som sedan kan föras över på datorn för att sedan kunna analyseras och transkriberas (Brinkmann & Kvale 2009). Vi har fört mindre anteckningar under intervjuerna men samtidigt varit aktsamma för att tappa fokus på intervjupersonen genom att föra alltför noggranna anteckningar. Fördelen med inspelning av en intervju är att intervjupersonens ton och pauser kan lyssnas på om och om igen. (Brinkmann & Kvale 2009)

Vi har transkriberat på ett sätt där vi tog med intervjupersonens uttryck. Att utelämna intervjupersonens känslor kan lätt leda till att det som sägs missuppfattas. Brinkmann, S., Kvale (2009) tar upp ett exempel där en dansk student intervjuas. På frågan om studenter ser ner på klasskamrater som har fått sämre betyg. Pojken svarar "No, no-no, one does not look down on anyone who gets bad grades that is not done. I do not believe that: well, it may be that there are some who do it, but I don't". Meningen kan lätt transkriberas som "One does not look down on others with low grades nor compete for grades" där de transkriberar missar den faktiska sanningen. Vi har därför transkriberat alla ord som sägs under en intervju då vi ville veta hur målgrupperna känner för sina checklistor samt hur dessa används, så kom känslor att spela en väldigt stor roll. Vi har också skrivit ner pauser, suckningar och skratt som intervjupersonen gjort, vilket kan innebära något annat än vad personen ifråga uttrycker i ord. Graden av djup på transkriberingen har lagts att uttryck skrivs ner men bara till den nivå att de finns, inte vilket uttal personen har eller dylikt. (Jacobsen 1993)

3.3 Reliabilitet

Självklart är det av högsta prioritet att vår uppsats skall ha så hög reliabilitet som möjligt. Enligt Holme, et al (1997) skall gärna en annan forskning med samma ämne bedrivas parallellt av en annan forskare vilket i vårt fall dessvärre inte sker. Ingen tidigare forskning har heller bedrivits inom området så vi kan därför inte heller jämföra med tidigare resultat. Holme, et al (1997) beskriver också hur man kan bära sig åt för att uppnå så hög trovärdighet som möjligt utan de förutsättningar som tidigare nämnts. Holme, et al (1997) förklarar att forskare bör utarbeta instruktioner och rutiner för de olika faserna i forskningen och vi kommer därför utarbeta frågor som är klara i form av en intervjuguide. Här får intervjupersonen sedan säkerställa vår reliabilitet genom att berätta för oss vad som är bra och vad vi bör ändra i vår intervjuguide för att öka respondentens förståelse. För att få fram samma svar från respondenten oavsett kontext så skall vi vara så subjektiva som möjligt och på minsta sätt styra vår intervjuperson till ett speciellt svar. Vi har också försökt beskriva vår metod så grundligt det bara går vilket Seale (1999) påstår öka reliabiliteten.

3.4 Validitet

Med validitet menas att det forskningsarbete som faktiskt bedrivs är relevant förhållande till frågeställning och syfte (Holme et al 1997). Vi har försökt lägga stor omsorg vid det här och försökt följa en röd tråd genom vårt arbete från det att vi skapade vår frågeställning, genom vårt teoretiska ramverk för att sedan vid val av metod säkerställa att denna är relevant just för att besvara vår frågeställning. Senare under utformningen av intervjuguide har vi försökt ställa frågor direkt kopplade till vår frågeställning med hjälp av det teoretiska ramverket för att vidbehålla sammanhållningen i uppsatsen. Även vid urval av intervjupersoner har vi försökt avgränsa oss till personal inom företaget med kunskap inom uppsatsområdet för att göra resultatet så trovärdigt som möjligt.

3.5 Etik

För att agera etiskt gentemot våra intervjupersoner har vi låtit de ta del av allt det material vi besitter. Vi har gjort vårt bästa för att inte undanhålla någon information från våra respondenter. Vi har också låtit de personer vi intervjuat läsa och granska den datasammanställning vi gjort för att säkerställa att vederbörande ger samtycke till detta.

När vi genomförde forskningen var vi förstådda med att intervjupersonerna var satta i en väldigt känslig position. Informanterna vill gärna inte säga vissa saker i rädsla av att bli utsatt på arbetsplatsen som sedan kan sluta i en uppsägning från personen eller avskedning från företagsledningen. (Siber 2001)

Vi började intervjuerna med att informera att dessa var frivilliga och att om de skulle välja att avstå så tog vi inte illa vid oss. Vi gav respondenterna tillfälle att avbryta intervjun om de kände sig obekväma samt att stryka intervjun ur vår undersökning. Vi gav också intervjupersonerna möjlighet att framträda anonymt om detta var ett önskemål från deras sida. Samtycket gavs antingen skriftligt eller muntligt innan intervjun började. Då valet av intervjuer varit att dessa skall vara semistrukturerade så ordnade vi även ett samtycke efter intervjun då en semistrukturerad intervju kan innehålla frågor som informanten från början inte visste skulle komma. (Kvale 1996)

I de fall det har behövts har vi i vår studie aidentifierat intervjupersonen. Detta har inneburit borttagning av information som direkt kan knyta respondenten till den specifika intervjun. Vi har även ersatt eller tagit bort de namn som har velat vara anonyma. Den informationen vi fått har aidentifieras så fort, som vi har haft möjlighet och informationen har bara funnits för vårt bruk. Vi förstår att genom att hålla den här informationen konfidentiell så försvårar vi möjligheten att replikera den här studien. (Kvale 1996).

4 Empirisk genomgång

Kapitlet innehåller beskrivning av de informanter vi har använt oss av samt en mindre företagsbeskrivning av de företag som deltagit i uppsatsen. Kapitlet innehåller också genomgång av våra intervjuer.

4.1 Presentation

Nedan följer presentationer av de olika intervjupersonerna samt mindre presentationer av de olika företagen vi har besökt. Genom att göra detta vill vi ge en klar bild för er varifrån vår information kommer och utifrån vilket perspektiv informanten analyserar deras checklistor. Intervjuerna har genomförts med hjälp av en intervjuguide som finns i bilaga 2.

4.1.1 Företagen

Online i Lund

Online har flertalet divisioner inom verksamheten vilket involverar så väl systemutveckling som telemarketing. De levererar skräddarsydda system till företag inom prenumerations- och postorderverksamheter. De åtar sig hela ansvaret för det levererade systemet och tillhandahåller it-support, underhåll och vidareutveckling av produkten.

Logica AB

Logica är ett företag som levererar tjänster. Detta involverar bl.a outsourcing samt vidare utveckling av produkter. De fokuserar på skräddarsydda lösningar till sina kunder och är därför väldigt nytänkande. Logica satsar på en bredd av lösningar och har därför specialister inom flertalet områden.

Volvo IT

Volvo IT är ett dotterbolag i koncernen Volvo AB. De utvecklar it-lösningar till olika fordonsföretag då oftast inom Volvo. De it-lösningar som finns är oftast baserade på SAP. De sysslar även med system för telemarketing och även en del med konsulttjänster.

4.1.2 Intervjupersoner

De intervjupersoner som valts ut har allihop spenderat ett antal år inom branschen. Rollerna skiljer intervjupersonerna åt. Vilket innebär att vi har försökt involvera informanter med olika bakgrunder samt med olika hierarkiska positioner inom respektive företag. Genom att göra detta har vi fått en bredd av informanter där vissa hjälper till att bestämma hur en checklista skall se ut samt personer som bara följer de checklistor som de får att arbeta med.

Informant 1 (*Johan Norman, Online i Lund*)

Arbetar som IT-chef på Online i Lund och har en bakgrund som systemutvecklare. Hans främsta arbetsuppgift är systemförvaltning men han uppehåller sig också med projektledning inom de projekt företaget håller på med. Då företaget är av mindre kaliber så är han involverad i alla projekten och känner att han är inblandad i allt som har med systemutveckling att göra på företaget. Johan är också ansvarig för strukturen i de projekt företagen arbetar med och har därför stor insyn i hur ett projekt vanligtvis går till väga.

Informant 2 (*Johan, Logica AB i Kristianstad*)

Arbetar med sitt första jobb inom systemutveckling men har en gedigen utbildning inom ämnet. Informanten arbetar med systemförvaltning inom organisationen där de håller på att nyutveckla redan befintliga produkter. Han har ingen medverkan i checklistornas utformning men kan fortfarande komma med förslag och idéer på hur de kan ändras. Johan har valt att utelämna sitt efternamn vid det informerade samtycket.

Informant 3 (*Thomas Farby Volvo IT*)

Informanten arbetar som Program Manager inom Volvo på en avdelning vilka levererar drivlinor och växellåder till Volvobilar. Han har också en roll som "Business solution manager" inom företaget där han ansvarar för företagets utveckling av SAP-applikationer. Thomas har arbetat 10 år inom organisationen och har således utvecklat ett gediget förstående för utvecklingsprocessen.

4.2 Systemförvaltning och utformandet av checklistor

Enligt Thomas Farby på Volvo IT arbetar de med formaliserade checklistor, i rollen som program-manager finns där manifesterat punkter som illustrerar vad som kan kallas en checklista. Dessa checklistor fungerar, i Thomas Farbys på Volvo IT's fall, som ett verktyg som hjälper dem med diverse faktorer, t.ex. uppföljning då alla möten, redovisningar har en röd tråd tack vare checklistan.

" I min roll som Program Manager, hjälper det mig genom att alla projekt följs upp på samma sätt, att alla gate-passager har en checklista som skall vara avprickad för passage och öppnade av nästa road-map." (Bilaga 5, (6), Thomas Farby, Volvo IT).

Vikten i att förstå checklistans kontext och vilka konsekvenser de har för användandet betonas av Thomas Farby på Volvo IT. De hävdar att en checklista är ett fördelaktigt verktyg men

måste hanteras med aktsamhet. En uppgifts/projekts kontext påverkar checklistans utformande och användning. Genom att förstå kontexten förstås checklistans användning.

”I de flesta fall är det givetvis en fördel att använda checklistor, men de måste alltid användas med förstånd för att inte skapa en onödig administration/onödiga kostnader.” (Bilaga 5, (7), Thomas Farby, Volvo IT)

Checklistorna är grundade på aspekter som en avdelning inom Volvo IT hanterar. Då SAP är deras integrerade system finns en grupp som hanterar checklistornas utformning, dock vet vi inte mer än så därför lämnar vi det ytligt.

”I Volvo IT finns en hel avdelning som arbetar med metoder. Inom Global SAP finns en grupp som arbetar med att SAP-anpassa det som utvecklas av metodgruppen.” (Bilaga 5, (9) Thomas Farby, Volvo IT)

4.3 Ändringshantering

Utvecklare värdesätter checklistorna stort i förhållande till användbarhetsaspekten eftersom det fungerar som styrmedel avseende coaching av oerfarna utvecklare.

Checklistornas utformning är som tidigare nämnt formaliserade. Med detta menas att enligt Thomas Farby har Volvo IT en struktur och strategi som manifesterats i de punkter som arbetarna följer. Föränderligheten sker således av högre ledningen vilket sätter vissa tyglar på den enskilda utvecklaren/projektledaren. Dock så spelar åter projektets kontext roll. Hanteringen av komplexiteten hos ett specifikt projekt medför på vilket sett du skall implementera checklistans innehåll. Däremot kan den ses som ett instrument som kräver obligatorisk närvaro vid olika sammanhang.

”Nej checklistorna och metoderna är obligatoriska och skall inte förändras. Givetvis skall dom också användas med ”sunt förnuft” och med hänsyn till storlek och komplexibilitet av det aktuella projektet.” (Bilaga 5, (11) Thomas Farby, Volvo IT)

Som en vidare påbyggnad av föregående fråga blir således arbetsflödet väldigt mekaniskt eftersom det sker en mekanisk styrning där arbetarna förutsätts arbeta efter checklistorna och inte tvärtom.

”Givetvis blir det en mekanik när arbetet i stor grad styrs av metoden/checklistorna. Det är ett problem att hitta utmaningar till erfarna projektledare som måste få gå vidare till mer komplexa projekt för att finna utmaningar.” (Bilaga 5, (12) Thomas Farby, Volvo IT)

Avdelningen som avser behandla checklistorna tar tillhanda förslag på så sätt att slutanvändare ger feedback eller förslag (kanske även kritik) mot ett visst moment som sedan behandlas av den grupp som avser arbete med metodens utformande. Slut användarens åsikt tas alltid tillvara då det är en förutsättning för en god användbarhet.

”Slutanvändarnas åsikter mottas givetvis, men förändringar styrs i hög utsträckning av metodgrupperna.” (Bilaga 5, (13) Thomas Farby, Volvo IT)

Thomas Farby på Volvo IT hävdar att deras fokus ligger på att kontinuerligt granska och reflektera sin(a) checklista(or) vilket medför att dem vinklar såklart kritik mot sig själv för att identifiera eventuella brister och sedan se det ur ett lösningsperspektiv. Kontinuitetsprincipen bygger på att dem gör uppföljning av varje projekt.

”Givetvis finns brister i användandet, men det är i regel i mindre projekt. Projekt över ett visst värde skall annonseras och följas upp i ett project Dashboard och samtliga projekt där följer alla checklistor” (Bilaga 5, (14) Thomas Farby, Volvo IT).

Johan på Logica anser att deras utvecklare ofta blir involverade i formandet av deras checklistor. Då de har erfarenheter från de tidigare iterationerna och har därför en väldigt stor talan om vad de vill ha gjort i systemet. Johan Norman på Online i Lund påstår att de har ett liknande koncept där de sköter ändringen av checklistorna internt. Detta sker alltid i dialog med utvecklarna och checklistorna blir därför dynamiska.

