



**EKONOMI  
HÖGSKOLAN**  
Lunds universitet

Institutionen för Informatik

# Analogi & Praktik

---

## Systemutvecklares förhållande till analogier

Kandidatuppsats, 15 högskolepoäng, INFK01 i Informatik

*Framlagd:* Juni, 2008

*Författare:* Andreas Nurbo  
Marie Ohlsson

*Handledare:* Claus Persson

*Examinatorer:* Lars Fernebro  
Agneta Olerup

# Analogi & Praktik

## Systemutvecklarens förhållande till analogier

© Andreas Nurbo och Marie Ohlsson

**Kandidatuppsats framlagd:** juni 2008

**Omfång:** 58 sidor

**Handledare:** Claus Persson

**Examinatorer:** Lars Fernebro  
Agneta Olerup

### *Abstrakt*

*Hur väljer personer som arbetar med systemutveckling att med hjälp av en analogi åskådliggöra sin systemutveckling och hur förhåller sig denna analogi till de utvecklingsmetoder de säger sig använda? Detta har undersökts med hjälp av ett frågeformulär med fokus på utveckling, utvecklingsmetoder och analogier. Undersökningens syfte var att se ifall det fanns ett samband mellan de sju informanternas analogi och den metod som de uppgav att de använde. Majoriteten av de tillfrågade använde sig av en analogi som hade en tydlig koppling till angiven metod. Vid behandling av enkätsvaren har vi observerat att informanternas systemtänkande var att betrakta som hårt och att de analogier som de tillfrågade hört talas om var uteslutande konstruktionsanalogier. Slutsatsen blev att tillämpad metod tillsammans med systemutvecklarens syn på utveckling bestämmer analogin.*

*Nyckelord: Analogi, systemutvecklingsmetod, systemutvecklare*

# Tack!

Under uppsatsens gång har vi stött på problem som vi inte kunnat förutse. Det skulle visa sig att ämnet om analogiers betydelse för systemutveckling som helhet skulle vara svårare att klarlägga än vad vi från början kunde ha föreställt oss. Tack till alla våra informanter på de företag som tagit sig tid att besvara våra frågor. Vi vill även passa på att tacka varandra under den tid som arbetet löpt, där vi trots motstånd lyckats hitta tillbaka igen på rätt spår. Tack familj och vänner som stöttat oss när det känts tungt och kommit med synpunkter som lett till förbättringar. Slutligen vill vi rikta ett stort tack till vår handledare Claus Persson.

Andreas Nurbo, Marie Ohlsson

# Innehåll

Kapitel 1: Inledning .....	1
1.1 Problemdiskussion .....	2
1.1.1 Syfte .....	3
1.1.2 Forskningsfrågor .....	3
1.2 Begrepp .....	3
1.3 Avgränsningar .....	3
Kapitel 2: Teoretisk grund .....	4
2.1 Analogier .....	4
2.1.1 Kategorisering .....	7
2.2 Systemutvecklingsmetodologi .....	8
2.2.1 Varför bör metoder användas? .....	8
2.2.2 Olika typer av metodologier .....	8
2.2.3 Val av metod för systemutveckling .....	10
2.2.4 Filosofisk övertygelse .....	12
2.3 Systemtänkande .....	13
2.3.1 Perspektiv på världen .....	13
2.3.2 Olika system .....	14
2.3.3 Typer av utveckling .....	15
2.4 Sammanbindning av teori .....	16
Kapitel 3: Tillvägagångssätt .....	18
3.1 Teoretiskt ramverk och informationssökning .....	18
3.2 Utformning av frågeformulär .....	19
3.3 Val av informanter .....	20
3.4 Enkätens genomförande och svarandemiljö .....	20
3.5 Analys av empirin .....	21
3.6 Metodreflektion .....	22
Kapitel 4: Presentation och analys av empirin .....	24
4.1 Analys av empirin .....	25
4.1.1 Metodanvändning .....	25
4.1.2 Syn på systemutveckling .....	27
4.1.3 Erfarenhet av och inställning till analogier .....	28
4.1.4 Analogier för systemutveckling .....	29
4.2 Sammanfattning av analysresultat .....	31
Kapitel 5: Diskussion .....	31
5.1 Analysresultatet .....	32
5.2 Iakttagelser .....	33
Kapitel 6 Avslutning .....	35
6.1 Slutsats .....	35
6.2 Tankar kring det hårda .....	35
6.3 Vidare forskning .....	36
6.4 Kommentar .....	36

Bilaga 1 Enkät.....	37
Bilaga 2 Empirisammanställning.....	38
Bilaga 3 Enkät svar .....	42
Referenser .....	57

## Tabeller

Tabell 2:1 Sewell och Sewells roller.....	5
Tabell 2:2 Avison och Wilsons roller.....	5
Tabell 2:3 Avison och Wilsons utvecklingssteg.....	5
Tabell 2:4 Kategorisering av analogier.....	7
Tabell 2:5 Systemteoretisk kategorisering av utvecklingstyper och metoder.....	17
Tabell 4:1 Svar på fråga 5 och 9.....	26
Tabell 4:2 Svar på fråga 7 och 8.....	26
Tabell 4:3 Svar på fråga 11.....	27
Tabell 4:4 Svar på fråga 12.....	28
Tabell 4:5 Svar på fråga 13 och 14.....	28
Tabell 4: 6 Svar på fråga 15.....	29
Tabell 4:7 Svar på fråga 16 och 17.....	30
Tabell 4:8 Sammanställning av enkätsvaren efter genomförd analys. ....	31

# Kapitel 1: Inledning

I en inledande diskussion om tänkbara uppsatsämnen kom vi av en händelse in på en bok av Sewell och Sewell, (2002) *The Software Architect's Profession: An Introduction*. I boken görs en liknelse, en *analogi*, mellan mjukvaruutveckling och att bygga hus. Sewell och Sewell argumenterar för ett antal roller i mjukvaruprojekt med ursprung i byggbranschen; såsom arkitekt, ingenjör och byggare.

- Mjukvaruarkitekten har samma principiella roll som en vanlig arkitekt
- Mjukvaruingenjörens uppgift motsvaras av byggnadsingenjören
- Programmeraren är byggaren

Vår ursprungliga tanke var att ta reda på hur praktiserande mjukvaruutvecklare förhåller sig till byggnadsanalogin, om de var positiva eller negativa till den och undersöka om deras roller och arbetssätt svarade mot analogin. Vid ytterligare diskussion om uppsatsämne framkom tanken, att det fanns risk att ämnet blev för trivialt och att vi därför borde utveckla ämnet mer. Det är ändå lämpligt att göra en kortare introduktion till byggnadsanalogin och diskussionen kring analogier.

Sewell och Sewell är inte de enda förespråkarna för byggnadsanalogin. Det finns även en organisation, The World Wide Institute of Software Architects (WWISA), som verkar för att sprida analogin. Andra förespråkare av byggnadsanalogin är Avison och Wilson fast med lite skillnader i jämförelse med Sewell och Sewell. Avison och Wilson (2001) överför i likhet med Sewell och Sewell (2002) byggindustrins roller till mjukvaruprojektet, men med den skillnaden att arkitektrollen överförs på en redan befintlig yrkesgrupp, nämligen systemvetaren och projektledaren.

Byggnadsanalogin är inte bara ifrågasatt i bloggösfären. Neil McBride (2003) har skrivit ett genmäle på Avison och Wilson artikel, där han tar upp två alternativa analogier som utgår från filmskapande och jazzensemble, som han anser bättre passar in på mjukvaruutveckling. På 2003 års Agile Development Conference hölls ett möte, Analogy Fest, vars enda syfte var att ta fram fler ändamålsenliga systemutvecklingsanalogier (Visible Workings, 2007). Inför mötet skrevs artiklar med temat ”*Om agil utveckling var som X? Hur skulle vi göra saker annorlunda?*”. Analogierna skulle inspirera till nytänkande inom agil utveckling. Analogier är inte det enda tänkbara tillvägagångssättet för att möjliggöra nytänkande. Dahlbom och Mathiassen (1992) skriver att filosoferande kring systemutveckling kan hjälpa personer att skaffa sig teknikerberoende förmågor och underlätta för personer att tänka systematiskt och kritiskt om utveckling.

Studerar vi systemutveckling i praktiken så finner vi att statistiken för projekt inte är särskilt smickrande för branschen.

- 40% av alla projekt försenades
- 40% av alla projekts budgetar överskreds
- 82% är misslyckade ur ett kundnöjdhetsperspektiv

(Projectplace International AB, 2007)

Detta kan vara ett skäl till att på ett mer oortodoxt sätt undersöka hur systemutvecklare betraktar sitt arbete.

## 1.1 Problemdiskussion

Avison och Fitzgerald (2006) tar upp 25 stycken olika metodologier. Avison och Fitzgerald är av åsikten att metodologier är mer än bara metoder, då metodologier även rymmer ett filosofiskt perspektiv. De 25 metodologierna har de identifierat olika, som de kallar det, perspektiv på utveckling. Exempel på sådana perspektiv är följande; ett system som *utnyttjar datorer väl* är en bra lösning och systemet som blir *implementerat tidigt* är en bra lösning.

Redovisning av olika sätt att betrakta systemutveckling saknas i mycket av den litteraturen om systemutveckling som vi kommit i kontakt med. Olika syn på systemutvecklingens kontext ligger dock till grund för kritik av exempelvis byggnadsanalogin, som enligt kritikerna inte tar hänsyn till det dynamiska förhållandet mellan olika parter i ett projekt (McBride, 2003).

Utvecklingsmetoder har sin grund i olika perspektiv på omgivningen, vilket innebär att de valt metodspecifika sätt att förhålla sig till omgivningen. Ett exempel som kan nämnas är *eXtreme Programming*, som försökt hantera intressenters skiftande behov och viljan till snabba resultat genom att utveckla systemet successivt (Chromatic, 2003).

Frågan är dock hur systemutvecklare beskriver sin utveckling med hjälp av analogier i jämförelse med de metoder de använder. Det har, som vi nämnde tidigare, framkommit ett antal analogier under systemutvecklingens historia. Av dessa verkar byggnadsanalogin vara en av de äldsta. Analogin ser systemutveckling som en sekvens av olika faser av färdigställande. Det råder delade meningar om byggnadsanalogin ger en korrekt bild av systemutveckling, vilket diskussioner på fora och bloggar för systemutvecklare visar klart och tydligt (Chen, 2007).

Frågar vi de ständiga diskussionerna kring analogiers vara eller icke-vara och flyttar fokus till förhållandet mellan systemutvecklarens systemutvecklingsanalogi och vad de faktiskt gör, så tror vi att detta kan leda till intressanta iakttagelser. Det kan vara så att analogier representerar utveckling som den *bör* vara, inte hur den *är*. Detta kan vara en av anledningarna till varför analogier ger upphov till livliga diskussioner. Kopplingen mellan hur systemutvecklare åskådliggör sin utveckling med hjälp av analogier och vad de uppger att de gör kan också vara intressant för studier rörande metodanvändning. Att beskriva sitt arbete med hjälp av analogier kan leda till nytänkande, vilket kan resultera i en bättre användning av metoder och verktyg.

Därmed anser vi att en undersökning rörande, och diskussion kring, ”bilden av verkligheten” kan fylla många funktioner.

### **1.1.1 Syfte**

Vi vill se ifall det finns ett samband mellan hur systemutvecklare åskådliggör sin systemutveckling med hjälp av analogi och hur de uppger sig arbeta. Samt hur de betraktar systemutveckling för att bättre förstå de konflikter som uppstår vid användning av analogier som medel för presentation av systemutvecklingsidéer och utvecklingsprocesser.

### **1.1.2 Forskningsfrågor**

För att kunna uppfylla vårt syfte formulerade vi följande huvudfråga som behövde besvaras.

*Hur överrensstämmer systemutvecklarens analogi med hur denne uppger sig arbeta och betrakta systemutveckling?*

Denna fråga hade i sig två delfrågor som behövde besvaras innan en jämförelse var möjlig. För att kunna studera förhållandet mellan analogi och praktik behövde vi få fram empiri som besvarar:

Hur definierar systemutvecklare utvecklingsprocessen och vilka metoder använder de?  
Hur beskriver systemutvecklare systemutvecklingsprocessen med hjälp av analogi?

## **1.2 Begrepp**

Vi använder i huvudsak två begrepp i detta arbete, *analogi* samt *praktik*. Med analogi avser vi en *liknelse mellan ett fenomen och ett annat*. Praktik använder vi i betydelsen *hur det är i verkligheten*. Praktik består i detta arbete av två delar. Systemutvecklarens faktiska utveckling samt hur de ser på utveckling. Med systemutvecklare avser vi personer som är delaktiga i framtagningen av ett system. Alltså inte bara systemarkitekter och liknande.

## **1.3 Avgränsningar**

Vi har endast frågat systemutvecklare med svensk utbildning och som skaffat sig merparten av sin erfarenhet av systemutveckling i Sverige. Anledningen till denna avgränsning är att det kan förekomma kulturella skillnader. Vi har även gjort en avgränsning med avseende på systemutvecklingsprocessen i praktiken och fokuserar där endast på använda metoder och inte det dagliga arbetet.



## Kapitel 2: Teoretisk grund

Detta kapitel presenterar de två huvuddelarna i vårt arbete, analogier och metodologier/metoder. Först går vi igenom de vanligast förekommande analogierna samt kritik som framkommit mot vissa av dem. Därefter går vi igenom olika kategorier av metodologier som vi senare kommer använda som underlag för vår analys. Efter genomgången av metodologier så presenterar vi den systemtänkande som ligger till grund för vår analys av empirin. Kapitlet avslutas med att analogi- och metodologidelen knyts samman med systemtänkande, samt hur systemtänkande kommer att användas i vår analys av det empiriska underlaget.

### 2.1 Analogier

Det finns ett antal analogier inom systemutvecklingsdiskursen som på olika sätt försöker åskådliggöra systemutvecklingsprocessen och de olika rollerna som finns inom den. Kunskap om olika populära systemutvecklingsanalogier är bra för att bättre kunna analysera de svar som framkommer i intervjuerna. En av de vanligast förekommande analogierna och den som verkar ha funnits med längst är byggnadsanalogin. Byggnadsanalogins närmaste konkurrent är jazzanalogin. Vi presenterar även filmanalogin som är en mer komplex analogi och en konkurrent till jazzanalogin. Förutom dessa större analogier förekommer det ett antal mer eller mindre spridda tankegångar vilka vi avslutar analogikapitlet med.

#### *Byggnadsanalogin*

Det som skiljer byggnadsanalogin från andra analogier såsom jazzanalogin, är att det finns tydliga roller inom byggsektorn att inspireras av. Sewell och Sewell (2002) förespråkar införandet av mjukvaruarkitekt i projekt. Mjukvaruarkitekt motsvarar den funktion som arkitekter har inom byggsektorn (Se Tabell 2:1). Till skillnad från Avison och Wilson (2001) så skriver de att arkitekten ska agera projektledare (Sewell & Sewell, 2002). Avison och Wilson (2001) skriver att rollen ska delas upp i två, projektledare som har hand om bland annat budget och tidsplanering och systemvetaren som designar systemet (Se Tabell 2:2).

Tabell 2:1 Sewell och Sewells roller

Byggnadssektorn	Systemutveckling
Arkitekt	Mjukvaruarkitekt
Byggnadsingenjör Byggare	Mjukvaruingenjör Programmerare

Tabell 2:2 Avison och Wilsons roller

Byggnadssektorn	Systemutveckling
Husköpare Arkitekt	Användare Projektledare Systemvetare
Byggnadsingenjör Byggare	Mjukvaruingenjör Programmerare

Avison och Wilson (2001) anser att det finns tre steg vid konstruktion av hus (Se Tabell 2:3). Första steget är att definiera kraven;

En person som vill ha ett nytt hus kontaktar en arkitekt. Kunden dokumenterar och beskriver vilka grundläggande krav som denne har på huset och arkitekten ställer frågor baserat på sin erfarenhet och kunskap för att få en mer detaljerad kravspekifikation. Denna information ligger sedan till grund för ett första utkast. Förslaget kommenteras och ändras fram tills dess att kunden är nöjd och därmed godkänner planritningen. (Avison & Wilson, 2001)

Tabell 2:3 Avison och Wilsons utvecklingssteg

Steg	Beskrivning
1	Definition av kraven
2	Design
3	Konstruktion (programutveckling)

Det andra steget, kännetecknas av att ritningen överlämnas till en byggnadsingenjör. Denne gör en detaljerad designspecifikation baserad på planritningen och tar fram specifikationerna för byggmaterial utifrån de regler som finns. I detta steg identifieras även problem och valmöjligheter som diskuteras med kunden, som får bestämma vad som ska göras. När specifikationen är klar får kunden godkänna alternativt komma med förslag på ändringar av specifikationen. (Avison & Wilson, 2001)

Det tredje steget är att bygga huset. Byggaren kan identifiera problem och valmöjligheter som sedan diskuteras med kunden. När huset är färdigbyggt får kunden inspektera och påpeka brister som rättas till. När allt är klart kan köparen flytta in. (Avison & Wilson, 2001)

### Jazzanalogin

Kritiker av byggnadsanalogin brukar påpeka, att analogin inte tar hänsyn till det dynamiska i systemutveckling eller osäkerheten. Av den anledningen har det uppkommit alternativa analogier. Dessa har dock inte lika stor uppbackning som förespråkarna av byggnadsanalogin. En av de större konkurrenterna verkar vara jazzanalogin<sup>1</sup>. Den har tillkommit efter kritik av

<sup>1</sup> Vårt intryck utifrån läsning av litteratur som berör ämnet.

byggnadsanalogin och brukar lyftas fram av personer som är anhängare av den agila systemutvecklingen (Visible Workings, 2003). Förespråkarna anser att systemutveckling är mer likt jazzimprovisation eftersom det är en dynamisk process med osäkra element (McBride, 2003).

Jazzimprovisation baseras på en uppsättning av melodier och rytm. Melodi och rytm skiftas sedan i ett samspel mellan bandmedlemmarna. Musikerna i jazzensemblen kan variera och experimentera eftersom den musikaliska strukturen är känd. De musicerar också inom en given ram av sociala normer med hög interaktion mellan parterna vilket möjliggör för personerna inom gruppen att ändra sin roll i förhållande till varandra (McBride, 2003; Sprez). Detta sker enligt McBride (2003) med hjälp av minimala strukturer som både kan vara sociala och tekniska. De sociala innefattar kommunikation, ledarskap och kulturella attityder till risktagande. De tekniska utgör repertoar, sångmallar, instrumentell kunskap och förmågan att reagera på medlemmarna i gruppen och publiken. I improvisationsjazz produceras musiken både gemensamt och med solon. Varje framträdande blir unikt utifrån de förutsättningar som finns (Sprez). Det finns ingen ledare som enhälligt fattar beslut utan musiken växer fram i samverkan utifrån de givna ramarna (McBride, 2003; Sprez)

### *Filmanalogin*

Avison och Wilson (2001) har förutom byggnadsanalogin även föreslagit filmproduktion som en tänkbar analogi, men de ger tyvärr ingen djupare beskrivning av den. Det gör dock McBride, en av förespråkarna av jazzanalogin. McBride (2003) skriver att det finns stora likheter mellan systemutveckling och filmproduktion. Han anser att filmanalogin belyser den dynamiska processen och den kreativa interaktionen mellan alla inblandade bättre än byggnadsanalogin. Filmanalogin tycks ha haft mindre genomslag än jazzanalogin.

McBride (2003), beskriver den filmskapande processen på följande sätt. Första steget är ett manus som baseras på en ny idé eller andra litterära källor. Manuset är grunden för filmen men det kan komma att ändras under hela filmningsprocessen. Det tolkas nämligen fortlöpande i filmprocessen av ett flertal personer med kreativa roller som till exempel regissörer och skådespelare. Innan filmningen kan påbörjas måste budget, tidsplanering, design av kulisser och rekvisita med mera bestämmas. Under inspelningens gång kan problem med manus upptäckas. Det kan även hända saker som inte går att ändra såsom väderförändringar, vilket kan leda till att manuset behöver ändras och även var inspelningen bör ske. Inspe­ling är ett kontinuerligt samspel mellan regissör, skådespelare, kameramän med flera genom hela filmningsprocessen. Efter inspelningen måste filmen editeras. Slutresultatet, filmen, är en produkt av ett antal människors interagerande (McBride, 2003).

