

**Dynamiska metoder för små systemutvecklingsprojekt**

**(HS-IDA-EA-03-402)**

**Reham Ahmed** (a00rehah@ida.his.se)

*Institution för datavetenskap  
Högskolan i Skövde, Box 408  
S-54128 Skövde, SWEDEN*

Examensarbete på det dataekonomiska programmet under vårterminen 2003.

Handledare: Camilla Andersson

### **Dynamiska metoder för små systemutvecklingsprojekt**

Examensrapport inlämnad av Reham Ahmed till Högskolan i Skövde, för Kandidatexamen (B.Sc.) vid Institutionen för Datavetenskap.

**2003-06-08**

Härmed intygas att allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och att inget material är inkluderat som tidigare använts för erhållande av annan examen.

Signerat: \_\_\_\_\_

## **Dynamiska metoder för små systemutvecklingsprojekt**

**Reham Ahmed (a00rehah@ida.his.se)**

### **Sammanfattning**

Litteratur inom systemutvecklingsområdet visar behov av nya snabbare systemutvecklingsmetoder. Orsaken är komplexiteten både i systemutvecklingsprojekten och i organisationer, där systemutveckling sker. Nu föredras små projekt, som omfattar färre än tio deltagare och genomförs på mindre än ett år. De traditionella systemutvecklingsmetoderna anses vara lämpliga för stora projekt. Det finns nya systemutvecklingsmetoder, som får benämningen dynamiska metoder, för små projekt. I detta examensarbete studeras faktorer för två dynamiska metoder som gör dem lämpliga för små projekt. De två dynamiska metoderna som kommer att studeras är Extreme Programming och en specifik dokumenterad anpassning av RUP.

Syftet med detta examensarbete är att studera metoddokumentationen för två dynamiska metoder för att reda ut i vilken utsträckning de passar för små projekt. Ett ytterligare syfte är att genom analys av andras erfarenheter om tillämpningar av de två metoderna redogöra för det som styrker argumenten om deras lämplighet för små projekt.

För att besvara frågeställningen studerades litteratur och ett antal artiklar. Två telefonintervjuer genomfördes också för att besvara en av de två delfrågorna, som tillsammans utgör frågeställningen för denna studie. Resultatet visar att den specifikt dokumenterade anpassade RUP inte följer alla aspekter för små projekt. Däremot följer Extreme Programming alla de aspekterna.

**Nyckelord:** Informationssystem, Systemutveckling, Små systemutvecklingsprojekt, Agile methods, Extreme Programming, Rational Unified Process

## Innehållsförteckning

<b>1</b>	<b>Introduktion .....</b>	<b>1</b>
<b>2</b>	<b>Bakgrund .....</b>	<b>3</b>
2.1	Systemutveckling .....	3
2.1.1	Traditionell systemutveckling .....	4
2.1.2	Hjälpmiddel i systemutvecklingsprocessen .....	5
2.1.3	Komplexitet i systemutveckling .....	6
2.2	Systemutvecklingsmetoder .....	8
2.2.1	Närliggande begrepp .....	8
2.2.2	Olika typer av systemutvecklingsmetoder .....	9
2.2.3	Uppfattningar kring metodtillämpningar .....	10
2.2.4	Behovet av alternativ till traditionella systemutvecklingsmetoder ....	11
2.3	Systemutvecklingsmetoder för små projekt .....	12
2.3.1	Manifestet och de tolv principerna för dynamiska metoder .....	14
2.3.2	Uppfattningar om de dynamiska metoderna .....	16
2.3.3	Två dynamiska metoder för små projekt .....	17
<b>3</b>	<b>Problemformulering .....</b>	<b>19</b>
3.1	Problembeskrivning .....	19
3.2	Problemprecisering .....	20
<b>4</b>	<b>Metod .....</b>	<b>21</b>
4.1	Översikt .....	21
4.2	Litteraturstudie .....	21
4.3	Intervjuer .....	22
4.4	Fallstudie .....	23
4.5	Metodtillämpning .....	24
<b>5</b>	<b>Systemutvecklingsmetoder för små projekt .....</b>	<b>25</b>
5.1	Aspekter för små projekt .....	25
5.2	Extreme Programming för små projekt .....	26
5.2.1	Principerna för ett litet XP-projekt .....	26
5.2.2	Arbetsprocessen för små XP-projekt .....	30
5.2.3	Sammanfattning av de karaktäristiska dragen som gör XP lämplig för små projekt .....	32
5.3	En specifik anpassad RUP för små projekt .....	32
5.3.1	Principer och antagande för ett litet RUP-projekt .....	32
5.3.2	Arbetsprocessen för ett litet RUP-projekt .....	34
5.3.3	Sammanfattning av de karaktäristiska dragen som gör den anpassade RUP lämplig för små projekt .....	36
<b>6</b>	<b>Erfarenheter från metodtillämpningar .....</b>	<b>37</b>
6.1	Erfarenheter kring XP för små projekt .....	37
6.2	Erfarenheter kring den specifikt dokumenterade anpassade RUP för små projekt .....	38
6.3	Intervjuer .....	39
6.3.1	Förberedelser inför intervjuerna .....	39
6.3.2	Resultat av intervjun om XP för små projekt .....	40
6.3.3	Resultat av intervjun om en anpassad RUP för små projekt .....	42
6.4	Sammanfattning av erfarenheter kring de två metoderna för små projekt .....	44

## 1. Inledning

<b>7 Resultat</b> .....	<b>45</b>
<b>8 Analys</b> .....	<b>53</b>
8.1 Analys av resultatet.....	53
8.1.1 Ramverkets lämplighet för att analysera resultatet.....	54
8.1.2 Djupare analys av resultatet .....	55
8.2 Relaterat arbete .....	56
<b>9 Slutsatser och diskussion</b> .....	<b>57</b>
9.1 Arbetets slutsatser .....	57
9.2 Reflektion .....	59
9.3 Förslag till framtida studier .....	60
<b>Referenser</b> .....	<b>62</b>

# 1 Introduktion

Vid utveckling av ett informationssystem behövs en metod som beskriver hur utvecklingsprojektet ska utföras och hur kraven för systemet ska hanteras (Avison och Fitzgerald, 2003, s. 21). En systemutvecklingsmetod är ett hjälpmedel för ett systemutvecklingsprojekt (Andersen, 1994, s. 94).

En begränsad tid och kostnad uppskattas för varje systemutvecklingsprojekt. Fowler (2002) hävdar att utvecklare vid ett systemutvecklingsprojekt förväntas arbeta snabbt samtidigt som det förväntas att systemet ska ha högre kvalitet än tidigare. Med andra ord är systemutveckling idag ofta förknippat med krav på att leverera system med extremt korta leveranstider, med hög kvalitet och med mindre resurser. Vidare påpekas att de traditionella systemutvecklingsmetoderna anses otillräckliga för de nya krav som ställs under systemutvecklingsprocessen (Fowler, 2002). Andersen (1994) påpekar att orsaken till att det är stor fokus på att förbättra metodanvändningen vid systemutveckling är att systemutvecklingsprojekt är mera komplexa och tidskrävande nu. En ytterligare orsak är att systemutveckling bedrivs idag i verksamheter som har komplexa miljöer (Andersen, 1994).

Systemutvecklingsmetoder är inte lika i innebörd, i täckning av olika delar och i täckning av olika aspekter, vilket skapar komplexitet vid val av metod (Avison och Fitzgerald, 1998). Avison och Fitzgerald (1998) anser att de traditionella systemutvecklingsmetoderna ofta är generellt beskrivna för att passa alla typer av organisationer och projekt. En ny grupp av dynamiska systemutvecklingsmetoder, *eng. agile methods*, har därför utvecklats (Fowler, 2002). Wallenquist m.fl. (2003) påpekar att begreppet agile översätts med flexibel, dynamisk och smidig. Eftersom det är svårt att hitta en korrekt översättning till begreppet "agile" valdes i detta examensarbete begreppet "dynamiska" som översättning för agile. De dynamiska metoderna fokuserar på att förbättra kravhanteringen och att spara tid samt andra resurser för att möta dagens behov under en systemutvecklingsprocess. Fowler (2002) anser att de dynamiska metoderna är lämpliga för små projekt, där antal deltagare är färre än tio.

Extreme Programming (XP) är en dynamisk metod. Därmed anses XP som en utvecklingsmetod för små projekt, där få personer deltar i projektet (Lindahl, 2003). Lindahl (2003) påpekar att XP snabbt levererar nya system till kunder och att det levereras kod med hög kvalitet. Han påpekar vidare att XP fokuserar på individen och har arbetsmetoder som främjar kommunikation och effektivt grupparbete.

Målet för Rational Unified Process (RUP<sup>®</sup>)<sup>1</sup>, som är en systemutvecklingsprocess, är att säkerställa bra kvalitet på systemet, så att projektet håller sig inom satta tids- och kostnadsramar (Kruchten, 2000a). RUP<sup>®</sup> är en kommersiell metod som är utvecklad och marknadsförd av Rational Inc., samma organisation som står bakom Unified Modelling Language (UML) (Fitzgerald m.fl., 2002). Enligt Hirsch (2003) anses RUP<sup>®</sup> i sin ursprungliga form vara mest lämplig för stora projekt. Han menar att RUP<sup>®</sup> innehåller en stor mängd information som kan vara svår att bearbeta, vilket ibland kan leda till att den försvårar snarare än underlättar arbetet för ett litet projekt.

---

<sup>1</sup> Rational, Rational Unified Process and RUP are trademarks or registered trademarks of Rational Software Corporation in the United States and/or other countries

## 1. Inledning

Vid ett litet projekt kan därför de delar som anses onödiga plockas ur RUP<sup>®</sup>. På det viset används, istället för RUP<sup>®</sup>, en anpassad RUP vid små projekt (Abrahamsson m.fl., 2002). Abrahamsson m.fl. (2002) anser att denna typ av anpassad RUP bör kännetecknas med snabbhet och smidighet. Därmed anses den, enligt Fowler (2002) vara en dynamisk metod som kan passa för små projekt. Enligt Fowler (2002) kan anpassningar och bearbetningar av en systemutvecklingsmetod som skall användas i ett litet projekt vara nödvändiga för att lösa ett specifikt problem.

Denna studie behandlar Extreme Programming (XP) och en specifikt dokumenterad anpassad RUP, som anses vara dynamiska metoder lämpliga för små projekt. Karaktäristiska dragen för de två metoder, som gör att de är lämpliga för små projekt, undersöks. Vidare analyseras redovisade erfarenheter, som finns från metod-tillämpningar respektive metod angående deras lämplighet för små projekt.

Fortsättningsvis presenteras en bakgrund till problemet. I kapitel tre redovisas problemet och de frågeställningarna som skall besvaras under detta examensarbete. Därefter redovisas metoden som skall användas följt av genomförandet, resultat och analys. Slutligen presenteras sammanfattade resultatet och slutsatser för detta examensarbete följt av diskussion.

## 2 Bakgrund

### 2.1 Systemutveckling

I detta avsnitt presenteras och förklaras begreppet systemutveckling. Vidare förklaras innebörden av en traditionell systemutveckling och av hjälpmedlen som används under ett systemutvecklingsprojekt. Slutligen diskuteras komplexitet i systemutveckling.

Enligt Avison och Fitzgerald (2003) stödjer och förbättrar ett informationssystem kommunikationen och beslutsfattandet. De förklarar vidare att det förbättrar intern effektivitet och ökar ett företags konkurrenskraft.

”Ett informationssystem är ett system för insamling, bearbetning, lagring, överföring och presentation av information” (Andersen, 1994, s. 15)

Andersen (1994) definition fokuserar på informationshantering och behandling av information. Andersen (1994) påstår att ett informationssystem ska ordna och hantera data på ett systematiskt sätt. Han förklarar djupare sin definition genom att behandla begreppen information och system separat. Enligt Andersen (1994) står information för kunskaper och upplysningar om faktiska eller tänkta förhållanden. Ett system är ett mönster, en ordning eller ett sammanhang (Andersen, 1994).

Arbetet med att skapa och utveckla ett informationssystem benämns systemutveckling. Systemutveckling är de aktiviteter som utförs i anknytning till en verksamhet med avsikt att förändra arbetsprocesserna och arbetsorganisationen inom verksamheten (Avison och Fitzgerald, 1998). Denna definition fokuserar på arbetsprocesserna i organisationen. Goldkuhl (1993) fokuserar på den mänskliga aspekten och dess roll för systemutveckling. Han anser att systemutvecklingen är människors arbete med att analysera, utforma och förändra verksamheter, där datorsystem ingår eller planeras ingå som integrerade delar.

Russo m.fl. (1996) påstår att systemutveckling är aktuellt när existerande system konsumerar en stor del av informationssystemets resurser. Avison och Fitzgerald (1998) hävdar att i utvecklingsprocessen av ett informationssystem ingår ett antal aktörer som har olika mål, språk, referensramar och så vidare. Kommunikationen mellan intressenter spelar en stor roll för att få systemutvecklingsprocessen att fungera, vilket leder till att experter i ämnet forskar med fokus på att få bättre kommunikation och mindre avstånd mellan aktörer (Avison och Fitzgerald, 1998).

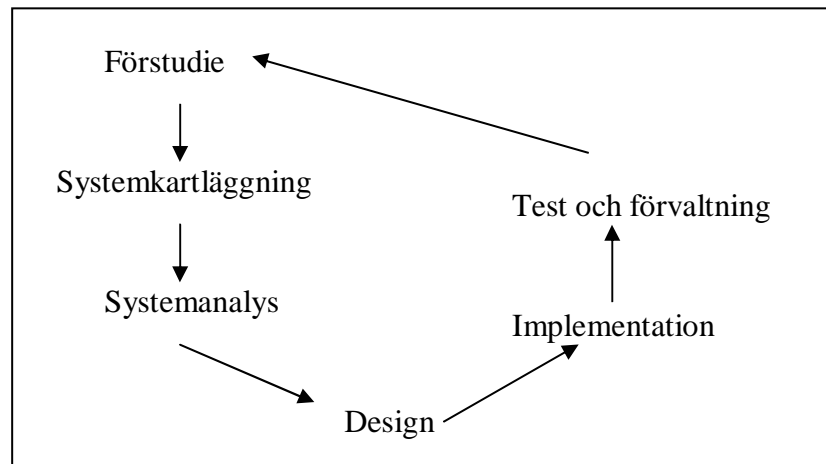
Andersen (1994) förklarar att det behövs en övergripande syn på hur informationssystemen utvecklas. Han berättar vidare att metoder och tekniker redogör för hur utvecklingsarbetet skall utföra. Innan utvecklingsarbetet påbörjas bör problem och möjligheter som verksamheten står inför diskuteras för att bestämma vilka typer av utvecklingsåtgärder som ska vidtas (Andersen, 1994).



## 2. Bakgrund

### 2.1.1 Traditionell systemutveckling

Nedan presenteras stegen för traditionell systemutveckling. En traditionell vattenfallsmodell för utveckling av ett system innehåller följande steg som presenteras i figur 1 nedan:



Figur 1. En traditionell livscykel för systemutveckling.

Enligt Avison och Fitzgerald (2003) fokuserar förstudiefasen på det aktuella systemet, dess problem och eventuella lösningar. Förstudiefasen ger en övergripande utredning av verksamheten och syftet till att få en uppfattning om önskade mål och önskade lösningar. Systemkartlägningsfasen presenterar kraven, begränsningar och regler. I systemkartlägningsfasen arbetas en mer detaljerad syn på hur verksamheten fungerar fram, såsom processer, arbetsrutiner och problem (Avison och Fitzgerald, 2003). Avison och Fitzgerald (2003) förklarar vidare att systemanalysfasen ger en mer detaljerad studie av vad informationssystemet ska göra genom att besvara en del frågor, bland annat:

- Varför har existerande problem uppstått?
- Varför valdes aktuella arbetsmetoder, som är avgörande för vad systemet skall ha för funktioner, att lösa existerande problem?
- Finns det alternativa arbetsmetoder?

I systemanalysfasen skall avgöras vad som kan göras bättre och vad som skall ingå i informationssystemet (Avison och Fitzgerald, 2003). Systemdesignsfasen omfattar utveckling och utformning av både manuella och datorbaserade rutiner i ett informationssystem. Bearbetning av rutiner underlättar genomförandet av det som är beslutat. Under denna fas presenteras gränssnittet för informationssystemet och vilka funktioner systemet ska ha. Vilka hård- och mjukvaror som ska användas i systemet och input och output diskuteras i systemdesignsfasen, som syftar till att realisera önskningarna som presenteras i förgående faser (Avison och Fitzgerald, 2003). Vid implementationsfasen skapas systemet, enligt Avison och Fitzgerald (2003). De påstår vidare att denna fas omfattar programmering, inköp av hård och mjukvara samt utbildning och mycket mer, vilket leder till att systemet förverkligas. I test och under-

## 2. Bakgrund

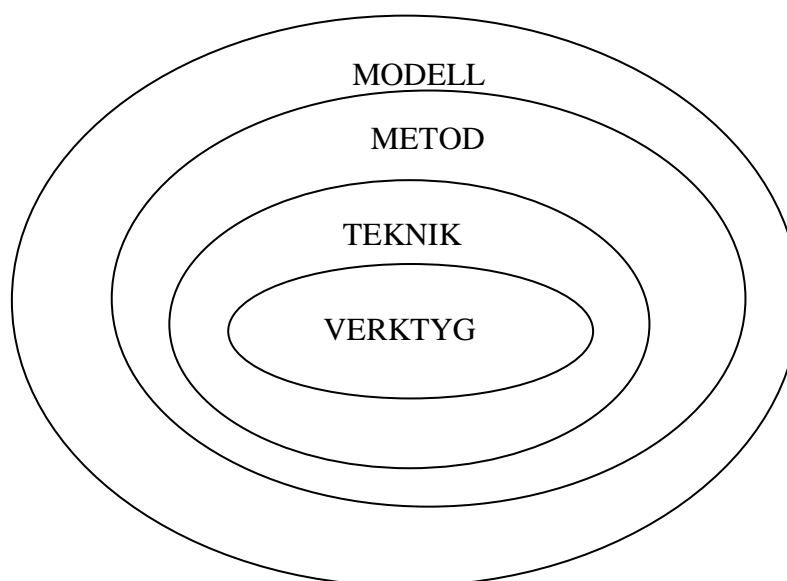
hållsfasen testas systemets funktioner för att säkerställa att de verkligen motsvarar de önskade funktionaliteten som presenterades i tidigare faser (Avison och Fitzgerald, 2003).