”Ja, alltså det sker ju internt och det sker ju på de forumen som finns så att säga. Det sker i samtliga fall i dialog. Errm för att bara för att det ser ut att vara en bra idé så behöver det ju inte vara det. Därför ska man ju ha så mycket input som möjligt så att det blir bra beslut som man inte behöver backa på. För det kostar bara tid och pengar.” (Bilaga 3, (202), Johan Norman, Online i Lund)

”Vi får oftast vara med redan när checklistorna skrivs, dom skrivs i hand med iterationerna.. så vi har all möjlighet att framföra åsikter och komma med förslag” (Bilaga 4, (45), Johan, Logica Ab)

4.4 Användarstöd

Utbildning av användarna av systemen är något som informanterna anser centralt. Enligt Johan på Logica finns det användarmanualer som används för att utbilda användarna av systemet. Han påpekar också att det är viktigt att de går igenom grundligt så systemet används så smidigt som möjligt, när det handlar om stora system. Johan Norman på Online i Lund anser själv att Online i Lund är usla på det. Han säger dock detta med glimten i ögat. Men han kändes dock när han berättade om vilka problem de hade med utbildning. Men där finns samtidigt stöd för de användare som har problem då i form av helpdesks etc. Han hävdar dock att det ofta inte behövs då det ofta handlar om mindre implementationer så skall det vara glasklart hur det skall göras.

”Dom har nån användarmanual, men vet inte om dom går efter den på utbildningen eller om det bara är en generell genomgång. Jag hoppas den går igenom bra i alla fall :) Så systemen används så smidigt som möjligt. Känns som det är mycket viktigt när det är ett stort system, där man kan göra samma sak på ett par olika sätt.” (Bilaga 4, (21), Johan, Logica Ab)

”Utbildning är en mycket viktig punkt och i de projekt jag ansvarar för brukar jag planera detta när projektresurserna tillsätts” (Bilaga 5, (8) Thomas Farby, Volvo IT).

”Det är vi ju otroligt usla på” (Bilaga 3, (140), Johan Norman, Online i Lund)

”Öhh, vi har viss utbildning och vi kommunicerar ut vad vi gör för någonting i releaserna. Till samtliga användare. Ermm, och uppmanar alla att ställa frågor om det är någonting de är intresserad av eller någonting de inte förstår. Ermm, vi utvecklar, dels utvecklar vi rent interna grejer. Som då används av Online. Men så har vi ju system som används som satslösningar alltså software service som används av externa parter. Ermm, och där kommunicerar vi mer, men oftast så är det en viss människa som har begärt en viss funktion och känner att nu har jag fått den. Så systemen i sig skall vara självklara till just det. Annars får man ju skriva manualer” (Bilaga 3, (142), Johan Norman, Online i Lund)

Thomas Farby på Volvo IT poängterar att de använder sig av metoder för att utbilda slutanvändare Ex. på metod: *Train-the-trainer* som bygger på att de som är involverade i utbildning av slutanvändare skall även medverka i projektets gång för att bygga en djupare förståelse för sin roll som utbildare. Dessa utbildare blir även nyckelpersoner i användandet av systemet när det väl är dags att implementera och sätta lösningen i akt.

”Jag använder också metoden ”Train the Trainer”, dvs att de som skall utbilda end-users skall också medverka i projektet, känna till sin roll som utbildare av slut-användare och vara en potentiell key-user av systemet efter Go Live.” (Bilaga 5, (8) Thomas Farby, Volvo IT)

Utbildningsmomentet förstärks även eftersom checklistan fungerar som underlag för att förmedla data till projektledare som inte besitter samma kompetens/erfarenhet som en annan (i detta fall Thomas som har 10 års erfarenhet som IT-chef på Volvo IT). De checklistor som existerar agerar även som underlag och trygghet som främjar kommunikation mellan olika utvecklare (även intressenter).

”Det är också en trygghet för mig i de fall när man har en mindre erfaren projektledare. Oftast beordrar jag en review före ett gate-möte, och då ser en erfaren projektledare igenom projekt-status och upprättar ett reviewprotokoll.” (Bilaga 5, (6) Thomas Farby, Volvo IT)

4.5 Användande av checklistor

Johan på Logica påstår att de jobbar med en iterativ process där de arbetar med mindre checklistor. Checklistorna innehåller ofta testresultat sedan innan så de vet vad som är fel och vad som bör åtgärdas. Varje iteration innebär ett antal mindre moment eller ett större.

”Vi har en iterativ utveckling, så jag får en iteration i taget med vilka funktioner som skall vara klara tills nästa iteration, samt testresultat från föregående iterationer. Iterationerna är ganska små och består oftast av ett par mindre uppgifter eller en större. En iteration är oftast bara nån vecka” (Bilaga 4, (8), Johan, Logica Ab)

Detta ger de också mindre tid mellan de olika testningarna och medför att det inte blir några jättechecklistor som akut måste fixas. De olika intervjupersonerna var också överens om att checklistorna ger en klar struktur att arbeta efter vilket underlättar deras arbete.

”De ger mig en klar uppfattning vad som skall göras och vad som behöver fixas” (Bilaga 4, (11), Johan, Logica Ab)

”Det är definitivt en fördel att ha en metod och den kan man ju då, i detta fallet, kalla en checklista givetvis. Då vet vi ju hur vi jobbar vi vet i vilket ordning vi ska göra vissa grejer.” (Bilaga 3, (101), Johan Norman, Online i Lund)

4.6 Användbarhet i checklistor samt värde på kvalitet

På frågan hur pass viktigt kvaliteten på en produkt är kontra deadlines, svarar respondenterna rätt så lika. Alla anser att kvalitet väger tyngst och det poängteras även om att ett system redan är i drift och fungerar tillfredställande så är kvaliteten ännu viktigare. Johan Norman på Online i Lund understryker dock hur pass viktigt checklistor och metoder är för att en produkt skall bli så pass bra som möjligt.

”Det beror nog på produkten, I vårt fall i och med den är verksam inom sjukvården så är nog kvalitén viktigare än deadline. Plus att den föregående versionen är redan i drift.” (Bilaga 4, (16), Johan, Logica Ab)

”För att du ska ha kvalitet, måste du ha någon form av metod eller checklista(or). Det måste du ha och du kommer inte runt det.” (Bilaga 3, (124), Johan Norman, Online i Lund)

Då vi frågar hur pass viktig användbarhet är så ser svaren ganska olika ut. Då Johan på Logica inte har någon speciellt stor insikt på just användbarhet då han arbetar som utvecklare så följer han bara checklistorna men håller med om att användbarhet är väldigt viktigt för att höja kvaliteten på ett system.

”Men är ett rätt intressant ämne att skriva om. Då det känns ganska viktigt för att höja kvaliteten på en produkt?” (Bilaga 4, (38), Kenny Olofsson)

”Jo det har du rätt i” (Bilaga 4, (39), Johan, Logica Ab)

Johan Norman på Online går in lite mer i detalj på just användbarhet och kvalitet då de sysslar med SCRUM som enligt Avison, D. & Fitzgerald, G. (2006) är en väldigt användarcentrerad utvecklingsmetod och anpassas mer eller mindre efter användarens önskemål med hjälp av ”use-cases”. Samtidigt så arbetar Logica också med anpassade checklistor för just det aktuella projektet och inte med några fastställda checklistor som t.ex. Volvo IT använder. Johan Norman berättar också hur pass viktig användbarhet egentligen är där han säger att det är A och O för att ett system verkligen skall bli bra.

”Alltså det gör dem ju därför att det ska finnas ”use-cases” och det ska testas. Så finns det många olika delar av kvalitet, eftersom det är ett luddigt begrepp.” (Bilaga 3, (118), Johan Norman, Online i Lund)

”Vi har helt projektstyrda checklistor.” (Bilaga 4, (43), Johan, Logica Ab)

”Ja, användbarheten är ju a och o. Därför att det minskar resurserna för supportkostnader” (Bilaga 3, (163), Johan Norman, Online i Lund)

Johan Norman på Online hävdar att deras ”use-case” blir i direkt översättning checklistor för just användbarhet då dessa involverar punkter som kunden vill ha i sitt system. Han påstår

också att det finns användbarhetskrav på en högre nivå som alltid skall tillfredställas men ett "use-case" visar på en lägre nivå och han hävdar därför att det blir en sub-checklista för användbarhet.

"De use-casen ni har skulle man kunna säga att de blir er checklista för användbarhet?" (Bilaga 3, (221), Kenny Olofsson)

"Ja så kan man säga. Det blir en checklista eller ja, en sub-checklista för användbarhet." (Bilaga 3, (222), Johan Norman, Online i Lund)

Han betonar att kvalitet kan mätas på flera olika sätt. Ett företag kan leverera ett system som fungerar vilket i sig är kvalitet samtidigt som systemet inte alls är sammanhängande och inte ser alls bra ut ur användbarhets perspektiv.

"Och det är också en viktig användbarhets grej som också spelar in, inget formulär på det här programmet ser likadant ut. Det ser för jävligt ut alltihop. Även om man har missat den här användbarhetsdelen så har man fortfarande levererat kvalité." (Bilaga 3, (136), Johan Norman, Online i Lund)

4.7 Prioritering

Johan på Logica påpekar att det ibland inkommer checklistor med kritiska punkter med akuta buggar. Dessa är därför väldigt viktiga att åtgärda snabbt och då styr checklistorna utvecklarna väldigt mycket. I annat fall så har utvecklarna ganska stor kontroll över hur checklistorna skall användas.

"Klart ibland får man en checklista med kritiska buggar i ett sidoprojekt, och då måste det göras kalassnabbt" (Bilaga 4, (55), Johan, Logica Ab)

Johan Norman på Online i Lund påstår att systemen aldrig blir färdiga och är därför i konstant behov av förvaltning. Nya parametrar dyker upp samt att verksamheten konstant förändras. Detta medföljer att systemet också måste ändras för att anpassas efter de nya behoven inom verksamheten. Johan på Logica anser att de arbetar på ett likande sett då deras utveckling är iterativ och detta kopplat med systemförvaltning betyder att det kommer upp nya punkter i den iterativa utvecklingen hela tiden.

"Ja, alltså det händer ju hela tiden. Systemen är levande vi har ju inget färdigt system. De kommer ju alltid att förändra sig eftersom verksamheten förändrar sig. Man kanske hittar en annan parameter i hur verksamheten funkar eller en förändring och nu måste vi jobba på ett annat sätt." (Bilaga 3, (218), Johan Norman, Online i Lund)

"Vi har en iterativ utveckling, så jag får en iteration i taget med vilka funktioner som skall vara klara tills nästa iteration, samt testresultat från föregående iterationer. Iterationerna är ganska små och består oftast av ett par mindre uppgifter eller en större. En iteration är oftast bara nån vecka" (Bilaga 4, (8), Johan, Logica Ab)

4.8 Beredning och brister i användandet av checklistor

Vad det gäller bitar som Logica är sämre på så säger Johan att de oftast bara har klassiska fel, så som dåligt kommenterad kod, dålig felhantering samt osammanhängande kod då det inte finns någon kodstandard utan alla kör med sina personliga stilar. Då Johan Norman på Online i Lund tidigare nämnt att deras största brist är sina utbildningar så går han in lite djupare på när vi frågar vilka dåliga sidor de har. Han ger också diverse lösningsförslag så som förändringar i processen och införande av helpdeskar för det aktuella systemet.

”Hehe du menar så, Det gamla vanliga. -Dåligt kommenterad kod -Dålig felhantering -Lite osammanhängande kodstandard då vi alla köra rätt personliga stilar” (Bilaga 4, (63), Johan, Logica Ab)

”De felen vi upptäcker, får ju åtgärdas genom att ändra processen helt enkelt. Genom att förändra processen så åtgärdas problem. Man kan införa helpdeskar man kan införa öhh bättre release dokumentation till exempel. Om det är det som behövs, ermm man kan också rikta release dokumentationen alltså ge avdelningen mer info om den utvärderingen som görs nu. Här genomförs ett nytt sätt för er att arbeta. Ja, då kanske man skall skriva en manual som bara ges till just dem. Vi har infört helpdesk för vissa externa projekt, så de mailar samma adress ringer samma nummer etc. Vi har uppdateringar alltså gjort bättre dokumentationer som ges till vissa avdelningar. För att vi upptäcker att de begär en funktion utan att upptäcka att där finns en.” (Bilaga 3, (214), Johan Norman, Online i Lund)

5 Diskussion och analys

I detta kapitel kommer vi föra en diskussion med hjälp av de empiriska finningar vi har gjort gentemot de teorier och den litteratur vi har funnit. Kapitlet är uppdelat på det viset att den följer rubrikerna i teori och analys för att ge en klar bild över den röda tråden i uppsatsen

5.1 Ändringshantering

Ändringshantering är något som kan ske på ett flertal olika sätt. Nordström (2005) hävdar att det finns ett flertal olika sorters sätt att angripa en ändring på. Dessa är att genomföra, avveckla, anpassa organisation och slutligen använda systemet på fel sätt. Då vi endast intervjuat personer som arbetar med att genomföra ändringarna och inget av de andra alternativen, så är det här genomföranden vi kommer fokusera på. Volvo IT har valt ett mer mekaniskt perspektiv på hur deras genomförande går tillväga. De har checklistor som är mer eller mindre statiska som utvecklarna skall följa. Logicas utvecklare jobbar som nämnt i analysen iterativt vilket gör att de har stor erfarenhet av systemet och vet hur det fungerar och ser ut och de får därför vara med att anpassa sina checklistor inför varje iteration. Online i Lund arbetar med SCRUM vilket är en användarbaserad utvecklingsmetod som anpassas i stort sett till 100 % efter kontexten. Dessa hanterar därför genomförande fasen väldigt flexibelt med hjälp av sina "use-cases".

5.2 Prioritering

Nordström (2005) påstår att en prioritering görs inom projektet. Denna görs iterativt under utvecklingen vilket vi kan se är ett tillvägagångssätt som Logica använder. De har en iterativ utveckling som gör att prioriteringarna kommer finslipas allt eftersom tiden går. Detta i sig tror vi är ett bra sätt att utveckla då stora missar oftast upptäcks och kan rättas till under nästa iteration. Johan Norman på Online i Lund hävdar också att ett system alltid är i rullning. Det blir helt enkelt aldrig färdigt då en verksamhet alltid måste anpassa sig till verkligheten. Detta speglar sig sedan ner i systemet där nya funktioner behövs läggas till eller företaget helt enkelt vill ha ett nytt sätt att arbeta.