### *Trädgårdsanalogin med flera*

Förutom de tidigare beskrivna analogierna finns det ett flertal andra som används i mer eller mindre utsträckning. Att utveckla mjukvara kan till exempel liknas vid att sköta en trädgård, bygga broar, driva restaurang, tillverkning, skapa ett uppslagsverk och så vidare (New Analogies For Software, 2007). Även mer extrema liknelser förekommer, som att jämföra mjukvaruutveckling med att "ostrukturerat" slå och sparka och kampsport. Dessa vill visa på

behovet att kombinera kodknackande med struktur och vetenskapligt angreppssätt för att uppnå effektivitet (Martial Arts As Software Development Metaphor, 2007). Vissa analogier är väldigt kortfattade och fokuserar endast på enstaka aspekter av systemutveckling. Ett exempel på detta är uppslagsverksanalogin. Varje ”artikel” skrivs separat och blir granskad medan den skrivs, den blir sedan kontrollerad igen och omstrukturerad för att passa ihop med alla de andra artiklarna i uppslagsverket (Creating An Encyclopedia Metaphor, 2007).

Många av analogierna försöker komma runt de problem som debattörerna anser finns med exempelvis bygg- och jazzanalogin. Trädgårdsanalogin lyfter fram behovet av kontinuerlig utveckling och underhåll av mjukvaran något som oftast inte diskuteras i samband med bygganalogin (Barnes, 2007). Trädgårdsanalogin vill framhäva att en mjukvara aldrig blir färdig.

Det finns andra analogier med likartade perspektiv som byggnadsanalogin men med fokus på andra sektorer inom byggbranschen. En av dessa är ”bygga bro”- analogin, som har många likheter med att byggnadsanalogin. Den har samma grundläggande konstruktionssyn. Analogier har även gett upphov till satiriska artiklar om nya sätt att bedriva systemutveckling på. En sådan artikel är Agile Bridge Building som beskriver det nya och gamla sättet att bygga broar på (Repici, 2007). Artikeln fokuserar på kampen mellan det gamla sättet bygga broar på jämfört med det nya agila sättet. Artikeln driver med den ”kamp” som finns mellan olika sätt att utveckla system. Den fokuserar på det nya agila och flexibla jämfört med det gamla och träiga sekventiella sättet som till exempel vattenfall.

### 2.1.1 Kategorisering

De olika analogierna har vi valt att kategorisera som *konstruktion*, *interaktion* eller *fokus*. Dessa grupper har vi själv tagit fram för att underlätta indelningen av olika analogier utifrån deras utmärkande drag. De som har typiska drag av byggande, det vill säga steg för steg arbete, har vi valt att beteckna som konstruktionsanalogier. Interaktionsanalogi benämner vi de analogier som sätter fokus på människorna i ett projekt och hur dessa samverkar och påverkar varandra, interagerar, i en dynamisk miljö. Här har vi placerat jazzanalogin och även filmanalogin. Anledningen till att vi väljer att placera filmanalogin här är pågrund av att den lyfter fram människors interagerande som det centrala, alltså hur skådespelarna förhåller sig till varandra. Den sista kategorin är fokusanalogierna. Dessa pekar på ett eller två specifika element eller uppgifter i systemutvecklingsprojekt. Här har vi placerat trädgårdsliknelsen vars fokus är underhåll, kampsport med fokus på formalisering och uppslagsverk med fokus på kodgranskning (Se Tabell 2:4).

Tabell 2:4 Kategorisering av analogier

<b>Konstruktionsanalogi</b>	<b>Interaktionsanalogi</b>	<b>Fokus</b>
Byggnadsanalogin	Jazzanalogin	Trädgårdsliknelse
Bro byggande	Filmanalogin	Kampsport
		Uppslagsverk

## 2.2 Systemutvecklingsmetodologi

Vad är egentligen en metodologi? Nationalencyklopedin (NE.se, 2007) förklarar innebörden av ordet enligt följande ”*planmässigt eller vetenskapligt förfarande*”. Vidare förklarar ordet metodik som, ”*olika vetenskapers tillvägagångssätt för att vinna kunskap eller lösa problem*”. Innebörden av ordet metodologi är omtvistat bland forskare och ordet metodologi har kommit att användas synonymt med ordet metod (Fitzgerald, Russo, & Stolterman, 2002). Vi väljer att i detta kapitel använda ordet metodologi som en kategori innehållande metoder.

Larsson (2004) anser att en metodologi skapas av metoder och därtill även ett ramverk för i vilka situationer dessa metoder bör användas. Vidare kan systemutvecklingsmetodologier ha olika perspektiv till exempel betona processer, data eller objekt och ha olika syn på, i vilken ordning faser och steg ska utföras. Detta resonemang kring metodologi stämmer väl med Crispen och Stuckey (1994), som väljer att i en artikel beskriva hur en metodologi ska understödja mjukvarudesigners i deras arbete

*A methodology should spell out general steps to follow. It should be specific enough to give guidance but be general enough to apply to most software situations. It should not be taken as a step-by-step way to develop entire systems; these recipes for how to get the work done simply do not exist.*  
(Crispen & Stuckey Jr, 1994, p. 272)

### 2.2.1 Varför bör metoder användas?

Janhager (2006) skriver, att genom användandet av metoder formaliseras och struktureras arbetet. Detta ger utvecklarna möjlighet, att lättare precisera tankar och fokusera på samt hantera de problem som framkommit under utvecklingen. Metoder underlättar för utvecklarna att ta fram bättre lösningar. Utöver detta säkerställs att viktiga aspekter beaktas som till exempel en bra dokumentation, vilket är viktigt för framtida underhåll och projekt samt som ett sätt att främja samarbetet i utvecklingsteamet. Argument mot användandet av metoder är att de riskerar att hämma kreativiteten och att fokus istället läggs på dokumentation, planering och ledning istället för de problem som uppstår. Holme och Solvang (1997) skriver, att en metod är att betrakta som ett redskap, det vill säga ett sätt att lösa specifika problem. Tillämpning av metod är nödvändigt men är inte tillräckligt utan grundläggande kunskap och förståelse av metoden hos utvecklarna.

### 2.2.2 Olika typer av metodologier

Avison och Fitzgerald (2006) har identifierat sex teman som inom olika metodologier, organisatoriska, människor, modellering, snabb och evolutionär utveckling, ingenjör samt extern utveckling. Av dessa så är modellering samt snabb och evolutionär utveckling de som är intressanta att lyfta fram i detta arbete. Förutom dessa så har vi valt att ta med Systems Development Life Cycle även kallat vattenfallsmodellen.

### *Vattenfallsmodellen*

Vattenfallsmodellen är en av de först publicerade modellerna inom mjukvaruutvecklingsprocesser och härstammar från ingenjörprocesser (Sommerville, 2001). Vattenfallsmodellen är en linjär utvecklingsprocess, vilket innebär att processen delas upp i ett antal steg som avlöser varandra. Resultatet av varje del måste godkännas innan nästa fas påbörjas då stegen informationsmässigt överlappar varandra. Problem som påpekats med vattenfallsmodellen är att utvecklingsprocessen inte är flexibel utan sekventiell och fasberoende. Överenskommelser måste av den anledningen göras tidigt i processen, vilket gör det svårt att senare göra ändringar om nya omständigheter skulle uppstå (Sommerville, 2001). På så vis menar Sommerville att vattenfallsmodellen endast är användbar då förutsättningarna är kända.

### *Snabb och Evolutionär utveckling*

Evolutionär utveckling är, till skillnad från vattenfallsmodellen, ingen linjär utvecklingsprocess där varje fas låses innan nästa påbörjas. De kännetecknas av att utvecklingen sker i iterationer och att systemets utveckling är inkrementell (Avison & Fitzgerald, 2006). Det finns olika varianter på evolutionär utveckling såsom prototyping, Rapid Application Development (RAD) och agil utveckling. Inkrementell utveckling, eller stegvis utveckling som den också kallas, kom på förslag 1980 (Sommerville, 2001). Syftet var att minska på den ständiga omarbetning som ofta förekom i utvecklingsprocessen. Den skulle även ge beställaren möjlighet att fördröja beslut gällande omgivningen i vilket systemet skulle verka tills denne skaffat sig bättre förståelse om systemet (Sommerville, 2001). Stegvis utveckling innebär att det först konstrueras en kärnprodukt med grundläggande funktioner. Produkten förbättrats sedan inför varje version tills systemet motsvarar beställarens önskemål.

Prototyping kom till under tidigt 1980-tal för att motverka de problem som fanns med vattenfallsmodellen (Chen, 2001, enligt Avison & Fitzgerald, 2006). Prototyping är en process där, i vissa fall avskalade, versioner av systemet tas fram så att beställaren har möjlighet att testa det under utvecklingen (Avison & Fitzgerald, 2006). Det finns två typer av prototyper evolutionära samt throw-away (kasta bort). Evolutionära prototyper utvecklas till att bli det egentliga systemet medans throw-away bara är till för att se ifall beställarens krav uppfyllts. Prototypen gör det möjligt att se hur den slutliga produkten kommer att se ut och till följd av det ger det möjlighet att ändra tidigare krav och önskemål (Avison & Fitzgerald, 2006).

De agila metoderna började ta form under 1990-talet. Exempel på tidiga agila metoder är Scrum, Crystal, DSDM (Dynamic Systems Development Method) och eXtreme Programming (XP). Den tidigare, mer traditionella mekanistiska världssynen, hade nu menar Sridhar och VenuGopal (2007), kommit att utmanas genom ett nytt så kallat agilt perspektiv. Där utvecklingsmetoder fokuserar på att erbjuda högt kundvärde genom tre principer; *snabb leverans av kvalitativ mjukvara, aktivt deltagande av berörda intressenter och märkbara förändringar till det bättre*. Enligt den agila utvecklingsteorin så är krav svåra att ta fram så dessa utvecklas allt eftersom i kombination med någon form av prototyper (Avison & Fitzgerald, 2006). Den agila utvecklingen välkomnar förändrade förutsättningar och planerar för detta genom iterationer med korta livscykler. Agil utveckling betonar att det är viktigare med individer och hur dessa interagerar

med varandra i förhållande till processer och verktyg (Avison & Fitzgerald, 2006). Exempel på agila utvecklingsmetoder är SCRUM och eXtreme Programming.

### *Modellering*

En modell är en representation av den verkliga världen. Enligt Avison och Fitzgerald (2006) finns det i huvudsak tre sätt att modellera inom systemutveckling dessa är; processmodellering, datamodellering samt objektmodellering. Det som kännetecknar modellering är att det går att bryta ner komplexa problem till mindre och mindre delar tills de motsvarar ett fåtal rader kod. Processmodellering beskriver den logiska analysen av processer och datamodellering strävar efter att modellera data från den verkliga världen (exempelvis en organisation) som representerar ”de fundamentala byggstenarna i system” (Avison & Fitzgerald, 2006). Den typ av modellering vi kommer att fokusera på är objektmodellering som slår ihop process och datamodellering till ett kombinerat objekt.

Ett objekt representerar något i den verkliga världen som har data (attribut) men kan också ha eller utföra diverse handlingar (Avison & Fitzgerald, 2006). Objekt kan också modelleras hierarkiskt och bestå av andra objekt. Dessa objekt kan representera olika saker i den verkliga världen såsom människor och olika objekt som interagerar. Exempel på metodologier är Object-orienterad analys samt Rational Unified Process (RUP). RUP är en arkitektur fokuserad process som jämför mjukvaru arkitektur med byggnadsarkitektur och den ritning som finns över en byggnad (Avison & Fitzgerald, 2006). RUP är även inkrementell och iterativ då dess skapare inte tror att krav kan definieras fullt ut i början av ett projekt.

### **2.2.3 Val av metod för systemutveckling**

En mjukvaruprocess består av ett antal aktiviteter som tillsammans leder fram till ett resultat, en mjukvaruprodukt. Mjukvaruutveckling är en komplex verksamhet som i likhet med all annan intellektuell verksamhet grundar sig på mänskliga bedömningar (Sommerville, 2001). Detta har lett till försök att automatisera delar av utvecklingsprocessen, vilket många gånger leder till bristfälliga resultat. Svårigheterna med de automatiserade utvecklingsprocesserna, menar Sommerville, ligger i att de metoder som används är dåliga på att fånga helheten. Ofta beror detta på att metoderna består av gammalmodiga tekniker och inte tar fördel av vad *best practice* (Sommerville, 2001). Organisationer och/eller systemutvecklare kan ha skilda synsätt på utvecklingsprocessen, detta gör att det inom en och samma organisation kan förekomma olika utvecklingsmetoder. Dessa metoder har enligt Sommerville följande aktiviteter gemensamt; *specificering, design, implementering och utvärdering* samt *testning*.

Sörman (2005) liknar systemutvecklingsprocessen vid en verktyglåda som ger användaren förutsättningar till att skapa en bättre struktur i utvecklingsarbetet. Metoder kan inte användas rakt av utan måste anpassas efter vad som ska göras. Genom att göra ett aktivt val av metod så läggs en bättre grund för utvecklingsarbetet (Mathiassen, Munk-Madsen, Nielsen, & Stage, 2001). Avison och Fitzgerald (2006) anser att valet av metod är så pass avgörande vid skapandet av ett användbart system, att de konstruerat ett ramverk som underlättar jämförelse samt utvärdering av olika metoder. Ramverket hjälper utvecklarna att göra ett mer underbyggt val av

metod. Ramverket innefattar sju element *philosophy, techniques and tools, scope, outputs, practice* och *product*. Ramverket är generellt och kan användas till att utvärdera systemutvecklingsmetoder samt till att bedöma värdet av andra metoder och discipliner. Av Avison och Fitzgerald (2006) sju element väljer vi att belysa *philosophy* eftersom det elementet har stor vikt för vår undersökning.

### *Systemutvecklingsmetoders filosofi*

Filosofin kan vara öppen, men i de flesta metodologier är den underförstådd, detta beror till stor del på att författarna av metodologierna sällan betonar dess filosofi, dock sägs det att alla metodologier grundar sig i en slags allmängiltigt synsätt. Avison och Fitzgerald (2006) menar att filosofibegreppet är att betrakta som ett antal principer, vilka tillsammans utgör metodologins bas. Dessa principer är *paradigm, objectives, domains* och *applications*.

Med paradigm menas att det går att identifiera två mönster som är av intresse, *vetenskapsmönstret* och *systemmönster* (Avison & Fitzgerald, 2006). Mönstret definieras "as a specific way of thinking about problems, encompassing a set of achievements which are acknowledged as the foundation of further practice." (Kuhn, 1962, enligt Avison & Fitzgerald, 2006, p. 598).

Vetenskapsmönstret innebär att helheten bryts ner i mindre delar som därefter undersöks i hopp om att öka förståelsen. Vetenskapsmönstret har haft en lång och framgångsrik historia men skulle komma att utmanas av det så kallade systemmönstret (Avison & Fitzgerald, 2006). Systemmönstret utvecklades som en reaktion på det vetenskapliga mönstret som mer var av reduktionismisk karaktär det vill säga ett synsätt ofta förknippat med orsakssamband. Systemmönstret ville till skillnad från vetenskapsmönstret "fånga hela bilden" det vill säga summan av alla delar och samverka dem i mellan.

Vetenskapsmönstrets bristande förståelse och oförmåga att hantera "levande system" vilket är mindre delsystem som tillsammans bildar en helhet och "människa aktivitetssystem" som innebär att systemet interagerar med människor (Avison & Fitzgerald, 2006). Dessa system har gemensamt problematiken att dess innebörd inte alltid kan förklaras i minsta beståndsdel och/eller detalj. Det är inte möjligt att bryta ner systemen i mindre bitar var för sig som individuella komponenter, menar Checkland (1981) och skriver att "helheten är större än summan av dess delar".

En tydlig ledtråd till en metodologis filosofi är de uppsatta målen, ledtråden visar till exempel vad som prioriteras i den valda metoden, det som Avison och Fitzgerald (2006) benämner *objectives*. Medan vissa metodologier fokuserar på datoriserade aspekter, finns det metodologier som har ett bredare perspektiv och därmed även väger in aspekter som inte lika lätt kan kontrolleras till exempel mänskliga aspekter. Skillnaderna i fokus bland metodologier bör beaktas av dem som bestämmer sig för att använda en viss metod eftersom detta kommer att avgöra och sätta gränserna av problemområdet och därmed ramarna för projektet.



Området varifrån metodologin har sitt ursprung och därmed ”satt sin prägel” är Avison och Fitzgerald (2006) *domains*. Tidiga metoder som till exempel vattenfallmodellen såg som sin uppgift att lösa, om inte inom ett specifikt område så åtminstone ett mycket begränsat sådant. Många gånger blir lösningen av sådana problem väldigt impulsiva och fungerar därför på sikt väldigt dåligt. Oftast uppstår ganska snart nya problem men som i grunden fortfarande härstammar från den tidigare, otillräckliga lösningen. Det är vanligt att de nya problemen uppstår då annan kringutrustning integreras med den redan befintliga.

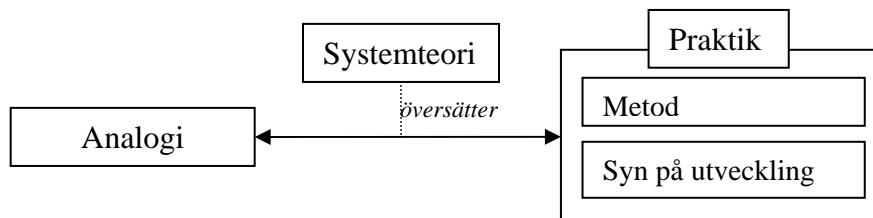
Även om utvecklingen av lösningar till olika problem har blivit väl koordinerat och att senare system har utvecklats med hjälp av den kunskapen är det trots det oftast så att lösningen på ett visst problem ger upphov till ett nytt (Avison & Fitzgerald, 2006). Detta lett till att många metoder anammar olika utvecklingsfilosofier. Dessa metoder tar ett mycket vidare perspektiv redan vid uppstarten av ett projekt, vilket innebär att det till en början inte fokuseras på att lösa specifika problem. *Target* är själva syftet med metodologin (Avison & Fitzgerald, 2006). En del metodologier är mer specifikt inriktade än andra. Metodologier kan vara särskilt bra på vissa typer av problem, omgivning, särskilda branscher och storlek på verksamheten.

Ämnet om metodologiers filosofiska underton är komplext (Avison & Fitzgerald, 2006). Detta gör i sin tur att det är viktigt att ha en diskussion kring valet av metod. De betonar vikten av att utvecklare bör vara medvetna om ”filosofibegreppet” och försöka välja en metod som matchar den egna övertygelsen. Det är viktigt att utvecklarens tankar kring hur ett uppdrag ska genomföras är den samma som hos författarna till den eller de valda metoder som används.

#### **2.2.4 Filosofisk övertygelse**

*Top-down* och *bottom-up development* är två filosofiska övertygelser om hur system bör utvecklas (Pizka & Bauer, 2004). *Top-down development* innebär att uppsatta krav bryts ner till en nivå som går att implementera. Kända metoder som har *top-down* perspektiv är *vattenfallsmodellen* och *Rational Unified Process* (RUP). I *bottom-up development* påbörjas utvecklingen från den tekniska basen som ska användas. Från denna nivå arbetar utvecklarna sig uppåt i nivåer av service och komponenter tills de uppsatta kraven nås. Metoder som har ett *bottom-up* perspektiv hör till de agila utvecklingsmetoderna. Båda utvecklingsfilosofierna går att använda, vad som skiljer dem åt är sättet att se på utvecklingsprocessen. Det slutliga valet bör inte underskattas, eftersom det har en avgörande roll på systemets kvalitet. Åtskilliga exempel visar på, skriver Pizka och Bauer (2004), att *bottom-up development* är att föredra, då denna teknik underlättar senare vidareutveckling.