Olle m.fl. (1991) bryter ner utvecklingsprocessen för ett informationssystem till endast fyra faser: informationssystemplanering, verksamhetsanalys, systemdesign och konstruktion. Innebörden av de fyra faserna liknar dock de faserna som Avison och Fitzgerald (2003) presenterar.

Systemutvecklingsprocessen ska vara en iterativ process (Goldkuhl, 1993), vilket innebär att en fas inte avslutas helt utan att det finns möjlighet att återkomma till fasen senare i systemutvecklingsarbetet. Detta kan jämföras med vattenfallsmodellen, där iteration mellan faserna saknas. Kruchten (2000b) hävdar att problemet med vattenfallsmodellen är att riskerna skjuts på framtiden. Resultatet blir att projektet inte håller budgeten eller tidsplanen eftersom det kostar mycket både i tid och i pengar att rätta till fel från tidigare faser sent i utvecklingsprocessen. Kruchten (2000b) presenterar iterativa och stegvis växande systemutvecklingsmetoder som ett alternativ till vattenfallsmodellen. Iterativa metoder innebär att utvecklaren identifierar risker och hanterar dessa tidigt i projektet. Detta sker samtidigt som utvecklaren fortsätter med att undersöka och utveckla nästa fas i systemutvecklingsprocessen (Kruchten, 2000b). Enligt Kruchten (2000b) kan problem från tidigare faser i systemutvecklingsprocessen tidigt upptäckas och lösas genom att utvecklaren går tillbaka direkt till den fas där problemet uppstår och rättar till felet. Genom att arbeta iterativt nås målen att hålla budgeten och tidsplanen lättare, vilket medför hög kvalitet på produkten (Kruchten, 2000b).

### 2.1.2 Hjälpmedel i systemutvecklingsprocessen

Enligt Andersen (1994) systemutvecklingsprocessens hjälpmedel är modell, metod, teknik och verktyg. Hjälpmedlen i utvecklingsarbetet presenteras i figur 2 nedan:



*Figur 2. Hjälpmedel i systemutvecklingsprocessen.*

## 2. Bakgrund

Hjälpmedlet som kommer att studeras mest i detta examensarbete är metod. Enligt Andersen (1994) och Avison och Fitzgerald (2003) är en systemutvecklingsmetod en översikt över utvecklingsarbetet och ett ramverk som beskriver vilket arbete som måste utföras och vem som bör utföra det. Brinkkemper (1996) skriver följande om en metod:

”A method is an approach to perform a systems development project, based on a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with corresponding development products” (Brinkkemper, 1996, s. 275).

Denna definition, som presenteras ovan, betonar att en systemutvecklingsmetod är en mer detaljerad beskrivning än en modell av sättet att lösa ett visst problem. Avison och Fitzgerald (2003) skriver att en systemutvecklingsmetod ska se till att kraven som ställs på systemet dokumenteras. Vidare skriver de att en systemutvecklingsmetod ska ge stöd, riktlinjer, olika faser, steg och tekniker för att fånga, utveckla och beskriva kraven för ett informationssystem. En bra systemutvecklingsmetod ska ge ett systematiskt tillvägagångssätt menar Avison och Fitzgerald (2003), vilket betyder att systemutvecklingsmetoden ska tala om för utvecklaren vart den befinner sig i processen och vad som skall göras.

En teknik är ett detaljerat arbetssätt och ett slags ”recept” som består av en uppsättning regler som säger hur en del av verkligheten kan uttryckas i en beskrivning (Andersen, 1994). Brinkkemper (1996) beskriver en teknik som en procedur i följande text:

”A technique is a procedure, possibly with a prescribed notation, to perform a development activity.” (Brinkkemper, 1996, s. 276).

Andersen (1994) hävdar att verktyg är ett fysiskt och ”tekniskt” hjälpmedel, exempel på enkla verktyg är papper och penna. Exempel på datorbaserade verktyg är CASE som står för Computer Aided Software Engineering. CASE är en programvara som ger stöd vid användning av en beskrivningsteknik. Brinkkemper (1996) påpekar att ett verktyg ger stöd till systemutvecklingsprocessen. Han skriver följande:

” A tool is a possibly automated means to support a part of a development process.” (Brinkkemper, 1996, s. 276).

### 2.1.3 Komplexitet i systemutveckling

I vår komplicerade värld, med hård konkurrens mellan olika verksamheter, är det viktigt att informationssystemet är utformat på rätt sätt för att bemöta komplexiteten på marknaden och i organisationsmiljön (Andersen, 1994). Fowler (2002) påpekar att vid ett systemutvecklingsprojekt är problemen för det mesta förknippade med komplexiteten i moderna verksamheter, där det alltid finns behov av bättre tekniker. Vidare förklarar han att problemen är också förknippad med komplexiteten i system-

## 2. Bakgrund

utvecklingsmetoderna, där det finns behov av en systemutvecklingsmetod som hanterar snabbt förändringar under ett systemutvecklingsprojekt.

Martin (2002a) påpekar att omfattande systemutvecklingsmetoder tar tid, omfattar onödiga möten och fokuserar bara på arkitekturen för mjukvaran. Utvecklarna behöver en sorts metodstruktur för att guida dem i olika projekt (Russo m.fl., 1996). De menar att systemutvecklingsmetoderna saknar beskrivning av funktioner och möjligheter som de erbjuder, vilket krävs vid tillämpning av dessa systemutvecklingsmetoder. Det sistnämnda är orsaken till att det är stort fokus på tydligare beskrivning av de systemutvecklingsmetoderna och tillämpningar av dem (Andersen, 1994).

Det går inte att minska komplexiteten i ett utvecklingsprojekt. Det går bara att söka nya sätt att hantera komplexiteten och att underlätta arbetsprocessen (Booch, 2001, s. 119). Booch (2001) påstår vidare att komplexiteten minskar ju mindre projektet är och ju färre antal deltagare som deltar i det. Det tar oftast mycket tid att få ett stort antal deltagare i ett systemutvecklingsprojekt att diskutera och ta viktiga beslut, vilket medför ineffektivitet i projektet (Artim m.fl., 1998).

Enligt Kruchten (2000a) kan ändrade kraven och ändrade behoven på systemet bero på misstolkning av användarnas krav. Han förklarar att misstolkningar behandlas sent i ett projekt och att kraven på ett system inte kan fastställas helt innan utvecklingsprocessen börjar. Detta beror på att kraven är föränderliga under hela utvecklingsprocessen (Kruchten, 2000a). Han påpekar alltså att förändringar på krav som ställs från användarna på systemet är ett problem under ett systemutvecklingsprojekt. Även Fowler (2002) påstår att utvecklarna anser att ett problem i ett systemutvecklingsprojekt, är att kraven på systemet som ställs av användarna alltid ändras. Beck och Fowler (2001) anser att användarna aldrig blir nöjda med resultatet av ett systemutvecklingsprojekt för att när systemutvecklingsprojektet avslutas upptäcks att användarna inte längre är intresserad av det som är planerat och infört under systemutvecklingsprojektet. Då behöver de något annat nytt.

Fowler (2002) påpekar att det inte bara är användarnas krav som förändras och leder till komplexitet i systemutvecklingsprojekt. Marknadens förändringar leder också till svårigheter att förutsäga framtiden och att planera tidsåtgången för hela projektet. Avison och Fitzgerald (2003) samt Fowler (2002) anser att marknaden idag kännetecknas av ökad konkurrens, globala utmaningar och omväxlingar i marknaden samtidigt som kontinuerlig, snabb och teknisk utveckling sker. De förklarar vidare att alla förutnämnda kännetecken leder till ökad komplexitet och dynamik i miljön, där ett system utvecklas.

Dåligt samarbete mellan olika parter i ett systemutvecklingsprojekt är orsaken till misslyckandet för olika projekt (Beck och Fowler, 2001). Beck och Fowler (2001) menar att användarna och utvecklarna försöker försvara sig mot varandra genom att bygga väggar mellan varandra. De delar inte kritisk information med varandra. Den kritiska informationen är nödvändig för att kunna leverera det nya systemet till användaren snabbt och med hög kvalitet. De förklarar vidare att utvecklarna och användarna måste erkänna att de behöver varandra och måste acceptera varandras rättigheter och ansvarsförmåga. Användarna måste också samarbeta vid ett systemutvecklingsprojekt (Beck och Fowler, 2001).

## 2. Bakgrund

Avison och Fitzgerald (2003) hävdar att orsaken till misslyckande vid ett systemutvecklingsprojekt är mänskliga eller organisatoriska problem och dålig planering som leder till stora kostnader. De påpekar vidare att genom att inte ta hänsyn till användarnas kunskap kan det leda till svårigheter för slutanvändarna att arbeta med systemet. Kruchten (2001) påpekar att en systemutvecklingsmetod inte ersätter användarnas roll och ersätter inte heller utvecklarnas erfarenhet. Han anser vidare att en väldefinierad systemutvecklingsmetod ger bättre kommunikation. Vidare påpekar Fowler (2002) och Pollice (2001a) att systemen växer i omfattning med tiden och det blir svårare att hantera och ha kontroll över arbetsprocessen i systemutvecklingsprojektet. Därför finns det behov av samarbete mellan systemets användare och projektdeltagare. Pollice (2001a) påpekar att en systemutvecklingsmetod inte behöver vara omfattande när organisationsmiljön är komplex, utan en dynamisk och smidig systemutvecklingsmetod med fokus på vad som anses viktigt och tillräckligt för utvecklingsprojektet kan räcka.

### 2.2 Systemutvecklingsmetoder

Nedan presenteras fakta om de olika traditionella systemutvecklingsmetoderna och dess roll under ett systemutvecklingsprojekt. Erfarenheter kring systemutvecklingsmetoderna och metodtillämpningens förväntade effekter kommer också att diskuteras i detta avsnitt. Slutligen presenteras alternativ till de traditionella systemutvecklingsmetoderna.

Avison och Fitzgerald (1998) påpekar att metoder innehåller filosofiska synsätt och de ska leda människor till att tänka rätt och inte att utesluta att nya idéer kommer fram. Problemen som hör till metoderna är att de inte täcker allt, att de ger lite stöd, att de styr för mycket, att de är för luddiga och att de är för detaljerade (Avison och Fitzgerald, 1998). Fitzgerald (1996) anser att metoderna används som ett hjälpmedel vid en stressfull och komplex miljö. Det betyder att systemutvecklingsmetoderna ska underlätta arbetet och leda deltagarna i systemutvecklingsprojektet. Jayaratna (1996) påpekar att förutom att en metod ska tala om vilka steg som ska gås igenom, ska den också tala om för användaren varför denne måste följa de steg som metoden omfattar. Detta tar han upp som en del av hans definition av en metod, som presenteras nedan:

”A methodology should tell us what steps to take, in what order and how to perform those steps but most importantly the reasons why the methodology user must follow those steps and in the suggested order.”  
(Jayaratna, 1996, s. 24)

Fitzgerald (1996) anser att systemutvecklingsmetoderna i framtiden kommer att ha som uppgift att genom god kommunikation informera användarna i organisationen om systemutvecklingsprojektets aktiviteter och göra det nya systemet tilltalande för användaren.

#### 2.2.1 Närliggande begrepp

Brinkkemper (1996) skiljer mellan begreppen metod och metodologi för systemutveckling. En metod beskriver sättet och regler att utveckla ett system (se

## 2. Bakgrund

även ovan 2.1.2). Brinkkemper (1996) definierar metodvetenskap som tekniken att designa, strukturera och anpassa metoder, tekniker och verktyg för utvecklingen av ett informationssystem. Metodologin däremot är teorin om en metodisk systemutveckling. Definitionen presenteras nedan:

” The methodology of information systems development is the systematic description, explanation and evaluation of all aspects of methodical information systems development.” (Brinkkemper, 1996, s. 276)

Karlsson (2002) förklarar att i viss litteratur behandlas begreppet metodologi som synonym till begreppet metod. Han anser att begreppet metodologi borde förklaras som teorin eller studier om metoder. I detta examensarbete diskuteras bara metoder för systemutveckling och inte metodologin för systemutveckling, om inget annat anges.

Begreppen process och metod används synonymt i detta examensarbete. De begrepp som kommer att användas i detta examensarbete är begreppet metod. Skillnaden mellan process och metod enligt Lind (2001) är att en process består av input som transformeras till output för en kund. Detta sker genom ett antal aktiviteter. En metod för systemutveckling förverkligar idéer i en modell genom en detaljerad beskrivning av de praktiska stegen som bör utföras i ett systemutvecklingsprojekt (Avison och Fitzgerald, 1998).

En process är en typ av metod där det finns stark fokusering på horisontella icke funktionsinriktade flöden (Lind, 2001). Lind (2001) menar att processtänkandet innebär en helhetssyn på hela verksamheten, där processerna spänner över olika funktioner i verksamheten. Goldkuhl (2000) hävdar också att processtänkandet innebär att arbetsprocessen och flödet genom verksamheten framhävs. Han förklarar vidare att processen är till för att lägga fokus på systemets användare och tona ned den funktionella synen. På det viset kan det finnas möjlighet att under projektets gång studera egenskaperna hos användarna för systemet, vilket kan leda till ökad motivation hos användarna. Goldkuhl (2000) sammanfattar egenskaper för en process genom följande punkter:

1. Serie av aktiviteter som medför ett lyckat projekt.
2. Repetitiva delprocesser som ger konsistens i projektet.
3. Användarefokus för att det är användaren som ska använda systemet.
4. Flödesorientering som ger framgång i projektet.
5. Förädling, stegvist värdeskapande, för att nå framgång.

De punkterna som presenteras ovan säkerställer att få ett lyckat systemutvecklingsprojekt, där förändringar på bland annat kraven som ställs på systemet hanteras.

### 2.2.2 Olika typer av systemutvecklingsmetoder

Avison och Fitzgerald (1998) hävdar att skillnaderna mellan de olika metoderna är att de täcker olika delar av systemutvecklingsprojektet och att de fokuserar på olika saker. Vidare hävdar de att systemutvecklingsmetoderna täcker olika aspekter av problemområdet, ger stöd för olika typer av aktörer och att de har olika ursprung.

## 2. Bakgrund

Avison och Fitzgerald (2003) beskriver att det finns olika typer av systemutvecklingsmetoder:

1. Processororienterade metoder, till exempel Jackson systems development (JSD).
2. Blandade metoder, som är formades från delar av de bästa av andra metoder, tekniker och verktyg. Exempel på blandade metoder är Information Engineering (IE).
3. Objektorienterade metoder, till exempel Rational Unified Process (RUP®).
4. Dynamiska metoder, till exempel Extreme Programming (XP).
5. Användarorienterade metoder, till exempel Effective technical and human implementation of computer-based systems (ETHICS).
6. Organisationsorienterade metoder, till exempel Soft systems methodology (SSM).
7. Strukturer, kallas inte egentligen för metoder eftersom de innehåller riktlinjer att välja metod, teknik och verktyg i ett systemutvecklingsprojekt. Exempel på strukturer är Multiview.

Avison och Fitzgerald (1998) hävdar att det är viktigt att välja rätt systemutvecklingsmetod för rätt organisation eller projekt, annars ökar komplexiteten i systemutvecklingsprojektet.

Russo m.fl. (1996) hävdar att många organisationer försöker tillämpa flera metoder på systemutvecklingsprojekten. Det beror på att organisationsmiljön är annorlunda nu än förr. Det påpekas att snabba förändringar i kraven från användarna, som ställs på systemet, leder till svårigheter att välja rätt metod (Fowler, 2002). Jayaratna (1996) skriver följande om val av en systemutvecklingsmetod, där han förklarar hur överensstämmelsen mellan deltagarna och vilken syn de har på olika typer av problemlösningsmetoder påverkar metodvalet:

”If there is only one methodology available or if methodology users are familiar with only one methodology, then there is no perceived problem of choice as there would be no alternatives to choose from.” (Jayaratna, 1996, s. 24).

### 2.2.3 Uppfattningar kring metodtillämpningar

Fitzgerald (1998) anser att fördelen med att använda metoder är att de ger möjlighet till att underlätta arbetet i ett komplex projekt genom att samordna och fördela aktiviteterna som projektet omfattar. En annan fördel är att de ger bättre möjlighet att

## 2. Bakgrund

kontrollera projektet genom att tydliggöra uppgifterna och jämföra resultaten av varje fas med gällande planer. En ytterligare fördel är att de ger möjlighet till standardisering av utförda arbeten under ett systemutvecklingsprojekt, vilket minskar personberoendets vikt i utvecklingsprojektet (Fitzgerald, 1998).