"Vi har en iterativ utveckling, så jag får en iteration i taget med vilka funktioner som skall vara klara tills nästa iteration" (Bilaga 4, (8), Johan, Logica Ab)

”Systemen är levande vi har ju inget färdigt system. De kommer ju alltid att förändra sig eftersom verksamheten förändrar sig. Man kanske hittar en annan parameter i hur verksamheten funkar eller en förändring och nu måste vi jobba på ett annat sätt.” (Bilaga 3, (218), Johan Norman, Online i Lund)

Nordström (2005) påstår att de ändringar som prioriteras som akuta ändringar fixas till snabbt då ett akut fel oftast drar ner effektiviteten och begränsar användarnas förmåga att arbeta i systemet. Hon nämner också något som hon kallar för normala ändringar vilket är det vi har stött på i vår empiri. Då de akuta ändringarna oftast sker lokalt på företaget av någon som är insatt i systemet på plats. Den enda av informanterna som nämner kritiska buggar är Johan på Logica som betonar att checklistorna styr utvecklarna väldigt mycket här då de i övriga fall har väldigt fria tyglar. Anledningen till detta är just att ändringen måste ske snabbt för att få systemet i drift så snabbt som möjligt.

”Klart ibland får man en checklista med kritiska buggar i ett sidoprojekt, och då måste det göras kalassnabbt” (Bilaga 4, (55), Johan, Logica Ab)

Detta är ett klokt sätt att resonera på då det är viktigt att bibehålla effektivitet samt att systemet kan vara i drift så fort som möjligt igen. Hade utvecklarna inte haft färdiga checklistor för det här kan det lätt bli så att de fastnar under utformandet av en checklista då vi kan se i analysen vilken tid det tar att faktiskt utforma en checklista som är anpassad till just ett specifikt tillfälle.

5.3 Systemförvaltning och utformandet av checklistor

Som Brandt (2005) säger så är systemförvaltning en process om hur utvecklare skall genomföra en vidareutveckling av ett system. Då rapporten skall ha perspektivet checklistor har vi försökt komma fram till hur våra intervjupersoner går tillväga när de skapar sina checklistor. Vi går sedan vidare med hur användandet av checklistorna går till i ett senare kapitel. Dessvärre finns det inga studier som gjort inom just checklistor vad vi har kunnat hitta och vi har därför inga större teoripunkter att stödja oss på. Thomas Farby säger att Volvo IT har en hel avdelning som bara arbetar med metoder och just utformandet av deras checklistor. Han hävdar att checklistor är att använda för att få struktur på arbetet men de bör dock användas med aktsamhet för att inte göra arbetet allt för omfattande. Checklistorna måste därför sättas i perspektiv till vilken kontext de skall användas i. Detta är något som Johan Norman på Online i Lund säger att de också och Johan på Logica följer samma spår. Fast de tacklar utformandet på olika sätt med hjälp av olika metoder så hävdar alla tre att det är viktigt att utforma en checklista utefter den kontext den faktiskt skall användas i.

5.4 Användande av checklistor

Användandet av checklistor med hänsyn till vår analys lyfter fram att kontexten avgör användandet av checklistor. Beroende på vilket företag vi intervjuat har vi fått olika svar som vi kan ta till oss. Logica & Volvo IT var relativt lika avseende användandet av checklista som metod, Online i Lund däremot var mer flexibelt orienterad, trots att de anser att checklistor är viktigt och är bra till olika avseenden.

”De ger mig en klar uppfattning vad som skall göras och vad som behöver fixas” (Bilaga 4, (11), Johan, Logica Ab)

”Det är definitivt en fördel att ha en metod och den kan man ju då, i detta fallet, kalla en checklista givetvis. Då vet vi ju hur vi jobbar vi vet i vilket ordning vi ska göra vissa grejer.” (Bilaga 3, (101), Johan Norman, Online i Lund)

”I de flesta fall är det givetvis en fördel att använda checklistor, men de måste alltid användas med förstånd för att inte skapa en onödig administration/onödiga kostnader.” (Bilaga 5, (7), Thomas Farby, Volvo IT).

Som bl.a. Thomas Farby på Volvo IT nämner i, stycket ovan så ser vi att det måste användas med sunt förnuft. Vi anser att en kontextbedömning av problemet eller utvecklingsprocessen borde göras. Avison, & Fitzgerald, G. (2006) tar upp att definitionen av en lättviktsmetod (agile som bl.a. Online i Lund jobbar med) är att den är lätt föränderlig och kan/behöver förändras beroende på utvecklingsmiljön och företagets kontext. Detta avser då systemutveckling. tunga, formaliserade utvecklingsmetoder har dock sina fördelar i frågan om komplexitet då deras ståndpunkt är att det finns en iteration och ständig upprepning som gör att även de mindre erfarna har möjligheter. (Fitzgerald, et al., 2002)

Däremot är det lätt att hamna efter i en bransch som är ständigt föränderlig (som IT branschen) därför förespråkas lättviktsmetoder som är gjorda för denna typ av förändring. Hänsynstagande till att agile kräver mycket mer av den enskilda utvecklaren. (Avison & Fitzgerald, G. (2006)

Online i Lund har en mindre integrerad enhet på färre antal personer/projekt. Då Online är det enda i vår studie som använder sig av agile (lättnviktsmetoder) som utvecklingsmetod så existerar det en koppling mellan teorin och fallen. Agile kräver mer av utvecklarna och anpassas till projektets kontext därför blir den en väldigt odefinierad lösning som passar bra in på Online i Lund då de är en mindre enhet som med ständig interaktion med varandra (och deras utvecklingssystem) kan finna sin ”skräddarsydd” lösning (Fitzgerald, et al., 2002). Det är förståeligt att större utvecklingsenheter som Volvo IT och Logica förlitar sig på formaliserade systemutvecklingsmetoder och checklistor som gör att de kan följa varje steg och säkerställa att det de gjort verkligen blivit rätt. Med dess uppföljning och dokumentering blir det även lättare att ta del av förvaltningsarbete. (Avison & Fitzgerald, G. 2006).

”Det är också en trygghet för mig i de fall när man har en mindre erfaren projektledare. Oftast beordrar jag en review före ett gate-möte, och då ser en erfaren projektledare igenom

projekt-status och upprättar ett reviewprotokoll.” (Bilaga 5, (6) Thomas Farby, Volvo IT)

5.5 Användbarhet i checklistor samt värde på kvalitet

I själva intervjuerna så har vi inte tagit upp termen ”värde” (eller value) alls. Detta eftersom det är den bakomliggande tanken i informanternas upplevelser av checklistor vi är ute efter och detta i sig går att knyta till vårt teori avsnitt om värde.

Även i samband med kvalitet så spelar kontexten roll. Produkten/tjänsten kan mycket väl vara avgörande för dispositionen därför att i vissa fall prioriteras snabba ”mindre kvalitativa” lösningar och i andra fall så är kvalitet vitalt.

”Det beror nog på produkten, I vårt fall i och med att den är verksam inom sjukvården så är nog kvaliteten viktigare än deadline. Plus att den föregående versionen är redan i drift.” (Bilaga 4, (16), Johan, Logica Ab)

Förhållandet mellan perceived benefits och perceived sacrifice avseende värde på Logicas produkt är i denna kontext större och komplex. Med det hävdar vi att det läggs mycket mer energi på att analysera förhållandet mellan de olika attributen för att få kvaliteten så bra som möjligt. Förhållandet, i Logica ABs fall, innebär att de upplevelser utvecklarna har kring värde måste hanteras. Hanteringen skall ske på ett sätt som gör att den mängd energi användarna (i sjukhuskontexten) antas lägga ner för att förstå det nya systemet ska minimeras. På ena sidan har vi attribut (Observera att dessa attribut vi nämner bygger på själva syftet med integration av IT inom sjukvården) som avser hjälpa personalen inom sjukvården hantera ett visst syfte, detta existerar i perceived benefits. På andra sidan har vi personalens egen insats till att förstå och korrekt använda systemet, detta hanteras på perceived sacrifice. Innebörden för utvecklarna på Logica AB blir att värdera balansen mellan attributen. Monroe (1991) hävdar i sitt förhållande att målet är att reducera perceived sacrifice genom att hantera dess attribut. Inom sjukvården handskas med människor och ett system/produkt/tjänst som inte fungerar som det avsett göra kan ha katastrofala konsekvenser. Därmed vill vi betona att även kontexten kan föra med sig en aspekt på hur värde kan hanteras. Vi kan även se i Preece, et al., (2002) studie att desto komplexare kontext checklistor appliceras på, desto svårare blir det att få grepp om det upplevda värdet av dess användning och sätta ramar för *hur* man ska gå tillväga avseende användbarhetsaspekterna.

Vidare tycker vi att man kan koppla samma förhållande perceived benefits och perceived sacrifice (med hänsyn till ovanstående stycke), av Monroe (1991) med Nielsen (1993) aspekter på användbarhet då de båda handlar om att bygga upp en form av ”trygghet” för slutanvändaren i olika avseende. Det vore intressant att se om Logica AB hanterat IT integration inom sjukvården på ett annorlunda sett om man använt de attribut kring användbarhet som Nielsen (1993) betonar. Applicering av dessa attribut på utvecklarnas upplevda värde inom sjukvården kan lyfta fram en tydligare balans mellan förhållandet perceived benefits och perceived sacrifice vill vi påstå.

I vårt fall så kan båda teorier appliceras på utvecklarnas (våra informanternas) tankar och deras arbetsprocess då t.ex. Johan Norman, Online Lund, hävdar att en checklista *är* ett måste för att uppnå användbarhet och god kvalitet. Vi vill därefter påpeka att även aspekterna kring värde

kan vara ett instrument för kvalitet.

”För att du ska ha kvalitet, måste du ha någon form av metod eller checklista(or). Det måste du ha och du kommer inte runt det.” (Bilaga 3, (124), Johan Norman, Online i Lund)

5.6 Beredning och brister i användandet av checklistor

Vad det gäller brister i användandet av checklistor så säger Johan på Logica en ganska intressant sak. Han säger att de brister de ser i sitt användande ofta är dåligt dokumenterad kod och inga kodstandarder. Detta är något som Nordström (2005) också beskriver som ett vanligt fenomen. Det kan då vara en idé att införa en kodstandard och också en standard för dokumentering då det uppenbarligen är ett väldigt förekommande fenomen så kan tacklas på ett bra sätt. Skriver alla likadan kod och är duktiga på att kommentera den så blir det också lättare för andra utvecklare inom projektet att läsa och förstå den. En standard inom företaget kan då skrivas ner i en form utav checklist som används för det här.

5.7 Användarstöd

Avison & Fitzgerald, G. (2006) hävdar att utbildningen av användarna skall ske redan innan implementeringen av själva systemet genomförs, för att få ut optimalt av implementeringen så fort som möjligt. Just utbildning av användare är något våra intervjupersoner har ställt sig väldigt olika till.

Som Nordström (2005) säger så göms ofta utbildningen i bakgrunden av allt annat. Utbildning kostar ofta en stor summa pengar och kan därför lätt glömmas bort. Det är något Online i Lund verkar ha gjort. De är medvetna om att utbildning av användarna är ytterst viktigt då det sparar resurser i längden men är ett problem de valt att förbise för tillfället och faktiskt påpekar att det är något Johan Norman anser de själva vara dåliga på. Detta är något som givetvis inte är bra då han motsäger sig själv då han betonar att utbildning är väldigt viktigt.

Nordström (2005) påstår också att helpdeskar bör införas då dessa underlättar supporten för systemet. Dock så kan en gedigen utbildning av användare minska trycket radikalt på helpdesken. Logica har valt att vad vi har sett inte använda någon helpdesk. De använder istället användarmanualer som går igenom på utbildningen. Nielsen, (1993) betonar *learnability* som hur snabbt användarna förstår och kan sätta igång med sitt arbete igen. *Learnability* blir då en parallell till vad Johan avser användarmanualerna existera för. Om *learnability*, enligt Nielsen (1993), hanteras på korrekt sätt, anser vi att även trycket på helpdeskar i Nordströms (2005) avseende minskar då en utbildning av användare existerar. Då Johan på Logica inte har någon större insikt i just utbildning av användarna har han fortfarande något att säga om hur pass viktigt han anser det är.

”Jag hoppas den går igenom bra i alla fall :) Så systemen används så smidigt som möjligt.” Känns som det är mycket viktigt när det är ett stort system, där man kan göra samma sak på ett par olika sätt.” (Bilaga 4, (21), Johan, Logica Ab)

Just när en uppgift i ett system kan utföras på flera olika sätt med hjälp av genvägar etc. tror vi att det är ytterst viktigt att användarna får ett klart perspektiv med hjälp utav en utbildning på hur användare bör gå tillväga för att arbeta så effektivt som möjligt. Johan Norman på Online i Lund påstår att då de vidareutvecklar redan befintliga system så skall systemen i sig vara självklara. De är också måna om att användarna skall ställa frågor om det är något de undrar över. Om något svårt implementeras så väljer de också att lösa detta via manualer precis som Logica. De tillmötesgår denna oerhört viktiga process av utvecklingen med en klackspark känns det som. Det går inte att förbise något som utbildning, vilket är en oerhört viktig del för att få ut maximal effektivitet av systemet i fråga. Volvo IT följer den teorin Avison & Fitzgerald, G. (2006) förespråkar. Där de försöker initiera utbildningen så fort som möjligt.

”Utbildning är en mycket viktig punkt och i de projekt jag ansvarar för brukar jag planera detta när projektresurserna tillsätts” (Bilaga 5, (8) Thomas Farby, Volvo IT).

5.8 Studiens brister

Studiens replikerbarhet kan ses som en brist. Vi anser dock att metoden i studien är replikerbar då vi försökt att i minsta detalj beskriva vårt tillvägagångssätt i metodkapitlet. Vi underlättar också replikering då vi bifogat vår intervjuguide. Det är dessvärre omöjligt att säga att resultatet kommer bli likadant.

Vi är medvetna om att studien kan kännas liten och kan därför vara svår att generalisera. Vi har försökt ta detta i anspråk och därför genomfört djupintervjuer med två av våra informanter för att gräva på djupet istället för bredden.