## 2.3 Systemtänkande



Figur 2:1 Förhållandet mellan de teoretiska delarna analogi och praktik i vårt arbete

Vi har tidigare i detta kapitel introducerat två sätt att beskriva systemutveckling på. För att kunna studera förhållandet mellan dessa så introducerar vi här systemtänkande. Systemtänkandet använder vi som en gemensam vokabulär och kategorisering. Systemtänkandet ligger även till grund för den ena delen av praktiken som var *syn på utveckling*. Genom att använda oss av systemtänkandet kan vi prata om förhållandet mellan *metoder*, *syn på utveckling* och *analogier* på samma sätt och möjliggöra analys på likvärdig grund. Vi har valt att hålla oss på en lägre teoretisk nivå för att undvika att introducera komplexitet som inte är nödvändig för den nivå av förhållandet som vi valt att studera.

### 2.3.1 Perspektiv på världen

Det finns två grundläggande sätt att betrakta världen på, *mekanistiskt* och *romantiskt*, men även kombinationer av dessa två synsätt finns också (Dahlbom & Mathiassen, 1993). Det mekanistiska sättet betraktar världen som en maskin. Eftersom en maskin arbetar enligt bestämda regler så kan vi också emulera denna i en dator. Det vill säga, kan vi specificera världen kan vi skriva ett program som återger den på ett korrekt sätt. Kännetecknande för det mekanistiska synsättet är att det antar att det går att få ett korrekt perspektiv på världen, det är möjligt att representera världen så nära verkligheten som möjligt. Grunden i detta tankesätt är en idé om att världen är stabil. Betraktas ett fenomen som en maskin så ligger det nära till hands att försöka dela upp det i mindre delar. Detta för att kunna uttolka vilka regler fenomenet följer. Är reglerna som styr kända så kan vi kontrollera maskinen. Det mekanistiska sättet har haft stor påverkan på hur vi betraktar programmering och systemutveckling (Dahlbom & Mathiassen, 1993).

Det mekanistiska perspektivet på systemutveckling har ett antal kännetecken. Formalisering är ett av dessa och innebär en beskrivning av vad som ska göras och hur det ska göras. Formella metoder som beskrivs i (Systemutvecklingsmetodologi 2.2) har ett mekanistiskt arbetssätt, men i de fallen är det för att säkerställa kvaliteten på det som utvecklas (Dahlbom & Mathiassen, 1993). Andra kännetecken på mekanistiskt tänkande är vikten av planering, tydliga och väldefinierade scheman för att kunna få fram bra mjukvarukvalitet. De mekanistiska metoderna utgår från att det finns en ordning och det går att ha kontroll.

Det andra sättet att betrakta världen på är det romantiska. Detta synsätt skiljer sig en hel del från det mekanistiska då det antar att allting förändras, världen är dynamisk. Det finns inget generellt fall på grund av världens föränderliga natur, varje problem är unikt. Det romantiska synsättet kännetecknas av att det inte är mängden information som är det viktiga utan förståelsen av den. För att förstå sakers natur så måste saker studeras på djupet så att de underliggande dolda förklaringarna hittas. Vilket i praktiken innebär att systemutvecklare måste se bortom det mekaniska, som ordning, stabilitet och kontroll, och studera maktkamper, arbetssätt som förändras och det kommande kaoset (Dahlbom & Mathiassen, 1993).

Det romantiska arbetssättet börjar med att ta fram en tolkning av världen, vilket för systemutvecklare innebär att studera exempelvis en verksamhet. Eftersom denna form av studie är subjektiv kan resultatet bli olika beroende på vem som studerar, i detta fall verksamheten. Detta skiljer sig från det mekanistiska som utgår från att det går att få fram en sann representation av världen.

### **2.3.2 Olika system**

Checkland (1981) beskriver två sätt att tänka kring system, *hårt* och *mjukt systemtänkande*. Hårt systemtänkande betraktar världen som ett stabilt system där idealet är att skapa en sann modell av världen. Det betraktar system som hierarkiskt organiserade delar, där en nivå kan beskrivas av delarna som utgör den underliggande nivån, ett system är summan av dess delar. Detta leder till, för att förstå helheten, att det måste finnas förståelse för den interna strukturen i systemet. Det hårda systemtänkandet anser att ett system är *väldefinierat* och kan betraktas som ett *objekt*. (Checkland, 1981; Dahlbom & Mathiassen, 1993). Detta tänkande utgår från att det går att modellera all information om ett system som krav genom att använda exempelvis Unified Modeling Language (UML).

Det mjuka systemtänkandet antar att världen kan representeras som ett antal system (Checkland, 1981). Idealet är att lära sig genom att konfrontera olika system med varandra (Dahlbom & Mathiassen, 1993). Till skillnad från hårt systemtänkande betraktar det mjuka system som en helhet med framväxande "uppgifter". Mjukt systemtänkande anser att det är opraktiskt att representera världen som "hårda system" då det inte är möjligt att modellera allt genom kravspecifikationer (Dahlbom & Mathiassen, 1993). Det finns andra saker som en systemutvecklare måste ta hänsyn till, till exempel bristande konsensus inom en organisation. Precis som det romantiska perspektivet, så utgår det mjuka systemtänkande från att världen består av olika människor som gör saker och att alla har sin egen tolkning av världen. Systemutvecklare måste välja en bit av denna värld och kalla det system. För att designa system behöver alla aktörer involveras. Varje aktör ger upphov till ett system eftersom varje tolkning av vad en aktör anser ska göras blir ett eget system. Den information som genereras analyseras med hjälp av en rationell process. Denna rationella process är iterativ och som genom diskussion leder till en delad förståelse av vad det är för system som ska utvecklas (Dahlbom & Mathiassen, 1993).

Dahlbom och Mathiassen (1993) introducerar *dialektiskt systemtänkande* som en kombination av det hårda och mjuka systemtänkandet. Dialektiskt systemtänkande utgår från att världen inte är

stabil, harmonisk, välstrukturerad eller rationell. Världen är under ständig förändring. Medan de andra sätten att tänka kring system leder till ett stort antal olika system försöker det dialektiska systemtänkande få fram ett system genom att studera de motsatsförhållanden som förekommer. Motsatsförhållanden är till exempel arbetarnas önskan om kortare arbetstid i motsats till ledningens vinstmål. Fenomenens motsatser måste inte vara som födsel/död, svart eller vitt. Genom att studera olika åsikters och fenomenets motsatser så kan utvecklare komma fram till en ny förståelse av helheten som inkluderar delarna av både för och emot argument.

### **2.3.3 Typer av utveckling**

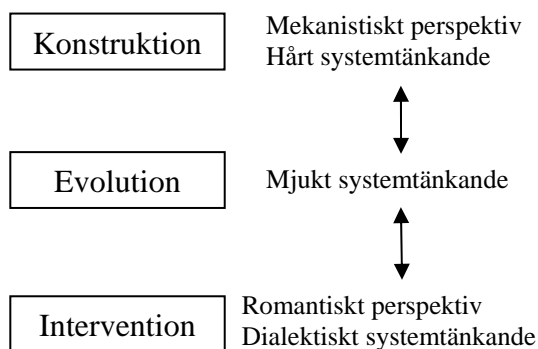
Det finns två kända, något förenklat, sätt att utveckla system på; *konstruktion* och *evolution*. Konstruktion är en stegvis process som tar sin början med analys och beskrivning av problemet i kravspecifikationer. Konstruktionsprocessen drivs av specifikationer med början på problemnivå ner till maskinnivå (Dahlbom & Mathiassen, 1993). Användare har en passiv roll i konstruktionsprocessen, deras enda syfte är att tillhandahålla information och godkänna beslut. Användarna intervjuas för att få fram information så att systemutvecklarna bättre kan formulera kravspecifikationer och kunna hitta lösningar utan att aktivt blanda in användarna. Systemutveckling genom konstruktion är ett byråkratiskt sätt att angripa problemet. Faser, dokumentationsstandarder och strukturerade tekniker är exempel på ett byråkratiskt tänkande. Det kännetecknas av fasta, på varandra beroende, faser undersökning av problemområde, analys, övergripande design och så vidare. Livscykelmodeller och metoder är exempel på konstruktionssynsätt. Konstruktionsprocessen är en top-down approach och är bra i de fall då problemet är väldigt komplext men osäkerheten låg (Dahlbom & Mathiassen, 1993).

Evolutionär utveckling använder sig av en bottom-up approach där lösningar på partiella problem ska leda fram till ett helt system. Detta angreppssätt är bra i de fallen med låg problemkomplexitet men med hög osäkerhet (Dahlbom & Mathiassen, 1993). Ägare av problemen och framtida användare är väldigt viktiga för evolutionära systemutvecklare. Detta då användare har en aktiv roll i utvecklingen genom att utvärdera designförslag och prototyper och ägarna av problemen är de som är med och tar fram problemformuleringen och fattar beslut rörande denna men de bestämmer även vilken kvalitet det färdiga systemet ska ha. Systemutvecklarna tar fram och föreslår olika lösningar och kommunicerar med ägarna om problemet och användarna genom hela denna process. Detta innebär att systemutvecklarna måste kunna kommunicera på ett sätt som problemägarna och användarna förstår (Dahlbom & Mathiassen, 1993).

Förutom de två sätt som beskrivits ovan introducerar Dahlbom och Mathiassen (1993) även ett tredje. Detta sätt att arbeta och förhålla sig till systemutveckling heter intervention. Konstruktion och evolution är utveckling för användare, intervention är utveckling tillsammans med och med hjälp av användarna (Dahlbom & Mathiassen, 1993). Intervention fokuserar på systemets kvalitet vid användning, hur det implementeras och anpassas i den faktiskt arbetsmiljön. Detta innebär, för att förbättra kvalitet så kan både datorsystemet och organisationen behöva ändras (Dahlbom & Mathiassen, 1993).

Interventionsutveckling fokuserar på de motsägelser som förekommer inom en organisation (Dahlbom & Mathiassen, 1993). Arbetarna vill en sak och cheferna en annan. Det kan även finnas motsägelser inom projektutvecklingsgruppen där en person vill göra på ett sätt och en annan dess motsats. De som arbetar enligt intervention betraktar dessa konflikter som möjligheter att ingripa och förändra och inte som problem. Detta sätt att arbeta är dialektiskt och väldigt romantiskt då synen på organisationen är att denna är full av konflikter, motsatser och personer med olika målsättningar (Dahlbom & Mathiassen, 1993).

För att kunna skapa en gemensam förståelse för organisationen och dess problem så tar interventionsutvecklaren hjälp av metaforer. Genom att använda sig av metaforer så kan användarna och andra i en organisation betrakta sitt arbete och sin roll på nya sätt vilket kan leda fram till ökad förståelse och bättre lösningar på olika problem inom organisationen och systemet som används.



Figur 2:2 Relationen mellan konstruktion, evolution och intervention. (Ortiz-Arroyo, 2006)

## 2.4 Sammanbindning av teori

Under rubriken Systemutvecklingsmetodologi 2.2 presenterades tre olika grupper/typer av metodologier, vattenfall, snabb och evolutionär utveckling samt modellering.

Vattenfallsmodellen, är av den systemteoretiska utvecklingstypen konstruktion. Den baserar sig på ett mekanistiskt perspektiv och systemet konstrueras utifrån ett hårt systemtänkande.

De snabba och evolutionära metodologierna kan betraktas systemteoretiskt som evolutionära. Evolutionär utveckling kännetecknas av ett romantiskt perspektiv på världen. Detta eftersom agil utveckling strävar efter att kunna anpassa sig snabbt om villkoren för utvecklingen ändras. Evolutionär utveckling har ett mjukt systemtänkande då dessa metoder förespråkar involvering av alla intressenter, tydlig koppling till kund och användare genom hela utvecklingsarbetet. Agil utveckling kan också kategoriseras som intervention. Särskilt om vi väljer att betona relationen mellan utvecklarna och intressenter. Det problematiska med denna tolkning är det dialektiska systemtänkandet samt interventions idé om ingripande i organisationen. Låter vi det dialektiska systemtänkandet stå tillbaka för ett mer mjukt systemtänkande så passar beskrivningen av intervention bättre in på agil utveckling än evolutionär.

Agil utveckling är ett gränsfall mellan evolutionär och intervention. Det ter sig som att det behövs en utvecklingstyp som är ett mellanting av evolutionär och intervention. På grund av svårigheterna att kategorisera agil utveckling så är det möjligt att personer klassificerar agil utveckling olika. Vi har dock valt att definiera agil utveckling som evolution med romantiskt perspektiv och utgår från detta i vår analys.

Den tredje gruppen av metodologier var modellering. De försöker i viss mån kompensera nackdelarna med vattenfallsmodellen med att införa iterativ och inkrementell utveckling. Modellering är svårare att kategorisera än de andra två grupperna då modellering kan ha både ett romantiskt och mekanistiskt perspektiv. Eftersom modellering betonar nedbrytning av världen med mera så har den ett hårt systemtänkande. Modellering och då främst objekt modellering med exempelvis RUP är att betrakta som en metod med både ett mekanistiskt och romantiskt perspektiv på världen då den antar att krav kan komma att ändras. Skillnaderna mot konstruktion är för stora för att kategorisera modellering som endast konstruktion. Vi har därför valt att kategorisera modellering som både evolution och konstruktion och då evolution med ett mekanistiskt perspektiv på världen.

*Tabell 2:5 Systemteoretisk kategorisering av utvecklingstyper och metoder.*

<b>Konstruktion</b>	<b>Evolution/Konstruktion</b>	<b>Evolution (romantisk)</b>
Vattenfallsmodellen	Modellering Objekt modellering m.fl. <ul style="list-style-type: none"> <li>• Objekt-orienterad analys</li> <li>• Rational Unified Process</li> </ul>	Evolutionär utveckling Agil utveckling m.fl. <ul style="list-style-type: none"> <li>• SCRUM</li> <li>• eXtreme Programming</li> </ul>

Under avsnittet Kategorisering 2.1.1 presenterades tre olika typer av analogier; Konstruktion, Interaktion samt Fokus. Dessa kan vi klassificera utifrån systemtänkandet på samma sätt som metoderna. Den enklaste analogitypen att definiera utifrån systemtänkande är konstruktionsanalogier. Dessa är precis som namnet säger en beskrivning utifrån utvecklingstypen konstruktion. Detta innebär att de har ett mekanistiskt perspektiv på världen samt ger uttryck för ett hårt systemtänkande. Interaktionsanalogier är lite svårare att kategorisera. Tydligt är dock att dessa inte är att betrakta som konstruktion. Deras fokus ligger på dynamik och en föränderlig omvärld och individer vilket gör att de har ett mer romantiskt perspektiv. Systemen som skapas utifrån dessa analogier är ofta beroende av flera olika interagerande delar allt från personer till verktyg. Av den anledningen är de därför mer att betrakta som uttryck för ett mjukt systemtänkande än ett hårt. Detta då ett av tecknen för hårt systemtänkande är reducerbarhet vilket saknas hos interaktionsanalogiernas slutsystem. Analogier ger även uttryck för en ständig strävan till förbättring vilket leder fram till att dessa typer av analogier är att betrakta som evolutionära. Ingen av de framförda analogierna ger uttryck för de ingripande tankar som dominerar intervention. Inte heller ger de uttryck för det dialektiska systemtänkandet. Varför detta verkar frånvarande från ”analogidebatten” ligger utanför ramen för denna uppsats.

## Kapitel 3: Tillvägagångssätt

Valet av metod är viktigt anser Holme och Solvang (1997) och diskuterar kring både de *kvantitativa* och *kvalitativa* undersökningsmetoderna. De skriver, att båda tillvägagångssätten har sina för- och nackdelar och valet av metod därför bör bestämmas efter studiens syfte, det vill säga den metod som bäst hanterar problemområdet. Den kvantitativa metoden kännetecknas av beskrivning och förklaring, medan den kvalitativa metoden karakteriseras av beskrivning och förståelse.

Vi valde att använda den kvalitativa undersökningsmetoden, då vi eftersträvade att undersöka föreställningar om analogi hos våra utvalda informanter. Vi ville också komma fram till vad dessa föreställningar grundade sig i. I likhet med vad Holme och Solvang (1997) skriver om den kvalitativa undersökningsmetoden, ville vi öka förståelsen kring vårt valda ämne.

### 3.1 Teoretiskt ramverk och informationssökning

Eftersom syftet med vår uppsats är att undersöka analogier och metaforers roll för gestaltning av systemutveckling valde vi att dela upp vårt teoretiska ramverk i tre delar *analogier*, *systemtänkande* och *utvecklingsmetodologier*. Mot dessa element har sedan det empiriska materialet utvärderats.

Den största del av litteratur som ligger till grund för vår uppsats kommer från ett stort antal artiklar som mestadels behandlar påstådda analogiers tillämpbarhet eller inte. Vi menar, att urvalet artiklar var relevant för vår studie, då författarna till dessa har goda kunskaper om ämnet. Annan litteratur i form av böcker har också använts, men detta har varit i de flesta fall för att bredda vår egen förståelse av ämnet som valts att behandla, samt för att verifiera information som framkommit i artiklar, Internet och från våra informanter inkomna enkäter.

Empiriskt material hämtat från Internet har bland annat bestått av besök på *bloggar* och *fora* där det för vårt valda ämne pågår intressanta diskussioner. En blogg kan liknas vid en virtuell dagbok, personen som skriver vill delge sina anteckningar till andra som är intresserade medan ett forum kan liknas vid en virtuell träffpunkt, där flera personer interagerar med varandra genom att själv skriva eller svara på varandras inlägg. Vanligtvis upprättas forum efter olika intressekategorier. Informationen vi tagit del av via bloggar och forum har varit av mycket skiftande kvalitet, då kunskap och erfarenhet varierat bland personers uttalanden.

Vi har genom enkäterna försökt återge en bild av informanternas uppfattning om hur ett system konstrueras med hjälp av en liknelse och sedan se hur pass väl denna analogi stämmer överrens

med de metoder man tillämpar. Vid enkätsvar, som vi uppfattat som ofullständigt eller på annat sätt känts oklart, har kontakt tagits där vi bett om förtydligande.

### 3.2 Utformning av frågeformulär

Den kvalitativa intervjuens styrka ligger i att intervjuformen sägs vara den som minst har påverkan både vad det gäller den som utför eller den som deltar i intervjun (Holme & Solvang, 1997). Utgångspunkten för en kvalitativ intervju är att de som utför studien vill veta hur en viss företeelse tar form, faran av att då använda sig av en intervjuform där det finns en överhängande risk att styra deltagarna i utfrågningsprocessen blir i sådana fall mycket olycklig (Holme & Solvang, 1997). Eftersom det ansågs nödvändigt att av våra intervjupersoner få så opåverkade svar som möjligt föll sig valet naturligt att använda oss av den kvalitativa metoden.

Frågeundersökningar i form av enkäter eller intervjuer hör till de vanligaste teknikerna enligt Holme och Solvang (1997). Skillnaden mellan dessa tekniker ligger i hur insamlingen av informanternas svar förts. Vid en utskickad enkät fyller den tillfrågade själv i svaren, vid en intervju är det personen som intervjuar som fyller i informantens svar. Den personliga intervjun det vill säga direktkontakt vid ett förutbestämt möte ger den fördelen att möta den tillfrågade ansikte mot ansikte (Holme & Solvang, 1997). Vi håller med om deras resonemang till viss del, men syftet med våra frågor var att till största möjliga mån undvika att påverka de tillfrågade genom ”tysta svar”, nickningar och instämmande ljud, och ledande frågor på grund av vår kunskap om både analogier och metoder. Vi anser även att en personlig intervju inte var nödvändig för att samla in nödvändig data eftersom avsikten inte var att på djupet undersöka hur systemutvecklare betraktar sitt arbete utan vi ville endast översiktligt undersöka förhållandet mellan analogi och praktik.