Fitzgerald (1998) anser att nackdelen med att använda en systemutvecklingsmetod är att utvecklaren bara fokuserar sig på att använda och följa en metod, vilket leder till att utvecklaren kan "slarva" med viktiga delar av ett systemutvecklingsprojekt. En annan nackdel är att metoderna också anses passa för alla typer av utvecklingsprojekt, vilket inte är korrekt eftersom ingen utvecklingssituation är den andra lik (Fitzgerald, 1998). En ytterligare nackdel är att metoderna inte tar hänsyn till kritiska faktorer såsom kreativitet, lärande och intuition. Det sistnämnda minskar värdering av människors bidrag som anses viktigt eftersom det är människor som utvecklar och använder systemen (Fitzgerald, 1998).

Avison och Fitzgerald (2003) hävdar att förutsättningar för en lyckad metodtillämpning är att metoden balanserar tekniska aspekter och människo-orienterade aspekter i verksamheten. En annan förutsättning för en lyckad metodtillämpning är att säkerställa att kraven är förutsägbara och stabila (Fowler, 2002). Fowler (2002) påstår att metoderna ska ge möjligheter till att systemutvecklingsarbetet blir mer förutsägbart och effektivt. Han menar att utan stabila krav kan inte en stabil plan framställas. Det är också viktigt att ha en uppfattning om hur metoden fungerar i praktiken för att få en lyckad metodtillämpning (Wynekop och Russo, 1995). Genom att veta hur systemutvecklingsmetoden fungerar i praktiken blir arbetet i systemutvecklingsprojektet mer förutsägbart. Andersen (1994) anser att om metoden är exakt beskriven kan en lyckad metodtillämpning erhållas. Systemutvecklingsmetoden är exakt beskriven när två personer kommer fram till samma resultat om båda personerna använder samma metod på samma problem (Andersen, 1994).

För stora projekt anses Rational Unified Process (RUP<sup>®</sup>), som är en traditionell tidskrävande och omfattande systemutvecklingsmetod, vara lämplig och användbar (Pollice, 2001). Strand (2002) anser att risken är stor vid införandet av RUP<sup>®</sup> för små projekt att göra mycket arbete under begränsad tid för att RUP<sup>®</sup> omfattar mycket arbete. Fowler (2002) anser också att om det är femtio personer eller fler som deltar i systemutvecklingsprojektet och om det existerar samarbete mellan olika parter, då skall en traditionell förutsägbar metod, som till exempel RUP<sup>®</sup>, införas. Ett sådant projekt anses vara ett stort projekt. Däremot i små projekt anses en snabbare och smidigare systemutvecklingsmetod vara lämplig särskild när kraven på systemet förändras ofta och antalet deltagare är få (Fowler, 2002; Pollice, 2001).

### 2.2.4 Behovet av alternativ till traditionella systemutvecklingsmetoder

Avison och Fitzgerald (1998) hävdar att utveckling av nya metoder eller bearbetning av existerande metoder uppstod ur ett behov av att kunna upprepa lyckade projekt, minska personalberoendet, kunna planera och tidsberäkna arbetet, kunna dela upp arbetet i hanterbara delar, kunna mäta framåtskridandet och underlätta framtida systemförvaltning.

Brinkemper (1996) påstår att alla systemutvecklingsprojekt är olika, vilket leder till att anpassning av de omfattande traditionella systemutvecklingsmetoderna är ett måste för att systemutvecklarna ska kunna få bättre stöd och kunna utföra system-



## 2. Bakgrund

utvecklingsprojektet snabbare. Fitzgerald (1996) hävdar att standardiserade checklistor med traditionella steg inte kommer att vara aktuella för utvecklarna i framtiden. Russo m.fl. (1996) anser att systemutvecklare är på väg mot användning av en metod där utvecklarna plockar och väljer ut olika delar ur metoden för att uppfylla de olika behoven som dyker upp under ett projekt. Fowler (2002) anser att anpassning och bearbetning av en metod till en snabbare metod är en lösning när kraven på systemet under ett utvecklingsprojekt förändras och är osäkra. Han menar att förändrade krav kan leda till att stabil design inte kan skapas och att en planerad metod inte kan följas. Enligt Wynekop och Russo (1995) är metदानpassning ett billigare sätt, än att utveckla en ny metod. Vidare påstår de att en anpassad metod kan möta förändringsbehoven. Även Brinkemper (1996) hävdar att komplexiteten och förändringar, som sker under ett utvecklingsprojekt, medger behov av en så kallad situationsmetod. En situationsmetod omfattar anpassning av en metod efter projektets behov och som går att förändra genom att byta ut olika delar (Brinkemper, 1996). Vidare påpekas att kontroll över förändringar i organisationsmiljön saknas i de traditionella metoderna och att det inte finns vetenskap om hur bra de anpassade metoderna fungerar. Detta är orsaken till att forskning i ämnet, om metoder generellt, krävs (Wynekop och Russo, 1995).

Fitzgerald (1996) anser att det finns behov av snabbare överlämningar av nya system än vad de traditionella systemutvecklingsmetoderna erbjuder, då de kräver många deltagare. Ökad komplexitet och ökade förändringar i verksamhetens aktiviteter leder till att det uppstår behov att utveckla system kontinuerligt (Fitzgerald, 1996). Även Russo m.fl. (1996) diskuterar hur de traditionella systemutvecklingsmetoderna misslyckas att möta de nya behoven i utvecklingsmiljön. Han menar att metoderna i dess existerande form tycks vara otillräckliga för utvecklarnas behov och att nya systemutvecklingsmetoder borde bearbetas för att uppfylla projektsituationens olika behov. Fitzgerald m.fl. (2002) anser att villkoren för systemutveckling har förändrats på grund av förändringar i organisationens affärsmiljö. Nu är det alltmer vanligt att det i organisationer, där systemutveckling sker, bara finns ett fåtal anställda och systemutvecklare som berör systemutvecklingsområdet, vilket är orsaken till att det finns behov av nya systemutvecklingsmetoder som kan användas med få deltagare. Det finns också behov av att systemutvecklingen sker snabbt. Kruchten (2001) anser att en liten och smedig metod stödjer organisationens förmåga att bearbeta förändringar i dess miljö.

### 2.3 Systemutvecklingsmetoder för små projekt

I detta avsnitt presenteras de dynamiska metoderna, som passar för små projekt. Grundläggande idéer för de dynamiska metoderna och deras egenskaper presenteras också. Vidare presenteras dynamiska metoders roll och syfte. Slutligen presenteras de två exemplen på dynamiska metoder för små projekt. De två metoderna kommer att studeras i detta examensarbete.

Hirsch (2003) påpekar att ett litet projekt bör omfatta mellan tre till åtta utvecklare och löpa mellan sex och tolv månader. Pollice (2001b) och Fowler (2002) anser att små projekt omfattar tolv eller färre antal deltagare, där projektet dröjer mellan några veckor till högst få månader. För denna studie anses ett litet projekt omfatta mellan tre till tio deltagare, där projektet genomförs på mindre än tolv månader.

## 2. Bakgrund

Fowler (2002) presenterar dynamiska metoder, *eng. Agile methods*, där Crystal och Extreme Programming (XP) utgör exempel. Fowler (2002) hävdar att de dynamiska metoderna är lösningen till dagens komplexa problem inom systemutveckling eftersom de erbjuder möjligheter att utveckla system snabbt med mindre resurser. Kruchten (2001) definierar begreppet ”dynamiska” som metodens förmåga att anpassas och tillämpas i en organisation, som omfattar förändringar i dess miljö. Wallenquist m.fl. (2003) påpekar att grunden i de dynamiska metoderna är att prioritera och värdesätta människan och de produktiva förändringarna i projektet framför tekniken och regelverket. Syftet med dynamiska metoder är att alla som arbetar i ett utvecklings-projekt skall få stöd för ett gemensamt synsätt, för att som team mer effektivt och konstruktivt kunna hantera förändringar under projektets gång (Wallenquist m.fl., 2003).

Booch (2001) påstår att genom tillämpningen av en dynamisk metod för små projekt, som omfattar färre än tio deltagare, med fokus på användaren möts kravet på snabbheten. Kravet på snabbhet uppfylls genom att arbetet i projektet effektiviseras. Abrahamsson m.fl. (2002) påpekar att dynamiska metoder är anpassade för projekt som omfattar få deltagare. De små projekten levererar små produkter (Abrahamsson m.fl., 2002).

En förutsättning för dynamiska metoder för små projektgrupper är att projektgruppen ska fokusera på kommunikation, användarna, vinster och feedback (Cockburn och Highsmith, 2002). Cockburn och Highsmith (2002, s. 148) diskuterar vad som gäller generellt för systemutvecklingsmetoder. Cockburn och Highsmith (2002) skriver följande om dynamiska systemutvecklingsmetoder för små projekt:

- Intresset för metoder ökar när de minskar kostnader för insamling och transferering av information genom goda kommunikationskanaler (Cockburn och Highsmith, 2002). Kostnader kan vara i form av tid och pengar. Dynamiska metoder ser till att deltagarna är nära varandra, vilket leder till bättre kommunikation. God kommunikation leder till att systemet blir lättare och billigare att utveckla. En av de tolv principerna, som ligger bakom manifestet för dynamiska metoder, betonar vikten av öga mot öga kommunikation mellan deltagarna i projektet (Highsmith, 2001). Öga mot öga kommunikation är en billig och snabb kanal för utbyte av information (Cockburn och Highsmith, 2002). Cockburn och Highsmith (2002) förklarar att när projektgruppen omfattar få antal deltagare finns det möjlighet att ha öga mot öga kommunikation. Därmed anses de dynamiska metoderna, som kräver öga mot öga kommunikation, vara lämpliga för små projekt, som omfattar högst tio deltagare. Det blir lättare att ställa frågor när en dynamisk metod tillämpas för ett litet projekt (Cockburn och Highsmith, 2002, s. 178). De vill säga projektdeltagarna stödjer varandra och kommuniceras bättre vid öga mot öga kommunikation.
- Fowler (2002) och Cockburn och Highsmith (2002) påpekar att projektdeltagarna spenderar mycket tid på dokumentationen under ett projekt, där tillämpas en traditionell metod. Dokumentation leder till att produktiviteten minskar för att om

## 2. Bakgrund

något förändras, omskrivas alla dokumenten. Antalet aktiviteter som skall genomföras ökar och tiden som avses för projektet är det samma, vilket förhindrar projektgruppen att slutföra projektet under den begränsade tiden. Därför anses dynamiska metoder, som kräver mindre dokumentation och få aktiviteter, vara en lösning för små projekt som genomförs under en kort tidsperiod (Fowler, 2002). Coldewey m.fl. (2000) påpekar att god kommunikation mellan projektdeltagarna ersätter dokumentation. Martin (2002c) förklarar att dokumenten, som anses viktiga för träning och transformering av kunskap till eventuella nya projektdeltagare, ersätts med att projektdeltagarna hjälper till med att träna och förklarar allt som omfattas av projektet för de eventuella nya projektdeltagarna.

Sammanfattningsvis kännetecknas de dynamiska metoderna med mindre dokumentation som sparar tid och med öga mot öga kommunikation som sparar tid och arbete. Därför anses de dynamiska metoderna vara lämpliga för små projekt.

### 2.3.1 Manifestet och de tolv principerna för dynamiska metoder

Highsmith (2001) presenterar historien bakom ”The Agile Alliance” som är en självständig grupp för dynamiska systemutvecklingsmetoder. Gruppdeltagarna konkurrerar ibland mot varandra eftersom deltagarna i gruppen förespråkar olika dynamiska metoder. Denna grupp framställde ett manifest för de dynamiska metoderna som förbättrar och utvecklar idéerna hos projektgruppen. Manifestet omfattar fokus på följande (Wallenquist m.fl., 2003, s. 2):

- Individer och samarbete framför processer och verktyg.
- Fungerande programvara framför omfattande dokumentation.
- Kundsamarbete framför kontraktsförhandlingar.
- Lyhördhet för förändring framför att följa en plan.

Abrahamsson m.fl. (2002) hävdar att en viktig aspekt som manifestet tar upp är vikten av ett nära förhållande och bra kommunikation mellan systemutvecklare och andra parter som tillhör verksamheten där systemet ska utvecklas. Orsaken till att denna aspekt är viktig är att en dynamisk metod kontinuerligt levererar små delsystem, vilket medger feedback. Feedback förbättrar kommunikationen mellan deltagarna i projektet, vilket förkortar tiden som avses för systemutvecklingsprojektet (Abrahamsson m.fl., 2002). God kommunikation minskar risken att spendera mycket tid på att få deltagarna att komma överens om ett beslut. Abrahamsson m.fl. (2002) påstår att den andra punkten i manifestet syftar till att minska dokumentationen genom att testa systemet kontinuerligt och allt som utvecklas under projektets gång. Fitzgerald m.fl. (2002) påpekar att nackdelen med att minska dokumentationen är att det blir svårt för eventuella nya deltagare i projektet att få reda på hur utvecklingsarbetet har gått till. Därmed blir risken stor att de nya deltagarna inte involveras aktivt. Dokumentationen minskas också genom att hålla koden enkel och okomplicerad. Vidare berättar Abrahamsson m.fl. (2002) att den tredje punkten som manifestet behandlar är samarbetet, vilket är en viktig aspekt för ett systemutvecklingsprojekt, då risken minskar att inte fullgöra det som avtalats. Slutligen påpekar de att fjärde punkten i manifestet syftar till att utvecklarna och representanter från verksamheten bör vara välinformerade och kompetenta för att kunna hantera förändrade kraven som dyker

## 2. Bakgrund

upp under utvecklingsprocessens livscykel. Med andra ord bör alla projektets deltagare förbereda sig och planera för att hantera förändringar i kraven som ställs på systemet under projektets gång (Abrahamsson m.fl., 2002).

De tolv principerna som definierar ramverket för de dynamiska metoderna är enligt Wallenquist m.fl. (2003):

1. *Vår högsta prioritet är att tillfredsställa kunden genom tidig och regelbunden leverans av värdefulla programvaror.* Systemet levereras snabbt med användbara funktioner för att kunden ska se tidigt att investeringen är lönsamt.
2. *Vi välkomnar förändringar i kraven även sent i utvecklingen.* Det är viktigt att vara dynamisk och inte se strikt till kravspecifikationen för att kunna hantera förändringarna till kundens konkurrensfördel. Projektet ska alltid snabbt ställas om vid förändringar så att det kan hantera de nya förutsättningarna.
3. *Vi levererar fungerande system ofta, med några veckor eller månaders mellanrum.* Fokus ligger på att snarast färdigställa och driftsätta nya versioner av systemet. På det viset kan värdefull feedback från användarna till projektgruppen samlas in. Den feedback som fås underlättar för projektgruppen att få nya insikter om hur ett optimalt system ska fungera.
4. *Verksamhetskunniga personer och utvecklare arbetar tillsammans dagligen genom hela projektet.* En tät dialog mellan projektdeltagarna skapas bäst genom att styrgruppen och användare ges tillträde till utvecklarnas projektrum.
5. *Vi bygger projektet på motiverade individer.* Projektmedlemmarna ska stödjas av projektledaren för att de ska lita på att de klarar av sin uppgift.
6. *Det bästa sättet att förmedla information till och inom ett team är genom samtal öga mot öga.* Öga mot öga kommunikation klassificeras som den rikaste formen av kommunikation och är ett effektivare kommunikationssätt än dokumentation.
7. *Fungerande programvara är det primära måttet på framsteg.* Dokument och modeller ska vara enkla för beställaren och slutanvändarna för att de ska kunna ge relevant feedback. Med andra ord krävs det en fungerande programvara som resulterar i enkla dokument och modeller. Programvaran ska kunna användas och testas i verkliga situationer.
8. *Smidiga processer verkar för långsiktighet.* Projektdeltagarna ska kunna upprätthålla ett konstant arbetstempo utan att förlora fokus eller produktivitet.

## 2. Bakgrund

9. *Kompromisslösa krav på hög teknisk kvalitet och god design ökar dynamiken.* Projektdeltagarna ska vara redo på att riva upp tidigare beslut om det passar in i projektet.
10. *Enkelhet – konsten att göra exakt rätt saker, varken mer eller mindre – är väsentlig.* Projektdeltagarna plockar bort allt som är onödigt.
11. *Den bästa kravformuleringen, arkitekturen och designen uppstår i en självorganiserande projektgrupp.* Projektgruppen får frihet att under givet budgetansvar rekrytera projektdeltagare baserat på kompetens och verksamhetskunskap.
12. *Med jämna mellanrum reflekterar projektgruppen över hur projektet kan bli effektivare, och anpassar sedan sitt handlande därefter.* Kontinuerliga diskussioner under projektets gång för att öka samhörigheten, motivationen och känslan av att kunna påverka projektet i en positiv riktning.

Wallenquist m.fl. (2003) påpekar att projektdeltagarna kan lägga till egna principer eller modifiera de befintliga som presenteras ovan.