Vi har bristande empiri då vi bara har tre informanter och av dessa tre var bara två utarbetade ansikte till ansikte. Vi hade för avsikt att utföra alla tre intervjuerna på samma sätt men slog mot många motgångar vid sökandet efter informanter pga. bristande intresse från företagen vi ringt men även att begreppet checklistor var för många företag okänt. Detta gav upphov till att en av tre intervjuer, blev elektroniskt utarbetad då det var omöjligt för oss (av geografiska skäl) att utarbeta ansikte till ansikte tilläggas skall göras att föreberedelser inför intervjun gjordes via flertalet telefonsamtal. Vi är dock hjärtligt tacksamma över att de informanter vi funnit har ställt upp och uppfyllt alla krav/frågor vi behövt hjälp med att besvara.

6 Slutsatser

I kapitlet behandlar vi de slutsatser vi funnit i vår diskussion som varit relevanta till vår frågeställning samt annat som kommit fram under studiens gång. Vi ger också förslag på fortsatta studier inom området.

Vår forskningsfråga löd:

I vilket avseende fungerar checklistor som instrument för förbättrad användbarhet/kvalitet vid systemförvaltning?

Delfråga: På vilket sätt värderar utvecklarna hanteringen av checklistor avseende användbarhet och kvalitet förhållande till det som skall utvecklas.

Utifrån våra tre intervjuer samt har vi kommit fram till följande: *först* och *främst* kan vi nämna att användandet av checklistor ansågs som ett fördelaktigt medel att använda överhuvudtaget! Alla informanter var överens om att checklistor är ett positivt verktyg vid utvecklingsarbete. Som *andra* punkt till detta vill vi nämna att det fanns olika avseende där checklistors positivitet speglades tydligare. Framför allt avseende strategi och struktur. T.ex. avseende iteration i arbetet, där checklisten agerar som manifest för de punkter som anses vitala i utvecklingsprocessen och samt i vilken ordning de kommer i. Ordning har att göra med avseendet struktur då våra informanter anser att checklistor fungerar som underlag för *hur* och *när* olika processer skall hanteras.

Checklistorna i sig hanterar inte utvecklarnas upplevda värde, avseende användbarhet och kvalitet direkt, utan mer *indirekt* har vi kommit fram till. Detta ser vi då vår teori, lika så vår empiri anser att felhantering (eller bugghantering som Johan på Logica kallar det) är ett moment avseende användbarhet. Vid ett sådant inträffande agerar checklistor som ett verktyg för att snabbt och effektivt åtgärda problemet.

Vi har även kommit fram till att användare av systemet är inte dem ända som blir berörda av *learnability* som Nielsen (1993) förespråkar om. Thomas Farby på Volvo IT anser att checklisten har hjälpt honom i inlärningsarbetet med de mindre erfarna projektdeltagarna. Checklisten avseende inlärningsprocessen blir då ett verktyg som förenklar utvecklingsprocessen för projektdeltagare som inte besitter samma kompetens och erfarenhet som andra. Användare av checklistor ska dock inte bli för rigida i sin användning då det kan leda till bl.a. onödiga administrativa kostnader så åter här spelar kontexten roll för användandet hävdar Thomas Farby på Volvo IT lika så Johan Norman på Online i Lund.

Då komplexiteten i kontexten bidrar till att olika prioriteringar görs och att implementeringen av checklistan blir svårare, blir även vetskapen kring användandet av checklistor komplext. Därför så finns det en formalitet i checklistor som snabbt och effektivt löser diverse problem. Detta beror såklart på produkten och vilken kontext den appliceras på.

Prioritering är något som växt fram utanför vår egentliga frågeställning som framstått som väldigt viktigt av val om hur en process faktiskt skall genomföras och vilken flexibilitet checklistorna skall ha. Vid tillfällen kommer det in kritiska buggar inom ett system då blir genast checklistorna väldigt formaliserade då detta måste hanteras fort. Lämnas det istället lite flexibilitet i de här fallen så kan det lätt dra ut på tiden vilket leder till förseningar för de som använder systemet. Då iteration förekommer så blir prioriteringen lättare. Förståelsen ökar för det faktiska fallet då utvecklarna har sett det några gånger tidigare. De vet vad som måste göras fort och vilket som kan prioriteras lite lägre.

Användarstöd är något som oftast anses väldigt centralt inom just systemförvaltning. Dock har vi inte hittat någon empiri som stödjer att det finns några checklistor över huvudtaget för just den här biten. Detta är väldigt märkligt då det kommer fram att det faktiskt sparar pengar och ökar effektiviteten hos användarna om de vet hur de faktiskt ska använda systemet. Logica visar viss indikation på att där finns ett strukturerat sätt för att utbilda användarna men det fanns inget konkret svar om där fanns checklistor som hjälpte till för att underlätta detta. Brister i det faktiska användandet av checklistor under utveckling är att koden i systemen ofta är osammanhängande och dåligt dokumenterad. Kodstandard och dokumenteringsmallar är därför något som bör inkluderas i checklistor avseende utveckling.

Bilagor

Bilaga 1 - Exempel på checklista

Checklista för kontroll av kravspecifikation ur Andersen, E.S. (1991) s. 454-455

Generellt

1. Fullständighet: Är alla kraven med, eller saknas det något krav för att man ska ha täckt de aktuella förändringsbehoven?
2. Korrekthet: Är beskrivningen av kraven utan fel?
3. Exakthet, otvetydighet, tydlighet: Är alla kraven entydiga och exakta, eller finns det oklarheter?
4. Konsistens: Är alla kraven i överensstämmelse med varandra, eller finns det oförenligheter bland dem?
5. Relevans: Är alla kraven helt nödvändiga för att man ska kunna täcka de aktuella förändringsbehoven? Är det möjligt att nyansera hur nödvändiga kraven är (absolut nödvändiga, önskvärda, tas med om kostnadsfritt), och är det i så fall gjort?
6. Kontrollbarhet: Är det möjligt att på objektivet sätt bedöma om ett krav är uppfyllt eller ej, när informationssystemet är implementerat?
7. Spårbarhet: Är det möjligt att spåra varje krav till förhållanden som behandlas i förändringsanalysen, eller i analysen av informationssystemet? Varför har kravet tagits med?
8. Genomförbarhet: Är alla krav möjliga att genomföra, det vill säga: klarar man utifrån den aktuella ekonomin, bemanningen och kunskapen att realisera alla krav?
9. Utan utformningsmässiga beslut: Är kraven fria från beslut eller utformning? Om det i ett krav fattas ett utformningsmässigt beslut, är då anledningen acceptabel?
10. Enkelt att göra ändringar: Är kravet formulerat på ett sådant sätt att det är lätt göra ändringar utan att det får konsekvenser av andra krav?
11. Nyta: Är kravspecifikationen en bra utgångspunkt för det fortsatta arbetet? Är den detaljerad nog?

Syftet med informationssystemet?

1. Precision: Är målbeskrivningen exakt?
2. Mätbarhet: Är det möjligt att göra en kvantitativ beräkning av vinsterna med informationssystemet?

En överordnad beskrivning av informationssystemet

1. Sammanhang: Är systemets koppling till andra system beskriven?
2. Krav på andra system: är kraven på de andra systemen beskrivna? Har man angett vad som sker om dessa inte är uppfyllda?
3. Krav på systemet: Är de krav som systemet måste anpassas till beskrivna (till exempel operativsystem, datakommunikationsprotokoll, maskinutrustning och annat)?
4. Intressenterna: Finns det en beskrivning av systemets intressenter?

Organisatoriska och personalmässiga förutsättningar

1. Pso-utveckling: Har man beskrivit vad som måste ske i frågan om personal- och organisationsutveckling, parallellt med systemutvecklingen? Vilka blir konsekvenserna om en sådan utveckling inte sker?

Funktionerna i informationssystemet

1. Fullständighet: finns det en komplett lista över systemets funktioner?
2. Kvaliteten på beskrivning: Är beskrivning av varje funktion exakt, otvetydig och detaljerad nog?
3. Enbart funktionsbeskrivning: Är beskrivningen av funktionen fri från egenskapskrav?

Informationssystemets allmänna egenskaper

1. Precision: Är egenskaperna exakt beskrivna?
2. Mätbarhet: Är egenskaperna mätbara, och är kraven kvantitativt angivna?
3. Prioritering: Är de kritiska egenskaperna, det vill säga egenskaper som det är helt nödvändigt att uppfylla, särskilt omnämnda?
4. Vanliga egenskaper: Är dessa vanliga egenskaper med?
 - åtkomst?
 - användarvänlighet?
 - pålitlighet?
 - säkerhet?
 - utvidgningsmöjligheter?
 - underhållsmöjligheter?
 - flyttbarhet?
 - om inte, varför?
5. Användargränssnitt: Är användargränssnittet beskrivet?

Funktionernas egenskaper

1. Fullständighet: Är funktionernas speciella egenskaper beskrivna?
2. Konsistens: Överensstämmer egenskapskraven för den enskilde funktionen och informationssystemet?
3. Vanliga egenskaper: Är dessa vanliga egenskaper med?
 - frekvens och spridning av rapporter?
 - svarstider (vid interaktiv databehandling)
 - kapaciteter (vilka belastningar funktionerna klarar)?

Manuella funktioner

1. Fullständighet: Är kraven på de manuella funktionerna beskrivna?

Dokumentation

1. Fullständighet; är kraven på följande dokumentation med:
 - systemdokumentationen?
 - beskrivningen av den tekniska lösningen?
 - användardokumentationen?
 - driftsdokumentationen
2. Enkelhet i fråga om åtkomst: Har man refererat till alla dokument och rapporter så att det är lätt att finna dem?

Utbildning

1. Fullständighet: Är kraven på utbildning beskrivna?

Bilaga 2 - Intervjuguide

Detta är intervjuguiden som vi använde vid e-intervjun. Den intervjuguide vi använt oss av vid de övriga intervjuerna har samma frågor men har inte innehållit presentationen, slutkommentarer och kontaktinformation.

Datum & Tid:

Informant:

Företag/Organisation:

Presentation

Vi är två studenter på Lunds Universitet. Vi studerar på Holger Crafoords Ekonomisentrum, där vi tillhör institutionen för Informatik. Vår kandidatexamen/C-uppsats är vår agenda.

Syfte med intervjun:

Vi håller på att undersöka de checklistor som används som verktyg vid uppdatering av era befintliga system. Vi vill se om det finns en koppling mellan användandet av just dessa checklistor, som verktyg, och användbarhet. Användbarheten är i detta fall hur ni (som utvecklare) anser att checklistorna hjälper er säkerställa att de kraven som ställs verkligen uppfylls. Checklistor kan även vara de ”punkter” eller ”kriterier” ni följer i en t.ex. metod eller dylikt.

Vi är inga experter inom området så därför är det möjligt att vissa frågor är svåra att svara på eller anses vara ”luddiga”. Vi hoppas att du har förståelse för detta och därför ger oss så utförliga svar som du kan. Du får gärna uttrycka mer än bara svaret på frågan för att hjälpa oss förstå vad du menar. Vissa frågor kanske kommer bli dubbelletter eftersom ett svar kan vara svar på fler än en fråga. Då är det bara att hänvisa till tidigare svar.

Dina svar kommer hjälpa oss fastställa en empiri i vår undersökning och vi vill därför förtydliggöra att du får såklart hoppa över frågor som du inte är bekväm att svara på och även om du önskar vara anonym. Om du önskas vara anonym så är det bara att skriva till det längst nere i guiden. Nedan följer frågorna.

Tack på förhand!

OBS! Du får gärna svara rakt i detta dokument sedan, spara, och skicka tillbaka!

INTRODUKTIONS FRÅGOR

- Vad är dina erfarenheter inom systemutveckling?
- Vad är din roll/befattning inom företaget?
- Hur länge har du arbetat i organisationen?
- Vad är dina arbetsuppgifter? (konkret)

CHECKLISTOR/METOD

*Hur ser checklistorna ut? Beskriv (om vi inte får se en eller få en bifoga)

*Hur arbetar ni med checklistorna?

*På vilket sätt värdesätter du checklistorna som verktyg för förenklat utvecklingsarbete?

*Vilken är den, anser du, största fördelen med att använda checklistor som ett verktyg i utvecklingsprocessen?

*Sätts checklistorna åt sidan ibland och punkter hoppas över?

*På vilket sätt tycker du checklistorna fungerar som verktyg för att säkerställa kvaliteten på det ni utvecklar?

*I vilka avseenden/faktorer skulle du anse att checklistor fungerar som ett verktyg för förbättrad kvalitet på just användbarhet? Finns där några speciella ”områden” eller ”faktorer” (ex, struktur, förändring, utveckling etc.) där checklistor på något sätt visat sig vara av fördel att använda än att inte göra det?

ORGANISATION & UTBILDNING

*Tar checklistorna upp någon sorts utbildning av användarna som skall använda det uppdaterade systemet i så fall, hur viktigt tycker du utbildning av användarna är?

*Vem är ansvarig för checklistornas utformande? Finns där en specifik roll för det arbetet? I så fall vilken.

ANVÄNDBARHET

*På vilket sätt värdesätter du användbarhetsaspekterna i checklistorna?

* Används checklisten(orna) som regelgrund för organisationen eller anpassas de till förändringar i externa faktorer (förändringar i marknaden, organisationen etc.)?

* Blir arbetssättet mer mekaniskt tack vare checklistorna? eller behåller man en dynamik i arbetsflödet? På vilket sätt i så fall?

* Hur mottogs förslag på förändringar i checklistornas "punkter" (om så existerar) och vem ställer dessa förslag? Är det bara utvecklarna eller mottogs slutanvändares åsikter också?

* Har ni identifierat någon form av brist i användandet, hur åtgärdas de? (om)

ÖVRIGA KOMMENTARER:

Ett stort tack för din medverkan!

Önskas anonymitet:

Slutkommentar

Vi tackar så mycket för din tid! Dina svar kommer att bidra till kunskap inom informatik!

Har du frågor eller dylikt angående uppsatsen är det bara att fråga.

Önskas det en kopia av intervjuens utformning i uppsatsen (som är i stort sätt samma som denna guide) så får du så klart det!