Holme och Solvang (1997) hävdar, att standardiserade frågeformulär bör undvikas i den kvalitativa intervjun då detta kan vara ledande, det vill säga styra den person som deltar i intervjun. Vi menar dock att våra, förvisso standardiserade frågor, (se Bilaga 1) inte styrt informanternas svar då frågorna besvarades via e-post med uppmaning om att återge utförliga svar. Vi menar därför, som Holme och Solvang skriver, att vi lyckats få fram svar från våra utvalda informanter som är att se som ett resultat av deras egna uppfattningar.

Sammantaget gäller det, skriver Holme och Solvang (1997) att motivera informanten men också försäkra sig om att denne är medveten om att hans eller hennes medverkan är avgörande. Vid första kontakt per telefon med den tillänkta informanten ställde vi därför frågan om han/hon ville delta genom att besvara vår enkät. Vi underströk särskilt att det var viktigt att noga överväga om man ansåg sig ha tid då det var avgörande för oss att samtliga förväntade informaters svar var oss till handa inom utsatt tidsram.

Vi valde med stor omsorg de frågor som skulle ingå i enkäten. Enligt Holme och Solvang (1997) är det viktigt att hålla intresset uppe för den svarande genom hela utfrågningen, detta var särskilt vanskligt vid enkäter där den tillfrågade sitter ensam med frågorna. Om formuläret är för omfattande, strukturen oklar och enkäten som helheten är slarvigt utförd är chansen större, enligt



Holme och Solvang (1997), att svarsresponsen bli låg. Av den anledningen börjar vår frågeguide med frågor som kräver lätta och korta svar, för att i mitten bli något mer komplicerade och mer omfattande och sist avslutades det hela med något mer lättsamma frågor. Enkätens frågor delades även in i *praktik* och *analogi* för att underlätta och öka klarheten i frågorna inom vilket område den tillfrågade rörde sig. Vi strävade efter att hålla nere antalet frågor för att underlätta för de svarande då för många frågor kan ge upphov till förvirring och oklarheter. Enkäten börjar med frågor rörande informantens bakgrund. Detta för att personens bakgrund, eventuellt, skulle kunna ligga till grund för de samband, eller brist på samband, mellan praktik och analogi som framkom i analys av enkätsvaren. Praktikdelen av enkäten bestod av två delar, metodanvändning samt syn på system och utveckling. Detta eftersom frågorna rörande dessa delar var nödvändiga för att uppfylla syftet med arbetet. Vi hade även med frågor gällande informantens inställning till valet av metod idag då det kunde vara så, att om det fanns ett missnöje med metoden skulle detta kunna ge upphov till en analogi som var väldigt olik den utvecklingsmetod som informanter sade sig använda. I analogidelen av enkäten har vi med frågor rörande informantens kunskap om analogier samt deras inställning till dessa. Eftersom det är något som kunde ligga till grund för den liknelse informanten använde sig av. En negativ inställning till analogier skulle kunna ge upphov till analogier av en mer färggrann och tillgjord analogi för att informanten ville ge uttryck för sitt ogillande av analogier. Så av den anledningen ville vi veta informantens inställning till analogier.

### 3.3 Val av informanter

Informant är den person som intervjuas och som är delaktig i det sammanhang som studeras skriver Holme och Solvang (1997). Urvalet av intervjupersoner är i högsta grad avgörande för undersökningsprocessen och bör därför genomföras med största omsorg. Eftersom avsikten med våra frågor var att skapa en djupare förståelse valdes sju personer som vi varit i kontakt med tidigare som tillsammans kunde utgöra en representativ grupp av denna yrkeskategori. De tillfrågade har skilda kunskaper och utbildning, olika lång erfarenhet inom systemutvecklingsbranschen och verkar inom olika branschområden som sträcker sig från små till medelstora organisationer vilket påverkar typen av system som utvecklas. Vi hävdar att denna heterogena grupp kan spegla den variation av olika sätt att utveckla system på som personer använder sig av. Detta då vi är intresserade av individer och deras sätt att betrakta systemutveckling. Vi är medvetna om att informanternas till största del tekniska inriktning på utbildning kan ha kommit att påverka resultatet åt det mer ingenjörsmässiga hållet.

Från vissa håll kom det förfrågningar, men också önskemål, om att få hålla företagsnamnet anonymt vilket vi respekterade.

### 3.4 Enkätens genomförande och svarandemiljö

Frågandesituationen i en kvalitativ metod är krävande för den som utfrågas, då det förväntas både beskrivande och resonerande åsikter (Holme & Solvang, 1997). Kvale (1997) skriver, att

det är viktigt att det tydligt framgår för den tillfrågade i vilket syfte denne medverkar i utfrågningen, eftersom det är frivilligt att delta. Vi bestämde oss därför att tidigt ta kontakt med våra tilltänkta informanter så snart utformningen av enkäten var klar. Kontakt togs via telefon där vi kort presenterade oss själva och anledningen till varför vi önskade personens medverkan. Den tillfrågade kunde därför efter vår kontakt i lugn och ro ta ställning till om han eller hon ville delta. Intervjuformuläret skickades omgående via e-post till respektive informant. Detta gjorde det möjligt att i god tid förbereda sig, något som vi ansåg vara en absolut förutsättning för att studien som helhet skulle bli meningsfull. Eftersom frågorna skickades omgående och vi hade ett väl tilltaget sista datum, kunde den som deltog hinna förbereda sig, det vill säga fundera på vad denna skulle svara på enkätfrågorna. Då den kvalitativa intervjun ”kräver” beskrivande och resonerande åsikter gör detta att de flesta, anser vi, vill fundera över *vad* man ska svara och *hur*.

Miljön i vilken utfrågningen sker bör beaktas. Förhållanden som tid och plats spelar en avgörande roll för informanternas svar på frågorna menar Holme och Solvang (1997). Vårt beslut att genomföra utfrågningen av våra informanter via utskickat e-postmeddelande innehållande bifogat frågeformulär grundade sig på att det för många av våra informanter var svårt att avsätta tid för ett möte där intervjun skulle kunna genomföras.

Vi valde att inte ha ett personligt möte med informanten då detta inte var nödvändigt för att få svar på vår typ av frågor. Att de som deltog i undersökningen har haft möjlighet att besvara våra frågor när de själva ansett sig ha tid har förhoppningsvis lett till att våra informanter känt sig mindre stressade och därför ställt sig mer positiva till att besvara våra frågor. Dessutom undviker vi, som nämndes tidigare, att från vår sida ge tysta svar.

Eftersom frågorna besvarades via e-post var det viktigt att sätta ett väl tilltaget sista datum för inlämning. Därigenom minskade risken för sent inkomna svar som av den anledningen inte skulle hinna ingå som underlag i vår analys.

### 3.5 Analys av empirin

Forskare som väljer att använda sig av den kvalitativa metoden utsätts ofta för kritik (Oates, 2006). Kritiken grundar sig i att den kvalitativa dataanalysen inte är ett förlopp som kan sägas ta sig från en punkt till en annan utan avbrott eller sidospår. Information saknas om hur arbetsgången av det analyserade materialet skett, detta eftersom det inte finns klara och fasta regler om hur analysen ska utföras. Detta medför att forskarens slutsatser kan komma att ifrågasättas, menar Oates. För att undvika denna kritik har vi strukturerat vårt analytiska tillvägagångssätt genom att använda systemtänkandet som en form av ”koppling” mellan analogi och metodologi för vilket vi utvärderat det empiriska materialet mot. Det sätt vi valt att behandla empirin åskådliggör också flödet i arbetet och gör det möjligt att återge en ökad förståelse för våra slutsatser.

Den kvalitativa metoden erbjuder färre tekniker jämfört mot den kvantitativa metoden då det gäller att analysera insamlad information (Oates, 2006). Medan den kvantitativa analysen baseras till stor del av mätbara resultat i form av siffror grundar sig den kvalitativa analysen i större grad

på forskarens egen förmåga att se mönster och teman i det empiriska materialet. För att underlätta identifiering av mönster har vi analyserat datan utifrån de olika perspektiv och systemtänkande som presenterade i kapitlet Systemtänkande. Ord har olika betydelse beroende på vem som uttrycker sig och måste sättas i ett sammanhang, skriver Oates. Därför har vi försökt, i de fall det varit möjligt, att argumentera för varför vi valt att kategorisera informanternas svar på ett visst sätt. Kategorierna består av de olika former av systemtänkande, hårt och mjukt, samt typer av utveckling, evolution och konstruktion, som presenterades i kapitlet 2.3 Systemtänkande. Både analogier och metoder kategorieras enligt systemtänkandet för att se ifall det fanns ett tydligt samband mellan de olika delarna. I de fall informationen gett upphov till alltför vittgående tolkningar har vi valt att inte ta ställning.

För att göra en bra analys av det empiriska materialet krävs det enligt Oates (2006) att datan förberetts. Det innebär, att så långt det är möjligt, få materialet i ett så lika format som möjligt. Vi har, som Oates föreslår, börjat vår analys med att gå igenom all insamlad data (se Bilaga 3), för att skaffa oss en generell bild och uppfattning. Genomgången som görs resulterar i en uppdelning av informationen som på detta vis kategoriseras efter hur relevant materialet är för det fortsatta arbetet. De frågor som berörde snarlika saker har så långt det varit möjligt presenterats tillsammans. Informanternas svar på enkätfrågorna valdes att presenteras i form av tabeller där de mest signifikanta delarna av svaren lyfts fram (se Bilaga 2). Detta gjordes för att se om det fanns tydliga mönster i den insamlade data innan data analyserats, Oates kallar denna indelning att *identifiera nyckelteman i datan*.

### 3.6 Metodreflektion

Holme och Solvang (1997) menar att användning av den kvalitativa undersökningsmetoden kan innebära risk avseende validiteten, det vill säga hur giltig den framtagna informationen är. Kvalitativa undersökningar ger förutom flexibilitet också en större närhet till det område som ska studeras, vilket kan ses som något positivt men också som ett problem. Det är omöjligt att helt göra sig fri från egna subjektiva tolkningar, eftersom dessa grundar sig i vår egen förståelse om ämnet vi valt att studera. Därför kan det omedvetet ha kommit att påverkat informanter genom våra enkätfrågor. Antalet tillfrågade och urvalet har varit svårt att avgöra för att öka studiens reliabilitet det vill säga tillförlitligheten av det analyserade materialet. Vår avsikt med undersökning är dock, att väcka intresse och bidra till en meningsfull ämnesdiskussion.

Intervjusituationen i en kvalitativ metod är krävande för den som utfrågas, eftersom det förväntas både beskrivande och resonerande åsikter (Holme & Solvang, 1997). Av den anledningen har vi tagit hänsyn till detta vid konstruktionen av intervjumallen och anpassat antalet frågor efter detta. Dock är den kvalitativa styrkan den att det är möjligt att hantera ett mindre antal frågor och ändå få ut maximalt av intervjufrågorna eftersom frågorna besvaras både detaljerat och beskrivande vis.

Enkäten som i god tid skickades till de tillfrågade kan ha påverkat de lämnade svaren. Dels har det gett dem tid att fundera, frågan som vi ställt kanske aldrig har funnits i personens tankar tidigare men nu har den väckts och det finns tid att fundera. Dels kan de tillfrågade ha frågat

andra personer i dennes omgivning om hjälp. Material som inhämtas från Internet är inte, understryker Svenning (2003), av sådan art att det i sig kan ses som en källa. Informationen har dock främst verkat som inspiration men, den har också frambringat ett stort värde då författarna av det publicerade materialet, anser vi, visat på stor kunskap. Vi har främst använt internetkällor för att hitta och granska populära systemutvecklingsanalogier.

## Kapitel 4: Presentation och analys av empirin

Nedan kommer en kort presentation av våra informanter. Dessa verkar alla inom systemutvecklingsbranschen. Spektrat av tillfrågade rör sig inom ett verksamhetsområde av företag från små till medelstora företag. Avslutningsvis ges en kort genomgång av vår empiri. I nästa delkapitel analyserar vi det insamlade materialet.

Tobias, Civilingenjör är med sina tio års erfarenhet inom mjukvarubranschen VD på börsnoterat bolag. Företaget som funnits sedan 1987 kan, menar de själva, räknas till föregångarna att använda PC-baserade lösningar inom produktion. Tillsammans med företagets partners levereras industriella IT- och automationslösningar baserade på produkter och komponenter av standardtyp. Företaget är verksamt inom områden som sträcker sig från tillverkande industri med bland annat bilindustri, allmännyttiga produktionsanläggningar som Vatten och Avlopp (VA) och energi till avancerad petrokemisk och tung processindustri.

Anellas utbildning består av en IT-examen och hon har fem års erfarenhet inom mjukvarubranschen. Hon arbetar som projektledare på ett av Europas ledande företag vid utveckling av informationssystem för järnvägsbunden transport. Företagets affärsområden täcks av tre centrala områden 1) Kostnadseffektiv planering och stimulerande system för transportföretag, 2) Effektivt kontrollarbetsystem för spår och 3) Stadsmässig offentlig transport och eftersatt signaleringsteknologi för sammankoppling av elektroniska små och medelstora stationer. Med mer än 20 års erfarenhet och med mycket specialiserad och djupgående kunskap inom järnvägsprocesser gör detta det möjligt, menar de själva, att vilja agera som pionjärer inom detta branschområde.

Erik har en civilingenjörsexamen och har arbetat med systemutveckling i nio månader. Företaget han arbetar på är ett litet bolag som tillverkar pins. Erik är anställd som programmerare/systemutvecklare och är med och utvecklar företagets design och beställningssystem. De som beställer pins är privatpersoner och alla pins är unika, så stor vikt har lagts vid att utveckla företagets webbaserade designverktyg som de numera även säljer konsulttjänster för.

Mattias är den som startat pin bolaget och han är också med och programmerar och utvecklar företagets programvara. Han är den som har minst utbildning av våra informanter med en Java kurs på komvux. Han uppger att han arbetat med systemutveckling i fyra år.

Linus och Mikael är båda två anställda som programmerare på ett mindre bolag. Bolaget utvecklar programvara för överföring av data mellan diverse datorbaserade produkter och mobiltelefoner. Företaget grundades 2004 och blev ett aktiebolag 2006. Båda två har civilingenjörsbakgrund och har arbetat lika lång tid på bolaget, sex månader.

Johan är för närvarande anställd på ett amerikanskt bolag som programmerare. Bolaget har som huvudsyssla att ta fram webbtjänster och den avdelning som Johan arbetar på utvecklar en Flashapplikation som ska göra det lättare att tillgodogöra sig statistik. ”Avdelningen” var förut ett svenskt bolag men blev uppköpt och inkorporerat i det amerikanska företaget och därmed flyttat till USA. Johans utbildningsnivå består av att nästan ha en magister i datalogi och en kandidat i informatik.

Våra informanter använde i stor utsträckning metoder som kategoriseras som agil utveckling. Det var endast en informant som inte använde sig av en agil utvecklingsmetod utan i hennes fall arbetade de efter vattenfallsmodellen. Synen på systemutveckling skiljde sig åt mellan informanterna och skillnaden kunde kopplas till hur stor erfarenhet av systemutveckling personen hade. Definition av system var överlag likvärdig vilket det står mer om i nästa kapitel där analysen av vår empiri presenteras. Informanterna var överlag positiva till användning av analogier och att tänka kring systemutveckling. Vårt underlag var inte tillräckligt stort för att se ifall det fanns något mönster mellan att vara positiv till analogier och analogiernas användbarhet.

## 4.1 Analys av empirin

Vi kommer här att gå igenom och kategorisera de svar vi fått på våra frågor. Först går vi igenom de mer praktikorienterade frågorna och kopplar svaren till systemtänkandet vi tidigare presenterat. En liknande genomgång görs av de svar som behandlar delen analogi. Systemtänkande används för att överbrygga de begrepp och beskrivningar som skiljer mellan svaren på frågorna rörande praktik och analogi. Detta så att det blir möjligt att använda en gemensam vokabulär vilket tydliggör skillnaderna och likheter i svaren.

### 4.1.1 Metodanvändning

En tydlig majoritet av de utvecklingsmetoder som de tillfrågade använder sig av är att kategorisera som typen evolution (Tabell 4:1). XP och Scrum är utifrån vår tidigare beskrivning av agil utveckling att betrakta som evolutionära med ett romantiskt perspektiv på världen. RUP är en svårare metod att klassificera eftersom hur den ska klassificeras beror på hur ett företag valt att implementera den i utvecklingen. Utifrån en helhetsbedömning av de svar vi fått från Tobias rörande metod, verktyg och faser så har vi valt att klassificera denna RUP implementation som både evolution och konstruktion i likhet med vad vi skrev i 2.4. Ett likartat problem får vi med den interna metod som Mikael använder. Han skriver att den är en odefinierad agil metod men då han förklarar agil med att den är iterativ så väljer vi att kategorisera denna interna metod som evolution med mekanistiskt perspektiv.

Endast ett företag använde sig av en metod som motsvarar utveckling enligt vattenfallsmodellen (Tabell 4:1 Anella). Eftersom utvecklingsprocessen sker enligt vattenfallsmodellen är metoden som företaget tillämpar ett typfall, anser vi, av konstruktionsutveckling. Konstruktion har, som vi skrivit i Systemtänkande 2.3, ett mekanistiskt perspektiv på världen och ett hårt systemtänkande.

Vi har hos våra informanter identifierat två typer av utvecklingsmetoder som kan klassificeras som antingen evolutionära eller konstruktion. Ingen av de svarande använder sig av metoder som vi kan klassificera som intervention enligt den definition som görs i kapitlet Systemtänkande 2.3.

Tabell 4:1 Svar på fråga 5 och 9.

Fråga 5: Vilken/vilka utvecklingsmetoder använder du dig av i arbetet?

Fråga 9: Vilka tekniker/verktyg använder ni i ert arbete såsom användarfall, klassdiagram, kravspecifikation?

	Metod	Tekniker
<b>Mattias</b>	XP	Parprogrammering, Planning game, User stories,
<b>Erik</b>	XP	Iterativt, TDD, User Stories
<b>Linus</b>	Ej definierad/ intern	Kravspec, Diagram, Användarfall
<b>Mikael</b>	Intern	Inga
<b>Johan</b>	SCRUM, XP	Produkt och iterationsbacklogsverktyg
<b>Tobias</b>	RUP, PPS	UML, Användarfall, Kravspec, CM, Testverktyg
<b>Anella</b>	CENELEC	Kravspec, användarfall

Utifrån de tekniker som informanterna säger sig använda kan vi inte göra någon tydlig klassificering utifrån vare sig perspektiv på världen eller systemtänkande. En del av de tillfrågade använder sig av kravspecifikationer, vilket visar på att det hos dessa också finns drag av ett mekanistiskt perspektiv i deras arbetsprocess (Tabell 4:2). Listningen av verktyg och tekniker underlättar dock klassificeringen av metoderna i oklara fall.

Tabell 4:2 Svar på fråga 7 och 8.

Fråga 7: Finns det en livscykel i ditt systemutvecklingsarbete såsom att ett projekt genomgår olika faser och i så fall vilka anser du finns?

Fråga 8: Om du skulle prioritera faserna i vilken ordning kommer dessa? Vilken fas är viktigast?

	Faser	Prioritering av faser
<b>Mattias</b>	Design, implementering, testning, release	Alla
<b>Erik</b>	Design, implementering, testning	Alla
<b>Linus</b>	Implementering, refaktorisering, testning, release	Beror på situation
<b>Mikael</b>	Utveckling, release	Båda
<b>Johan</b>	Brainstorming, prototyping, implementering, release, underhåll	Designbeslut, processen bör vara flexibel
<b>Tobias</b>	Analys, design, implementering, testning, release	Analys, testing, release, design, implementering
<b>Anella</b>	CENELEC*	Analys

\*Koncept, systemdefinition, riskanalys, kravspecificering, design och implementering, med flera 14 st totalt.