### 2.3.2 Uppfattningar om de dynamiska metoderna

Huvudorsaken till behov av tillämpning av så kallade dynamiska metoder är svårigheten att följa svåra och omfattande metoder som kräver många deltagare (Coldewey m.fl., 2000). Att tillämpa de dynamiska metoderna för små projekt resulterar i system med hög motivation bland deltagarna under ett systemutvecklingsprojekt och i system med hög kvalitet (Martin, 2002a; Coldewey m.fl., 2000). De dynamiska metoderna för små projekt värderar och analyserar arbetet i ett systemutvecklingsprojekt för att leverera bra produkter till användarna (Fowler, 2002). När de dynamiska metoderna tillämpas existeras inte långsiktiga planer för framtiden utan för ofta möten görs, där kortsiktiga planer framställs. Orsaken till kortsiktiga planer är att kraven som ställs från användarna på systemet anses vara inte stabila och förändras hela tiden (Noyes, 2002). Abrahamsson m.fl. (2002) anser att de dynamiska metoderna bygger på att använda få projektregler som anses tillräckliga. De anser vidare att detta leder till ett lyckat projekt som tar så lite tid så möjligt. God kommunikation och mindre dokumentation är en förutsättning för en lyckad tillämpning av en dynamisk metod (Fowler, 2002; Highsmith, 2001).

De dynamiska metoderna kräver kontinuerligt feedback från användarna för att få ett lyckat resultat (Fowler, 2002). En dynamisk metod medför iterativ utveckling, vilket är nödvändigt för hantering av oförutsägbara krav på systemet under ett systemutvecklingsprojekt (Fowler, 2002). Hirsch (2003) påpekar att vissa små projekt som tillämpat en dynamisk metod misslyckades trots alla fördelarna som är förknippat med den. Projektet misslyckades på grund av dålig kommunikation mellan de olika deltagarna i systemutvecklingsprojektet. En annan orsak till misslyckandet är avsaknandet av tidig feedback från användarna. Hirsch (2000) påpekar att en kund, som presenteras organisationen, ska involveras tidigt i projektet när en dynamisk metod tillämpas i ett systemutvecklingsprojekt. Kunden anses vara den viktigaste tillgången till information som behövs för ett litet systemutvecklingsprojekt (Highsmith, 2001).

## 2. Bakgrund

Med andra ord involveras kunden aktivt för att få en lyckad metodtillämpning av en dynamisk metod. Kunden bör också ha god kännedom om organisationen och till möjligheter som det nya systemet kan erbjuda (Fowler, 2002).

### 2.3.3 Två dynamiska metoder för små projekt

Fowler (2002) påpekar att Extreme Programming, XP, är en dynamisk metod. XP anses vara lämplig för små projekt, som omfattar högst tio deltagare och genomförs snabbt (Fowler, 2002). Namnet Extreme Programming kan lätt tolkas som att koda ostrukturerat och snabbt (Jeffries, 2001). Jeffries (2001) anser att anledningen till detta namn för metoden är att ordet "eXtreme" associeras med huvudmålet för XP som är att minimera riskerna som projektet utsätts för. Däremot ordet "Programming" associeras med det faktum att det är programmet, särskild källkoden, som tillför värde för användaren. Det vill säga att projektets syfte är att producera fungerade källkod och biprodukter, som används för att underlätta skapandet av källkod (Jeffries, 2001). Fowler (2002) påpekar att XP, som får mest uppmärksamhet bland de dynamiska metoderna, är en populär dynamisk metod för små och enkla projekt. Martin (2002a) anser att organisationer, där omfattande systemutvecklingsmetoder används, bör ha fokus på mjukvarutillverkning. Han menar att genom användning av en lätt och dynamisk metod så som XP kan bättre system med bättre mjukvara och kod fås. Alla projekt behöver inte omfattande metoder. Cockburn och Highsmith (2002) anser att en välstrukturerad XP med tio deltagare kan lösa stora problem bättre än en annan systemutvecklingsmetod som kräver trettio deltagare. Det beror på att XP löser problemen, som ska lösas under ett systemutvecklingsprojekt, genom att dela upp problemen till små problem (Cockburn och Highsmith, 2002). Martin (2002a) påstår att XP resulterar i system som är lätta att modifiera och lätta att utveckla. XP leder till att både tid och pengar sparas.

Ett annat exempel på dynamiska metoder för små projekt är dokumenterade anpassningar av RUP<sup>®</sup>. Noyes (2002) och Fowler (2002) påpekar att RUP<sup>®</sup> i sin ursprungliga form anses vara en omfattande och tung systemutvecklingsmetod med många faser och omfattande dokumentation. Därför anses RUP<sup>®</sup> vara lämplig för stora projekt, där kunden inte involveras aktivt på samma sätt som i de dynamiska metoderna för små projekt och där kraven är stabila och oförändrade (Noyes, 2002; Fowler, 2002). RUP<sup>®</sup>:s olika och självständiga delar ger möjlighet till anpassning av den genom att plocka bort det som inte anses behövs för ett litet projekt (Abrahamsson m.fl., 2002). Även Karlsson (2001) påpekar att ett viktigt koncept i RUP<sup>®</sup> är att den anses vara som ett ramverk och bör anpassas. Han förklarar vidare att anpassning av systemutvecklingsmetoden ger bättre fördelar som är mer påtagliga än att använda metoden så som den är. Vidare påpekar han att anpassning innebär att de delarna i metoden som bedöms viktiga betonas. Även Kruchten (2001), Noyes (2002) och Fowler (2002) anser att RUP<sup>®</sup> har en flexibel struktur som kan anpassas efter organisationsbehov och därmed anses som en dynamisk metod för små projekt.

Larman (2002) påstår att RUP<sup>®</sup> är en produkt som lanseras av företaget Rational. Härmed påpekas att i detta examensarbete behandlas RUP<sup>®</sup> så som en systemutvecklingsmetod. Koncepten, innebörd och definitioner som nämns i detta examensarbete anses vara generella för RUP<sup>®</sup>. Larman (2002) presenterar olika koncept att anpassa och bearbeta RUP<sup>®</sup> för att få den att bli en dynamisk metod. Ett av de koncepten är att välja ut och generalisera en del av metodens aktiviteter och arte-

## 2. Bakgrund

fakter. Detta koncept följs av Hirsch (2003) i hans dokumentation för den specifikt dokumenterade anpassade RUP. Dokumentationen är skriven år 2003 men Hirsch påpekar i den att den anpassade RUP används sedan år 1997 och den senaste version är sedan 2002. Den specifikt dokumenterade anpassade RUP bygger på att välja ut för ett litet projekt ett antal aktiviteter ur RUP<sup>®</sup> och resten av aktiviteterna är alternativa. Hirsch (2003) påstår att den dokumenterade anpassningen av RUP anses vara lämplig för små projekt. Han påstår också att den leder till effektiva systemutvecklingsprojekt om den införs och hanteras rätt.

Sammanfattningsvis finns det behov av att utveckla system i små projekt. Ett litet projekt kräver mindre omfattande systemutvecklingsmetoder än de traditionella systemutvecklingsmetoderna. De dynamiska metoderna passar för små projekt, som uppfyller kraven på snabba leveranser av utvecklade system med färre antal projekt-deltagare. Exempel på de dynamiska metoderna, vilka studeras i detta examensarbete, är XP och en specifik dokumenterad anpassning av RUP.

## 3 Problemformulering

### 3.1 Problembeskrivning

Komplexiteten som moderna verksamheter omfattar är ett problem i ett systemutvecklingsprojekt, eftersom moderna verksamheter kännetecknas av osäkra planer för framtiden och ökad konkurrens (Avison och Fitzgerald, 2003, s. 7). Osäkra planer och ökad konkurrens medför förändrade krav på systemet under ett systemutvecklingsprojekt. Martin (2002a) hävdar att traditionella systemutvecklingsmetoder, som används för utveckling av ett informationssystem är komplexa och tidskrävande, vilket inte passar verksamheters krav på snabbare leverans av system till kunden och användarna. Booch (2001) påstår att det inte går att minska komplexiteten i ett utvecklingsprojekt. Det går bara att förbättra sättet att hantera den. Han påstår vidare att komplexiteten minskar ju mindre projektet är och ju färre deltagare som deltar i det.

Huvudorsaken till behovet av lätta och snabbare systemutvecklingsmetoder är svårigheten för deltagare i små projekt att följa svåra och omfattande systemutvecklingsmetoder, som kräver tid och resurser av olika slag (Coldewey m.fl., 2000). Dynamiska metoder kännetecknas av flexibilitet och färre antal deltagare (Abrahamsson m.fl., 2002). Booch (2001) anser att system inte utvecklas snabbare genom att använda stora och omfattande systemutvecklingsmetoder. Användning av lätta systemutvecklingsmetoder med fokus på slutanvändarna skulle möta kravet att leverera system snabbt och med hög kvalitet (Booch, 2001). Exempel på en lätt och dynamisk metod, *eng. agile method*, är eXtreme Programming (XP) som tillämpas i små projekt med högst tio deltagare (Fowler, 2002). Abrahamsson m.fl. (2002) påstår att XP, som är en dokumenterad dynamisk metod, är benämnd i många arkitekturer och erfarenhetsrapporter.

Larman (2002) påstår att Rational Unified Process, RUP<sup>®</sup>, är en produkt som lanseras av Rational. Kruchten (2001), Noyes (2002) och Abrahamsson m.fl. (2002) anser att RUP<sup>®</sup> har en flexibel struktur som kan underlättas och anpassas efter organisationens behov. De hävdar också att de delar i RUP<sup>®</sup> som anses onödiga för systemutvecklingsprojektet kan plockas bort för att anpassa RUP<sup>®</sup> till små projekt. Därmed kan en anpassning av RUP<sup>®</sup> anses vara en dynamisk metod (Fowler, 2002). Noyes (2002) hävdar också att RUP<sup>®</sup> kan, om den styrs och anpassas rätt, leverera system snabbt så som eXtreme Programming, XP. Orsaken är att en rätt anpassad RUP kan ge bättre kravhantering, analys och design (Noyes, 2002). Med bättre kravhantering menas att förändringar av kraven behandlas under systemutvecklingsprojektet. Larman (2002) påstår att det finns tre olika koncept att anpassa och bearbeta RUP, för att få den till en dynamisk metod. Ett av de koncepten är presenterat av Hirsch (2003) i form av en specifik dokumenterad anpassning av RUP, som därigenom kan anses vara en dynamisk metod. Hirsch (2003) påpekar att denna specifika dokumenterade anpassning av RUP leder till ökad effektivitet i systemutvecklingsprojektet om den införs och hanteras rätt. Det som är speciellt kritiskt i en lyckad metodtillämpning av den specifikt dokumenterade anpassade RUP för små projekt behöver studeras närmare, eftersom vissa små projekt som tillämpade den specifikt dokumenterade anpassade RUP, som Hirsch (2003) presenterar, misslyckades. Hirsch (2003) påpekar att ett litet projekt bör omfatta mellan tre till åtta utvecklare, där projektet genomförs



### 3. Problemformulering

på mellan sex och tolv månader. Pollice (2001a) och Fowler (2002) anser att små projekt omfattar tolv eller färre antal deltagare, där projektet omfattar mellan några veckor och högst några få månader. I detta examensarbete anses små projekt omfatta mellan tre till tio deltagare, där projektet genomförs på mindre än tolv månader.

#### 3.2 Problemprecisering

Kraven på att snabbt leverera mindre resurskrävande system är orsaken till att existerande systemutvecklingsmetoder utvecklas till mindre omfattande metoder. I detta examensarbete studeras de dynamiska metoderna för små projekt. En tillämpning av denna typ av metoder kännetecknas av att involvera få deltagare och av krav på snabb leverans till kunden och användarna. Förutom dokumenterade systemutvecklingsmetoder omfattar de dynamiska metoderna även dokumenterade anpassningar av RUP, som är avsedda för små projekt. De två väldokumenterade exemplen på dynamiska metoder för små projekt som studeras i detta examensarbete är XP och Hirsch:s (2003) specifika anpassningen av RUP. De metoderna är valda utifrån starka påståenden i respektive metodbeskrivning om dess förträfflighet och tillämpbarhet. Vidare är de två metoderna vanligt förekommande vid små systemutvecklingsprojekt. Specifikt studeras följande delfrågor:

- Vad är de dynamiska metodernas dokumenterade karaktäristiska drag som gör att de är lämpliga för små projekt?
- Vilka erfarenheter finns från metodtillämpningar med de två metoderna som styrker argumenten från metodlitteraturen avseende dess lämplighet för små projekt?

Resultatet av examensarbetet förväntas vara intressant för verksamheter som planerar att införa denna typ av systemutvecklingsmetoder för små projekt.

## 4 Metod

### 4.1 Översikt

För att besvara frågorna som presenteras i problempreciseringen för detta examensarbete blir kvalitativa forskningsmetoder ett naturligt val. Orsaken är att materialet som används är textmaterial och därmed är denna typ av metoder lämpliga. Kvalitativa metoder kännetecknades av att skapa förståelse för ett visst fenomen och att studera det på djupet, vilket gör att de täcker hela problempreciseringen. För att kunna hitta de karaktäristiska dragen för de två dynamiska metoderna, som ska studeras, kan följande metoder anses vara rimliga alternativ: litteraturstudie, besöks- och telefonintervjuer samt fallstudie.

Metoderna som valdes för detta examensarbete bestod av en kombination av litteraturstudie och intervju. Första frågan i problempreciseringen, som är ”Vad är de dynamiska metodernas dokumenterade karaktäristiska drag som gör de lämpliga för små projekt?”, kom att besvaras med hjälp av litteraturstudie. Den andra frågan i problempreciseringen, som är ”Vilka erfarenheter finns från metodtillämpningar med de två metoderna som styrker argumenten från metodlitteraturen avseende dess lämplighet för små projekt?”, kom att besvaras genom en kombination av litteraturstudie samt telefonintervjuer för att täcka aspekter som inte finns i litteraturen.

### 4.2 Litteraturstudie

Litteraturstudie är relevant för detta examensarbete då de två dynamiska metoderna för små projekt som studerades är väldokumenterade. För att reda ut om en systemutvecklingsmetod är lämplig för ett litet projekt krävdes det en litteraturstudie som studerar de dynamiska metodernas dokumentation. En litteraturstudie kan ge tillgång till all generell, relevant och dokumenterade information som täcker undersökningsämnet (Patel och Davidsson, 1994). Därför genom litteraturstudie kan den första frågan, som är ”Vad är de dynamiska metodernas dokumenterade karaktäristiska drag som gör att de är lämpliga för små projekt?”, besvaras. Litteraturstudie kan också vara lämplig för att besvara den andra frågan, som är ”Vilka erfarenheter finns från metodtillämpningar med de två metoderna som styrker argumenten från metodlitteraturen avseende dess lämplighet för små projekt?”, för att erfarenheter kring de två dynamiska metoderna för små projekt också är dokumenterade. Litteraturstudien kan också ge möjlighet till att skapa underlag för de eventuella intervjufrågorna. Intervjufrågorna kan komma att ställas i eventuella intervjuer för mera aktuella erfarenheter kring de två dynamiska metoder för små projekt.

Kunskap behövs för att kunna bearbeta problempreciseringen för detta examensarbete. Kunskapen byggs genom att studera vad kända författare inom ämnet har skrivit om samma eller liknande problem. Enligt Patel och Davidsson (1994) är böcker, artiklar som är publicerad i vetenskapliga tidskrifter och rapporter de vanligaste källorna där kunskap hämtas. Materialet som ska samlas in för att besvara detta examensarbets problemprecisering är från böcker, artiklar, forskningsrapporter och dokument. Böcker och dokument som behandlar ämnet skulle kunna ge en helhetssyn och ge möjlighet att skapa grundläggande fakta om ämnet. Däremot

## 4. Metod

forskningsrapporter och artiklar bidrar med den senaste informationen om ämnet som är relativt nytt. Systematisk genomgång av litteraturen som behandlar dynamiska metoder för små projekt kan ge en överblick av ämnet. För att informationssökning via litteraturstudie inte ska bli tidskrävande är det viktigt att i samband med läsningen av böcker anteckna den information som är relevant för arbetet. Även Patel och Davidsson (1994) påpekar att informationssökning via litteraturstudie är tidskrävande för att finna relevant information som undersökaren kan använda sig av.

### 4.3 Intervjuer

Anledning till att intervjuer kan användas för detta examensarbete är att de ger möjlighet till att reda ut överensstämmelse mellan vad som sägs om de karaktäristiska dragen med aktuella erfarenheter kring tillämpning av de två dynamiska metoderna. Med andra ord kan den andra delen av frågeställningen besvaras. En annan anledning kan vara att intervjuerna kan ge möjlighet till att reda ut erfarenheter kring de två dynamiska metoderna, där detaljer och aspekter om tillämpningar av dessa dynamiska metoder, som inte täcks av litteraturen, kunde redas ut. En intervjuundersökning kan ge möjlighet till att samla in aktuella praktiska erfarenheter från vanliga personer som tillämpar de två dynamiska metoderna för små projekt, och inte bara från kända författare i ämnet. På det viset kan ur praktiska erfarenheter avgöras de två dynamiska metodernas lämplighet för små projekt.

Den andra frågan i problempreciseringen kan helt besvaras med hjälp av intervjuundersökningar om det kan bli lätt att få tag på tillräcklig många och relevanta intervjupersoner. Eftersom det kan kanske inte vara lätt att få tag på tillräcklig många intervjupersoner kan den andra frågan besvaras genom kombination av litteraturstudie och intervjuer. Första frågan är inte relevant att besvaras med intervjuer för att den är olämplig metod att hitta de dokumenterade karaktäristiska dragen för de dynamiska metoderna genom intervjuer.