Om Ni önskar att få ta del av den slutgiltiga uppsatsen så skickar vi gärna över en kopia till er!

Åter, ett stort tack ska Ni ha! //

Sasan Zanganeh

Kenny Olofsson

Kontakt:

Sasan_Zanganeh@hotmail.com

0737 29 43 41

Kenny_monn3@hotmail.com

0701 46 19 5

Bilaga 3 - Intervju med Johan Norman, Online i Lund**Datum & Tid:** 3/12 – 08, kl 11.00 – 11.50.**Plats:** Online, Lund.**Informant:** Johan Norman, IT-Chef på Online, Lund.

Transkriberingen börjar! På vänster kolumnen ser du siffran som motsvarar vilken mening vi är på i intervjun, 2 kolumnen (från vänster) ser man namnet på personen som talar och den resterande kolumnen representerar vad som har sagts.

1	Sasan	Vi tänkte bara börja med en kort introduktion, har du satt igång miken?
2	Kenny	mm. Det har jag.
3	Sasan	Jättebra! Då tänkte jag bara börja med att fråga vad din tidigare erfarenhet, av systemutveckling, har varit? Berätta lite om det.
4	Johan	Mina tidigare erfarenheter?
5	Sasan	Aa... Eller erfarenheter överhuvudtaget skulle jag vilja säga. Berätta om...Du kan börja, istället att berätta, vad är din roll inom denna organisationen?
6	Johan	I dagsläget är jag IT-Chef.
7	Sasan	IT-chef. (sade han instämmande)
8	Johan	Men har bakgrund i systemutveckling ehm, som sträcker sig från och till. Så nu är vi ett par år fram då.
9	Sasan	Om jag förstått dig rätt så har du också läst på Lunds? Informatik.
10	Johan	Jajamen! Några poäng har jag läst ;) (sade han med *glimten i ögat*).
11	Sasan	Några poäng har du läst ☺. Se där. (sade han instämmande).
12	Kenny	Hehehe..
13	Johan	Min uppsats finns nog på biblioteket där uppe fortfarande och handlade om internethandling och skrevs 98 skulle jag tro.
14	Sasan	Den kan man få tag på i "OLLE" då? Gå in där och titta eller? ;) (sade han lite skämtsamt).
15	Johan	Jajamen! ;) (sade han med åter, *glimten i ögat*).
16	Sasan	Härligt att höra ☺ (sade han glatt). Skulle du, bara kort,

		kunna berätta om dina arbetsuppgifter konkret?
17	Johan	Mina arbetsuppgifter idag är, förutom systemförvaltning alltså infrastruktur förvaltning, så är det också projektledning och arkitektur i de systemutvecklingsprojekten vi har.
18	Sasan	Mm. Lite nyfikenhet då från mig, vad gör man egentligen som IT-chef?
19	Johan	Frågan är nog "Vad gör jag <u>inte</u> "? ;) (sade han lite skämtsamt)
20	Kenny & Sasan	Haha! ☺
21	Johan	Nä men det är mycket formulering, och det är mycket strategi som vart ska vi, med våra system. Vart ska vi med vår infrastruktur, vilka tekniker ska vi använda och vilka teknologier ska vi använda. Vad ska vi INTE göra. Och det är mycket prioriteringsarbete. Vad ska vi göra nu, vad ska vi INTE göra nu. Vad ska vi göra lite senare och vad ska vi göra MYCKET senare. Vad ska vi försöka undvika att göra alls och så har man begränsat med resurser.
22	Kenny	Mm.
23	Johan	Har man INTE begränsat med resurser, så har man för mycket folk.
24	Sasan	Nä ok!
25	Johan	Du kan nästan aldrig ha precis med lagom med resurser för då har man gjort någonting fel, antingen för lite krav från organisationen eller för lite från uppdragsgivaren. Ehm.. Skulle vi då ha för lite resurser, så har vi bra tempo. Helt enkelt
26	Sasan	Det är så det fungerar alltså?
27	Johan	Alltså det är ju klart att varje dag är inte så att man liksom kommer kl 08.00 och går kl 20.00 på kvällen för att man har så j*vla mycket att göra.
28	Sasan	Ah ok.
29	Johan	Men över tiden så ska man ju ha tillräckligt mycket att göra.
30	Sasan	Jaja såklart!
31	Johan	Annars så stagnerar det på något sätt.
32	Kenny	Mm.
33	Johan	Men jag kan gå över på att beskriva lite på vår

		utvecklingsmetodologi istället.
34	Sasan	Gärna det! Det var faktiskt min nästa fråga ☺ (sade han lite glatt).
35	Johan	Vi är inte renodlade, utan vi ingick i ett projekt för ett gäng år sedan, vid sekelskiftet ehm, som kom att resultera i en doktorsavhandling på LTH. På telekomm. Där. Det är de som ju också sysslar med systemutveckling på LTH.
36	Kenny	Mm.
37	Johan	Där vi körde "eXtreme programing" och testade ut det. Och våra jobbare här var också ett gäng systemvetare, som jobbade extra hos oss. Så de fick köra "eXtreme programing" i detta projektet som gjorde nu man var 8 pers. totalt. Ehm.. Och vi, eller jag snarare, fastnade rätt mycket i "agile".
38	Sasan	Varför då?
39	Johan	Därför att det är effektivt. Deras struktur, vi jobbar i platta organisationer. Kanske ibland något luddigare krav, något luddigare mål från kanske inte 100 % beställarkompetens. Och inte för många i projekten.
40	Sasan	Mm.
41	Johan	Men i dags läget nyttjar vi nog något som mer kan liknas vid SCRUM.
42	Sasan	Den är väldigt populär nu har jag hört, just SCRUM. Det är väll många modernare organisationer som använder den?
43	Johan	SCRUM har fördelar jämfört med ren XP, sen har ju XP sina fördelar. Sen beror det även på vem man lyssnar på, enligt vissa människor så får man inte plocka bort någonting eller plocka på sig godbitar medans andra tycker det är helt ok. Sen så är alla överens om att det man ska ha är en bra produkt.
44	Kenny	Jo.
45	Johan	Så vi ligger någonstans mot SCRUM hållet. Vi jobbar med "sprintar". Vi jobbar med "cases & issues". Det vi däremot, i dagsläget, är extremt dåliga på är test-driven programing. Test-driven development.
46	Sasan	Jaså ok.
47	Johan	Och det är egentligen på grund av, eller inbyggda, svårigheter med att köra TDD och ha databasintensiva applikationer. Därför att TDD bygger ju på att du har en given in-data måste du ha en given för svar.
48	Sasan	Mm.

49	Johan	Med andra ord, om du ska köra mot en databas, så måste du ha.....
50	Sasan	Ursäkta jag avbröt dig men vad menar du med "TDD"?
51	Johan	TDD, Test-driven development.
52	Sasan	Test-driven development! Det var det du mena!
53	Johan	Men en databas förändrar ju sig. Inte i struktur men i innehåll.
54	Sasan	mm.
55	Johan	Och då att verifiera att det fortfarande funkar blir att det blir vääääldigt många "cases" som hela tiden måste skrivas om i "test-cases".
56	Sasan	Mm.
57	Johan	Så det är snarare en lathet att vi inte verkligen gör detta. Just därför att databasen alltid måste återställas till sitt ursprung för annars kan man ju inte köra testet igen.
58	Kenny	Ja precis.
59	Johan	Annars blir det ju inte samma in-data.
60	Sasan	Nä det har du rätt i.
61	Johan	Så att den biten har vi liksom... Kanske inte använder kanske inte så mycket däremot så har vi ju "use-cases". Såklart.
62	Kenny	Mm.
63	Johan	Trycker man "här" så ska följande händer
64	Kenny	Mm.
65	Johan	Och gör det inte det så är det ju inte ok. Och viss "sånt" som om man kanske kört "unit-tests", dvs. "den här in-datan är detta, kör vi den först och främst manuellt.
66	Kenny	Ok?
67	Johan	Ja det är en skog eftersom Test-driven development är det svåraste i "agile".
68	Sasan	Det är det?
69	Johan	Ja det svåraste att få folk att vänja sig vid. Bara det att det fenomenet att du har ett "case" jag ska göra detta, vad gör du först? Jo, du skriver dina "test-cases". Sen kör du dem och dem kommer ju att faila därför funktionerna gör ju ingenting! ☺ (sade han med

		glimten i ögat)
70	Sasan & Kenny	Nä precis! ☺ (lite smått skrattande)
71	Johan	Sen så bygger du dem tills du gjort alla dina ”unit-tests” går igenom. Då är du faktiskt klar. Under förutsättning att du faktiskt skrivit dit alla ”unit-tests” från första början ☺ (*glimten* är åter tillbaka)
72	Sasan & Kenny	Haha!
73	Johan	Men vi använder ju sprintar, vi använder ”cases”, vi väljer ”cases” inför varje release, eller sprint då. Vi eh.. nyttjar par programmering till viss omfattning och sådana grejer. Hm jag vet inte riktigt var ”checklistor” ska komma in i det här?
74	Sasan	Nä checklistor är alltså... det är ju som så, det är ju mycket svårare att hitta någon konkret, vad ska man säga, ”fast punkt” eller fasta, vad ska man säga, ”budord” i ”agile” metoden för där är man ju väldigt flexibel i och...
75	Johan	Aa just det!
76	Sasan	Kunna skifta sig och till och med kunna förändra, men även det i sig är ju en checklista.
77	Johan	Man har ju en arbetsgång som man inte går ifrån. Alltså då att man inför varje release plockar med sig ”vad ska vi göra nu?”. Hur ser de här ”casen” ut? Är det något vi måste splitta? Är det nästan samma så det handlar om ”duplicate”? Är det något som vi faktiskt ska göra just nu i alla fall? Hur lång ska sprinten vara? Och nu kör vi!
78	Sasan	Jag tror att du svara på många frågor i ditt tal nyss så det medför en risk att jag upprepar det du redan sagt men det är för att jag ska förstå.. eh ja. Skulle du kunna beskriva, kanske lite konkret, hur ser en checklista ut egentligen i en organisation som denna.
79	Johan	Vi har faktiskt inte ”skrivna” checklistor.
80	Sasan	Nä utan ni följer snarare en metodologi som t.ex. SCRUM då?
81	Johan	Vi arbetar bara något åt SCRUM hållet. Och så blir det ju så eftersom när spliten är mot sitt slut. Alla ”cases” är klara, då ”stage”ar man. Dvs. lägger ut några test databaser med så skarp data som möjligt och testar så att det verkligen fungerar eftersom det som fungerar i en utvecklingsmiljö är en sak. När man har gjort alla

		ändringarna. Om du sedan bara klonar upp din produktionsdatabas, lägg till de 2 fälten som ska in. Ändra den config data som ska ändras och sen kör man och så upptäcker man att "Oj! Det funkar ju inte i en "stage"ing miljö", halvskarp miljö, därför att vi har glömt att göra detta och detta och detta. Bra, då fixar man det och så får man den skarp, och då vet man att det funkar. Så det är ju liksom slutfasen där och det är ju en del av, vad man skulle kunna kalla, en checklista.
82	Sasan	Ah ok.
83	Johan	Det SKA "stage"as. Det ska det och sen ska det köras genom alla testerna igen i en "stage"ing miljö.
84	Sasan	Då har du, i stort sätt, svarat på "Hur ni arbetar" med checklistor också kan man väll säga?
85	Johan	Ja, sen att de checklistorna är mer "icke skriven" kunskap, kanske man skall ändra på? Det kanske man borde göra.
86	Kenny	Mm.
87	Johan	Samtidigt så vet ju hur vi jobbar så att det funkar ☺ hehe.
88	Sasan	Nä precis, det är ju som sagt agilt, det handlar om att anpassa. Fungerar det så fungerar det ju, ingen fara då.
89	Kenny	Nä men ehm. Samtidigt så finns det inte så j*kla mycket checklistor ute heller hos företag verkar det som. Det är mer som liknar detta här än ett intränat arbetssätt. Man vet att det funkar så.
90	Johan	Det som jag har upplevt när jag har pratat med andra är också det här att agile är så tufft idag.
91	Kenny	Mm.
92	Johan	Men det är väldigt få som verkligen kör agile och det krävs väl rätt mycket av ledningen för att få lov att göra det. Man måste förstå poängerna med det. För det är ju ingen hel aa.. det löser ju inte alla problem. Utan det löser vissa problem. Går man sedan till de riktigt stora företagen, ehm som elbolagen, ABD, dem här som kör med vattenfallsmetoden eller någon av de varianterna, där har man ju checklistor. I stora tjocka pärmar. Men det gör ju å andra sidan att där börjar du projekten och sen avslutar du dem och då är du klar med hela projektet oavsett om systemet gör vad det är tänkt eller inte. Men där är ju helt andra sorters checklistor.
93	Kenny	Mm.
94	Johan	Och sen beror det ju på vilka processer man har internt. Om ni har intervjuat utvecklingsföretagen, som IT-systems, Logika eller någon av de andra. Så gör de ju

		åt någon annan. Det är ett sätt att jobba.
95	Kenny	Mm.
96	Johan	Medan dem interna projekten är tilldelade på ett annat sätt. Som stora företag där dem har egna utvecklingsavdelningar osv. Som gör att dem bara har interna kunder där dem dessutom kan verksamheten på ett helt annat sätt.
97	Sasan	Ja då blir det ju genast mycket lättare att arbeta.
98	Johan	Ja förutom för uppdragsgivarna som då ska lägga det internt de har inga påtryckningsmedel. De kan inte avsluta kontraktet. De kan bara hota med att gå till en extern part och det får de inte policy mässigt. Hehe.
99	Kenny	Nä det är sant ☺ (skratta han instämmande).
100	Sasan	Denna frågan ställs så här, ”hur värdesätter du checklistor, och då pratar vi om era agila metoder, som ett verktyg för förenkling i utvecklingsprocessen?” Jag menar, ser du på det som en definitiv fördel?
101	Johan	Det är definitivt en fördel att ha en metod och den kan man ju då, i detta fallet, kalla en checklista givetvis. Då vet vi ju hur vi jobbar vi vet i vilket ordning vi ska göra vissa grejer. Vi kan också kommunicera hur vi gör våra grejer.
102	Sasan	Ursäkta jag avbryter dig, så det är avseendet struktur vi pratar om framför allt? Att det är då den går in och hjälper.
103	Johan	Ja du får också en struktur i arbetet. Så man inte hoppar fram och tillbaka hela tiden. Så strukturen är viktig så att alla vet vad de ska göra.
104	Sasan	För att bygga vidare på den frågan, vilken anser du vara den största fördelen med just struktur och nackdelen, om det skulle finnas.
105	Johan	Om man är för rigid i sin struktur, vilket då blir motsats till agile, så kan dem ju lika väl glida mot ett hinder för verksamheten i stort. ”Vi kan inte fixa detta idag eftersom vi måste gå enligt process” den klassiska byråkrati ledningen. Man måste vara flexibel, oavsett om man kör i en väldigt väldigt strukturerad projektmiljö. Man måste vara flexibel plus att, om du inte t.ex. har en release förens om 2,5 månader som händer i vattenfallsprocesser, ska vi verkligen inte fixa den nu? Så att strukturen är viktig men den måste vara flexibel och den största fördelen med att ha en struktur är när den är flexibel för folk vet vad dem ska göra, folk vet vart vi är på väg. Folk vet hur det funkar. Så att den största nackdelen blir när man är för rigid i den.
106	Sasan	Mm.