De faser som informanterna menar att deras projekt följer är ungefär samma hos alla tillfrågade utom två (Tabell 4:2 Tobias, Anella). Faser i sig säger inget om perspektiv på världen eller typ av utvecklig, de hjälper dock till att komma till rätta med oklarheter i klassificering av metod. När det gäller prioritering av faser är detta något som endast de äldre och mer erfarna gör då de väljer att lyfta fram analysen som den viktigaste fasen (Tabell 4:2). Om analysen ska kopplas till mekanistiskt eller romantiskt perspektiv beror inställningen till analys och hur länge denna analys är giltig. Det vill säga om analys är något som sker kontinuerligt eller endast förekommer i början av ett projekt. Detta framkommer inte i frågorna och svaren.

#### 4.1.2 Syn på systemutveckling

Det finns stora skillnader i hur informanterna väljer att kategorisera sin systemutveckling. De äldre och mer erfarna systemutvecklarna valde att betrakta systemutveckling som en kombination av konstruktion och evolution (Tabell 4:3). De som valde att utveckling är en evolutionär process var yngre och mindre erfarna. Erik framhöll dock att typen av projekt kan påverka vilken typ av utvecklingstyp, konstruktion eller evolution, som bedrivs.

Tabell 4:3 Svar på fråga 11.

Fråga 11: Vilket/vilka av följande alternativ tycker du speglar din syn på systemutveckling? Systemutveckling är

- konstruktionsprocess där systemet utvecklas stegvis utifrån en abstrakt helhetsbild av systemet, kravspecifikation, som delas upp i mindre delar och förfinas
- evolutionärprocess där partiella problem identifieras och lösningar utvärderas allteftersom med hjälp av användarna vilket leder till en helhetsbild av systemet?

	Passande beskrivning av utveckling	Motivering
<b>Mattias</b>	Evolution	-
<b>Erik</b>	Evolution	Nuvarande utveckling. Beror på projekt.
<b>Linus</b>	Konstruktion	Tydliga krav från början
<b>Mikael</b>	Evolution	-
<b>Johan</b>	Evolution & Konstruktion	Utveckling är en kombination av båda och kan bero på situation.
<b>Tobias</b>	Evolution & Konstruktion	Utveckling är en kombination av båda.
<b>Anella</b>	Konstruktion	-

Linus hänvisar till tydliga krav från början vilket är något som lyfts fram i konstruktionsmetoder. Svaren på fråga 12, hur de definierar ett system, är svåra att klassificera utifrån de världsperspektiv och systemtänkande vi använt som mall. Mikaelns definition kan tolkas som ett mjuk systemtänkande då det ger uttryck för många olika tolkningar och involvering av en massa olika personer (Tabell 4:4). Mikaelns systemdefinition är dock något som är resultatet av någon form av analysprocess därför är det svårt att kategorisera denne som antingen hårt eller mjukt. Tobias, Anella och Johan ger uttryck för ett hårt systemtänkande (Tabell 4:4). Anledningen till att vi väljer att definiera detta som tillhörande av det hårda systemtänkande är deras tankesätt kring komponenter, det vill säga att ett system är uppbyggt av flera beståndsdelar som tillsammans svarar för en helhet.



I svaren förekom också två helt motstående systemdefinitioner men som vi valt att klassificera som hårt systemtänkande. Erik, som anser att endast fysiska komponenter kan utgöra system och Mattias, som anser att de digitala komponenterna utgör systemet. Ingen av definitionerna involverar människor och båda kan brytas ner i mindre bitar vilket är anledningen till att vi klassificerar dessa som hårt systemtänkande.

Tabell 4:4 Svar på fråga 12.

Fråga 12: Hur definierar du ett system i systemutveckling?

	Definition av system
<b>Mattias</b>	Det man kodar.
<b>Erik</b>	Summan av projektets fysiska komponenter i den miljö som den faktiskt används.
<b>Linus</b>	-
<b>Mikael</b>	Allting som på något sett rör det man arbetar med.
<b>Johan</b>	En mängd komponenter som samverkar för att lösa ett informationshanteringsproblem.
<b>Tobias</b>	Ett system för mig är ett urval av funktioner som hänger ihop och utför en uppsättning av funktioner för att uppnå ett definierat resultat. Ett system har alltid en motor eller hjärta som gör att det hela tiden tickar, som jag ser det skiljer just detta det från en komponent som inte själv lever utan får impulser och sedan lämnar ett svar.
<b>Anella</b>	Ett system är uppbyggt av moduler som i sin tur är uppbyggt av block.

### 4.1.3 Erfarenhet av och inställning till analogier

Det var endast fyra av de tillfrågade som hade hört talas om någon liknelse (Se Tabell 4:5). Av dessa så var två av typen konstruktion. Bilkonstruktion är av samma typ som husbygge då stegen och faserna är likartade. Båda går att bryta ner i mindre delar och så vidare Linus hade hört talas om talasätt rörande Lego. Dessa avviker från de andra liknelserna då de inte rör utvecklingsprocessen och är då av en helt annan typ och inte särskilt intressant för oss. Kan ju

Tabell 4:5 Svar på fråga 13 och 14

Fråga 13: Vilken är den mest vanligt förekommande liknelsen du hört talas om vid utveckling av system?

Fråga 14: Varför tror du att denna liknelse görs?

	Liknelser	Motivering
<b>Mattias</b>	-	-
<b>Erik</b>	Husbygge	För att det behövs samarbete och att det är ett komplicerat projekt med kända problem och metoder.
<b>Linus</b>	Systemutveckling är som Lego, man får akta sig vart man sätter fötterna. Systemutveckling är som att bygga med Lego, man får inga flickor.	För att det är viktigt att vara systematisk och noggrann.
<b>Mikael</b>	-	-
<b>Johan</b>	Bilkonstruktion	Det är alltid svårt att beskriva abstrakta fenomen och genom att göra jämförelse med kända fenomen så hoppas man underlätta förståelse.
<b>Tobias</b>	Husbygge	För att det är en komplex process bestående av flera specialiteter och gränssnitt mellan olika subsystem.
<b>Anella</b>	-	-

påpekas att användningen av Lego som metafor kan vara ett uttryck för hårt systemtänkande.

På frågan om varför de trodde att dessa liknelser gjordes så hade informanterna olika motiveringar. De informanter som hade hört talas om liknelser hade olika motiveringar till varför sådana gjordes (Tabell 4:5). Alla motiveringar berörde dock på ett eller annat sätt komplexitet vid systemutveckling.

Generellt sett ställer sig de tillfrågade positiva till allmänt filosoferande kring själva utvecklingsprocessen (Tabell 4:6). Trots den positiva inställningen, menar några att det finns uppenbara risker med att använda sig av filosofiskt tänkande. Argument som läggs fram är till exempel faran av att låsa sig i tänkandet kring metaforer och att tillämpningen av liknelser och/eller metaforer i sig skulle medföra en informationsförlust på detaljnivå som är viktigt för hanteringen av det faktiska problemet. Ett annat exempel är risken att istället tillföra information som är mindre intressant eller av vilseledande karaktär och att detta i sig kan leda till en felaktig uppfattning om vilka åtgärder som behövs och om hur dessa bör utföras. Informanternas inställning till användning av liknelser och metaforer sträcker sig från bra till dålig och i vissa fall till rentav skadligt. De yngre är mer negativt inställda till allmänt filosoferande än de äldre systemutvecklarna. Flera skriver att det är viktigt att vara medveten om att det krävs både kunskap och förståelse av det man vill abstrahera för att kunna tala om det i bredare termer. Detta menar också Dahlbom och Mathiassen (1993) när de beskriver hur utvecklare bör använda och förhålla sig till analogier.

*Tabell 4:6 Svar på fråga 15.*

*Fråga 15: Tror du liknelser, metaforer och allmänt filosoferande kan vara ett bra sätt att öka förståelsen av sin egen systemutveckling och systemutveckling generellt? Är det något du själv använt och känt dig hjälpt av?*

<b>Förståelse kring den egna utvecklingen med hjälp av liknelser och metaforer.</b>	
<b>Mattias</b>	Det har pedagogiskt och mnemoniskt värde och framförallt, är väldigt kul. Men i övrigt tror jag mest att det är skadligt att utgå ifrån liknelser.
<b>Erik</b>	Nej, det tror jag faktiskt inte. Kan leda till en felaktig uppfattning.
<b>Linus</b>	Tror att det liknelser och metaforer kan vara bra för, är att lära sig och diskutera olika principer. Men det ska inte överdrivas.
<b>Mikael</b>	Tror man lätt kan måla in sig i metaforer, men det kan vara ett bra verktyg.
<b>Johan</b>	-
<b>Tobias</b>	Det är bra sätt att beskriva skeenden.
<b>Anella</b>	Det tror jag absolut.

#### **4.1.4 Analogier för systemutveckling**

På frågan hur de tillfrågade själva skulle beskriva utveckling av ett system med hjälp av en liknelse skiljer sig svaren (Tabell 4:7). En del använde sig av för oss kända liknelser men det framkom en del nya jämförelser som vi inte hade hört talas om tidigare, men vilka kunde härledas genom utmärkande drag till de mer kända. Vissa valde att framhålla interaktionen mellan människor medan andra helt saknar människor i sina liknelser. Majoriteten av de

tillfrågade ansåg att deras liknelse endast beskrev utvecklingsprocessen och inte det slutliga systemet. Mattias vill likna systemutveckling med julens förberedelser. På grund av hur analogin beskrivs går det inte att klassificera den. Den kan tillhöra alla perspektiv och systemtänkande sett utifrån tillfrågad persons inställning till julen, det vill säga, svaret är subjektivt. Vi väljer dock att tolka Mattias svar från ett grundläggande romantiskt perspektiv, det vill säga att julförberedelser verkar i en kaotisk miljö. Erik liknar utvecklingsprocessen med byggande av sandslott. Detta överensstämmer till viss del med McBride (2003) jazzanalogi och dess känsliga struktur vars framgång bygger på, i detta fall, projektgruppens förmåga till teamwork. Resultatet kan bli väldigt bra men det kan också bli väldigt dåligt som när slottet förvandlas till en hög med sand.

Mikael menar att systemutveckling kan liknas vid att bygga ett hus genom att först bygga en koja (Tabell 4:8). Vid en första anblick är det lätt att tro att denna liknelse skulle gå under byggnadsanalogin och därmed ett konstruktionstänkande, men då huset byggs till och om allteftersom ser vi tydliga paralleller till Barnes (2007) trädgårdsanalogi som kännetecknas av en ständigt pågående utveckling och underhåll. Den dynamiska och inte kravstyrda processen gör att vi väljer att beteckna den som en evolutionär process.

Johans jämförelse med att likna systemutvecklingen vid byggandet av en snögubbe i lag menar vi kan härledas till McBride (2003) filmanalogi som hanterar en mycket hög interaktion av och med människor. Särskilt tydligt framgår parallellen mellan projektledarrollen och regissören. Liknelsen betecknar vi som evolutionär då den saknar egentliga faser eller formalisering, utan fokus ligger istället på interaktionen mellan människorna genom hela processen fram till färdig produkt. Därigenom menar vi att den ger uttryck för ett mjukt systemtänkande. Att Johan anser att system också kan förändras ger stöd för att kategorisera analogin som evolutionär.

*Tabell 4:7 Svar på fråga 16 och 17.*

*Fråga 16: Beskriv med hjälp av liknelse (illustrativt) din systemutveckling.*

*Fråga 17: Är slutresultatet i din liknelse även hur du betraktar system, ex "grytan" i analogiexemplet, eller är din analogi endast en beskrivning av utvecklingsprocessen?*

	<b>Liknelser</b>	<b>Stämmer slutresultatet med den gjorda liknelsen.</b>
<b>Mattias</b>	Systemutveckling är som julen. Börjar ett par månader innan den ska vara klar, alla pratar och det och stressar upp sig över det hela.	Nej, grytan är för mig de fulla, förättna släktingarna (ryde strax efter deadline)*.
<b>Erik</b>	Mjukvaruprojekt är som sandslott, ju mer det växer desto större är risken att du en dag upptäcker att det du har inte längre är ett vackert slott, utan bara en stor hög med sand	Ja.
<b>Linus</b>	-	-
<b>Mikael</b>	Planlösningen utvecklas snarare med huset.	Det är en beskrivning av utvecklingsprocessen.
<b>Johan</b>	Bygga en jättestor snögubbe i ett lag.	Nog processen, men också att själva snögubben är något som kan formas kontinuerligt ...
<b>Tobias</b>	När man skall bygga ett hus har man ett bygglag bestående av flera olika typer av experter och utförare. ...	Yes!
<b>Anella</b>	Jag ser det som att vi bygger ett hus med moduler som vi bygger ihop för att slutligen få vårt färdigbyggda hus.	Analogin är en beskrivelse av processen och systemet.

\* Ryde är en arbetskamrat

Tobias och Anella väljer att använda sig av byggnadsanalogin, vilken de menar representerar och sammanfattar deras systemutveckling på ett bra sätt. Vi har tidigare kategoriserat byggnadsanalogin som tillhörande konstruktionsanalogier och som namnet antyder så tillhör denna Dahlbom och Mathiassen (1993) konstruktionsmetod. Detta innebär att analogin intar ett mekanistiskt perspektiv på världen och ger uttryck för ett hårt systemtänkande.

## 4.2 Sammanfattning av analysresultat

Ställer vi upp de olika kategoriseringar vi gjort i analysen så får vi fram det resultat som visas i Tabell 4:8. Det framgår klart och tydligt att metoder som kan kategoriseras som evolutionära dominerar hos de tillfrågade. Vid synen på systemutveckling ser vi en jämn fördelning av utvecklingstyp, från evolution till både evolution och konstruktion. Linus och Anella avviker något då de valde annorlunda i jämförelse med övriga i samma åldersgrupp. Erik var av åsikten att utvecklingstyp berodde på situation men han var ”evolutionär” just nu. Vi har därför valt att lyfta fram evolution hos honom.

I alla fall, där det var möjligt att jämföra, så överrensstämde analogin helt eller delvis med de tillfrågades val gällande syn på systemutveckling som konstruktion och/eller evolution. Frågar vi världsperspektivet hos de evolutionära metoderna, så fanns det stora likheter mellan metod och analogi i alla fall. Tobias utmärkte sig dock då han hade en metod som var både konstruktion och evolution och använde en konstruktionsanalogi. Han hade dock en metod med mekanistiskt perspektiv och detta mekanistiska perspektiv delas av hans konstruktionsanalogi. Utifrån det menar vi att det finns ett klart samband mellan metod och analogi.

Ser vi till hur informanterna definierat system så är det tydligt att det hårda systemtänkande är gällande oavsett använd metod, val av systemutvecklingsbeskrivning och analogi. Mikael var den ende med vad vi tolkat som ett mjukt systemtänkande. Denna tolkning är dock väldigt subjektiv och det är möjligt att en annan läsare kommit fram till en annan tolkning därför valde vi att inte göra en slutgiltig kategorisering.

Tabell 4:8 Sammanställning av enkätsvaren efter genomförd analys.

	Metod	Syn på systemutveckling	Systemdefinition	Analogi	Inställning till analogier
<b>Mattias</b>	Evolution (romantiskt)	Evolution	Hårt	-	Positiv
<b>Erik</b>	Evolution (romantiskt)	Evolution	Hårt	Evolution	Negativ
<b>Linus</b>	Ej klassificerbar	Konstruktion	-	-	Positiv
<b>Mikael</b>	Evolution (mekanistiskt)	Evolution	-	Evolution	Positiv
<b>Johan</b>	Evolution (romantiskt)	Evolution/Konstruktion	Hårt	Evolution	-
<b>Tobias</b>	Evolution/Konstruktion	Evolution/Konstruktion	Hårt	Konstruktion	Positiv
<b>Anella</b>	Konstruktion	Konstruktion	Hårt	Konstruktion	Positiv

## Kapitel 5: Diskussion

Nedan följer en diskussion kring det resultat som framkommit vid analysen av empirin samt intressanta svar och samband som visade sig i de svar informanterna lämnade. Vi behandlar inte bara resultatet av analysen då vi tycker att det framkommit saker under arbetet som är värt att belysas och reflekteras över.

### 5.1 Analysresultatet

Det som utmärker sig i analysresultatet är att alla informanter utom en gjort en systemdefinition utifrån det hårda systemtänkandet. Mikael's definition av system kunde vi inte kategorisera. Sett i relation till metod så avviker definitionerna från det systemtänkande som metoderna baseras på i alla fall utom en. Anella är den enda som ger uttryck för hårt systemtänkande och mekanistiskt perspektiv över hela spektrat. Anella är den ende av informanterna som har en tydlig röd tråd i sina svar. Det som utmärker Anellas situation är att hon arbetar med utveckling som har ett uttalat krav på en vattenfallsmodell. Detta kan ha kommit att påverka hennes syn på systemutveckling.

Mattias och Erik avviker då de har systemdefinition som inte stämmer överrens med vare sig metod, kategorisering av systemutveckling eller analogi. (Bör påpekas att Mattias inte skrivit en analogi som vi har kunnat kategorisera). Denna avvikelse kan bero på att de flesta oavsett bakgrund har ett hårt systemtänkande när de ska definiera ett system. Ett utökande av antal frågor samt mer definierade frågor, skulle kunna ge upphov till en bättre bild av en persons syn på systemutveckling. Det skulle då, gissningsvis, vara möjligt att ge ett svar på denna avvikelse. Detta är dock något som i så fall får göras i en annan undersökning.

Det var endast två informanter, Tobias och Johan, som ansåg att systemutveckling var en kombination av evolution och konstruktion. Deras metoder skiljde sig dock åt då Johan hade en evolutionär och Tobias en som kategoriserades som både konstruktion och evolution. De hade även en systemdefinition som visade på ett hårt systemtänkande. Det är intressant att båda valde element av både evolution och konstruktion i synen på systemutveckling men ändå valde att endast belysa "ena halvan", så att säga i analogin. Det mest beskrivande skulle ha varit att ha en analogi som visade på både evolution och konstruktion om synen på systemutveckling var det som bestämde analogin. En möjlig förklaring till skillnaden av analogier mellan Johan och Tobias kan vara sättet att se på faser. Tobias lyfte fram analys främst och hade en konkret prioritering av faser i jämförelse de övriga. Johan prioriterade designbeslut men ansåg att processen ska vara flexibel. Dock ser vi att den mest trovärdiga förklaringen sett utifrån det material som ligger till grund för den här undersökningen är att använd metod och dess perspektiv har betydelse för vad personen väljer att lyfta fram i analogin. En mer djupgående analys av använda metoder samt fler frågor rörande den praktiska användningen skulle kunna ge svar på skillnaderna mellan "praktiken" och analogin.

Om vi ser till helheten så tycks det som att både metod och syn på systemutveckling (konstruktion/evolution) har inverkan på hur en person väljer att åskådliggöra sin systemutveckling med hjälp av en analogi. En metod med mekanistiskt perspektiv i kombination med att betrakta systemutveckling som konstruktion eller både konstruktion och evolution ger upphov till att en person väljer att åskådliggöra systemutveckling med en konstruktionsanalogi.

Då flera av de tillfrågade har varit inblandade i valet av metod så kan det vara så att hur personen anser att systemutveckling ska bedrivas påverkat metodvalet (Bilaga 2 Tabell 4). Detta kan vara en anledning till att det inte finns tydligare skillnad mellan syn på systemutveckling och använd metod. Det är möjligt att ett mycket högre antal informanter med en högre andel som inte varit delaktiga i valet av metod skulle visa på större skillnader i förhållandet mellan metod och analogi. Detsamma gäller relationen syn på systemutveckling och analogi. Utöver det ovan nämnda skulle det behövas ytterligare frågor rörande metodandvändning och syn på systemutveckling. Fler frågor rörande den praktiska användningen av en metod skulle ge upphov till säkrare klassificering av både RUP och den interna metoden. Det vill säga högre upplösning på frågorna rörande praktik.