I en intervjuundersökning kan intervjuaren vid registrering av svaren avgöra vad som är relevant för undersökningen, vilket kan ses vara en nackdel om intervjuaren missar viktiga delar av svaren. Ett problem till som kan bli aktuellt är att insamlingen av information genom intervjuer kan upplevas som krävande både i tid och i arbete. Därför är det viktigt att tänka under undersökningsarbetet på att tiden som avses för detta examensarbete är begränsat.

För detta examensarbete påpekas att en relativt strukturerad intervju, där det finns utrymme till följdfrågor beroende på vad respondenten svarar, kan bli mest relevant för att täcka hela den andra frågan i problempreciseringen på ett tydligt sätt. En strukturerad intervju ska ge möjlighet till att intervjun förberedas, där frågorna kan framställas innan intervjun. För att reda ut om de karaktäristiska dragen för de två dynamiska metoderna gäller generellt kan intervjufrågorna formuleras utan fokus på några specifika dokumenterade karaktäristiska drag. Denna typ av frågor kallas för öppna frågor. Alternativ till öppna frågor är slutna frågor. Slutna frågor för detta examensarbete kan formuleras efter de karaktäristiska dragen som hittas när första frågeställningen besvaras. Med andra ord kan intervjufrågorna formuleras efter de dokumenterade karaktäristiska dragen som hittas ur litteraturstudie eller vara i form av öppna frågor om erfarenheter kring de karaktäristiska dragen. På det viset kan det

## 4. Metod

som styrker argumenten från metodlitteraturen avseende dess lämplighet för små projekt redas ut. Besöksintervjuer i kombination av strukturerade intervjuer kan bli relevanta för detta examensarbete för att de kan ge möjlighet till att samla information från respondenter medan de är i arbetsmiljön. Besöksintervjus fördel är att det kan gå att ställa många, komplicerade och öppna frågor, där eventuella oklarheter i frågorna kan enkelt redas ut. Nackdelen med besöksintervjuer för detta examensarbete kan bli att de kanske ska kräva att intervjuaren reser långt för att intervjua respondenter.

Ett alternativ till besöksintervjuer är telefonintervjuer. Intervjufrågorna kan bli slutna för att underlätta för intervjupersonen att besvara dem via telefon. Slutna frågor kan vara en vägledning till relevanta svar. Telefonintervjuer passade detta examensarbete för att de är ett sätt att undvika resor. Erfarenheter om det praktiska införandet att jämföra med vad litteraturen säger om de två dynamiska metoderna för små projekt kunde vara begränsat och svårt att hitta i regionen. Därför anses att telefonintervjuer vara relevanta, då de inte kräver att intervjuaren reser långt eller blir låst till vissa intervjupersoner som befinner sig i regionen. Nackdelar förknippat med telefonintervjuer för detta examensarbete är att det kan bli svårt att ställa omfattande frågor som tar lång tid att besvara och att det kan finnas risk att inte få tag på sökta personer.

### 4.4 Fallstudie

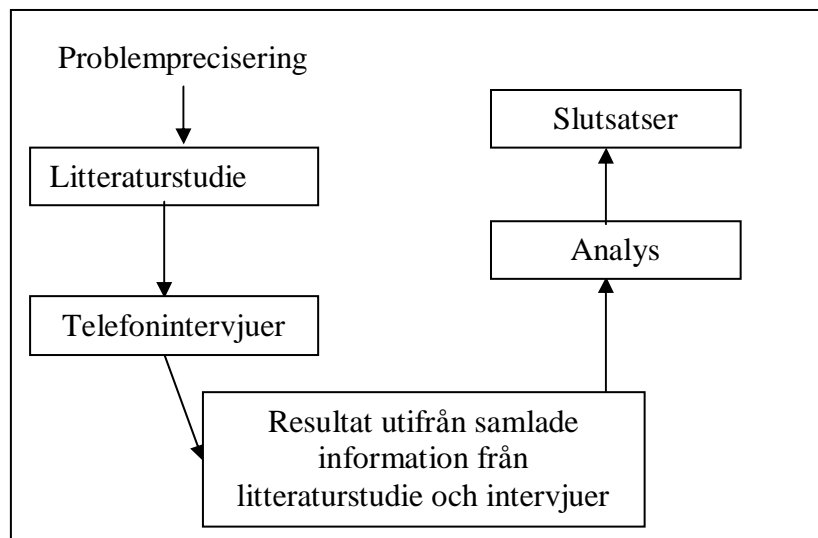
Fallstudien kan bli relevant för detta examensarbete för att den kan ge möjlighet till att kunna studera och utreda de karaktäristiska dragen för de två dynamiska metoderna på en avgränsad grupp, till exempel ett specifikt företag. Gruppens erfarenheter av tillämpning av de två dynamiska metoderna kan också studeras genom en fallstudie. Fallstudie i kombination av litteraturstudie kan bli ett sätt att genomföra detta examensarbete och att besvara både frågorna som presenteras i problempreciseringen. Litteraturstudie kan komplettera fallstudie genom att utreda och förtydliga de dokumenterade karaktäristiska dragen. Med andra ord kan den första frågeställningen besvaras med fallstudier i kombination med litteraturstudie för att reda ut och förklara de dokumenterade karaktäristiska dragen som gör dynamiska metoder lämpliga för små projekt. Den andra frågeställningen kan besvaras helt av fallstudier för att de ger möjlighet till att samla erfarenheter kring de två dynamiska metoderna i praktiken genom att studera tillämpningar av de två dynamiska metoderna.

Två fallstudier är aktuella i detta examensarbete för att undersöka erfarenheter kring tillämpning av de två dynamiska metoderna för små projekt. Det är inte troligt att ett företag eller en projektgrupp använder båda systemutvecklingsmetoderna. Nackdelen med de två fallstudierna är att de ska ta mycket tid och ska kanske bli svårt att genomföra på den begränsade tiden som avses för detta examensarbete. Fallstudiernas fördel för detta examensarbete är att resultatet från studierna om påstående och erfarenheter kring de karaktäristiska dragen för de dynamiska metoderna för små projekt kan generaliseras.

## 4.5 Metodtillämpning

I detta avsnitt presenteras de metoder som användes för detta examensarbete. Orsaken till varför de valdes presenteras också.

Genom telefonintervjuer och litteraturstudie kunde informationen analyseras och sammanfattas för att ge ett resultat för detta examensarbete. Fallstudie valdes bort för att det inte fanns ett fall, en organisation, att studera. Dessutom skulle det kanske behövas två fall för att studera för att det inte skulle vara möjligt att hitta en organisation som tillämpar de två dynamiska metoderna, som skulle studeras, samtidigt. För att tydligare beskriva genomförandet för detta examensarbete presenteras genomförandet grafiskt i figur 3.



Figur 3. Studiens genomförande

Litteratur som var aktuell för detta examensarbete är skrivna av kända författare i ämnet. Litteratur var ur olika författare, som har olika bakgrund, men behandlar samma ämnesområde. Undersökaren förhåll sig kritisk till all litteratur som användes i detta examensarbete för att kunna göra en bedömning om fakta eller upplevelser var sannolika eller inte. Undersökaren förhöll sig kritisk genom att söka fakta om samma sak i flera litteraturer av olika kända författare och genom att söka fakta av lämpliga intervjupersoner.

## 5 Systemutvecklingsmetoder för små projekt

### 5.1 Aspekter för små projekt

Nedan presenteras aspekter för små projekt. Syftet med detta avsnitt är att klarlägga de kritiska framgångsfaktorerna för små projekt. Med små projekt avses projekt som omfattar färre än tio deltagare och som genomförs på mindre än ett år.

Eckstein och Josuttis (2002, s. 2) påpekar att korta feedbackcykler är lätt att etablera när det är små projektgrupper. Feedback ger möjlighet att upptäcka och lösa problemen tidigt, vilket skapar effektivitet i projektet (Eckstein och Josuttis, 2002). The Standish Group (1994, s. 3) påstår att små projekt, som genomförs på korta tidsperioder, kännetecknas av tidig feedback och av att systemet levereras ofta till användarna. De förklarar vidare att små projekt resulterar i en iterativ utveckling av design, prototyper, utveckling och test för små element. Tidiga leveranser av systemet, vilket ger möjlighet till feedback, anses stödja att utvecklingen blir iterativ.

Eckstein och Josuttis (2002) påpekar också att en kund involveras i ett litet projekt för att presentera organisationen och bestämma vilka krav som ska behandlas. De påpekar vidare att antalet kunder, som deltar i projektet för att presentera organisationen, troligen ökar när projektgruppen är stor. Detta kan leda till konkurrens och konflikter mellan kunderna, vilket kan ta lång tid att hantera. Det påpekas att det finns en aspekt för små projekt som syftar till att bara en kund deltar i projektet. Deltagandet av en kund underlättar för de andra projektdeltagarna att förstå organisationsbehov, hur beslut tas och vilka funktioner som finns i organisationen (Craig och Jassim, 1995, s. 62). Wisén och Lindblom (1998, s. 127) påpekar att ett projekt som omfattar små projektgrupper är lätt att styra om projektdeltagare har tid med projektarbetet. Kunden, som involveras i små projekt, ska ha tid för projektarbetet och ska vara aktiv.

Cockburn och Highsmith (2002, s. 151) förklarar att projektstorlek avgör möjligheten att ha öga mot öga kommunikation, som klassificeras som god kommunikation. God kommunikation är viktigt mellan projektdeltagarna i ett litet projekt. De menar vidare att när projektet är litet med få antal deltagare finns det möjlighet att ha öga mot öga kommunikation, där alla projektdeltagarna arbetar i ett arbetsrum. När projektdeltagarna är få blir lätt att kommunicera och samarbeta i ett rum (Cockburn och Highsmith, 2002). I en projektgrupp, som omfattar mellan 80 och 100 deltagare är risken stor att deltagarna blir splittrade i olika avdelningar, olika byggnader eller olika städer. Detta leder till att det blir svårt att reda ut vem som gör vad. Det blir också svårt att förhindra projektdeltagarnas arbete att överlappa varandra. Öga mot öga kommunikation leder till att kommunikationskvaliteten ökar, att kostnaden för kommunikation blir låga och att deltagarna i projektet förstår varandra (Cockburn och Highsmith, 2002).

Craig och Jassim (1995, s. 85) påstår att en person kan ha flera roller i små projekt för att uppfylla projektets olika behov. Vidare påstår de att de tekniska uppgifterna i små projekt utförs av projektdeltagarna och ingen central support behövs. Craig och Jassim (1995) tycker att varje input och output i ett litet projekt ska definieras tydligt och bara det som måste göras görs. Armano och Marchesi (2000, s. 4) påstår innehållet för att små projekt måste kunna omstruktureras för att kunna hantera förändringar. De påstår vidare att ett litet projekt ska omfatta verktyg som hjälper projektdeltagarna att skapa

## 5. Genomförande

effektivitet i projektet för att kunna hinna med det som skall göras. Smith (1999) påpekar att resurserna som kommer att utnyttjas under ett litet projekt skall förutsägas lätt och snabbt.

Nedan presenteras sammanfattning av aspekterna för små projekt:

- Tester och feedback för att utvecklingsprocessen ska vara iterativ.
- Leverera delsystemen ofta till användarna.
- En kund måste delta aktivt och ska ha tid för projektet.
- God kommunikation är viktigt och är bäst genom öga mot öga kommunikation.
- En projektdeltagare ska ha flera roller.
- Enkelt att reda ut vad som ska göras och vem som gör vad.
- Att bara göra det som måste göras.
- Omstruktureringen av projektet sker när förändringar äger rum.
- Verktyg eller principer som skapar effektivitet.
- Under ett litet projekt förutsägs resursbehovet snabbt.

Aspekterna för små, som presenteras ovan, är ur olika källor. De olika källorna är överens om att behoven i ett litet projekt skiljer sig ifrån behoven i ett stort projekt.

### 5.2 Extreme Programming för små projekt

Nedan presenteras det som betonar de karaktäristiska dragen som gör Extreme Programming, XP, lämplig för små projekt.

#### 5.2.1 Principerna för ett litet XP-projekt

Jeffries (2001, s. 1) och Fowler (2002) påstår att enkelhet, kommunikation, feedback och mod är grundbegreppen för XP. Enkelhet innebär att bara det som behövs görs. Enkelhet stöder därmed små projekt, som syftar på att bara det absolut nödvändigaste görs för att minska komplexiteten i projektet. Jeffries (2001) påpekar vidare att det är enkla uppgifter som genomförs i ett XP-projekt. Avison och Fitzgerald (2003) och Jeffries (2001) påpekar att XP ger möjlighet för deltagarna i projektet att kommunicera bättre genom kontinuerlig testning, vilket försäkras systemets kvalitet och ger viktig feedback från användarna. Feedback ger information från användarna, vilket hjälper projektdeltagarna att utreda vart de befinner sig i utvecklingsarbetet (Jeffries, 2001). En av de aspekterna, som presenteras i kapitel 5.1, för små betonar vikten av feedback. Feedback som kommunikationen medför stödjer små projekt. Abrahamsson m.fl. (2002) påpekar att XP har utvecklats efter problemen som har uppstått på grund av de långa utvecklingscyklerna i de traditionella systemutvecklingsmetoderna. De förklarar vidare att alla nyckelprinciper som avses för XP inte är nya för systemutvecklingsmetodens område. De nyckelprinciperna för XP samlades och valdes ut efter dynamiska metodens principer (se även Fowler, 2002). Wells (1999) förklarar att XP har få regler och viktiga principer att följa, vilket säkerställer effektivitet i små systemutvecklingsprojektet.

XP är en kollektion av tolv principer som delvis kan anses vara bekanta med andra principer för andra systemutvecklingsmetoder (Martin, 2002b; Fowler, 2002). De

## 5. Genomförande

principerna stödjer de fyra grundbegreppen för XP, som är enkelhet, kommunikation, feedback och mod (Pollice, 2001b, s. 2). De fyra grundbegreppen har samma syfte som en del av de aspekterna. Därför anses de principerna som karaktäristiska drag för en lyckad metodtillämpning av XP för små projekt. Nedan presenteras principerna som gör XP lämplig för små projekt:

*En kund deltar i projektet:* Jeffris (2001) påpekar att en kund är representant för organisationen i ett XP-projekt. I XP är varje deltagare i projektet en integrerad del i hela projektgruppen. Det är bara en kund som involveras aktivt och dagligen i systemutvecklingsprojektet (Jeffries, 2001). Jeffries (2001) påstår att kundens uppgift i projektet är att bestämma vad som ska ingå i systemet och vad som inte ska ingå. Med andra ord presenterar kunden vilka krav som ställs från användarna på systemet. Kunden ska också ha tid för att lösa konflikter och svara på alla frågor som berör organisationsbehov (Abrahamsson m.fl., 2002; Fitzgerald m.fl., 2002). Denna princip presenteras som en aspekt för små projekt (se kapitel 5.1). Därför anses principen om deltagandet av en kund i ett projekt gör XP lämplig för små projekt. Cockbrun och Highsmith (2002) påpekar att aktivt kunddeltagande ger feedback snabbt, vilket ger möjlighet till de andra projektdeltagarna att under hela projektet förstå behoven i organisationen. Pollice (200b) förklarar att tillsammans med aktivt kunddeltagande finns två andra villkor som kännetecknar ett litet XP-projekt. De två andra villkoren är att projektdeltagarna inte ska vara mer än tio och att samarbetet ska vara god mellan deltagarna. Risken minskar att få ett lyckat XP-projekt om en av de villkoren saknas (Pollice, 2001b).

*Små delsystem:* Fitzgerald m.fl. (2002) anser att det är enkla system som tillverkas snabbt i ett XP-projekt. De påpekar vidare att sedan nya versioner på de enkla systemen utvecklas och överlämnas till användarna under korta tidsperioder. Varje delsystem, som presenterar ett mål, ska vara så liten som möjligt och ska uppfylla kraven som ställs från användarna på systemet (Fitzgerald m.fl., 2002). Varje delsystem är resultat av en samling av iterationer, där varje iteration genomförs på cirka två veckor (Martin, 2002b). Små delsystem ger möjlighet att få feedback som anses vara viktig en aspekt för ett litet projekt.

*Planeringsknep:* Ett snabbt beslut tas om omfattning och resurser för nästa delsystem, *eng. release*. Beslutet är baserat på möjligheterna som organisationen och tekniken erbjuder (Fitzgerald m.fl., 2002). Jeffries (2001) påpekar att XP-projektgruppen använder en enkel form av planering för att bestämma vad som skall göras och för att förutsäga när projektet anses vara fullgjord. Detta planeringsknep, som är att förutsäga resurserna och omfattning snabbt och enkelt, är ett av de karaktäristiska dragen som gör XP lämplig för små projekt. Anledningen är att förutsäga resurserna snabbt och enkelt, anses vara en aspekt för ett litet projekt.



## 5. Genomförande

Testning: Tester anses vara en viktig del i XP eftersom de måste vara avklarade innan implementationsfasen (Fitzgerald m.fl., 2002). Jeffries (2001) påpekar att XP har en testdriven design, där programmerare skriver mjukvara steg för steg. Först skrivs en liten test för en kod därefter utvecklas koden, sedan skrivs en test till och så vidare. Därför anses XP vara en iterativ metod och därmed anses vara lämplig för små projekt. Tester, som leder till iterativ utvecklingsprocess, anses vara en aspekt för ett litet projekt. Fitzgerald m.fl. (2002) anser att det finns två kategorier av tester i XP. Det är enhetstester, som programmerare använder för att reda ut om koden fungerar. Funktions- eller acceptanstester används för att reda ut om systemet möter kraven som ställs från kunderna (Martin, 2002b).