107	Johan	Är man liksom för flexibel så blir det bara kaos. Och då blir det ingenting gjort tillräckligt snabbt och de inblandade blir bara frustrerade. "Ska vi aldrig releasea det här? F*n vad tråkigt, vi vill göra nya grejer".
108	Kenny	Ja det är sant.
109	Sasan	Så som jag har uppfattat det sa du att ni använder SCRUM till viss del?
110	Johan	Vi använder någonting som skulle kunna likna SCRUM ja.
111	Sasan	Som skulle kunna likna SCRUM, just de, då innebär det att ni har tittat i SCRUM och där finns punkter som ni kan välja bort och punkter som ni kan välja in inte sant?
112	Johan	Ja.
113	Sasan	Då har vi i stort sätt tagit den frågan också, det är jättebra ju... och även denna ☺ hehe. Då ska vi se här
114	Johan	Haha.
115	Kenny	Jag tänkt mest på dem hära... på vilket sätt tycker du checklistorna fungerar verktyg för att säkerställa kvalitet på produkten?
116	Johan	Ja det gör de ju på..... ja det vete f*n..
117	Sasan	Det är en svår fråga eftersom kvalitet är så svår definierat.
118	Johan	Alltså det gör dem ju därför att det ska finnas "use-cases" och det ska testas. Så finns det många olika delar av kvalitet, eftersom det är ett luddigt begrepp.
119	Kenny	Aa, precis.
120	Johan	Gör applikationerna och what ever var de är tänkta att göra är en kvalitet, men gör de det på ett bra sätt, är även en kvalitet i sig dvs. är det användarvänligt? t.ex. Om jag gör denna uppdateringen, låser det hela databasen? Det är en annan sorts kvalitet. Sen finns det kvalitet i själva deliverans.
121	Kenny	Mm.
122	Johan	Vad händer om någon slarvar? Glömmer uppdatering av en config. Fil eller flytta data i databasen för att den nya funktionen ska fungera det är också en del av kvalitet att det inte är slarv, men det är där checklistor kommer in så att man har sina uppgifter. Det ska testas. Det ska "stage"as. För att se om det funkar utanför utvecklarnas egen miljö.
123	Sasan	Mm.

124	Johan	För att du ska ha kvalitet, måste du ha någon form av metod eller checklista(or). Det måste du ha och du kommer inte runt det.
125	Sasan	Aha ok!
126	Johan	Det är ju vad agile säger också egentligen, vi har vårt sätt att jobba. Om du säger ???15.04 use-cases och staging osv för annars kan du inte leverera kvalitet. Det går inte.
127	Sasan	Nej, de kvaliteter som vi har som du säger den parametern som du säger mellan just användbarheten och tid och pengar
128	Johan	Ja
130	Sasan	Om checklistor är ett hjälpmedel för att förminska den här parametern
131	Johan	Absolut, mm det är där. Sen ahr du ju vissa delar där man säger "ermm" att man leverera faktiskt när man ska på ett bra sätt, err 15.27??? Sen har du ju också från uppdragsgivaren där man har skrivit att vi vill ha det här. "Pause" Då måste man ha i sin metod att måste vi ställa ytterligare frågor? Eller ska vi blindt ta det exakt som det står här. Men det ska man ju inte.
132	Sasan & Kenny	Mm, nej precis
133	Sasan	Det är precis som du sa att det blir lite luddigare i agile
134	Johan	Det blir lite luddigare. Å andra sidan för att kunna göra det sen måste man kolla, ska knappen här vara grön eller blå? Enligt standard ska den vara grön. Men du har skrivit att du vill ha en blå knapp. Varför det? Det är ju ett ansvar utvecklingsavdelningen måste ta, ett ansvar där man måste se till att det faktiskt blir rätt. Vare sig om det är från beställarens sida, vem det nu än må vara och jämfört med systemet i övrigt. Man kan liksom inte bara slänga in en uppdatera knapp när det heter spara överallt annars.
135	Kenny	Ne, precis
136	Johan	Och det är också en viktig användbarhets grej som också spelar in, inget formulär på det här programmet ser likadant ut. Det ser för jävligt ut alltihop. Även om man har missat den här användbarhetsdelen så har man fortfarande levererat kvalitet.
137	Sasan	Hehe, jo det kanske gick fort och kosta lite pengar
138	Johan & Sasan	Men bra blev det inte. (alla skrattar)
139	Kenny	Ja, har ni någon utbildning av era användare som skall använda system när det är levererat?

140	Johan	Det är vi ju otroligt usla på. (Sagt på ett glatt sätt)
141	Kenny	Ok (skrattar)
142	Johan	Öhh, vi har viss utbildning och vi kommunicerar ut vad vi gör för någonting i releaserna. Till samtliga användare. Ermm, och uppmanar alla att ställa frågor om det är någonting de är intresserad av eller någonting de inte förstår. Ermm, vi utvecklar, dels utvecklar vi rent interna grejer. Som då används av Online. Men så har vi ju system som används som satslösningar alltså software service som används av externa parter. Ermm, och där kommunicerar vi mer, men oftast så är det en viss människa som har begärt en viss funktion och känner att nu har jag fått den. Så systemen i sig skall vara självklara till just det. Annars får man ju skriva manualer
143	Kenny	Mm
144	Johan	Eller dokumentation, hur använder du den här funktionen för att inte sänka systemet eller för att det ska blir som du har tänkt dig. Om de är ganska kraftfulla och det är hur man ska kommunicera ut det. För att så att säga regelrätta utbildningar har vi ju inte. Det är också en sån grej att när du kör agile så skall du ha så korta sprinter och releaser som möjligt och det är vad många rekommenderar och du kan ju inte köra nya utbildningar var fjärde vecka.
145	Kenny	Nej, det går ju inte
146	Johan	Samtidigt så är det ju så små ändringar att, ja nu funkar den här knappen eller nu har det tillkommit en knapp som ändrar den här statusen för det har ni behövt öhh öhh göra och så va och då är det ganska enkla grejer.
147	Kenny	Ja, precis
148	Sasan	Existerar det någon interaktionsdesignroll här på online eller allihopa har den här kunskapen?
149	Johan	Det, det ska utvecklarna sköta själva.
150	Sasan	Då pratar jag mest om det här när externa kunder kommer in och beställer vissa varor eller system.
151	Johan	Nej, någon regelrätt interaktionsdesign finns det inte och vi är ju inte så många heller ”skratt”.
152	Sasan	Nej, det är ju sant
153	Johan	Ja, ni såg ju avdelningen, den va inte så stor
154	Kenny	Ja, hur många va ni 5-6 st?
155	Johan	Ja 6 stycken är vi

156	Kenny	Det är ju trevligt när man är lite mindre. Då känner man ju alla i alla fall.
157	Johan	Ja, precis
158	Kenny	Ja, är det någon som är ansvarig för er systemutvecklingsmetod checklistor skall se ut?
159	Johan	Ja, det är ju jag då, men det gör jag ju då i samråd med de som jobbar med det.
160	Sasan	Ja så då är det även agilt där kan man säga
161	Johan	Ja, precis även agilt där för fungerar det inte för de, alltså de som ska jobba med det. Då blir det inte bra. Alltså du måste sälja in det. En del med att sälja in det till användarna av metoden är att de faktiskt skall tycka att det är bra och det kvittar om det är användarna som i det här fallet det är utvecklarna eller om det är användarna som är beställarna. Det är ju de som vet oftast hur man bäst gör det. Pause och det är en del av det agila också. Användarna är ju de som driver de som beställer, användarna är ju de som bestämmer det är inte it-avdelningen som säger att vi ska göra så här.
162	Sasan	Då mäter du användbarhet som något väldigt stort
163	Johan	Ja, användbarheten är ju a och o. Därför att det minskar resurserna för supportkostnader
164	Kenny	Ja, jo det är ju klart ju.
165	Johan	Om en funktion har rätt namn och folk vet hur de ska använda den och inte tror att han gör någonting annars (skrattar) så slipper du hur mycket support som helst.
166	Kenny	Mm, ja
167	Johan	Tex.
168	Kenny	Ermm, hur gör ni om ni känner att en kund har en fullständigt korkad idé så har ni någonting bättre som förslag?
169	Johan	Då för vi ju en dialog med uppdragsgivaren eller beställaren.
170	Kenny	Ja
171	Johan	Ermm, vi kallar det beställaren. Då för vi en dialog med de. Du vill faktiskt kunna se det här och du vill faktiskt kunna göra det här.
172	Kenny	Ja
173	Johan	Och hittills har vi ju oftast haft rätt.
174	Kenny	Ja, för det känns som man kan missa en hel del grejer

		om man lyssnar blindt på beställaren.
175	Johan	Ja och det är också ett ansvar som ligger på systemutvecklarna i agila metoder. Du måste ha en lokal kännedom, du måste veta vad du sysslar med det är inte blackbox.
176	Kenny	Nej, precis
177	Sasan	Exakt
178	Johan	För då gör du bäst grejer, känner du domänen åtminstone ytligt så där, så levererar du bättre kvalitet helt enkelt. Men det är ju med analysarbetet och use-case arbetet att föra en dialog med uppdragsgivaren. Hur viktigt är det? Alltså bara en så pass enkl fråga. Hur viktigt är detta faktiskt för dig? ”ja inte jätte viktigt, jag kom bara på det igår så jag kunde lika gärna slänga iväg en specifikation till er. Men, nej nej det är inte viktigt”.
179	Kenny	Mm
180	Johan	Men sånt är bra då behöver vi inte prioritera det speciellt högt. Samtidigt är det jättebra att få in alla de här grejerna. För att är det någonting man upptäcker när man jobbar med cases så är det, ja vi ska jobba med de här grejerna. Vänta lite här nu, här ligger fyra case. Som är liksom bara. En liten, liten extension på var och en av de här. Då kan vi lika bra göra de nu också. Johans telefon ringer och en pause tas.
181	Johan	Var va vi?
182	Sasan	Vi pratade lite om användbarhetsaspekterna och det var a och o.
183	Johan	Ja alltså användarheten är viktigt och man måste föra en dialog med beställaren.
184	Sasan	Mm
185	Johan	Det är ju att man ska få så mycket utväxling i systemet som möjligt. Om man säger cases som bara är extensions, varför skall vi vänta med det här grejerna till kanske nästa release? När vi kanske redan har gjort hälften. Ja men då gör vi ju de också och levererar mer till kunden.
186	Sasan	Precis
187	Sasan	Förändras checklistorna mycket efter det som händer ute på markanden?
188	Johan	Ja det gör vi ju definitivt och framförallt hur det ser ut för oss.
189	Sasan	Ja precis och intraorganisatoriskt menar jag då.

190	Johan	Ja precis, det måste vara en levande process
191	Sasan	Mm
192	Johan	För som it-avdelning så tycker jag man måste vara en serviceorganisation. Ermm, ja företaget fungerar inte utan it men det är inte vi som producerar. Alltså eftersom vi sysslar med call-center verksamhet så är det ju call-centret som är själva produktionsenheten. Men de fungerar ju inte utan it, alltså blir ju it en servicen.
193	Sasan	Ja
194	Kenny	Mm
195	Sasan	Anledningen att jag ställer frågan är att många har använt sina checklistor som en nisch. Alltså de har behållt det gamla utan att anpassa sig. Så de på något sätt kan referera till att så här gör vi och det ser bra ut utåt.
196	Johan	Ja, alltså metoder och checklistor måste anpassas efter hur övriga organisationer förändrar sig. Men, öh. Det måste ju göras på ett intelligent sätt. Alltid, bara för att någon tycker att det funkar bättre för öhh den. Alltså att process borde fungera så här så kanske man inte ska ändra process. För det kanske för organisationen i helhet blir dyrare. Man måste ju alltid räkna i pengar också.
197	Sasan & Kenny	Mm
198	Johan	Om vi gör som it-avdelningen vill här så minimerar vi resursanvändandet det blir lite mer jobb för dig. Öhh jag vill inte ha mer jobb jag vill ha mindre jobb. Då kostar det en hel extra tjänst att kunna lösa det på det sättet.
199	Kenny	Mm
200	Johan	Och då blir det ju fel. För att man ska ju inte göra något så det blir för dyrt. Utan det måste ju vara dynamiskt där också. För att produktionen möter ju alltid nya krav och därför måste ju även processerna ändra sig. I vissa fall. Det är ett kontinuerligt arbete.
201	Kenny	Japp, Hur mottags förslag i checklistornas punkter? Och vem lägger fram förslagen efter det som kommit fram så antar jag att ni göra det tillsammans?
202	Johan	Ja, alltså det sker ju internt och det sker ju på de forumen som finns så att säga. Det sker i samtliga fall i dialog. Ermm för att bara för att det ser ut att vara en bra idé så behöver det ju inte vara det. Därför ska man ju ha så mycket input som möjligt så att det blir bra beslut som man inte behöver backa på. För det kostar bara tid och pengar.