## 5.2 Iakttagelser

### *Analogiernas syfte*

Det är intressant att informanterna överlag var positiva till användandet av analogier om än med viss reservation. Det var endast en informant som klart uttala sig negativ till användning av analogier och metaforer. De som var positiva ansåg att analogier kan ha ett visst värde då de används för att underlätta förståelsen för något. Ingen av informanterna lyfte fram att liknelser kunde vara något att inspireras av som är syftet med byggnadsanalogin det vill säga rollerna och processen från en bransch flyttas till en annan. De som använder sig av konstruktionsanalogier lyfter ofta fram de olika rollerna som finns inom ett byggprojekt. Detta saknas i exempelvis interaktionsanalogin jazzband, där fokus är på dynamiken. Fokus på rollerna var högre i de som använde sig av konstruktionsanalogier. De övriga hade valt att antingen inte ha med personer i analogin eller som Johan gjorde i sin snögubbeanalogi, lyfte fram vikten av kommunikation mellan projektdeltagarna.

Informanternas syn på vad deras analogier beskrev skiljde sig åt. De personer som valde att beskriva med hjälp av konstruktion ansåg att slutprodukten också var en analogi av ett system. I deras fall, att ett system är att likna vid ett hus. Det är möjligt att detta hör ihop med det hårda systemtänkandet som fanns hos dessa informanter. Hos de som använde sig av evolutionära liknelser skiljde det sig åt mellan beskrivning av processen endast, och redogörelsen av både processen och systemet. Vad som ligger till grund för denna avvikelse kan vi endast spekulera kring. Vi kan dock se en koppling till att det hårda systemtänkandet dominerade hos alla parter. De informanter som ansåg att analogin beskrev både process och system hade alla ett hårt systemtänkande vid definition av system. Det var endast en av informanterna, Mikael, som skrev att hans analogi endast beskrev systemutvecklingsprocessen. Vi hade vissa svårigheter att definiera Mikael's systemdefinition tillskillnad från Johans. Det är mycket möjligt att hur

personen ifråga betraktar system, också påverkar personens syn på analogier och hur denne väljer att åskådliggöra systemutveckling med hjälp av analogi. Detta kan vara en anledning till varför författare av två evolutionära liknelser betraktar slutprodukten olika, som fallet med Johan och Mikael. Båda två hade dock ett system som kunde förändras i sina analogier

### *Vem är en byggare? Jag är en arkitekt*

Av de olika analogierna som vi tagit del av från informanterna så var Anellas kommentar om rollerna i ett projekt särskilt intressant. I beskrivningen av konstruktion av system så skrev hon att programmerarna var att betrakta som arkitekter och att dessa gjorde en ritning. Liknande tankegångar har framkommit i diskussioner på Internet (Chen, 2007). Denna tanke är väldigt intressant då den pekar på en stor skillnad mellan att bygga hus och konstruera system. När det gäller att konstruera system ligger kostnaden i framtagningen av ritningen till skillnad från konstruktion av hus där den stora kostnaden ligger i uppförandet av byggnaden. Skillnaden i kostnad beror på att byggandet och testandet av system, exempelvis mjukvaran, görs på väldigt kort tid. Byggandet av mjukvara sköts i de flesta fall av en kompilator eller liknande och testandet av särskilda testverktyg som kan automatiseras. Vid uppförande av hus så görs detta manuellt med hjälp av byggare med flera. Det finns inte heller möjlighet för några större ändringar då byggnadnaden är färdig.

### *Konstruktion, evolution, intervention och ...*

En iakttagelse som framkommit vid vår analys av empirin och diskussion kring resultatet är att det kan vara lämpligt att införa ytterligare en utvecklingstyp. Detta för att ha en utvecklingstyp som (bättre än konstruktion, evolution och intervention) representerar de attribut som utmärker den agila utvecklingen. Vi har i detta arbete definierat den agila utvecklingen som evolution med romantiskt perspektiv och mjukt systemtänkande. Hade vi haft tillgång till en utvecklingstyp som bättre passade in på agil utveckling så skulle detta kunnat leda fram till andra samband än de som identifierats i detta arbete. Det hade också minskat förvirringen då det finns både evolutionär utveckling och agil.

## Kapitel 6 Avslutning

I det här kapitlet kommer vi att redogöra för de resultat vi kom fram till i uppsatsen samt förslag till vad som kan tänkas vara intressant att studera vidare om. Vi kommer även att peka på de brister vi identifierat under arbetets gång.

### 6.1 Slutsats

Vårt syfte med detta arbete var att undersöka hur systemutvecklare åskådliggör systemutveckling i förhållande till deras syn på utveckling och de metoder och tekniker de uppgav sig använda. Majoriteten av de tillfrågade använde sig av liknelser som efter analys kunde kopplas till den metod de uppgav sig använda. Det framkom dock skillnader i relationen mellan deras val av vilken utvecklingstyp, konstruktion/evolution, som bäst motsvarade deras syn på systemutveckling och den analogi som de gjorde. De som valde att betrakta systemutveckling som både konstruktion och evolution beskrev analogier som var antingen konstruktion eller evolution. Denna åtskillnad kunde kopplas till skillnaden i perspektiv på världen hos de metoder som informanterna i fråga använt. Så vår slutsats är att använd metod ihop med ens sätt att betrakta systemutveckling bestämmer vilken typ av analogi som används. Denna slutledning är dock osäker då urvalet var litet och inblandning i valet av metod var blandat hos informanterna. Ett större urval av informanter och högre upplösning i frågorna hade kunnat visa att det i de flesta fall antingen är använd metod eller syn på systemutveckling som avgör analogi, till skillnad från denna undersökning som inte såg några större skillnader.

### 6.2 Tankar kring det hårda

I arbetet har vi observerat att sättet att betrakta systemutveckling ändras med erfarenhet. Vi har också sett att byggnadsanalogin främst har gehör hos äldre och mer erfarna utvecklare. Vårt urval av informanter är dock för litet för att detta ska vara säkerställt.

Vi kunde observera att det dominerande tänkandet kring system är att betrakta som hårt systemtänkande och att de analogier som personerna hört talas om var konstruktionsanalogier eller gav uttryck för ett hårt systemtänkande. Det hårda systemtänkandet rimmar illa med de metoder som flera av våra informanter använde. I inledningen så presenterade vi statistik från 2007. Statistiken föranleder oss till att påstå, att det behövs en djupare kontinuerlig diskussion om hur systemutvecklare betraktar system och vad det är en systemutvecklare egentligen gör. Diskussionen finns men vi har i studien lagt fram en annan väg att representera systemutvecklarens arbete och åsikter. En diskussion som tar avstamp i detta skulle kunna leda



fram till ett nytänkande kring systemutveckling. Det kan vara så att projekt misslyckas för att systemutvecklare har ett hårt systemtänkande kring system, vilket leder till en felaktig tillämpning av de metoder de använder.

### 6.3 Vidare forskning

Utifrån den diskussion vi fört i detta arbete har vi identifierat ett antal olika forskningsmöjligheter. Det första som slog oss var att det borde göras en undersökning av hur systemutvecklare ser på systemutveckling i förhållande till hur erfarna dessa är. Det skulle vara intressant att se efter hur många års erfarenhet denna skillnad uppkommer och om den ens uppkommer.

Det kan även vara intressant att i en större studie undersöka hur systemutvecklare väljer att definiera system. Har nästan alla ett hårt systemtänkande och endast i undagsfall ett mjukt eller dialektiskt eller finns det koppling till deras roll inom projekt och typen av projekt?

Förutom ovanstående systemteoretiska forskningsförslag så kan vidare studier av systemutvecklingsanalogier nog ge upphov till en hel del tänkvärda idéer kring systemutveckling och de olika rollerna inom projekt. I vår undersökning var det främst Anella som hade en nyanserad syn på programmerare som arkitekter istället för det mer dominerande tänket om systemarkitekter.

Avslutningsvis så tror vi att en systemteoretisk undersökning som inkluderar ett flertal olika kulturer kan belysa om systemtänkandet är ”universiellt” eller ifall det är kulturberoende. Det är möjligt att vi i västvärlden domineras av ett hårt systemtänkande medan länderna i östvärlden domineras av ett mjukt systemtänkande eller rent av ett dialektiskt.

### 6.4 Kommentar

Vi har under arbetets gång identifierat ett antal brister i vår undersökning. Dessa brister rör främst vårt val av frågor samt formulering av dessa frågor. Frågor rörande vilken typ av projekt som informanterna arbetade med samt storleken och komplexiteten upptäckte vi kunde vara avgörande för att förklara skillnaderna mellan utvecklingstyp (konstruktion/evolution) samt analogi. Fler frågor rörande använd metod skulle också hjälpt till att underlätta analysen i de fall där använd metod var svår att kategorisera. För att underlätta analysen så borde några frågor ha åtföljts av en uppmaning till informanterna att motivera sitt val och/eller sitt ställningstagande. Det hade varit lämpligt att göra en enklare förstudie för att tidigare hitta brister i frågorna och hur informanter tolkade dessa.

## Bilaga 1 Enkät

### Bakgrund

1. Hur gammal är du?
2. Vilken utbildning har du?
3. Hur lång erfarenhet har du inom mjukvarubranschen?

### Praktik

4. Utvecklar du för organisationens interna bruk eller direkt mot kund?
5. Vilken/vilka utvecklingsmetoder använder du dig av i arbetet?
6. Vad låg till grund för valet av de/den metoden? Var det ett medvetet val?
7. Finns det en livscykel i ditt systemutvecklingsarbete såsom att ett projekt genomgår olika faser och i så fall vilka faser du finns?
8. Om du skulle prioritera faserna i vilken ordning kommer dessa? Vilken fas är viktigast?
9. Vilka tekniker/verktyg använder ni i ert arbete såsom användarfall, klassdiagram, kravspecifikation?
10. Finns det andra metoder som du hellre skulle använda, vilka och varför i så fall?
11. Vilket/vilka av följande alternativ tycker du speglar din syn på systemutveckling.  
Systemutveckling är en
  - a. konstruktionsprocess där systemet utvecklas stegvis utifrån en abstrakt helhetsbild av systemet, kravspecifikation, som delas upp i mindre delar och förfinas.
  - b. evolutionärprocess där partiella problem identifieras och lösningar utvärderas allteftersom med hjälp av användarna vilket leder till en helhetsbild av systemet?
12. Hur definierar du ett system i systemutveckling?

### Analogi

13. Vilken är den mest vanligt förekommande liknelsen du hört talas om vid utveckling av system?
14. Varför tror du att denna liknelse görs?
15. Tror du liknelser, metaforer och allmänt filosoferande kan vara ett bra sätt att öka förståelsen av sin egen systemutveckling och systemutveckling generellt? Är det något du själv använt och känt dig hjälpt av?
16. Beskriv med hjälp av liknelse (illustrativt) din systemutveckling.  
ex Systemutveckling är som matlagning, projektledaren är kökschefen och kockarna programmerarna,... receptet är planeringen och specifikation ... och det färdiga systemet är en gryta där de olika ingredienserna samverkar för att skapa en helhet.
17. Är slutresultatet i din liknelse även hur du betraktar system, ex ”grytan” i analogiexemplet, eller är din analogi endast en beskrivning av utvecklingsprocessen?

## Bilaga 2 Empirisammanställning

Tabell 1

Fråga 1: Hur gammal är du?

Fråga 2: Vilken utbildning har du?

Fråga 3: Hur lång erfarenhet har du inom mjukvarubranschen?

Fråga 4: Utvecklar du för organisationens interna bruk eller direkt mot kund?

	<b>Ålder</b>	<b>Utbildning</b>	<b>Arbetat</b>	<b>Primär utveckling</b>
<b>Mattias</b>	23	Kurs på komvux	4 år	Internt
<b>Erik</b>	24	Civilingenjör	9 månader	Internt
<b>Linus</b>	24	Civilingenjör	6 månader	Externt
<b>Mikael</b>	24	Civilingenjör	6 månader	Internt och externt
<b>Johan</b>	30	Magister datalogi, kandidat informatik	6 år	Internt och externt
<b>Tobias</b>	35	Civilingenjör	10 år	Internt och externt
<b>Anella</b>	37	IT-examen	5 år	Externt

Tabell 2

Fråga 5: Vilken/vilka utvecklingsmetoder använder du dig av i arbetet?

Fråga 9: Vilka tekniker/verktyg använder ni i ert arbete såsom användarfall, klassdiagram, kravspecifikation?

	<b>Metod</b>	<b>Tekniker</b>
<b>Mattias</b>	XP	Parprogrammering, Planning game, User stories,
<b>Erik</b>	XP	Iterativt, TDD, User Stories
<b>Linus</b>	Intern	Kravspec, Diagram, Användarfall
<b>Mikael</b>	Intern	Inga
<b>Johan</b>	SCRUM, XP	Produkt och iterationsbacklogsverktyg
<b>Tobias</b>	RUP, PPS	UML, Användarfall, Kravspec, CM, Testverktyg
<b>Anella</b>	CENELEC	Kravspec, användarfall

Tabell 3

Fråga 7: Finns det en livscykel i ditt systemutvecklingsarbete såsom att ett projekt genomgår olika faser och i så fall vilka anser du finns?

Fråga 8: Om du skulle prioritera faserna i vilken ordning kommer dessa? Vilken fas är viktigast?

	Faser	Prioritering av faser
<b>Mattias</b>	Design, implementering, testning, release	Alla
<b>Erik</b>	Design, implementering, testning	Alla
<b>Linus</b>	Implementering, refaktorisering, testning, release	Beror på situation
<b>Mikael</b>	Utveckling, release	Båda
<b>Johan</b>	Brainstorming, prototyping, implementering, release, underhåll	Designbeslut, processen bör vara flexibel
<b>Tobias</b>	Analys, design, implementering, testning, release	Analys, testing, release, design, implementering
<b>Anella</b>	CENELEC *	Analys

\*Koncept, systemdefinition, riskanalys, kravspecificering, design och implementering, med flera 14 st totalt.

Tabell 4

Fråga 6: Vad låg till grund för valet av de/den metoden? Var det ett medvetet val?

Fråga 10: Finns det andra metoder som du hellre skulle använda, vilka och varför i så fall?

	Medvetet val	Motivering	Annan metod
<b>Mattias</b>	Ja	Litet team	Nej, förbättra implementering av nuvarande
<b>Erik</b>	Ja	Litet team	SCRUM
<b>Linus</b>	Nej	-	Mer strukturerade metoder
<b>Mikael</b>	Nej (Chefens beslut)	Flexibilitet	Nej, nuvarande passar
<b>Johan</b>	Ja	Strukturerad, flexibel, iterativ utveckling.	Nej
<b>Tobias</b>	Ja	Kunskap om metoderna sen tidigare	Nej, men sträva efter förbättring.
<b>Anella</b>	Nej (krav)	Ett krav	Inget val

Tabell 4

Fråga 11: Vilket/vilka av följande alternativ tycker du speglar din syn på systemutveckling? Systemutveckling är

- konstruktionsprocess där systemet utvecklas stegvis utifrån en abstrakt helhetsbild av systemet, kravspecifikation, som delas upp i mindre delar och förfinas
- evolutionärprocess där partiella problem identifieras och lösningar utvärderas allteftersom med hjälp av användarna vilket leder till en helhetsbild av systemet?

	Passande beskrivning av utveckling	Motivering
<b>Mattias</b>	B	-
<b>Erik</b>	B	Nuvarande utveckling. Beror på projekt.
<b>Linus</b>	A	Tydliga krav från början
<b>Mikael</b>	B	-
<b>Johan</b>	A och B	Utveckling är en kombination av båda och kan bero på situation.
<b>Tobias</b>	A och B	Utveckling är en kombination av båda.
<b>Anella</b>	A	-

Tabell 6

Fråga 12: Hur definierar du ett system i systemutveckling?

Definition av system	
<b>Mattias</b>	Det man kodar.
<b>Erik</b>	Summan av projektets fysiska komponenter i den miljö som den faktiskt används.
<b>Linus</b>	-
<b>Mikael</b>	Allting som på något sett rör det man arbetar med.
<b>Johan</b>	En mängd komponenter som samverkar för att lösa ett informationshanteringsproblem.
<b>Tobias</b>	Ett system för mig är ett urval av funktioner som hänger ihop och utför en uppsättning av funktioner för att uppnå ett definierat resultat. Ett system har alltid en motor eller hjärta som gör att det hela tiden tickar, som jag ser det skiljer just detta det från en komponent som inte själv lever utan får impulser och sedan lämnar ett svar.
<b>Anella</b>	Ett system är uppbyggt av moduler som i sin tur är uppbyggt av block.

Tabell 7

Fråga 13: Vilken är den mest vanligt förekommande liknelsen du hört talas om vid utveckling av system?

Fråga 14: Varför tror du att denna liknelse görs?

	Liknelser	Motivering
<b>Mattias</b>	-	-
<b>Erik</b>	Husbygge	För att det behövs samarbete och att det är ett komplicerat projekt med kända problem och metoder.
<b>Linus</b>	Systemutveckling är som Lego, man får akta sig vart man sätter fötterna. Systemutveckling är som att bygga med Lego, man får inga flickor.	För att det är viktigt att vara systematisk och noggrann.
<b>Mikael</b>	-	-
<b>Johan</b>	Bilkonstruktion	Det är alltid svårt att beskriva abstrakta fenomen och genom att göra jämförelse med kända fenomen så hoppas man underlätta förståelse.
<b>Tobias</b>	Husbygge	För att det är en komplex process bestående av flera specialiteter och gränssnitt mellan olika subsystem.
<b>Anella</b>	-	-

Tabell 8

Fråga 15: Tror du liknelser, metaforer och allmänt filosoferande kan vara ett bra sätt att öka förståelsen av sin egen systemutveckling och systemutveckling generellt? Är det något du själv använt och känt dig hjälpt av?

Förståelse kring den egna utvecklingen med hjälp av liknelser och metaforer.	
<b>Mattias</b>	Det har pedagogiskt och mnemoniskt värde och framförallt, är väldigt kul. Men i övrigt tror jag mest att det är skadligt att utgå ifrån liknelser.
<b>Erik</b>	Nej, det tror jag faktiskt inte. Kan leda till en felaktig uppfattning.
<b>Linus</b>	Tror att det liknelser och metaforer kan vara bra för, är att lära sig och diskutera olika principer. Men det ska inte överdrivas.
<b>Mikael</b>	Tror man lätt <b>kan</b> måla in sig i metaforer, men det kan vara ett bra verktyg.
<b>Johan</b>	-
<b>Tobias</b>	Det är bra sätt att beskriva skeenden.
<b>Anella</b>	Det tror jag absolut.

Tabell 9

Fråga 16: Beskriv med hjälp av liknelse (illustrativt) din systemutveckling.

ex Systemutveckling är som matlagning, projektledaren är kökschefen och kockarna programmerarna,... receptet är planeringen och specifikation ... och det färdiga systemet är en gryta där de olika ingredienserna samverkar för att skapa en helhet.

Fråga 17: Är slutresultatet i din liknelse även hur du betraktar system, ex "grytan" i analogiexemplet, eller är din analogi endast en beskrivning av utvecklingsprocessen?

	Liknelser	Stämmer slutresultatet med den gjorda liknelsen.
<b>Mattias</b>	Systemutveckling är som julen. Börjar ett par månader innan den ska vara klar, alla pratar och det och stressar upp sig över det hela.	Nej, grytan är för mig de fulla, förättna släktingarna (ryde strax efter deadline)*.
<b>Erik</b>	Mjukvaruprojekt är som sandlott, ju mer det växer desto större är risken att du en dag upptäcker att det du har inte längre är ett vackert slott, utan bara en stor hög med sand	Ja.
<b>Linus</b>	-	-
<b>Mikael</b>	Planlösningen utvecklas snarare med huset.	Det är en beskrivning av utvecklingsprocessen.
<b>Johan</b>	Har inte funderat så mycket på detta, men något jag kommer att tänka på av någon anledning är att bygga en jättestor snögubbe i ett lag.	Nog processen, men också att själva snögubben är något som kan formas kontinuerligt ... .
<b>Tobias</b>	När man skall bygga ett hus har man ett bygglag bestående av flera olika typer av experter och utförare.	Yes!
<b>Anella</b>	Jag ser det som att vi bygger ett hus med moduler som vi bygger ihop för att slutligen få vårt färdigbyggda hus.	Analogin är en beskrivelse av processen och systemet.