Enkel design: Designen i XP ska vara enkel och bara omfattar det som är nödvändigt (Abrahamsson m.fl., 2002; Fitzgerald m.fl., 2002; Martin, 2002b). Designen ska vara exakt anpassad efter systemets funktionalitet, där inget extra och onödigt arbete ska göras (Jeffries, 2001). Jeffries (2001) påstår att vid en iterativ metod som XP är det viktigt att ha en väl strukturerad design för att behålla effektiviteten och god kommunikation i projektet. God kommunikation och effektiviteten anses vara karaktäristiska drag som gör XP lämplig för små projekt.

Kontinuerlig integration: Martin (2002b) och Abrahamsson m.fl. (2002) anser att kontinuerlig integration i XP innebär att programmerare integrerar och testar koden flera gånger per dag och efter varje ändring. Stora kodförändringar på en gång är inte att rekommendera. Fördelen med kontinuerlig integration är att felet upptäcks snabbt och att koden testas flera gånger tills den fungerar till 100 procent. Mindre integrationer och tester leder till svårigheter att tidigt reda ut var felet är när felet uppstår (Jeffries, 2001). Kontinuerlig integration ger feedback, vilket kan anses som en fördel för ett litet projekt som genomförs på ett begränsat tid.

Kollektivt kodägande: Koden är kollektiv och ägs av deltagarna i projektet, vilket ger möjlighet till bättre tillgång till hela koden och till bättre kommunikation (Abrahamsson m.fl., 2002; Jeffries, 2002; Martin, 2002b). Verktyg eller principer som skapar effektivitet är en aspekt för ett litet projekt. Kollektivt kodägande skapar effektivitet och produktivitet i projektet (Jeffries, 2002), vilket kan anses som fördel för ett litet projekt med begränsat tid och antal deltagare. Jeffries (2002) påpekar att daglig kontakt med koden förbättrar kvalitén och leder till att det blir mindre fel på koden. Vidare påpekar han att kollektivt kodägande leder till stora problem om deltagarna i projektet inte vet vad de gör. XP undviker de stora problemen genom att programmering sker i par och genom enhetstester.

Ett programmeringsspråk: Denna princip skapar effektivitet, vilket är en aspekt för ett litet projekt. Denna princip leder också till god

## 5. Genomförande

kommunikation, som anses vara en aspekt för ett litet projekt. Martin (2002b) anser att kommunikationen förbättras mellan deltagarna när bara ett programmeringsspråk används under projektets gång, då kommunikationen sker på samma sätt och på samma språk. Kodningen blir standardiserad genom att bara ett programmeringsspråk som blir aktuellt i projektet. Standardkodning är för att skapa konsistens i projektet och för att ge möjlighet till kollektivt ägandet av koden (Jeffries, 2002).

*Omstrukturering:* Martin (2002b) anser att när oväntade förändringar i systemet blir aktuella stämmer inte designen längre och att den försämras, därför bör projektets innehåll omstruktureras snabbt efter förändringar. Han anser vidare att genom omstrukturering rättas designen, som är ett kommunikationsmedel, för att få den att stämma med verkligheten under projektets gång. Omstrukturering undviker redundans, minskar oanvändbara funktioner, behåller koden ren och behåller designen enkel (Wells, 1999), vilket leder till enkelhet under ett litet projekt. Snabb omstrukturering efter förändringar anses vara en aspekt för ett litet projekt.

De principerna, som presenteras ovan, har formulerats efter erfarenheter från införandet av traditionella systemutvecklingsmetoder (Fitzgerald m.fl., 2002, s. 65). Fitzgerald m.fl. (2002) påstår att de traditionella systemutvecklingsmetodernas principer formulerades efter vad som borde fungera teoretisk, utan att testa om de verkligen passar in i systemutvecklingsmiljön. Vidare skriver han att XP:s principer kompletterar och förstärker varandra genom att de överlappar varandra.

Crocker (2002) påpekar att i ett XP-projekt sker programmeringen i par. Han förklarar vidare att för att kunna programmera i par ska det vara enkelt att kommunicera. När det är få deltagare blir det enkelt att kommunicera och programmera i par eftersom det är lätt då att få projektgruppen samlade i samma lokal (Pollice, 2001b). Det förutnämnd förklarar varför god kommunikation i form av öga mot öga kommunikation existerar i ett litet XP-projekt. God kommunikation genom öga mot öga kommunikation anses vara en aspekt för ett litet XP-projekt. Crocker (2002) förklarar att om antal deltagarna ökar blir risken stor att projektgruppen att blir distribuerat. Projektgruppen kan också bli fördelad till flera projektgrupper som löser delproblem istället för hela problemet, vilket kan försämra produktens kvalitet. De olika projektgrupperna kan då möjligen fokuserar sig bara på sitt, vilket leder till mindre effektivitet (Crocker, 2002). Därför är det bara en kund som deltar i ett litet XP-projekt. Det finns en aspekt för små projekt som syftar till att bara en kund som deltar i projektet.

Beck (1999) förklarar att det är viktigt att teknologin, som används i XP, snabbt hanterar eventuella förändringar för att kunna hålla tidsramen för projektet. Teknologin ska inte kräva omfattande feedback från kunden och användarna. Orsaken är att det kan försämra motivationen hos kunden och användarna, vilket i sin tur kan leda till förseningar i systemutvecklingsprojektet (Beck, 1999). Beck och Fowler (2001) påpekar att dagliga korta möten med alla projektdeltagarna i ett XP-projekt ger möjlighet till att reda ut vem som gör vad under dagen.

### 5.2.2 Arbetsprocessen för små XP-projekt

Scott (2002) påpekar att projektgruppen för ett XP-projekt ska vara högst tolv deltagare. Han menar att det är vad ett XP-projekt klarar av för att hålla hela projektgruppen i ett och samma rum. Det finns olika roller för ett XP-projekt. De rollerna har olika uppgifter i utvecklingsprojektet (Beck, 1999). Jeffries (2001) anser att ett XP-projekt omfattar bland annat testare som hjälper kunden att utföra acceptanstester, analytiker som analyserar det som genomförs och ledare som leder arbetet i projektet. Förutom de förutnämnda rollerna kan ett XP-projekt omfatta ett antal roller som ska fördelas på ett begränsat antal deltagare. Enligt Beck (1999) är de rollerna utredare, konsult och chef. En utredare ger feedback på det som genomförs under projektets gång och utreder hur bra systemet är. En konsult är en extern deltagare som hjälper till med tekniska problem när det behövs och en chef fattar olika beslut under projektets gång. Varje individ behöver inte bara ha en roll och behöver inte vara specialist för denna roll, utan alla deltagarna kan ha flera roller (Jeffries, 2001). Det finns en aspekt för ett litet projekt som syftar till att en deltagare ska ha flera roller. En deltagare i ett XP-projekt kan ha flera roller, vilket är en anledning till att XP anses vara lämplig för små projekt.

Livscykeln för XP omfattar ett antal faser, som Beck (1999) diskuterar, presenteras nedan. Dessutom presenteras vad som är i de faserna som styrker XP:s lämplighet för små projekt. Aspekterna för små projekt, som presenteras i kapitel 5.1, är utgångspunkten vid redogörelsen av XP:s faser. Faserna är:

1. Utforskningsfas: Beck (1999) påpekar att under denna fas skriver kunderna berättelser på vad de ställer för krav på systemet efter utvecklingsarbetet. Varje berättelse beskriver en egenskap som systemet kommer att ha. Wells (1999) påpekar att skillnaden mellan en berättelse och ett kravdokument, som presenterar kraven på systemet från kunden, är beskrivningens detaljnivå. Han påpekar vidare att en berättelse anses vara underlag att planera tiden som krävs och definiera det som ska vara input i projektet. Med andra ord är en berättelse ett hjälpmedel att förutsäga behoven. Under denna fas ska hela projektgruppen bilda en gemensam uppfattning och bekanta sig med projektets verktyg, teknologi och principer (Beck, 1999). Det kräver god kommunikation mellan projektdeltagarna för att kunna bilda en gemensam uppfattning. God kommunikationen bör fås via öga mot öga kommunikation. Arbetet för denna fas beräknas dröja några veckor, beroende på hur lång tid det tar för programmerare att bekanta sig med teknologin som ska användas för projektet (Beck, 1999). Sammanfattningsvis förutsägs resursbehovet och öga mot öga kommunikation blir aktiv mellan deltagarna under denna fas. De förutnämnda anses vara aspekter för ett litet projekt.
2. Planeringsfasen: Argumenten och berättelser för det första delsystem måste vara avklarade Under denna fas (Beck, 1999). Beck (1999) anser att därefter framställer utvecklaren ett enkelt schema, där bara det som måste göras är planerat in i schemat. En av de aspekterna för små projekt syftar till att bara det som måste göras görs. Under denna fas bara det som måste göras klarläggas och

## 5. Genomförande

planeras i schemat. Beck (1999) påpekar vidare att under denna fas inte brukar ta mer än några dagar.

3. Iterationersfasen: Schemat, som är skapat i planeringsfasen, formuleras om till ett antal iterationer, där var och en av de iterationerna ska ta mellan en till högst fyra veckor (Beck, 1999). Iterationer leder till att utvecklingsprocessen blir iterativ, vilket anses vara en aspekt för ett litet projekt. Beck (1999) förklarar att kunden deltar när beslutet tas om vilka berättelser ska projektet omfatta. Kunden deltar aktivt också när beslutet tas om vilka berättelser ska sorteras för varje iteration. Vidare skriver han att funktionstester, som skapas av kunden, används i slutet av varje iteration för att kunna få feedback från användaren. I slutet av den sista iterationen är delsystemet redo för produktion (Beck, 1999).
4. Produktionsfasen: Beck (1999) förklarar att denna fas omfattar extra tester för att bedöma utförandet av delsystemet innan det överlämnas helt till användaren för produktion och användning. Tester medför feedback som är viktigt för små projekt. Beslut om eventuella nya förändringar skall inkluderas i delsystemen eller inte tas under denna fas (Beck, 1999). Beck (1999) påpekar att om de ska inkluderas genomförs iterationerna snabbt, där varje iteration beräknas att dröja högst en vecka. Sammanfattningsvis framgår under denna fas tester, som ger feedback och omstruktureras projektets innehåll efter förändringar. De förutnämnda anses vara två aspekter som följs av deltagarna under ett litet projekt.
5. Underhållsfasen: När det första delsystemet överlämnas helt till kunden och användarna, måste projektdeltagarna behålla systemet i användnings- och underhållsform om nya iterationer ska produceras (Beck, 1999). Genom att behålla systemet i användnings- och underhållsform skapas effektivitet. Med andra ord följs under denna fas en princip som syftar till att skapa effektivitet. En aspekt av de aspekterna för små projekt syftar till att det ska finnas i projektet principer som skapar effektivitet.

Dessutom finns en avslutningsfas. Avslutningsfasen börjar då inga mera berättelser, som skrivs av kunden, är kvar att bearbeta (Beck, 1999). Beck (1999) påpekar att avslutningsfasen inte påbörjas förrän det nya systemet överensstämmer med vad kunden önskade innan utvecklingsarbetet påbörjades.

Avison och Fitzgerald (2003, s. 444) påstår att XP bara omfattar fyra faser, vilka är planering, design, utveckling och produktion. Omfattning av de faserna liknar i stort sett det som Beck (1999) presenterar. Men omfattning av avslutningsfasen benämns inte av Avison och Fitzgerald (2003).

### 5.2.3 Sammanfattning av de karaktäristiska dragen som gör XP lämplig för små projekt

Utifrån de som presenterade ovan i detta kapitel om XP, anses de karaktäristiska dragen som gör XP lämplig för små projekt är följande:

- Kontinuerlig leverans av systemet till användarna för testning, vilket ger feedback.
- XP är en iterativ utvecklingsmetod.
- En kund deltar aktivt i projektet.
- Det finns möjlighet att tidigt och snabbt förutsäga resurserna som ska förbrukas under projektets gång.
- Projektets innehåll omstruktureras snabbt om något förändras.
- Koden är tillgänglig för alla projektdeltagarna, vilket anses vara en princip som leder till effektivt under ett litet projekt.
- Bara ett programmeringsspråk som används under ett litet XP-projekt för att säkerställa god kommunikation och effektivitet under ett litet projekt.
- Fokus på att projektet ska kännetecknas med god kommunikation mellan alla projektdeltagarna genom öga mot öga kommunikation.
- En projektdeltagare får ha flera roller.
- Bara det som anses viktigt för att få ett lyckat projekt görs.
- XP syftar till att alla projektdeltagarna ska veta vad de ska göra och vem som gör vad. Det utredas genom bland annat dagliga möten.

De karaktäristiska dragen, som presenteras ovan, kan jämföras med aspekterna för små projekt, som presenteras i kapitel 5.1. Utifrån jämförelsen konstateras att de karaktäristiska dragen följer aspekterna för små projekt.

## 5.3 En specifikt dokumenterad anpassad RUP för små projekt

Nedan presenteras de grundläggande principer och den grundläggande strukturen för en specifikt dokumenterad anpassad RUP. Syftet är att redogöra de karaktäristiska dragen som gör den specifikt dokumenterade anpassade RUP lämplig för små projekt.

### 5.3.1 Principer och antaganden för ett litet RUP-projekt

RUP<sup>®</sup>, Rational Unified Process, i sin ursprungliga form omfattar olika delar som kan plockas ut för varje projekt efter behoven och storleken på projektet (Hirsch, 2003). Det finns sex principer, som skall följas av mjukvarumetoder. De sex principerna presenteras av Kruchten (2000) och Hirsch (2003). Alla de principerna används i RUP<sup>®</sup> och i den specifikt dokumenterade anpassade RUP. De sex principerna fungerar mycket bra i praktiken (Hirsch, 2003). Principerna som stöder de aspekter som presenteras i kapitel 5.1 för små projekt presenteras nedan:

Iterativ utveckling. Systemet växer allt eftersom genom iterationer och att inget arbete i ett projekt genomförs utan anledning (Hirsch,

## 5. Genomförande

2003). Varje iteration lägger till en funktion till systemet. Hirsch (2003) påpekar att varje projekt bör omfatta mellan sex och tolv iterationer, där varje iteration i små projekt genomförs på cirka två veckor. Kruchten (2000) anser att utvecklingen i den traditionella vattenfallsmodellen är linjärt, vilket enligt Andersen (1994) leder till att modellen bara passar utvecklingsprojekt som saknar förändringar i bland annat ställda kraven från användarna på systemet. Därför anses en iterativ utveckling vara ett bättre alternativ till den traditionella vattenfallsmodellen när kraven, som ställs från användarna på systemet, förändras under projektets gång. Hirsch (2003) påpekar att feedback fås snabbt vid en iterativ utvecklingsmetod. På grund av tidiga feedback finns det möjlighet till utvecklarna att upptäcka tidigt arkitekturens och designens problem (Hirsch, 2003). En iterativ utveckling anses vara en aspekt för små projekt.

Kravhantering. Kraven på systemet är viktiga då de måste uppfyllas under projektets gång (Hirsch, 2003). Hirsch (2003) anser att dålig kravhantering leder till misslyckade projekt. Vidare påpekar han att kravhanteringen handlar om att kraven, som projektet ska omfatta, definieras ordentligt och att en kund deltar i projektet. En av aspekterna för små projekt syftar till att en kund ska delta i projektet för att presentera kraven.

Kontrollera projektets kvalitet. Detta innebär att kontinuerliga tester görs under projektets gång och ger feedback. Detta görs för att upptäcka misstag tidigt. Då kostar det mindre att åtgärda misstaget (Hirsch, 2003). Testning som ger feedback anses vara en aspekt för små projekt.

En ytterligare princip som hör hitt är: *Hantera förändringar av projektet.* Hirsch (2003) påpekar att denna princip syftar till att projektgruppen ska ha kontroll i ett systemutvecklingsprojekt, vilket innebär bland annat att rutinerna skall följas. Det är viktigt att klarlägga tidigt i projektet vad som förändrar vad, när och hur för att skapa rutiner som ska följas (Hirsch, 2003). Med andra ord hanteras förändringarna i ett litet RUP-projekt genom att klarlägga tidigt i projektet vad som förändrar vad. Det vill säga att den specifikt dokumenterade anpassade RUP hanterar förändringar men inte genom att omstrukturera projektets innehåll, utan genom att undvika förändringar.

Förutom de principerna presenterar Hirsch (2003) ett antal antagande för den specifikt dokumenterade anpassade RUP för små projekt. Två av antaganden stödjer aspekterna för små projekt för att de syftar till att en kund involveras aktivt i projektet och att leverera ofta systemet till användarna under projektets gång. Kunden ska involveras i projektet för att skriva tester, hantera kraven aktivt och snabbt (Hirsch, 2003). Kunden är också viktig för att snabbt kunna avgöra vad som eventuellt ska förändras och vad som är bra. Hirsch (2003) påpekar också att systemet ska levereras till användarna ofta för att projektdeltagarna ska kunna få feedback som är viktigt för att skapa en iterativ utvecklingsmetod. Vidare påpekar han att det är viktigt att behålla kvalitén när

## 5. Genomförande

produktiviteten är hög för att undvika förändringar och misslyckande som kan kräva mera tid i framtiden att behandla.