203	Kenny	Ja
204	Johan	För det viktigaste är ju trots allt att tjäna pengar. (Med sarkastisk röst) ”Annars kan man ju inte ha anställda och då kan inte de få lön, hehe, Ja just det”.
205	Kenny & Sasan	Skrattar
206	Johan	Men det är ju så, det låter lite synsikt men ett företag som inte är ute efter att tjäna pengar det blir ju inte långvarigt.
207	Sasan	Nej, det är ju klart.
208	Kenny	Nej, det går ju inte.
209	Sasan	Ja, pengar är ju en satisfaktionsfaktor för att faktiskt leverera. Det är det bara att acceptera
210	Johan	Ja, precis
211	Sasan	Öhh, här kommer kanske en liten fräck fråga men. Har ni identifierat några brister i ert eget sätt att arbeta, förutom då som du sa att ni var lite svaga på vissa bitar. Du får ursäkta att jag säger det, men som du sa väldigt dåliga på internutbildningar
212	Johan	(skrattar) Väldigt usla tror jag att jag sa. (Alla skrattar)
213	Sasan	Hur åtgärdas bristerna i alla fall. Det är det vi är ute efter.
214	Johan	De felen vi upptäcker, får ju åtgärdas genom att ändra processen helt enkelt. Genom att förändra processen så åtgärdas problem. Man kan införa helpdeska man kan införa öhh bättre release dokumentation till exempel. Om det är det som behövs, ermm man kan också rikta release dokumentationen alltså ge avdelningen mer info om den utvärderingen som görs nu. Här genomförs ett nytt sätt för er att arbeta. Ja, då kanske man skall skriva en manual som bara ges till just dem. Vi har infört helpdesk för vissa externa projekt, så de mailar samma adress ringer samma nummer etc. Vi har uppdateringar alltså gjort bättre dokumentationer som ges till vissa avdelningar. För att vi upptäcker att de begär en funktion utan att upptäcka att där finns en.
215	Kenny	Mm
216	Johan	Men de har inte tid att utbilda sina egna anställda och då fixar vi det åt de. Det gäller att vi i en dialog hittar de svagheter som finns eller kanske uppstår. För det kanske inte var en svaghet för ett halvår sedan. Men nu är det en svaghet för att andra grejer har ändrat sig. Då för man en dialog så får man ändra det efter hand.
217	Sasan	Händer det ofta att man levererar ett system som kommer tillbaka mycket senare och säger att detta fungerar inte eller vi skulle vilja ha detta. Är det lätt

		genom förvaltningsprocessen att gå in och lägga till den här funktionen där.
218	Johan	Ja, alltså det händer ju hela tiden. Systemen är levande vi har ju inget färdigt system. De kommer ju alltid att förändra sig eftersom verksamheten förändrar sig. Man kanske hittar en annan parameter i hur verksamheten funkar eller en förändring och nu måste vi jobba på ett annat sätt.
219	Sasan	Ja, precis det var det jag ville höra. Att systemen aldrig blir färdiga
220	Johan	Ja då har det ju stagnerat på något sätt. Inte ens vår storprogramvara blir ju någonsin färdig. Den fungerar ju visserligen bättre än vad den gjorde för 10år sedan. Det är ju på den nivån, det finns alltid nya grejer att göra i systemen. Därför att verkligheten runt om kring oss förändrar sig.
221	Kenny	Ja, en sista fråga. De use-casen ni har skulle man kunna säga att de blir er checklista för användbarhet?
222	Johan	Ja så kan man säga. Det blir en checklista eller ja, en sub-checklista för användbarhet.
223	Sasan	Då tackar vi så mycket för intervjun. Har du några frågor till oss?
224	Johan	Nej.

Bilaga 4 - Intervju med Johan, Logica AB**Datum & Tid:** 12/1 – 09, kl 21.30 – 22.00.**Plats:** Hemma hos honom, i Broby.**Informant:** Johan, Utvecklare på Logica i Kristianstad.

Transkriberingen börjar! På vänster kolumnen ser du siffran som motsvarar vilken mening vi är på i intervjun, 2 kolumnen (från vänster) ser man namnet på personen som talar och den resterande kolumnen representerar vad som har sagts.

1	Kenny	Vi kör igång med en kort bakgrundsfråga och jag hoppas du orkar trots den sena timmen. Vad har du för bakgrund, alltså vilken utbildning och tidigare jobb inom systemutveckling?
2	Johan	Detta är mitt första jobb inom systemutveckling. Har läst två år ren utveckling på eftergymnasial nivå, samt tagit mcp i C# .net utveckling. Sen gick jag 3 år på högskola på hkr, som systemutvecklare.
3	Kenny	Hur länge har du arbetat i organisationen?
4	Johan	1år och och 4månader
5	Kenny	Vad är din roll inom organisationen?
6	Johan	Är ren systemutvecklare, nyutvecklare en befintlig applikation
7	Kenny	Ok, det låter ju intressant då vår uppsats skall handla om just vidareutveckling. Vilka sorters checklistor arbetar du med? och hur ser en typisk sådan ut?
8	Johan	Vi har en iterativ utveckling, så jag får en iteration i taget med vilka funktioner som skall vara klara tills nästa iteration , samt testresultat från föregående iterationer. Iterationerna är ganska små och består oftast av ett par mindre uppgifter eller en större. En iteration är oftast bara nån vecka
9	Kenny	Ok, Trivs du med det sättet ni arbetar med checklistor?
10	Johan	Ja, det funkar hur bra som helst. Är skönt att inte ha så stora mängder att göra mellan varje testning.
11	Kenny	Jo de kan jag tänka mig(skrattar). På vilket sätt värdesätter du checklistorna?
12	Johan	Hehe, ja du. De ger mig en klar uppfattning vad som skall göras och vad som behöver fixas .

13	Kenny	På vilket sätt tycker du checklistorna fungerar som verktyg för att säkerställa kvaliteten på det ni utvecklar?
14	Johan	Det ökar kvaliteten innan produkten kommer till kunden, men tar längre tid att färdigställa den
15	Kenny	Ok, vilket skulle du säga väger tyngst då? En färdig produkt som är klar snabbt/på deadline eller en med god kvalitet
16	Johan	Det beror nog på produkten, I vårt fall i och med den är verksam inom sjukvården så är nog kvaliteten viktigare än deadline. Plus att den föregående versionen är redan i drift.
17	Kenny	Mjo det har du nog rätt i. Känns sådär om ett sjukvårssystem skulle komma ut med flertalet buggar i som kostar människoliv
18	Johan	Hehe, jo det är nog inte speciellt uppskattat(skrattar).
19	Kenny	Har ni nån utbildning av användarna av systemet?
19	Johan	Japp, det har vi, är inte så insatt i hur mycket dock
20	Kenny	Ok, du har inte koll på om det finns nån checklista för just utbildningen? Det kanske är en svår fråga om du inte är så insatt i det, men hoppas du kan svara på den.
21	Johan	De har någon användarmanual, men vet inte om dom går efter den på utbildningen eller om det bara är en generell genomgång. Jag hoppas den går igenom bra i alla fall :) Så systemen används så smidigt som möjligt.
22	Kenny	Hehe, jo det kan ju vara bra. Ok, nästa fråga. Hur viktigt anser du att en utbildning av användarna är?
23	Johan	Känns som det är mycket viktigt när det är ett stort system, där man kan göra samma sak på ett par olika sätt.
24	Kenny	Ja precis, så de gör det på bästa sätt osv med tex. genvägar.
25	Johan	Exakt.
26	Kenny	Har ni checklistor som fokuserar speciellt på användbarhet?
27	Johan	Ja, vi har några generella checklistor, t.ex att allting skall kunna göras med enbart tangentbord. annars är det mest sådana punkter de kommit fram till genom att använda det gamla systemet.
28	Kenny	Ok, så ni har utvärderingar och grejer av det gamla eller? Just ja, kör ni me nån officiell metod eller med

		egen? och använder ni usecases o sådär
29	Johan	Vi har mest utvärderingar av det gamla systemet. Plus att det vi producerar visas med jämna mellanrum på användarträffar för superanvändare vilket är användare som kan gamla systemet utantill, vilket diskuteras samt går igenom.
30	Kenny	Mm
31	Johan	Mm det är någon officiell metod vet jag inte, men är väl en sorts usecase iom kunden är med o tutar
32	Kenny	Hehe jo(reser på ögonbrynen) , låter rätt najs me användarträffar, ni är de första jag hör om som har de så
33	Johan	Ja, fast jag har aldrig varit på nån sån => Är chefen och utbildarna som kör de. Skulle gärna vilja vara på en dock då det hade varit lärorikt
34	Kenny	Hehe, ok. Hur mycket värdesätter du användbarhetsaspekterna i checklistorna?
35	Johan	Jäklar vad det var mycket om checklistorna =P Just användbarhets aspekterna är väl dom jag använder minst, då det mesta är ganska basic, men ibland finns det några udda krav så
36	Kenny	Ok Jo uppsatsen ska ju handla om de
37	Johan	Aha, den tråkigaste biten i utveckling =P
38	Kenny	Haha, jo. Men är ett rätt intressant ämne att skriva om. Då det känns ganska viktigt för att höja kvaliteten på en produkt
39	Johan	Jo, det har du rätt i, är inte direkt frågor som jag har kanonkoll på då jag mest sysslar med utvecklingen. Men skall ge så bra svar jag kan.
40	Kenny	Hehe, tycker du har klarat dig fint så här långt. Används checklisten(orna) som regelgrund för organisationen? Eller anpassas det och kan förändras med förändringar runt organisationen?
41	Johan	Erm, vad menar du med det?
42	Kenny	Vi säger såhär istället. Förändras checklistorn efter vilket projekt det handlar om?
43	Johan	Ja, det gör dom , vi har helt projektstyrda checklistor.
44	Kenny	Ok, om ni har åsikter om de direktiven ni får i checklistoran får ni ändra grejer då? eller det är det stenhårt

45	Johan	Vi får oftast vara med redan när checklistorna skrivs, dom skrivs i hand med iterationerna.. så vi har all möjlighet att framföra åsikter och komma med förslag
46	Kenny	Ok, coolt. Verkar vara ett schysst sätt att arbeta.
47	Johan	Jo då, det frunkar bra som tusan.
48	Kenny	Blir arbetssättet mer mekaniskt tack vare checklistorna? eller behåller man en dynamik i arbetsflödet?
49	Johan	Det är väldigt dynamiskt, ofta har vi ett par grejer att snurra på samtidigt så.
50	Kenny	Hehe, ni verkar ha det fint. De flesta andra jag hört/pratat med verkar ha det väldigt styrt uppifrån.
51	Johan	Ja, det är de bästa jag har varit på så, fast bara kört truck innan så(skrattar) Okej, jo jag har hört det av dom från ericsson med. Så har nog hittat en guldgruva(skrattar och verkar väldigt lycklig).
52	Kenny	Är det ni som styr hanteringen av checklistorna eller är det checklistorna som hanterar er?
53	Johan	Hehe i vårt fall styr vi rätt mycket oss själva.
54	Kenny	Ok
55	Johan	Klart ibland får man en checklista med kritiska buggar i ett sidoprojekt, och då måste det göras kalas snabbt
56	Kenny	Mm. Hur mottags förslag på förändringar i checklistornas "punkter"?
57	Johan	Har vi bara några vettiga förslag så är det alltid positivt. och även om det inte går igenom så blir dom alltid glada att vi kommer med egna förslag. Aldrig något negativt
58	Kenny	Ok, Sista frågan nu en liten fräckis ^^
59	Johan	Hehe, ok.
60	Kenny	Har ni identifierat någon form av brist i användandet, hur åtgärdas det? (om)
61	Johan	Jag hängde inte med redigt där(skrattar lite) Om vi gör något fel i det som kommer ut eller så?.
62	Kenny	Nej, inte riktigt. Tycker du det är något ni är dåliga på, ngt som brukar komma i checklistorna upprepanande som

		inte är så bra?
63	Johan	Hehe du menar så, Det gamla vanliga. -Dåligt kommenterad kod -Dålig felhantering -Lite osammanhängande kodstandard då vi alla köra rätt personliga stilar
64	Kenny	Mm. Så verkar det vara lite överallt i branschen.
65	Johan	Jo, det är ju vanliga standard fel.
66	Kenny	Det var alles, jag vill tacka för intervjun och hör av mig om vi skulle vilja ställa någon kompletterande fråga.
67	Johan	Det är bara att fråga på när du vill via mail eller telefon om du har problemt att ta dig hit.
68	Kenny	Ok.

Bilaga 5 - Intervju med Thomas Farby, Volvo IT

Datum & Tid: 19/12 - 08

Informant: Thomas Farby

Företag/Organisation: Volvo IT

Presentation

Vi är två studenter på Lunds Universitet. Vi studerar på Holger Crafoords Ekonomisentrum, där vi tillhör institutionen för Informatik. Vår kandidatexamen/C-uppsats är vår agenda.

Syfte med intervjun:

Vi håller på att undersöka de checklistor som används som verktyg vid uppdatering av era befintliga system. Vi vill se om det finns en koppling mellan användandet av just dessa checklistor, som verktyg, och användbarhet. Användbarheten är i detta fall hur ni (som utvecklare) anser att checklistorna hjälper er säkerställa att de kraven som ställs verkligen uppfylls. Checklistor kan även vara de "punkter" eller "kriterier" ni följer i en t.ex. metod eller dylikt.

Vi är inga experter inom området så därför är det möjligt att vissa frågor är svåra att svara på eller anses vara "luddiga". Vi hoppas att du har förståelse för detta och därför ger oss så utförliga svar som du kan. Du får gärna uttrycka mer än bara svaret på frågan för att hjälpa oss förstå vad du menar. Vissa frågor kanske kommer bli dubletter eftersom ett svar kan vara svar på fler än en fråga. Då är det bara att hänvisa till tidigare svar.