\* Ryde är en arbetskamrat.

## Bilaga 3 Enkät svar

### Mattias

#### Bakgrund

1. Hur gammal är du?  
23
2. Vilken utbildning har du?  
*Nån rutten javakurs på komvux.*
3. Hur lång erfarenhet har du inom mjukvarubranschen?  
*Jag har arbetat som programmerare i fyra år, och hade innan dess pysslat med det i kanske tre. Nerd lifestyle.*

#### Praktik

4. Utvecklar du för företagets interna bruk eller direkt mot kund?  
*Endast internt bruk.*
5. Vilken/vilka utvecklingsmetoder använder du dig av i arbetet?  
*Parprogrammering, Test-driven (test-first, red-green-refactor) utveckling, Lagerseparation, iterativ utveckling & planering, gemensamt kodägarskap. Kund (XP) i teamet.*
6. Vad låg till grund för valet av de/den metoden? Var det ett medvetet val?  
*Vi kände att många av XP-metoderna passade oss, eftersom vi bygger ett webbprojekt och releaser därmed är väldigt billiga, eftersom vi är ett litet team och har möjlighet att kommunicera mycket verbalt och eftersom vi arbetar väldigt, väldigt nära kund och därmed har extremt flukturerande krav. Test-first är en metod som Mattias tyckte passade hans psyke, och därmed forcerade på utvecklingsteamet i början. Anledningen till att vi FORTSÄTTER använda den har nu mer att göra med att vi tycker det vore helt befängat att inte använda det, nästan till den grad att vi ifrågasätter kompetensen hos den som inte gör det.*
7. Finns det en livscykel i ditt systemutvecklingsarbete såsom att ett projekt genomgår olika faser och i så fall vilka faser anser du finns?  
*När projektet byggdes hade vi en designfas på ett par veckor där vi lade upp en detaljerad specifikation. Sedan följde fasen där vi implementerade systemet, en feature i taget, och slängde specifikationen som snart visade sig värdelös. Sedan gjorde vi en beta-release när vi hade minimiuppsättningen funktionalitet. Det som kanske är lite mer intressant är den cykel vi följer vid varje iteration, som varar i två veckor. Vi börjar med ett planning game på måndagen, där våra två Kunder (kundtjänst och produktionsansvariga på företaget) får prioritera nya features och fixar på sidan som skrivs ned som små korta historier på a5-kort, även känt som XP-stories. Sedan sätter vi oss ned och skriver test och kod. Under veckan skriver våra kunder även acceptanstest, som, innan release i slutet av iterationen, körs på testservern. Release sker nästan alltid i slutet av iterationen - om någon feature inte hinns med så skalar vi bort den från iterationen*

*och releasas i alla fall. I undantagsfall så crammar vi några dagar (läs: helg) extra, men det hör till undantagen och vi bedömer det inte som god praxis.*

8. Om du skulle prioritera faserna i vilken ordning kommer dessa? Vilken fas är viktigast?  
*Iterativ utveckling, baby. Inte så mycket faser. Faser suger. Min viktigaste fas på dagen är frukost*
9. Vilka tekniker/verktyg använder ni i ert arbete såsom användarfall, klassdiagram, kravspecifikation?  
*Vi har ingen speciell teknik för planering annat än xps traditionella Planning game. Vi gör ibland mer detaljerade planer för tekniskt komplexa funktioner, alltså story cards som kräver många tasks, men för det mesta upplever vi att detaljerad planering inte är speciellt användbar.*
10. Finns det andra metoder som du hellre skulle använda, vilka och varför i så fall?  
*Egentligen är jag ganska nöjd med vår process - det jag skulle vilja ha är bättre, löpande kommunikation med kunderna och mer definierad process för testing, där acceptanstesterna skrivs tillsammans med kunden innan man påbörjar skrivandet.*
11. Vilket/vilka av följande alternativ tycker du speglar din syn på systemutveckling. Systemutveckling är en
  - a. konstruktionsprocess där systemet utvecklas stegvis utifrån en abstrakt helhetsbild av systemet, kravspecifikation, som delas upp i mindre delar och förfinas.
  - b. evolutionärprocess där partiella problem identifieras och lösningar utvärderas allteftersom med hjälp av användarna vilket leder till en helhetsbild av systemet?

*Två, baby\**

*\* Här avses svar b.*

12. Hur definierar du ett system i systemutveckling?  
*Skiten man kodar.*

## **Analogi**

13. Vilken är den mest vanligt förekommande liknelsen du hört talas om vid utveckling av system?  
*Jag har aldrig hört nån använda en liknelse vid beskrivning av kod och programmeringsstrukturer, faktiskt. I det här fallet var ryde tvungen att förklara konceptet för mig. Under den tid jag programmerat har jag aldrig hört talas om begreppet liknelser. I shit you not.*
14. Varför tror du att denna liknelse görs?  
---
15. Tror du liknelser, metaforer och allmänt filosoferande kan vara ett bra sätt att öka förståelsen av sin egen systemutveckling och systemutveckling generellt? Är det något du själv använt och känt dig hjälpt av?  
*Vi använder till viss del metaforer för saker och ting när vi programmerar, till exempel DataElephant, MessageOwl och CardMachine, eftersom det har pedagogiskt och mnemoniskt värde, och framförallt är väldigt kul, men i övrigt tror jag mest att det är skadligt att utgå ifrån liknelser såsom att utveckling är som att bygga ett hus, etc. eftersom man försöker pådyvla en verklighets modell på en annan i stället för att se varje unika situations krav. Lite som patterns - de kan vara användbara, men alltför många programmerare tänker på det som något man utgår ifrån, inte det som det är - ett sätt att lösa ett problem som uppstår. Om man håller på och utgår från modeller och metaforer och crap så är risken stor att man skapar problem i stället för att låta dem uppstå naturligt.*



16. Beskriv med hjälp av liknelse (illustrativt) din systemutveckling.  
ex Systemutveckling är som matlagning, projektledaren är kökschefen och kockarna programmerarna,...  
receptet är planeringen och specification ... och det färdiga systemet är en gryta där de olika ingredienserna samverkar för att skapa en helhet.  
*Systemutveckling är som julen. Den börjar ett par månader innan den ska vara klar, alla pratar och det och stressar upp sig över det hela. Sen när man äntligen når release så är alla bara trötta på hela grejen.*
17. Är slutresultatet i din liknelse även hur du betraktar system, ex ”grytan” i analogiexemplet, eller är din analogi endast en beskrivning av utvecklingsprocessen?  
*Nej, grytan är för mig de fulla, förätna släktingarna (ryde strax efter deadline).*

## Erik

### Bakgrund

1. Hur gammal är du?  
24
2. Vilken utbildning har du?  
*Civilingenjör Datateknik, LTH*
3. Hur lång erfarenhet har du inom mjukvarubranschen?  
*Jag har programmerat sedan jag var 13, varit inblandad i diverse projekt under min studietid, och sedan utexamineringen arbetat med det på heltid sedan april i år.*

### Praktik

4. Utvecklar du för företagets interna bruk eller direkt mot kund?  
*Internt bruk.*
5. Vilken/vilka utvecklingsmetoder använder du dig av i arbetet?  
*Vi arbetar primärt enligt principer hämtade från XP. Iterativt, testdrivet och med täta releaser.*
6. Vad låg till grund för valet av de/den metoden? Var det ett medvetet val?  
*Det föll sig naturligt då teamet var litet.*
7. Finns det en livscykel i ditt systemutvecklingsarbete såsom att ett projekt genomgår olika faser och i så fall vilka faser anser du finns?  
*Eftersom projektet är evolutionärt och fortfarande pågår så är det snarare fråga om en cykel snarare än en vattenfallsliknande uppsättning faser. I början av projektet hade vi naturligtvis större betoning på kravhantering och design. Nu består arbetet av iterationer som inleds med att prioritera ny funktionalitet i samband med våra interna intressenter (kunder), varpå vi arbetar med att implementera det i två veckor. I slutet av iterationen så körs acceptanstest manuellt för att se att det implementerade uppfyller kraven som definierats. Eftersom vi programmerar testdrivet så har testning ingen separat fas utöver det, utan är en naturlig del av implementeringsarbetet.*
8. Om du skulle prioritera faserna i vilken ordning kommer dessa? Vilken fas är viktigast?  
*En sådan jämförelse är inte så meningsfull i vårt fall. Givetvis bör kravhanteringen skötas med eftertanke så att vi utvecklar rätt sak, men det är inte viktigare än att lägga tid på att göra det ordentligt, vältestat eller bra. Vi försöker eftersträva att arbeta så att ingen del försummas.*
9. Vilka tekniker/verktyg använder ni i ert arbete såsom användarfall, klassdiagram, kravspecifikation?  
*Vid planeringsmötet inför iterationer så skriv stories för vad den nya funktionaliteten ska innefatta på små kort som prioriteras. Dessa prioriteras sedan och förs in i våra delade dokument. Oftast görs ingen närmare design än så, iaf. inte av typen som underhålls eller sparas i form av klassdiagram el. dyl.*
10. Finns det andra metoder som du hellre skulle använda, vilka och varför i så fall?  
*SCRUM, för att kunna berätta för folk att vi använder det och se deras oförstående blickar.*
11. Vilket/vilka av följande alternativ tycker du speglar din syn på systemutveckling.  
Systemutveckling är en
  - a. konstruktionsprocess där systemet utvecklas stegvis utifrån en abstrakt helhetsbild av systemet, kravspecifikation, som delas upp i mindre delar och förfinas.

- b. evolutionärprocess där partiella problem identifieras och lösningar utvärderas allteftersom med hjälp av användarna vilket leder till en helhetsbild av systemet?

*Det där är en ledande och dum fråga. Systemutveckling som koncept kan vara båda delarna beroende på projektet. I vårt fall är den evolutionär.*

12. Hur definierar du ett system i systemutveckling?  
*Det gör jag inte. Jag arbetar istället. ;) Men annars: Summan av projekts fysiska komponenter i den miljö som den faktiskt används?*

### Analogi

13. Vilken är den mest vanligt förekommande liknelsen du hört talas om vid utveckling av system?  
*Den enda jag hört talas om är den av ett husbygge. Jag har stött på den primärt inom ramen för kurser jag läst under min utbildning.*
14. Varför tror du att denna liknelse görs?  
*För att den beskriver något som skapas med hjälp av ett team med människor som måste samarbeta, och att den representerar ett komplicerat projekt med kända problem och metoder.*
15. Tror du liknelser, metaforer och allmänt filosoferande kan vara ett bra sätt att öka förståelsen av sin egen systemutveckling och systemutveckling generellt? Är det något du själv använt och känt dig hjälpt av?  
*Nej, det tror jag faktiskt inte. Liknelser innebär att man måste abstrahera något komplicerat till en minsta gemensamma nämnare med någonting annat. Detta medför att man förlorar mycket av den underliggande detaljnivån på det faktiska problemet och tillför en orelaterad detaljnivå för ett annat problem vilket kan leda till en felaktig uppfattning om komponenternas interagerande. En testsvit är inte det samma som en bärande vägg, eller en purjolöksoppa, eller en lastbil på en bro. Jag tror att det är viktigare att man utgår ifrån en förståelse av domänen och om man vill prata i abstrakta termer fortfarande gör det, fast utan att lägga på en liknelse. Detta kräver att man har erfarenhet av det man vill abstrahera för att kunna prata om det i bredare termer.*
16. Beskriv med hjälp av liknelse (illustrativt) din systemutveckling.  
ex Systemutveckling är som matlagning, projektledaren är kökschefen och kockarna programmerarna,... receptet är planeringen och specifikationen ... och det färdiga systemet är en gryta där de olika ingredienserna samverkar för att skapa en helhet.  
*Mjukvaruprojekt är som sandlott, vaje dag bygger man vidare på det, men ju mer det växer desto större är risken att du en dag upptäcker att det du har inte längre är ett vackert slott, utan bara en stor hög med sand.*
17. Är slutresultatet i din liknelse även hur du betraktar system, ex ”grytan” i analogiexemplet, eller är din analogi endast en beskrivning av utvecklingsprocessen?  
*Ja.*

## Linus

### Bakgrund

1. Hur gammal är du?  
*24 år.*
2. Vilken utbildning har du?  
*Civilingenjör i datorteknik.*
3. Hur lång erfarenhet har du inom mjukvarubranschen?  
*4 månader.*

### Praktik

4. Utvecklar du för företagets interna bruk eller direkt mot kund?  
*Direkt mot kund.*
  5. Vilken/vilka utvecklingsmetoder använder du dig av i arbetet?  
*Inga stora väldefinierade metodiker.*
  6. Vad låg till grund för valet av de/den metoden? Var det ett medvetet val?  
*Metodiken är inte ett medvetet val.*
  7. Finns det en livscykel i ditt systemutvecklingsarbete såsom att ett projekt genomgår olika faser och i så fall vilka faser anser du finns?  
*Implementering av ny funktionalitet, refaktorisering, testning och stabilisering inför releaser.*
  8. Om du skulle prioritera faserna i vilken ordning kommer dessa? Vilken fas är viktigast?  
*Alla faser är nödvändiga, vilken fas som behöver prioriteras är beroende av situationen.*
  9. Vilka tekniker/verktyg använder ni i ert arbete såsom användarfall, klassdiagram, kravspecifikation?  
*Ganska informella kravspecifikationer. Olika diagram efter behov. Användarfall vid implementering av ny funktionalitet och testning.*
  10. Finns det andra metoder som du hellre skulle använda, vilka och varför i så fall?  
*Skulle vilja använda lite formellare metoder.*
  11. Vilket/vilka av följande alternativ tycker du speglar din syn på systemutveckling. Systemutveckling är en
    - a. konstruktionsprocess där systemet utvecklas stegvis utifrån en abstrakt helhetsbild av systemet, kravspecifikation, som delas upp i mindre delar och förfinas.
    - b. evolutionärprocess där partiella problem identifieras och lösningar utvärderas allteftersom med hjälp av användarna vilket leder till en helhetsbild av systemet?
- A, Tycker om att utgå från så detaljerade krav som är rimligt att ta fram, eftersom många designval är svåra att göra sent i utvecklingen.*
12. Hur definierar du ett system i systemutveckling?  
-

## Analogi

13. Vilken är den mest vanligt förekommande liknelsen du hört talas om vid utveckling av system?  
*Systemutveckling är som att bygga med Lego, man får akta vart man sätter fötterna.  
Eller Systemutveckling är som att bygga med lego, man får inga flickor.*
14. Varför tror du att denna liknelse görs?  
*För att det är viktigt att vara systematisk och noggrann.*
15. Tror du liknelser, metaforer och allmänt filosoferande kan vara ett bra sätt att öka förståelsen av sin egen systemutveckling och systemutveckling generellt? Är det något du själv använt och känt dig hjälpt av?  
*Jag tror att det liknelser och metaforer kan vara bra för att lära sig och diskutera olika principer, men det ska inte överdrivas. Allmänt filosoferande över arbetsmetoderna ska däremot inte underskattas.*
16. Beskriv med hjälp av liknelse (illustrativt) din systemutveckling.  
ex Systemutveckling är som matlagning, projektledaren är kökschefen och kockarna programmerarna,...  
receptet är planeringen och specification ... och det färdiga systemet är en gryta där de olika ingredienserna samverkar för att skapa en helhet.  
*Kan inte komma på en meningsfull liknelse.*
17. Är slutresultatet i din liknelse även hur du betraktar system, ex ”grytan” i analogiexemplet, eller är din analogi endast en beskrivning av utvecklingsprocessen?  
---

# Mikael

## Bakgrund

1. Hur gammal är du?  
*24*
2. Vilken utbildning har du?  
*Civilingenjör i datateknik så väl som på fyra tentor.*
3. Hur lång erfarenhet har du inom mjukvarubranschen?  
*4 års studier, ett halvår yrkesaktiv.*

## Praktik

4. Utvecklar du för företagets interna bruk eller direkt mot kund?  
*Båda delar.*
5. Vilken/vilka utvecklingsmetoder använder du dig av i arbetet?  
*En odefinierad agil metod. odefinierat som flytande, det finns ingen uttalad metodiken det fungerar agilt i och med att vi bygger på allteftersom istället för att arbeta mot en färdig spec.*
6. Vad låg till grund för valet av de/den metoden?  
*Var det ett medvetet val? Flexibilitet, valet är gjort av chefen.*
7. Finns det en livscykel i ditt systemutvecklingsarbete såsom att ett projekt genomgår olika faser och i så fall vilka faser anser du finns?  
*I dagsläget har vi nog bara faserna utveckling och release.*
8. Om du skulle prioritera faserna i vilken ordning kommer dessa? Vilken fas är viktigast?  
*De är minst lika viktiga båda två eftersom de utesluter varandra eftersom det inte är någon idé att releasa utan kod och ingen idé att koda utan att det blir färdigt, båda faserna är nödvändiga.*
9. Vilka tekniker/verktyg använder ni i ert arbete såsom användarfall, klassdiagram, kravspecifikation?  
*Ingadera.*
10. Finns det andra metoder som du hellre skulle använda, vilka och varför i så fall?  
*Nej, i dagsläget känns det här systemet vettigt, men det lär ändras med tiden.*
11. Vilket/vilka av följande alternativ tycker du speglar din syn på systemutveckling. Systemutveckling är en
  - a. konstruktionsprocess där systemet utvecklas stegvis utifrån en abstrakt helhetsbild av systemet, kravspecifikation, som delas upp i mindre delar och förfinas.
  - b. evolutionärprocess där partiella problem identifieras och lösningar utvärderas allteftersom med hjälp av användarna vilket leder till en helhetsbild av systemet?
12. Hur definierar du ett system i systemutveckling?  
*Allting som på något sätt rör det man arbetar med.*

## Analogi

13. Vilken är den mest vanligt förekommande liknelsen du hört talas om vid utveckling av system?  
*Vet ej.*
14. Varför tror du att denna liknelse görs?  
---
15. Tror du liknelser, metaforer och allmänt filosoferande kan vara ett bra sätt att öka förståelsen av sin egen systemutveckling och systemutveckling generellt? Är det något du själv använt och känt dig hjälpt av?  
*Till en viss grad, tror man lätt kan måla in sig i metaforer. Men det kan vara ett bra verktyg för att förklara något avancerat.*
16. Beskriv med hjälp av liknelse (illustrativt) din systemutveckling.  
ex Systemutveckling är som matlagning, projektledaren är kökschefen och kockarna programmerarna,... receptet är planeringen och specification ... och det färdiga systemet är en gryta där de olika ingredienserna samverkar för att skapa en helhet.  
*Okej, vår utvecklingsmetod är väl som att man skulle bygga ett hus genom att först bygga en koja, sedan komma på att det tex. vore trevligt med ett extra rum, då ta ned en vägg och bygga till det, sedan konstatera att ett kök vore bra och lägga till det också samtidigt som man modifierar de tidigare rummen för att anpassa sig efter tillkomsten av köket. Så fortsätter man allteftersom och bygger om huset ett rum i taget utan att ha en överskådlig planlösning över hela konstruktionen. Planlösningen utvecklas snarare med huset. Det finns heller ingen gräns för hur stort huset kan bli utan det finns alltid möjlighet att lägga till något nytt bara man bygger om det gamla*
18. Är slutresultatet i din liknelse även hur du betraktar system, ex ”grytan” i analogiexemplet, eller är din analogi endast en beskrivning av utvecklingsprocessen?  
*Det är en beskrivning av utvecklingsprocessen.*

## Johan

### Bakgrund

1. Hur gammal är du?  
*30 år*
2. Vilken utbildning har du?  
*Nästan magisterexamen i datalogi och nästan kandidatexamen i informatik.*
3. Hur lång erfarenhet har du inom mjukvarubranschen?  
*6 år.*

### Praktik

4. Utvecklar du för företagets interna bruk eller direkt mot kund?  
*För tillfället internt.*
5. Vilken/vilka utvecklingsmetoder använder du dig av i arbetet?  
*Agil metod, inslag av scrum och extreme programming.*
6. Vad låg till grund för valet av de/den metoden? Var det ett medvetet val?  
*Medvetet val för att få en strukturerad men samtidigt flexibel, iterativ och evolutionär utveckling.*
7. Finns det en livscykel i ditt systemutvecklingsarbete såsom att ett projekt genomgår olika faser och i så fall vilka faser anser du finns?  
*Brainstorming, prototypning, implementering, första utkast, fortsatt iterativ utveckling till release, därefter underhåll och vidareutveckling.*
8. Om du skulle prioritera faserna i vilken ordning kommer dessa? Vilken fas är viktigast?  
*Tidigare designbeslut kan vara avgörande, men viktigast är en process som tillåter stora förändringar och samtidigt ha ett fungerande system.*
9. Vilka tekniker/verktyg använder ni i ert arbete såsom användarfall, klassdiagram, kravspecifikation?  
  