### 5.3.2 Arbetsprocessen för ett litet RUP-projekt

Rollerna för ett projekt som använder den specifikt dokumenterade anpassade RUP fördelas på ett begränsat antal deltagare (Hirsch, 2003). Hirsch (2003) påpekar att en roll kan utföras av flera personer och en person kan utföra flera roller. Enligt Hirsch (2003) leder få projektdeltagare till ett effektivare systemutvecklingsprojekt då det är lätt för ett fåtal gruppmedlemmar att följa projektplanen. Vidare påpekar Hirsch (2003) att projektdeltagarna ska vara duktiga, skickliga och motiverade för att klara det som skall göras under projektets gång. Denna specifikt dokumenterade anpassade RUP omfattar tre kontrakt som görs i samarbete med kunden och som presenterar fyra faser för projektet. Hirsch (2003) påpekar att i små projekt brukar oftast påbörjande- och utvecklingskontraktet slås ihop. De tre kontrakten är för att öka kunskapen om kraven och tekniska lösningar. Nedan presenteras de tre kontrakten och på vilket sätt de stöder aspekterna som presenteras i kapitel 5.1, enligt Hirsch (2003):

Påbörjandekontraktet presenterar en viktig fas, som är påbörjandefasen. I påbörjandefasen diskuteras mellan projektets olika intressenter vad som skall ingå respektive inte ingå i projektet. Diskussionen leder till att en hel bild på omfattningen för utvecklingsprojektet fås. Vidare definieras visionen och nyttan för produkten samt kraven från användarna på systemet för att undvika och hantera konflikter. Planen för projektets olika delar och primära prototyper skapas utifrån de prioriterade kraven för att kunna förutsäga riskerna och behoven under projektets gång. I detta kontrakt är för att förutsäga behovet av tid och material för projektet med avseende på vad kunden har för budgetgräns. Att förutsäga tidigt resursbehovet för projektet anses vara en aspekt för små projekt. Denna fas genomförs på två veckor för små projekt.

Utvecklingskontraktet presenterar utvecklingsfasen, som ser till att ha stabila planer för projektet och minskar riskerna. Detta sker genom djupare analys av problemen och genom att skapa en grundläggande och stabil arkitektur, vilket är grunden för en säker bedömning av resursförbrukning, kostnader och tiden under projektets gång. Systemet ska levereras till användarna varje månad, vilket betonar att systemet ska levereras ofta till användarna för att få feedback. En av de aspekterna som anses vara för små projekt syftar till att systemet ska levereras ofta till användarna. Denna fas genomförs på minst en månad och högst sex månader.

Konstruktions- och övergångskontraktet presenterar både konstruktions- och övergångsfasen. Detta kontrakt omfattar styrning av processen för att fastställa kostnader och planer på optimal nivå. För att kunna fastställa material och tiden för projektet ska kraven vara stabila och tydliga. Kunden deltar i alla möten och alla iterationsplaner. Systemet levereras också ofta till användarna under denna fas av projektet. Sammanfattningsvis deltar en kund aktivt under denna fas av projektet och systemet levereras ofta till användarna, vilka anses

## 5. Genomförande

vara aspekter för små projekt. I slutet av denna fas överförs systemet till användarna i verksamheten för användning. Detta sker så fort produkten anses vara färdig.

Förutom kontrakten innehåller den specifikt dokumenterade anpassade RUP ett antal artefakter (Hirsch, 2003). Artefakter är inte slutprodukter, utan de produceras och användas under ett projekt (Kruchten, 2003). De används för att styra informationen och de kan utgöras av ett dokument, som kallas för Business Case, eller i form av en modell, som kallas för Use-Casemodell. Hirsch (2003) påpekar att artefakter också kan vara riktlinjer, design och källkod som i sin tur presenteras också i dokument. Arbetsflöden, som beskriver vad som ska göras, för den specifikt dokumenterade anpassade RUP är åtta av de nio som avses för RUP® (Hirsch, 2003). Nedan presenteras de åtta arbetsflöden och vad de har för roll, enligt Hirsch (2003):

Krav har som syfte att få alla intressenter att vara överens om vad systemet skall göra. Därför anses god kommunikation mellan de olika projektdeltagarna vara viktigt. Systemets gränser, beräkningar av kostnader och tidsåtgången för projektet diskuteras.

Analys och design har som mål att föra alla kraven som uppstår till det blivande systemet. Med andra ord utreder systemutvecklaren hur kraven ska implementeras i detta arbetsflöde. Här utvecklas en robust arkitektur för systemet.

Implementation definierar hur källkoden, som systemet omfattar, organiseras. Komponenterna byggs av klasser och objekt. Komponenterna sätts samman till ett exekverbart system.

Test kontrollerar att komponenterna är integrerat på rätt sätt och att alla krav har blivit implementerade och att alla objekt fungerar som de ska. Tester ger feedback.

Driftsättning säkerställer att slutprodukten blir tillgänglig för dess slutanvändare. Detta innebär bland annat att installationen av programvaran kontrolleras och att slutanvändarna ska träna programvaran.

Konfigurations- och förändringsstyrning syftar till att kontrollera versioner av dokument och kod, samt förändringar i kravbilden. Arbetsflödet löper genom hela projektets gång.

Projektstyrning syftar till att övervaka och kontrollera projektet, att sköta utvecklingsplanering och att projektet slutförs. Varje projektdeltagare ska veta vad den ska göra. Möjligheter och risker utvärderas och den iterativa principen följs.

Förhållanden hanterar anpassningen av processen i verksamheten. Syftet är att skapa en god utvecklingsmiljö för att stödja projektdeltagarna.



## 5. Genomförande

Ett arbetsflöde resulterar i ett antal artefakter och är en samling av aktiviteter (Kruchten, 2002). Hirsch (2003) presenterar ett antal nödvändiga och ett antal alternativa artefakter för varje arbetsflöde. Anledningen är att bara det som avses viktigt för varje projekt skall göras, vilket anses vara en aspekt för små projekt.

### 5.3.3 Sammanfattning av de karaktäristiska dragen som gör den anpassade RUP lämplig för små projekt

Utifrån de som presenterar ovan i detta kapitel om den specifikt dokumenterade anpassade RUP, anses de som gör den specifikt dokumenterade anpassade RUP lämplig för små projekt är följande:

- Kontinuerlig leverans av systemet till användarna för testning, vilket ger feedback.
- Den specifikt dokumenterade anpassade RUP ger möjlighet till god projektstyrning i ett litet projekt, vilket leder till att alla projektdeltagarna vet vad de ska göra.
- Den specifikt dokumenterade anpassade RUP ger möjlighet till att hantera förändringar genom att klarlägga vad som förändrar vad, när och hur.
- En kund involveras aktivt för att projektdeltagarna ska kunna hantera kraven aktivt.
- En projektdeltagare kan ha flera roller.
- Diskussioner och samarbete mellan deltagarna och kunden, vilket leder till god kommunikation.
- Fokus på att förutsäga resurserna som ska förbrukas under projektet gång.
- Väl definierade riktlinjer och information om de olika delarna i projektet genom artefakter, vilket minskar missförstånd mellan projektdeltagarna.
- Bara det som anses viktigt för att få ett lyckat projekt görs.

De karaktäristiska dragen, som presenteras ovan, kan jämföras med aspekterna för små projekt, som presenteras i kapitel 5.1. Utifrån jämförelsen konstateras att de karaktäristiska dragen följer en stor del av aspekterna för små projekt.

## 6 Erfarenheter från metodtillämpningar

### 6.1 Erfarenheter kring XP för små projekt

Nedan presenteras erfarenheter från införandet av XP och den anpassade RUP för små projekt från litteratur och intervjuer med två intervjupersoner. Syftet är att reda ut ur praktiska erfarenheter det som styrker hur XP och den anpassade RUP är lämpliga för små projekt.

Abrahamsson m.fl. (2002) och Beck (1999) förklarar att ur praktiska erfarenheter om XP är god kommunikation mellan deltagarna i små XP-projektet viktigt, där alla deltagarna ska utnyttja varandras kunskap. Vidare påpekar Abrahamsson m.fl. (2002) ur praktiska erfarenheter om XP att ett XP-projekt ska omfatta mellan tre och högst tjugo personer för att en god kommunikation ska kunna säkerställas. De förklarar vidare att motståndet mellan deltagarna, kunden och styrelse hanterades genom god kommunikation. Beck (1999) påpekar ur praktiska erfarenheter om XP att i XP-projekt följdes XP:s principer utan några problem, vilket säkerställde ett lyckat projekt. Scott (2002) påpekar att XP-projekt inte behöver omfatta mer än tolv deltagare. Anledningen var att när projektet omfattar få projektdeltagare blir det lättare att hålla reda på vem som gör vad. Att reda ut vem som gör vad anses vara en aspekt för små projekt. Vidare förklarar Scott (2002) ur praktiska erfarenheter om XP att är bara det en tillgänglig kund som deltar i ett XP-projekt. Han menar att det inte går att öka antal kunder hur som helst för att det kan leda till konflikter mellan dem.

Abrahamsson m.fl. (2002) påpekar att i genomförda små XP-projekt är det nästan ingen uppmärksamhet på projektstyrning. De individuella principer, som presenteras i kapitel 5.2.1, tar uppmärksamhet i XP (Abrahamsson m.fl., 2002). Orsaken är att det finns behov av att fokusera på projektgruppen som har mycket att göra och som ska effektivisera arbetet i projektet.

Beck och Fowler (2001) förklarar ur praktiska erfarenheter om XP att varje iteration i ett litet XP-projekt för det oftast genomförs på två veckor. Orsaken till att försöka genomföra iterationer på så kort tid så möjligt är att ha kontroll över projektet. Korta iterationer som ger feedback är kännetecknen på framgång eftersom de visar hur långt deltagarna har kommit i projektet (Beck och Fowler, 2001). Beck och Fowler (2001) påpekar också att iterationer i slutändan ledde till iterativ utveckling. Tester och feedback som leder till iterativ utveckling, anses vara en aspekt för små projekt.

Karlström (2002) skriver om erfarenheter kring små XP-projekt. Han påstår att så länge det var ett litet projekt med få antal projektdeltagare fungerade projektstyrningen bra. Orsaken var att det var lätt att hålla en liten grupp i en och samma lokal. Karlström (2002) menar att risken av att projektdeltagarna skriver över varandras kod och att de missförstår varandra var liten vid ett litet XP-projekt. God kommunikation spelade en stor roll under iterationer och överlämning av delsystem till kunden (Karlström, 2002). God kommunikation försäkrade att projektet inte fortsätter i fel riktning. Karlström (2002) påpekar vidare att enkelheten, som kännetecknar ett litet XP-projekt, ledde till effektivitet. Onödig kod implementerades inte i XP-projekt, vilket var för att bara det som måste bli gjort gjordes. Karlström (2002) anser att deltagandet av en kund var viktigt för att projektgruppen ska kunna

## 6. Genomförande

snabbt modifiera och lägga till de delsystem som överlämnas senare till kunden och användarna. På det viset fick projekt-deltagarna snabbt och tidigt reda på exakt vad kunden ville utan missförstånd, vilket ledde till att tiden som var avsedd för projektet minskade.

Griffin (2002) skriver om erfarenheter kring XP för små projekt. Hon påstår att det genomförda XP-projektet gav ett positivt resultat, där kundens krav uppfylldes. Vidare förklarar hon att XP gav möjlighet till kunden, som deltar i projektet genom bland annat att vara med i dagliga möten, att känna sig aktiv och meningsfull under projektets gång. Det var viktigt med aktivt kunddeltagandet för att det är kunden som presenterar kraven som ställs från användarna på systemet. Griffin (2002) påstår ur praktiska erfarenheter om XP att ett lyckat XP-projekt kännetecknas med att projekt-deltagarna ska dela med sig och med kunden viktig information och kunskap. Projektdeltagarna ska också hantera förändringar och samarbeta med kunden under hela projektet.

Reifer (2002) redovisar i en artikel positiva erfarenheter kring dynamiska metoder, särskilt XP, i åtta företag. De projekten, som denna artikel behandlar, genomfördes på mindre än ett år och antalet projektdeltagarna var färre än tio. Sammanfattningsvis förklarar Reifer (2002) att XP-projekten karakteriserades av att projektgruppen var motiverad, kraven var tydliga, arkitekturen var stabil och det var hög grad av utvecklingsflexibilitet på grund av att kunden, som deltog i projektet, hade tid för projektets arbete.

### **6.2 Erfarenheter kring den specifikt dokumenterade anpassade RUP för små projekt**

Nedan presenteras erfarenheter om införandet av den anpassade RUP för små projekt. Ur praktiska erfarenheter redas ut det som styrker hur lämplig den specifika anpassade RUP för små projekt.

Hirsch (2000) har tillämpat den specifika anpassade RUP för ett projekt som omfattade mellan tre till åtta deltagare. Projektplanen för detta projekt kännetecknades med att det var fokus på resultatet mer än på uppgifterna, att artefakterna definierades väl och att iterationsplanen var detaljerad. Detta var anledningen till att alla projekt-deltagarna visste vad de skulle göra. I detta projekt var det möten varje vecka och iterationerna var en månad långa för att kunna behålla kontrollen över hela projektets omfattning. Projektgruppen omfattade erfarna utvecklare och en kund som involverades tidigt i projektet. Deltagarna fick feedback ofta från användarna och från kunden under projektets gång. Detta projekt kännetecknades med feedback från användarna för att delsystem levererades ofta. Detta projekt genomfördes på två månader kortare än denna tid som var avsedd för det. Resultatet för detta projekt var positivt för att alla kraven som ställdes av kunden blev uppfyllda (Hirsch, 2000).

Hirsch (2000) tillämpade den specifika anpassade RUP i ett annat litet projekt. I detta projekt saknades feedback från kunden. Kunden var hela tiden upptagen med annat och inte med projektets uppgifter. Projektet misslyckades på grund av mindre feedback från kunden, vilket ledde senare till motstånd från användarna. Det är kunden som är representant för organisationen och om han inte ger feedback kommer viktig

## 6. Genomförande

information för projektet saknas. Informationen kan tala om vad användaren önskar sig. Därför anses feedback vara viktigt för att minska motstånd från användarna.

Hirsch (2000) påpekar att ur praktiska erfarenheter om den specifika anpassade RUP för små projekt att det är viktigt att involvera en kund i projektet för att hantera oklara och förändrade krav (Hirsch, 2000). Kunden måste ha tid för projektet och måste vara tillgänglig. Med andra ord lönar det sig inte att använda den specifika anpassade RUP för mindre projekt om kunden inte har tillräckligt med tid för utvecklingsprojektet. Hirsch (2000) påpekar vidare att motiverade och ansvarsfulla systemutvecklare är en nödvändig förutsättning för den specifika anpassade RUP för att kunna klara alla systemutvecklingsuppgifterna i ett litet projekt. Vidare förklarar han att systematiska kontinuerliga tester på systemet, som ger feedback kontinuerligt och som behövs för att projektdeltagarna ska kunna få reda på om de är på rätt riktning tidigt, saknades i små RUP-projekt. Mindre feedback ledde till ineffektivitet i projektet.

### 6.3 Intervjuer

Nedan presenteras de genomförda intervjuer för detta examensarbete. För att få tydligare information om hur de två dynamiska metoderna är lämpliga för små projekt genomförs telefonintervjuer med två personer som är erfarna i XP och en anpassade RUP.

#### 6.3.1 Förberedelser inför intervjuerna

Problempreciseringen för detta examensarbete besvaras genom bland annat intervjuer. Intervjupersoner är systemutvecklare för att de är mest relevanta av projektgruppen, som deltar i mer än ett projekt. Flera företag inom IT-branschen i Stockholm och Göteborg kontaktades per telefon för att kunna få tag på relevanta intervjupersoner. Utifrån ett antal företag valdes två som tillämpar de två dynamiska metoderna genom subjektivt urval. Subjektivt urval innebär att intervjuaren själv väljer ut vilka individer som skall ingå i urvalet utifrån sin egen bedömning (Hellevik, 1987). Intervjupersonerna, som valdes för detta examensarbete, ansågs vara insatta i ämnet och hade erfarenhet om en anpassad RUP och XP för små projekt. De två intervjupersonerna som blev aktuella för detta examensarbete kontaktades senare via e-brev. Problemformulering för detta examensarbete och frågorna skickades till intervjupersonerna innan telefonintervjun. Syftet med att skicka frågorna var att ge en bild till intervjupersonerna om vad telefonintervjun kommer att behandla.

Intervjufrågorna som blev aktuella för detta examensarbete är slutna frågor. Slutna frågor blev relevanta för att undvika fokus på icke relevanta fakta. Aspekterna för små projekt (diskuterad i kapitel 5.1) blev ett underlag för frågorna som ställdes för att hitta erfarenheter som styrker vad som står i metodlitteraturen om de två dynamiska metodernas lämplighet för små projekt. Intervjufrågorna är uppdelade till två delar. Den ena delen behandlar XP och den andra delen behandlar de anpassade RUP (se bilaga 1 och 2). Orsaken till uppdelningen av frågorna är att det var två intervjuer om två olika dynamiska metoder som genomfördes. Varje del av de intervjufrågorna var uppdelat i tre frågenivåer som är inledande frågor, huvud- och slutfrågor. Följdfrågorna var inspirerade av svaren från intervjupersonerna på huvudfrågorna. Följdfrågorna dök upp under genomförandet av intervjuerna.