Dina svar kommer hjälpa oss fastställa en empiri i vår undersökning och vi vill därför förtydliggöra att du får såklart hoppa över frågor som du inte är bekväm att svara på och även om du önskar vara anonym. Om du önskas vara anonym så är det bara att skriva till det längst nere i guiden. Nedan följer frågorna.

Tack på förhand!

OBS! Du får gärna svara rakt i detta dokument sedan, spara, och skicka tillbaka!

INTRODUKTIONS FRÅGOR

1. Vad är dina erfarenheter inom systemutveckling?

Jag har jobbat inom systemutveckling som IT-konsult i en supply-roll under de senaste tio åren. Tidigare har jag till och från arbetat inom systemutveckling på kundsidan i en demand-roll.

2. Vad är din roll/befattning inom företaget?

Sedan ett antal år är jag Program Manager för en Business Unit inom Volvo, Volvo Powertrain, som levererar drivlinor (motor, växellådor, fram- och bakaxlar) till Volvos produktprogram. Jag har tidigare arbetat som lösningskonsult och projektleddare inom Volvo.

Jag har också en roll som "Business Solution Manager" inom OTD-området (Megaprocessen Order to Delivery, egentligen Volvos Fabriker) när det gäller SAP-applikationer.

3. Hur länge har du arbetat i organisationen?

Jag har arbetat 10 år inom Volvo IT.

4. Vad är dina arbetsuppgifter? (konkret)

I min roll som program manager är min roll enl. nedan:

I min roll som BSM är min roll enl nedan:

SD&PO – Business Solution Manager

Mission	Promote, develop, manage and maintain SAP solutions portfolio supporting Volvo group processes and secure early watch on new SAP products.
Ownership	<ul style="list-style-type: none"> • SAP Solutions through the lifecycle (project, maintenance).
Activities	<ul style="list-style-type: none"> • Perform SAP Solution/process governance • Be part of: <ul style="list-style-type: none"> - the proposal team - the project team (at least in early phases) - the Change Control Board with the Business Owner during both project and maintenance phases • Participate & review to general design, taking care of integration with other solutions, aligned with IS/IT strategies of the process. • Promote SAP products portfolio. • Ensure that we have the right level of usage of SAP and reuse of solutions. • Benchmark solutions with companies and best practices. • Cooperate with CSD, GBSM, System Owners and Process Owners. • Contribute to capacity and competence planning (Internal, External and Competitive sourcing). • Ensure that service portfolio is adapted to customers needs and is competitive. • Ensure global SAP commitments regarding GBSM directives and give recommendations to the GBSM. • Manage solution competence network together with Practice resources.
Roles	<ul style="list-style-type: none"> • Business Solution Manager.

SD&PO – Program Manager

Mission	<p>Promote and provide advices to customer in order to ensure SAP solutions consistency.</p> <p>Coordinate and secure delivery of Global SAP proposals, solutions and services to</p>
Ownership	<ul style="list-style-type: none"> • Global SAP Customer requests, projects and services portfolio. • Delivery of proposals, solutions and services (cost/time/scope/quality). • Advises to customer to secure application functionalities and architecture.
Activities	<ul style="list-style-type: none"> • Manage Global SAP customer requests, projects and services portfolio. • Promote SAP solutions and provide advises regarding application functionalities and architecture. • Identify and review opportunities with GAM and AM. • Support the customer to define his business case. • Deliver SAP solution based proposals for the received requests assisted by a proposal team. • Ensure the staffing for the delivery. • Initiate project with Project Managers, coach and support them. • Member/Chairman of the IT Delivery Committee. • Control and Monitor the delivery (cost, time, scope and quality) in conjunction with GAM and AM. • Coordinate or manage delivery status to the customer. • Manage customer activity forecast including Licenses and provide the resource prognosis to the practices. • Ensure the hand-over from the project to the Maintenance.
Roles	<ul style="list-style-type: none"> • Customer Service Delivery, Program Manager.

Du har säkerställt att du är välkommunikatör med dem du har ett relation.

utvecklar?

I min roll som Program Manager, hjälper det mig genom att alla projekt följs upp på samma sätt, att alla gate-passager har en checklista som skall vara avprickad för passage och öppnade av nästa road-map. Det är också en trygghet för mig i de fall när man har en mindre erfaren projektledare. Oftast beordrar jag en review före ett gate-möte, och då ser en erfaren projektledare igenom projekt-status och upprättar ett reviewprotokoll.

7. I vilka avseenden/faktorer skulle du anse att checklistor fungerar som ett verktyg för förbättrad kvalitet på just användbarhet? Finns där några speciella "områden" eller "faktorer" (ex, struktur, förändring, utveckling etc.) där checklistor på något sätt visat sig vara av fördel att använda än att inte göra det?

I de flesta fall är det givetvis en fördel att använda checklistor, men de måste alltid användas med förstånd för att inte skapa en onödig administration/onödiga kostnader.

ORGANISATION & UTBILDNING

8. Tar checklistorna upp någon sorts utbildning av användarna som skall använda det uppdaterade systemet isf, hur viktigt tycker du utbildning av användarna är?

Utbildning är en mycket viktig punkt och i de projekt jag ansvarar för brukar jag planera detta när projektresurserna tillsätts. Jag använder också metoden "Train the Trainer", dvs att de som skall utbilda end-users skall också medverka i projektet, känna till sin roll som utbildare av slut-användare och vara en potentiell key-user av systemet efter Go Live.

9. Vem är ansvarig för checklistornas utformande? Finns där en specifik roll för det arbetet? I så fall vilken.

I Volvo IT finns en hel avdelning som arbetar med metoder. Inom Global SAP finns en grupp som arbetar med att SAP-anpassa det som utvecklas av metodgruppen.

ANVÄNDBARHET

10. på vilket sätt värdesätter du användbarhetsaspekterna i checklistorna?

Givetvis väldigt viktigt, och inom Volvo IT finns en grupp som hjälper till att coacha oerfarna användare.

11. Används checklisten(orna) som regelgrund för organisationen eller anpassas dem till förändringar i externa faktorer (förändringar i marknaden, organisationen etc.)?

Nej checklistorna och metoderna är obligatoriska och skall inte förändras. Givetvis skall dom också användas med "sunt förnuft" och med hänsyn till storlek och komplexibilitet av det aktuella projektet.

12. Blir arbetssättet mer mekaniskt tack vare checklistorna? eller behåller man en dynamik i arbetsflödet? På vilket sätt i så fall?

Givetvis blir det en mekanik när arbetet i stor grad styrs av metoden/checklistorna. Det är ett problem att hitta utmaningar till erfarna projektledare som måste få gå vidare till mer komplexa projekt för att finna utmaningar.

13. Hur mottags förslag på förändringar i checklistornas "punkter" (om så existerar) och vem ställer dessa förslag? Är det bara utvecklarna eller mottags slutanvändares åsikter också?

Slutanvändarnas åsikter mottas givetvis, men förändringar styrs i hög utsträckning av metodgrupperna.

14. Har ni identifierat någon form av brist i användandet, hur åtgärdas de? (om)

Givetvis finns brister i användandet, men det är i regel i mindre projekt. Projekt över ett visst värde skall annaonseras och följas upp i ett project Dashboard och samtliga projekt där följer alla checklistor.

ÖVRIGA KOMMENTARER:

Bifogar ett antal templates som ni gärna kan använda. Bifogar också ett par exempelsom jag vore tacksam för att ni behåller för er själva. Sänder detta för jag tycker det är viktigt att ni kan få ett exempel för att förstå hur vi använder oss av templates.

Först den höga nivån om process specificaton



\\Vcn.ds.volvo.net\ it-got\home06\V0731
\\Vcn.ds.volvo.net\ it-got\home06\V0731

Ett stort tack för din medverkan!

Önskas anonymitet:

Slutkommentar

Vi tackar så mycket för din tid! Dina svar kommer att bidra till kunskap inom informatik!

Har du frågor eller dylikt angående uppsatsen är det bara att fråga.

Önskas det en kopia av intervjuens utformning i uppsatsen (som är i stort sätt samma som denna guide) så får du så klart det!

Yepp

Om Ni önskar att få ta del av den slutgiltiga uppsatsen så skickar vi gärna över en kopia till er!

Yepp på kontaktinfo thomas.farby@volvo.com

Åter, ett stort tack ska Ni ha! //

Sasan Zanganeh

Kenny Olofsson

Kontakt:

Sasan_Zanganeh@hotmail.com
0737 29 43 41

Kenny_mon3@hotmail.com

Referenser

- Alvesson, M. & Svenningsson, S. (2007): *Organisationer, ledning och processer*, Studentlitteratur, Lund.
- Andersen, E. S. (1991): *Systemutveckling – principer, metoder och tekniker*, Studentlitteratur, Lund.
- Avison, D. & Fitzgerald, G. (2006): *Information Systems Development: Methodologies, Techniques and Tools*. 4th ed. McGraw-Hill, London.
- Backman, J. (1998): *Rapporter och uppsatser*. Studentlitteratur, Lund
- Brandt, P. (2005): *Systemförvaltningsmodeller*. KFS AB, Lund, ISBN 91-628-6641-9
- Brenner, M. (2006): Classifying ITIL Processes; A Taxonomy under Tool Support Aspects. IEEE Business-Driven IT Management, 2006. BDIM '06. The First IEEE/IFIP International Workshop on. pp. 19-28.
- Brinkmann, S. & Kvale, S. (2009). *InterViews : learning the craft of qualitative research interviewing*. Sage Publications, Los Angeles.
- Bryman, A. (2002): *Samhällsvetenskapliga metoder*. 3. uppl., Liber, Malmö.
- Clacy, B. & Jennings, B. (2007): Service Management: Driving the Future of IT. IEEE Comput. Soc., Vol. 40, Issue. 2. pp. 98-100.
- Dix, A., Finlay, J., Abowd, G. & Beale, R. (1997), *Human-Computer interaction*, Pearson Education Limited, London.
- Engel, J.F., Blackwell, R.D. & Miniard, P. W. (1990), *Consumer Behavior*, 6th ed., Dryden, Chicago, IL.
- Fitzgerald, G., Russo, N. L. & Stolterman, E. (2002) *Information System Development: Methods in Action*. McGraw-Hill, London.
- Gulliksen, J. & Göransson, B. (2002): *Användarcentrerad systemdesign en process med fokus på användare och användbarhet*. Studentlitteratur, Lund.
- Haverblad, A. (2004): *IT service management i praktiken*, Studentlitteratur, Lund.
- Holme, I. M. & Solvang, B. K. (1997): *Forskningsmetodik, om kvalitativa och kvantitativa metoder*. Studentlitteratur, Lund.
- Jacobsen, D.I. (2002): *Vad, hur och varför? Om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen*. Studentlitteratur, Lund
- Jacobsen, J. K.(1993): *Intervju - konsten att lyssna och fråga*, Studentlitteratur, Lund.

-
- Jiangping, W., Yunfeng & W., Chuwei Z. (2007): *Research on IT Service Management Knowledge Support Structure*. 2007 International Conference on Wireless Communications, Networking and Mobile Computing. pp. 6613-6616
- Kvale, S. (1996): *InterViews: An Introduction to Qualitative Research Interviewing*. Sage Publications. Thousand Oaks, London.
- Lawson B. (2006): *How Designers Think. The design process demystified*. 4th ed. Elsevier, Amsterdam
- Li, L., Mathiassen, L., Keil, M. & Guangzhi Z. (2008): The influence of checklists and roles on software practitioner risk perception and decision-making, *The Journal of Systems and Software*, Vol. 81, pp. 81
- Lycett, M. & Giaglis, G-M. (2000): *Component-Based Information Systems: Toward a Framework for Evaluation*. Citeseer, Brunel University, Uxbridge.
- Macheridis, N. (2005): *Projektspekter – kunskapsområde för ledning och styrning av projekt*, Studentlitteratur.
- Mariani, L. & Pezze, M. (2005) A Technique for Verifying Component-Based Software, *Electronic Notes in Theoretical Computer Science*, Vol. 116
- Monroe, K.B. (1991), *Pricing – Making Profitable Decisions*, McGraw-Hill, New York, NY.
- Nielsen, J. (1993), *Usability Engineering*, AP Professional, Cambridge, MA., USA
- Nordström, M (2005): *Styrbar systemförvaltning*, Linköpings Tryckeri AB, ISBN: 91-85297-60-7
- Ottersten, I. & Berndtsson, J. (2002): *Användbarhet i praktiken*, Studentlitteratur, ISBN: 9144041225
- Peter, J.P. & Olson, J.C.(1993), *Consumer Behavior and Marketing Strategy*, 3rd ed., Irwin, Homewood, IL.
- Preece, J., Rogers, Y. & Sharp, H., (2002), *Interaction Design. Beyond Human-Computer Interaction*, Wiley.
- Qureshi, M.R.J. & S.A. Hussain. (2007): A reusable software component-based development process model, *Advances in Engineering Software*, Vol. 39, Issue. 2.
- Ravald, A. Grönroos, C. (1996): The value concept and relationship marketing. *European Journal of Marketing*, Vol. 30 No. 2, 1996, pp. 19-30.

Robey, D., Boudreau, MC., (2000) Organizational consequences of information technology: dealing with diversity in empirical research. In Zmud, R. W. Editor, *Framing the Domains of IT Management: Projecting the Future... Through the Past*. Cincinnati, Ohio: Pinnaflex Educational Resources.

Rothenberger M. A. & Dooley, K. J., Kulkarni, U. R., Nada N. (2003): Strategies for software reuse: A principal component analysis of reuse practices. *IEEE Transactions on Software Engineering*, Vol. 29 No. 9, pp. 825-837.

Seale, C. (1999): *The quality of qualitative research*. Sage, London.

Siber, J. E. (2001): Protecting Research Subjects, Employees and Researchers: Implications for Software Engineering. *Empirical, Software Engineering*, Vol. 6, No. 4.

Zaltman, G. & Wallendorf, M. (1983), *Consumer Behavior: Basic Findings and Management Implications*, John Wiley and Sons, New York, NY.

Zeithaml, V.A. (1988), "Consumer perceptions of price, quality and value: a means-end model and synthesis of evidence", *Journal of Marketing*, Vol. 52, pp. 2-22