*Verktyg för att hantera produkt och iterationsbacklog.*
10. Finns det andra metoder som du hellre skulle använda, vilka och varför i så fall?  
*Nej det verkar fungera bra för tillfället.*
11. Vilket/vilka av följande alternativ tycker du speglar din syn på systemutveckling. Systemutveckling är en
  - a. konstruktionsprocess där systemet utvecklas stegvis utifrån en abstrakt helhetsbild av systemet, kravspecifikation, som delas upp i mindre delar och förfinas.
  - b. evolutionärprocess där partiella problem identifieras och lösningar utvärderas allteftersom med hjälp av användarna vilket leder till en helhetsbild av systemet?

*Det kan bero på vad som ska göras. Om det är glasklart hur den färdiga produkten ska fungera så är det snarare en konstruktionsprocess. Min erfarenhet är dock att det först under utvecklingen framgår vad som ska göras och då blir det mer evolutionärt.*



12. Hur definierar du ett system i systemutveckling?

*En mängd komponenter som samverkar för att lösa ett informationshanteringsproblem.*

### Analogi

13. Vilken är den mest vanligt förekommande liknelsen du hört talas om vid utveckling av system?

*Vet inte riktigt, men bil-analogier förekommer ju alltid. Tex att det måste finnas ritningar först, och att hela systemets komplexitet är för stort för en person att förstå osv.*

14. Varför tror du att denna liknelse görs?

*Det är alltid svårt att beskriva abstrakta fenomen och genom att göra jämförelser med redan bekanta företeelser hoppas man att kunna förmedla sin förståelse.*

15. Tror du liknelser, metaforer och allmänt filosoferande kan vara ett bra sätt att öka förståelsen av sin egen systemutveckling och systemutveckling generellt? Är det något du själv använt och känt dig hjälpt av?

---

16. Beskriv med hjälp av liknelse (illustrativt) din systemutveckling.

*ex Systemutveckling är som matlagning, projektledaren är kökschefen och kockarna programmerarna,... receptet är planeringen och specifikation ... och det färdiga systemet är en gryta där de olika ingredienserna samverkar för att skapa en helhet.*

*Har inte funderat så mycket på detta, men något jag kommer att tänka på av någon anledning är att bygga en jättestor snögubbe i ett lag. Projektledaren är den enda som står på lite avstånd och tittar och håller inte på med snön själv. Men han ser helheten från där han står. De som bygger själva snögubben ser detaljerna och kan behöva guidning om var det ska till/tas bort snö för att formen ska bli fin. De som bygger måste också ta ett par steg tillbaka ibland för att se hur långt de har kommit samt kommunicera med de andra så att de har en gemensam uppfattning om hur den färdiga snögubben ska se ut.*

17. Är slutresultatet i din liknelse även hur du betraktar system, ex "grytan" i analogiexemplet, eller är din analogi endast en beskrivning av utvecklingsprocessen?

*Nog processen, men också att själva snögubben är något som kan formas kontinuerligt till skillnad från ett traditionellt hus där när man väl har fått balkarna på plats och spikat på brädorna inte kan börja flytta runt stommen.*

## Tobias

### Bakgrund

1. Hur gammal är du?  
*35 år.*
2. Vilken utbildning har du?  
*Civ.ing*
3. Hur lång erfarenhet har du inom mjukvarubranschen?  
*10 år.*

### Praktik

4. Utvecklar du för företagets interna bruk eller direkt mot kund?  
*Båda*
5. Vilken/vilka utvecklingsmetoder använder du dig av i arbetet?  
*RUP, UML samt PPS(projektmodell från TietoEnator)*
6. Vad låg till grund för valet av de/den metoden? Var det ett medvetet val?  
*Valet berodde på tillgång och befintlig kunskap i gruppen / Ja*
7. Finns det en livscykel i ditt systemutvecklingsarbete såsom att ett projekt genomgår olika faser och i så fall vilka faser anser du finns?  
*Specning(Analys)/Design/Implementation/Test/Driftsättning.*
8. Om du skulle prioritera faserna i vilken ordning kommer dessa? Vilken fas är viktigast?  
*Spec/Test/Driftsättning/Design/Implementation i denna ordning från Vänster..*
9. Vilka tekniker/verktyg använder ni i ert arbete såsom användarfall, klassdiagram, kravspecifikation?  
*Use cases, Kvarspecifikationer, CM, Testverktyg.*
10. Finns det andra metoder som du hellre skulle använda, vilka och varför i så fall?  
*De är ok men måste självklart hela tiden förfinas.*
11. Vilket/vilka av följande alternativ tycker du speglar din syn på systemutveckling. Systemutveckling är en
  - a. konstruktionsprocess där systemet utvecklas stegvis utifrån en abstrakt helhetsbild av systemet, kravspecifikation, som delas upp i mindre delar och förfinas.
  - b. evolutionärprocess där partiella problem identifieras och lösningar utvärderas allteftersom med hjälp av användarna vilket leder till en helhetsbild av systemet?

*Jag anser att det är en blandning av båda. Utgångspunkten är (a) och det är på det sättet man hittar sin modell. Sedan kommer problem och upptäcker på vägen att leda till att man behöver gå tillbaka och definiera om och utvärdera nya lösningar och modeller. Det blir automatiskt iterativt men man startar alltid med en förhoppning om att man skall kunna komma igenom på ett varv.*

12. Hur definierar du ett system i systemutveckling?

*Ett system för mig är ett urval av funktioner som hänger ihop och utför en uppsättning av funktioner för att uppnå ett definierat resultat. Ett system har alltid en motor eller hjärta som gör att det hela tiden tickar, som jag det skiljer just detta det från en komponent som inte själv lever utan får impulser och sedan lämnar ett svar.*

### Analogi

13. Vilken är den mest vanligt förekommande liknelsen du hört talas om vid utveckling av system?

*Bygga ett hus.*

14. Varför tror du att denna liknelse görs?

*För det är en komplex process bestående av flera specialiteter och gränssnitt mellan olika subsystem.*

15. Tror du liknelser, metaforer och allmänt filosoferande kan vara ett bra sätt att öka förståelsen av sin egen systemutveckling och systemutveckling generellt? Är det något du själv använt och känt dig hjälpt av?

*Jag tycker generellt det är bra med analogier och liknelser för att beskriva skeenden.*

16. Beskriv med hjälp av liknelse (illustrativt) din systemutveckling.

*ex Systemutveckling är som matlagning, projektledaren är kökschefen och kockarna programmerarna,... receptet är planeringen och specifikationen ... och det färdiga systemet är en gryta där de olika ingredienserna samverkar för att skapa en helhet.*

*När man skall bygga ett hus har man ett bygglag bestående av flera olika typer av experter och utförare typ snickare, murare, målaren(systemarkitekter, specialister, programmerare...) osv... varje sådan grupp har en arbetsledare(delprojektledare). Byggmästaren(projektledaren) ansvarar för att huset blir vad arkitekten(säljaren) sålde till kunden(kund). Bygget startar med att arkitekten överlämnar projektet till byggmästaren och då har byggmästaren möjlighet att ha invändningar och funderingar kring vad som överenskommit(projektöverlämnade). Byggmästaren gör då en projektplan för byggnationen och lägger upp en plan för när olika saker behöver göras, när saker skall beställas illexempel. virke(PC, operativ osv...). Sedan startar bygget med att nyckel personerna i bygget, arbetsledare och experter möts och har ett första byggmöte(projektmöte) där hela planen presenteras och fastslås. Sedan har man regelbundna byggmöten där progress och prognoser presenteras samt man diskuterar problem och lösningar. Sedan görs detaljgenomgångar av alla delar för att se att allt är specificerat och löst(Specifikation och Design). Inom varje arbetsgrupp hålls andra möten för att planera sina aktiviteter och vem som skall göra vad. Byggmästaren går igenom leveransen vart efter rummen blir klara och tar kunden på visning ett antal gånger under resan för att säkerställa att huset blir som det är tänkt. Detta har två syften, säkerställa att man uppfattat den ursprungliga ritningen rätt(kontraktet/specifikationen) och att utöka projektet genom att låta kunden ändra på saker som den vill och därigenom tjäna bättre på bygget samt göra kunden extra nöjd. När allt är klart hålls en besiktning där kunden, arkitekten och byggmästaren går igenom hela leveransen och ser att är levererat enligt planen.*

17. Är slutresultatet i din liknelse även hur du betraktar system, ex "grytan" i analogiexemplet, eller är din analogi endast en beskrivning av utvecklingsprocessen?

*Yes!*

## Anella

### Bakgrund

1. Hur gammal är du?  
*37 år*
2. Vilken utbildning har du?  
*IT-examen, KY Malmö Yrkeshögskola*
3. Hur lång erfarenhet har du inom mjukvarubranschen?  
*4år*

### Praktik

4. Utvecklar du för företagets interna bruk eller direkt mot kund?  
*Direkt mot kund*
5. Vilken/vilka utvecklingsmetoder använder du dig av i arbetet?  
*CENELEC, europeisk järnvägsstandard.*
6. Vad låg till grund för valet av de/den metoden? Var det ett medvetet val?  
*Det är inget val, det är ett måste för att få systemet säkerhetsgodkänt.*
7. Finns det en livscykel i ditt systemutvecklingsarbete såsom att ett projekt genomgår olika faser och i så fall vilka faser anser du finns?  
*CENELEC har fasta faser.*
8. Om du skulle prioritera faserna i vilken ordning kommer dessa? Vilken fas är viktigast?  
*Jag anser att kravfasen är den viktigaste för där läggs grunden för ett system som överensstämmer med kundens krav.*
9. Vilka tekniker/verktyg använder ni i ert arbete såsom användarfall, klassdiagram, kravspecifikation?  
*Kravspecifikation och användarfall.*
10. Finns det andra metoder som du hellre skulle använda, vilka och varför i så fall?  
*Vi har inga större valmöjligheter här, vårt arbete handlar mest om lågnivåspråk och vi har guidelines som ska följas.*
11. Vilket/vilka av följande alternativ tycker du speglar din syn på systemutveckling.  
Systemutveckling är en
  - a. konstruktionsprocess där systemet utvecklas stegvis utifrån en abstrakt helhetsbild av systemet, kravspecifikation, som delas upp i mindre delar och förfinas.  
*JA*
  - b. evolutionärprocess där partiella problem identifieras och lösningar utvärderas allteftersom med hjälp av användarna vilket leder till en helhetsbild av systemet?  
*NEJ*
12. Hur definierar du ett system i systemutveckling?  
*Ett system är uppbyggt av moduler som är uppbyggt av block*

### Analogi

13. Vilken är den mest vanligt förekommande liknelsen du hört talas om vid utveckling av system?  
*Har inte hört talas om någon liknelse.*

14. Varför tror du att denna liknelse görs?

---

15. Tror du liknelser, metaforer och allmänt filosoferande kan vara ett bra sätt att öka förståelsen av sin egen systemutveckling och systemutveckling generellt?

*Det tror jag absolut. Är det något du själv använt och känt dig hjälpt av? Jag har alltid en tekniskt ansvarig för de olika delsystemen men jag vet att liknelser och metaforer inte är något som i större omfattning används av våra civilingenjörer på företaget. Det är teknik och åter teknik som är i fokus.*

16. Beskriv med hjälp av liknelse (illustrativt) din systemutveckling.

ex Systemutveckling är som matlagning, projektledaren är kökschefen och kockarna programmerarna,... receptet är planeringen och specifikationen ... och det färdiga systemet är en gryta där de olika ingredienserna samverkar för att skapa en helhet.

*Jag ser det som att vi bygger ett hus med moduler som vi bygger ihop för att slutligen få vårt färdigbyggda hus. Det svåra är att se till att man inte glömt något så enkelt som ett handtag på ytterdörren för då kommer huset inte att kunna uppfylla den funktion man tänkt sig och som kunden beställt. Projektledaren planerar upp arbetet efter att arkitekterna som är programmerarna har en färdig ritning.*

17. Är slutresultatet i din liknelse även hur du betraktar system, ex ”grytan” i analogiexemplet, eller är din analogi endast en beskrivning av utvecklingsprocessen?

*Analogin är en beskrivelse av processen och systemet.*

## Referenser

- Avison, D., & Fitzgerald, G. (2006). *Information systems development methodologies, techniques & tools* 4th ed. London, McGraw-Hill
- Avison, D., & Wilson, D. (2001). *A viewpoint on software engineering and information systems: What we can learn from the construction industry?* Information and Software Technology, vol 7 (pp 795-799)
- Barnes, K. (2007). The code garden (an analogy that sucks less). <http://codecraft.info/index.php/archives/82/> 2007-11-30
- Checkland, P. (1981). *Systems Thinking, Systems Practice*. Wiley
- Chen, Y. (2007) *Software != Construction (II - The Differences)*. Continuous Learning [www.yinsochen.com/blog/2007/01/04/software-is-not-construction-ii](http://www.yinsochen.com/blog/2007/01/04/software-is-not-construction-ii) 2007-10-12
- Chen, J. (2001) Building web applications, *Information Systems Management*, Vol 18, No 1
- Chromatic. (2003). *Extreme Programming, Pocket Guide*. Sebastopol: O'reilly
- Citerus AB, [www.citerus.se/kunskap/omlattrorligutveckling.4.a939951052f77843580006986.html](http://www.citerus.se/kunskap/omlattrorligutveckling.4.a939951052f77843580006986.html), 2007-11-30
- Creating An Encyclopedia Metaphor, [www.c2.com/cgi/wiki?CreatingAnEncyclopediaMetaphor](http://www.c2.com/cgi/wiki?CreatingAnEncyclopediaMetaphor) 2007-11-30
- Crispen, R. G., & Stuckey Jr, L. D. (1994). *Structural model: Architecture for software designers*. Proceedings of the Conference on TRI-Ada '94 (pp. 272 - 281). New York, ACM
- Dahlbom, B., & Mathiassen, L. (1992). Systems Development Philosophy, *Computers & Society*, vol 22 (pp 12-23)
- Dahlbom, B., & Mathiassen, L. (1993). *Computers In Context: The Philosophy and Practise of Systems Design*. Cambridge, Blackwell
- Fitzgerald, B., Russo, N., & Stolterman, E. (2002). *Information System Development: Methods in practise*. London, McGraw-Hill
- Founders, *Worldwide Institute of Software Architects*: <http://www.wwisa.org/wwisamain/board.htm> 2007-11-12
- Hjelm, M. & Weinander, N. för Dataföreningen, Stockholmskretsen, nätverket Agile. [www.agilesweden.org/omagile.htm](http://www.agilesweden.org/omagile.htm) 2007-11-30
- Holme, I. M., & Solvang, B. K. (1997). *Forskningsmetodik – Om kvalitativa och kvantitativa metoder*. Lund, Studentlitteratur
- Janhager J, (2006). *Systematisk konceptutveckling*, [www.md.kth.se/edu/mce/4F1901/Systematisk konceptutv.pdf](http://www.md.kth.se/edu/mce/4F1901/Systematisk_konceptutv.pdf)
- Kvale, S. (1997). *Den kvalitativa forskningsintervjun*. Lund, Studentlitteratur
- Kuhn, T. S (1962). *The Structure of Scientific Revolutions* 2 end. Chicago
- Larsson M, (2004). *Introduktion*, [www2.dis.uu.se/dsv/education/courses/ht04/systemutveckling/lectures/Lecture%201%20-%20Introduction.pdf](http://www2.dis.uu.se/dsv/education/courses/ht04/systemutveckling/lectures/Lecture%201%20-%20Introduction.pdf)
- Löwgren, J., & Stolterman, E. (2002). *Design av informationsteknik- materialet utan egenskaper*. Lund, Studentlitteratur

- Martial Arts As Software Development Metaphor,  
[www.c2.com/cgi/wiki?MartialArtsAsSoftwareDevelopmentMetaphor](http://www.c2.com/cgi/wiki?MartialArtsAsSoftwareDevelopmentMetaphor), 2007-11-30
- Mathiassen, L., Munk-Madsen, A., Nielsen, P. A., & Stage, J. (2001). *Objektorienterad analys och design*. 2 uppl Lund, Studentlitteratur
- McBride, N. (2003). A viewpoint on software engineering and information systems: integrating the disciplines. *Information and Software Technology*, vol 45 (pp 281-287)
- NE.se,  
[www.nationalencyklopedin.se/jsp/search/search.jsp?h\\_search\\_mode=simple&h\\_advanced\\_search=false&t\\_word=m](http://www.nationalencyklopedin.se/jsp/search/search.jsp?h_search_mode=simple&h_advanced_search=false&t_word=m)  
etodologi, 2007-11-30
- New Analogies For Software, <http://www.c2.com/cgi/wiki?NewAnalogiesForSoftware>, 2007-11-30
- Oates, B. J. (2006). *Researching Information Systems and Computing*. London, Sage
- Ortiz-Arroyo, D. (2006), *Philosophy of System Development*, [http://www.cs.aue.auc.dk/~do/Teaching/F06/PSD/PSD Notes \(3slides-page\).pdf](http://www.cs.aue.auc.dk/~do/Teaching/F06/PSD/PSD%20Notes%20(3slides-page).pdf)
- Pizka, M., & Bauer, A. (2004). *A brief top-down and bottom-up philosophy on software evolution*. Proceedings of the 7th International Workshop on Principles of Software Evolution (pp. 131-136)
- Projectplace International AB. (2007),  
[files.projectplace.com/swedish/reports/lyckat\\_misslyckat\\_it\\_projekt\\_2007.pdf](http://files.projectplace.com/swedish/reports/lyckat_misslyckat_it_projekt_2007.pdf)
- Repici, D. J. *Agile Software Development and Bridge Building*;  
<http://www.creativyst.com/Doc/Articles/Mgt/AgileBridges/AgileBridges.htm> 2007-12-10
- Role of the Software Architect, *Worldwide Institute of Software Architects*;  
<http://www.wvisa.org/wwisamain/role.htm> 2007-11-12
- Sewell, M. T., & Sewell, L. M. (2002). *The Software Architect's Profession: An Introduction*. Upper Saddle River, Prentice Hall
- Sommerville, I. (2001). *Software Engineering* 6th ed. Essex, Pearson
- Sprez, *Symphony or Jazz Band Metaphor for Software Development*;  
[www.sprez.com/articles/symphonyorjazzband.html/](http://www.sprez.com/articles/symphonyorjazzband.html/), 2007-11-30
- Sridhar, N., & VenuGopal, B. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM*, vol 50 (pp 79-83)
- Svenning, C. (2003). *Metodboken*. 5 uppl Eslöv, Conny Svenning och Lorentz Förlag
- Sörman, S. (2005). Systemutvecklingsprocess LIME. Version 1. Stockholm, Medicinsk informatik  
[http://courses.ki.se/kommentarer\\_-\\_systemutvecklingsprocess\\_lime.pdf?node=76609](http://courses.ki.se/kommentarer_-_systemutvecklingsprocess_lime.pdf?node=76609)
- Visible Workings, *Analogy Fest*; <http://www.visibleworkings.com/analogyfest>, 2007-11-30