## 6. Genomförande

Inledande frågorna för både intervjuerna (se bilaga 1 och 2) var för att få fakta om intervjupersonen och dennes erfarenhet om ämnet. Dessutom omfattade inledande frågor för den anpassade RUP:s intervjun en ytterligare fråga som behandlar likheter och olikheter mellan detta sättet intervjupersonen anpassar RUP för små projekt och den specifikt dokumenterade anpassade Hirsch:s RUP för små projekt (se bilaga 2). Det var inte lätt att hitta ett företag som tillämpar den specifikt dokumenterade anpassningen av RUP i regionen. För att kunna avgöra i hur stor grad de två anpassningar av RUP är lika blev den tredje frågan i intervjun för den anpassad RUP aktuell. Vidare avgjordes om intervjupersonens anpassning av RUP är relevant för detta examensarbete eller inte.

Huvudfrågorna för XP:s och RUP:s intervjun blev lika i antal och behandlade i stort sett samma sak (se bilaga 1 och 2). Huvudfrågorna behandlade hur de två dynamiska metoderna följer aspekterna för små projekt när de genomfördes i små projekt. Orsaken till att ta upp de frågorna var att reda ut på djupt erfarenheter som styrker argumenten från metodlitteraturen avseende de två dynamiska metodens lämplighet för små projekt.

Den första slutfråga (se bilaga 1 och 2) gav möjlighet till intervjupersonen att sammanfattningsvis förklarar ur dennes erfarenhet andra karaktäristiska drag som gör den dynamiska metoden lämplig för små projekt. Denna fråga kan ge möjlighet till att upptäcka andra karaktäristiska drag som litteraturen kanske inte tar upp. Den andra frågan utredde vad som är viktigt vid tillämpning av de två dynamiska metoderna i ett litet projekt. Sista frågan utredde i hur stor grad lyckades de två dynamiska metoderna att stödja ett lyckat litet projekt.

### 6.3.2 Resultat av intervjun om XP för små projekt

Intervjupersonen som valdes för denna intervju arbetar på Konkret IT. Konkret IT är ett konsultföretag för bland annat systemutveckling och mjukvaruutveckling. Intervjupersonen har arbetat med XP för små projekt i två år. Sammanfattning av svaren på huvud- och slutfrågorna, som ställdes till intervjupersonen om praktiska erfarenheter om de karaktäristiska dragen som gör XP lämplig för små projekt, presenteras nedan i form av en punktlista:

- Feedback fås ofta under ett litet XP-projekt eftersom många delsystem levereras ofta. De är viktiga för att de styr projektet i rätt riktning. Nackdelen med för ofta feedback är det kan resultera i mycket arbete att göra, vilket i sin tur kan leda till att projektdeltagarna tappar fokus på viktiga saker. Fördelen med för ofta feedback är att projektdeltagarna får möjlighet att reda ut felen i rätt tid.
- Kunden är en viktig deltagare i ett litet XP-projekt. En kund i projektet är fördelaktigt för att den ger möjlighet till att användarna får det som är tänkt att få. Kunden är en länk mellan utvecklarna och användarna. Kunden bestämmer vad som skall göras eftersom den presenterar vad som efterfrågas av användarna. Kunden arbetar heltid i projektet och definierar delmomenten i utvecklingen. Det är viktigt att kunden är kunnig, har tid för

## 6. Genomförande

projektet och är motiverat för ett litet XP-projekt. Nackdelen med att en kund deltar aktivt i ett litet XP-projekt är att kunden kan fokusera sig inte bara på verksamhetskraven utan också på tekniken som används i projektet. Med andra ord fokuserar kunden på andra saker mer än att fokusera på kraven som spelar en stor roll för att få ett lyckat litet XP-projekt. Först och främst ska kunden fokusera ordentligt på att presentera kraven tydligt.

- Öga mot öga kommunikationen och att sitta i samma arbetsrum är avgörande för ett lyckat litet XP-projekt. Små projektgrupper kan lättare kommunicera i ett och samma rum. Öga mot öga kommunikation medför direkt kommunikation med berörda personer, vilket minskar risken för missförstånd och skapar effektivitet i projektet. Öga mot öga kommunikation är viktigt vid dagliga möten, som är för att diskutera vad som skulle göras på samma dag och planeras schemat för projektet. Möten är viktiga för att avgöra vart projektet befinner sig. Öga mot öga kommunikation ersätter dokumentation för att den leder till god kommunikation mellan projektdeltagarna. Projektdeltagaren kan uppleva ibland att det är svårt att koncentrera sig på det den gör på grund av alla projektdeltagarna arbetar i samma arbetsrum.
- XP tillåter att en deltagare får ha flera roller i ett litet XP-projekt. Fördelarna med att en deltagare kan ha flera roller är att arbetet blir effektivare, projektdeltagaren har alltid något att göra och projektets olika behov uppfylls. Däremot är nackdelen att deltagarna kan lätt känna sig splittrat mellan de olika rollerna.
- Kraven, som ställs från användarna och presenteras i berättelser, avgör vad projektet kommer att omfatta. Tiden, kostnader och vad tekniken klarar av är tre faktorer som avgör vad som skall göras i ett litet XP-projekt. Det skiljer sig inte mycket mellan hur projektdeltagarna avgör vad som skall göras för ett litet och för ett stort projekt. Det är samma faktorer som avgör vad som skall göras för både små och stora projekt.
- Förändringar i ett litet XP-projekt hanteras väl genom att alltid programmera i par. När programmering sker två och två blir det lättare att inte tappa nödvändig information om en av deltagarna försvinner. Kraven förändras snabbt i ett litet XP-projekt. Kraven, som ska uppfyllas under projektets gång, dokumenteras med hänsyn till att ett litet XP-projekt ska omfatta lagom mängd arbete för att behålla kvalitet och motivation. Mängden av systemet som levereras till användarna för test och för att projektdeltagarna ska kunna få feedback ska vara lagom mycket. XP egentligen anses vara för små system som utvecklas av små projektgrupper, där förändringar studeras väl innan projektledaren och kunden beslutar om att de skall hanteras eller inte. Därefter omstruktureras projektets innehåll och designen om förändringarna ska hanteras.

## 6. Genomförande

- XP tillämpas med ett programmeringsspråk i ett litet projekt men den kan tillämpas med flera programmeringsspråk i ett litet projekt för att hitta flera lösningsförslag. Om ett programmeringsspråk används blir det bättre kommunikation mellan deltagarna och effektiviteten ökar i projektet. Flera programmeringsspråk ger dock möjlighet att bättre lösa uppgifterna eftersom möjligheterna då är större.
- Det skiljer sig inte mycket från ett litet XP-projekt och andra projekt när det handlar om att förutsäga resurserna som kommer att användas under projektets gång. XP omfattar en princip för hantering av resursbehovet, som kan vara i form av tid och personal, för små projekt. Principen bygger på att utvärdera det som är genomfört inom en vecka i poäng. Om poängen är låg förutsägs hur mycket resurser som behövs mera i den närmaste tiden för att förbättra arbetsprocessen.

Slutligen påpekar intervjupersonen att kontinuerliga tester på system, som är under hela projektets gång, är ett karakteristiskt drag som gör XP lämplig för små projekt. Intervjupersonen anser att XP är absolut en fördel för små projekt och stödjer små projekt i högsta grad eftersom den är väldokumenterad.

### 6.3.3 Resultat av intervjun om en anpassad RUP för små projekt

Intervjupersonen som telefonintervjuades har cirka två års erfarenhet om små RUP-projekt. Intervjupersonen jobbar i Lindysoft, som är ett konsultföretag för bland annat systemutveckling. Intervjupersonen påstår att skillnaden mellan den anpassade RUP han tillämpade för små projekt och den specifikt dokumenterade anpassade RUP, som Hirsch presenterar, är inte så stor. Skillnaden är att den RUP för små projekt som han använder inte är lika väldokumenterad som Hirsch:s. Intervjupersonen känner igen innehållet som Hirsch presenterar. Han påstår att en stor del av principerna och innehållet av vad Hirsch presenterar följs i hans små RUP-projekt men de är inte lika tvingande som Hirsch:s dokumentation. Intervjupersonen sa att RUP som han använder för små projekt fokuserar sig på användningsfallen som presenterar kraven. Därefter sker modelleringen efter dessa användningsfall. Senare framställs riskhanteringsplan och designen, som är efter användningsfallen. Slutligen förutsägs tidsresurser och kostnader för projektet. Nedan presenteras vad intervjupersonen påpekade ur praktiska erfarenheter kring den anpassade RUP för små projekt. De som presenteras nedan är svaren från intervjupersonen på huvud- och slutfrågor. Presentationen sker i form av en punktlista:

- Testerna är viktiga och spelar en stor roll för att få feedback under projektets gång. Intervjupersonen förklarade att han deltog i ett litet RUP-projekt som misslyckades tack vare brist på feedback. Feedback är viktigt för att undvika stress och problem vid slutet av projektet. Det finns inga nackdelar med för ofta feedback under ett litet RUP-projekt. För ofta leverans av delsystemet till användarna, som ger feedback, under ett litet RUP-projekt är fördelaktigt.
- Det måste finnas ett aktivt kunddeltagande under ett litet RUP-projekt. Kunden ska vara tillgänglig hela tiden för att försäkra att

## 6. Genomförande

projektdeltagarna får de viktiga feedback, som behövs för att reda ut om projektet är i rätt riktning.

- God kommunikationen mellan projektdeltagarna anses mest viktigast för att de ska kunna dela med sig information som inte dokumenteras och för att förmedla information vidare till de eventuella nya deltagarna i projektet. Projektdeltagarna ska vara duktiga och ska arbeta i samma arbetsrum. Därför anses öga mot öga kommunikation vara viktigt under ett litet RUP-projekt. Orsaken är att ett litet projekt bygger på att det ska vara enkelt att bland annat ställa frågor och hålla samman i projektgruppen. Det finns inga nackdelar med öga mot öga kommunikation för små RUP-projekt.
- En projektdeltagare kan få flera roller i ett litet RUP-projekt för att kunna klara projektets olika behov. Det är viktigt att ha duktiga systemutvecklare vid tillämpning av en anpassade RUP för små projekt. Nackdelen med att en projektdeltagare får ha flera roller i ett litet RUP-projekt är att det blir lätt för projektdeltagarna att fokusera på rollerna och inte på aktiviteterna. En ytterligare nackdel är att projektdeltagaren fokuserar sig mindre på att försöka göra allt, den gör bara sitt.
- Användningsfallet, där presenteras kraven som ställs från användarna, avgör vad som ska göras i projektet. Förändringar av kraven presenteras av kunden och hanteras av projektledare och arkitekten. Det är de som bestämmer om förändringen är möjlig.
- I praktiken användes bara ett programmeringsspråk under ett litet RUP-projekt men det är inte omöjligt att använda flera programmeringsspråk. Orsaken till att ett programmeringsspråk användes är inte att skapa effektivitet i projektet utan var att det inte behövdes ha flera programmeringsspråk. Schemat och projektet planeras i ett litet RUP-projekt. Uppdatering av projektplanen sker vid varje möte, som äger rum varje vecka under projektets gång. Enkelheten för ett litet RUP-projekt är förknippat med att göra färre aktiviteter och inte med att göra lätta aktiviteter.
- Intervjupersonen påpekade också att det är lika jobbigt för alla projekt att förutsäga förbrukning av tiden och behovet av projektdeltagarna. I små RUP-projekt förutsägs resurserna bara en gång. I den anpassade RUP finns det möjlighet för projektdeltagarna att tydligt utreda vilka aktiviteter som ska utföras för varje iteration, vilket stödjer deltagarna att förutsäga resursförbrukning under projektet.

Han påpekade vidare att arbetsflöden, användningsfall och fokus på analys och design är karaktäristiska drag som gör den anpassade RUP lämplig för små projekt. Intervjupersonen tycker att den anpassade RUP har för lite fokus på programmeringen. Slutligen tyckte han att RUP kan ge ett lyckat resultat om den anpassas efter små projekt men det fungerar säkert bättre om ett litet projekt



genomförs med en systemutvecklingsmetod som är avsedd för små projekt från början. Orsaken är att det är svårt att anpassa RUP för ett begränsat projekt utan nackdelar. Nackdelarna beror på att kraven förändras under projektets gång samtidigt som kraven är stora på att systemet ska ha hög kvalitet.

### **6.4 Sammanfattning av erfarenheter kring de två metoderna för små projekt**

I detta avsnitt sammanfattas erfarenheterna som finns från metodtillämpningar med de två dynamiska metoderna som styrker argumenten från metodlitteraturen avseende dess lämplighet för små projekt. Syftet är att kombinera åsikterna från de olika källorna, som presenteras i kapitlen 6.1, 6.2 och 6.3, under ett avsnitt.

I ett XP-projekt är god kommunikation mellan deltagarna viktigt för ett lyckat projekt. God kommunikationen säkerställs tack vare få antal projektdeltagare som arbetar öga mot öga och som mötas dagligen. Projektdeltagarna delar med sig viktig information tack vare god kommunikation. Alla deltagarna i ett XP-projekt arbetar i ett arbetsrum, vilket ger möjlighet att reda ut vem som gör vad. En kund deltar aktivt i ett litet XP-projekt för att undvika missförstånd och skapa effektiviteten i projektet. Kunden ska ha tid för projektet för en lyckad tillämpning av XP för små projekt. Fokus ligger på projektdeltagarna och effektivitet i ett projekt. Ett XP-projekt kännetecknas med för korta iterationer och med för ofta leverans av systemet till användarna, vilket ger feedback från användarna till projektdeltagarna. En projektdeltagare i ett litet XP-projekt ska ha flera roller för att klara projektets olika behov. Ett litet XP-projekt måste inte ha bara ett programmeringsspråk utan projektet kan ge samma resultat som om flera programmeringsspråk används. Tester är viktiga under ett litet XP-projekt för att de ger snabb feedback som hjälper projektdeltagarna att avgöra om projektet är på rätt riktning. I XP utvärderas arbetet varje vecka och därefter avgörs resursbehovet för den närmaste tiden. Slutligen anses XP vara en lämplig systemutvecklingsmetod för små projekt.

I ett litet RUP-projekt är det viktigt att definiera artefakterna och planera iterationerna väl för att undvika missförstånd mellan deltagarna. En kund involveras aktivt i projektet och ser till att relevant feedback fås ofta under projektets gång. Systematiska kontinuerliga tester på systemet saknas i ett litet RUP-projekt. God kommunikation är viktig under ett litet RUP-projekt för att kunna skapa effektivitet. God kommunikation säkerställs genom öga mot öga kommunikation och genom att projektdeltagarna arbetar i samma arbetsrum. En projektdeltagare bör vara duktig och ha flera roller för att kunna klara projektets olika behov. Inga specifika principer följs i ett litet RUP-projekt för att skapa effektivitet. I ett litet RUP-projekt kan en eller flera programmeringsspråk användas men inte i syfte med att skapa effektivitet. I ett litet RUP-projekt aktiviteterna utredas väl för att senare ska projektdeltagarna kunna förutsäga resurserna som ska förbrukas under projektets gång. Under små RUP-projekt fokuserar projektdeltagarna på analys och design av användningsfallen som presenterar kraven som ställas från användarna på systemet. Alla små RUP-projekt lyckas inte på grund av bland annat brist på feedback och brist på aktivt kunddeltagande. Ett litet projekt lyckas bäst om systemutvecklingsmetoden som tillämpas är avsedd för små projekt från början.

# 7 Resultat

I detta avsnitt presenteras undersökningsresultatet. Egen reflektion på resultatet presenteras också i detta kapitel.

De karaktäristiska dragen som de två metoderna lämpliga för små projekt presenteras separat i kapitlen 5.2.3 och 5.3.3. De karaktäristiska dragen sammanfattas och presenteras nedan. De karaktäristiska dragen konstateras utifrån litteraturstudie. De är:

- Båda metoderna syftar till att systemet leveras kontinuerligt till användarna för testning, vilket ger feedback.
- Båda metoderna är iterativa utvecklingsmetoder.
- Båda metoderna syftar till att en kund deltar aktivt i projektet.
- Båda metoderna syftar till att det ska finnas möjlighet att tidigt och snabbt förutsäga resurserna som ska förbrukas under projektets gång.
- Projektets innehåll för ett XP-projekt omstruktureras snabbt om något förändras.
- Den specifikt dokumenterade anpassade RUP ger möjlighet till att hantera förändringar genom att klarlägga vad som förändrar vad, när och hur.
- Koden är tillgänglig för alla projektdeltagarna i ett litet XP-projekt, vilket anses vara en princip som leder till effektivitet.
- Bara ett programmeringsspråk som används under ett litet XP-projekt för att säkerställa god kommunikation och effektivitet under projektet.
- Fokus för båda metoderna ligger på att projektet ska kännetecknas med god kommunikation mellan alla projektdeltagarna genom öga mot öga kommunikation.
- Diskussioner och samarbete mellan deltagarna i ett litet RUP-projekt och kunden, vilket leder till god kommunikation.
- Båda metoderna syftar till att en projektdeltagare får ha flera roller.
- Båda metoderna syftar till att alla projektdeltagarna ska veta vad de ska göra och vem gör vad.
- Väl definierade riktlinjer och information om de olika delarna i ett litet RUP-projekt genom artefakter, vilket minskar missförstånd mellan projektdeltagarna.
- Båda metoderna syftar till att bara det som anses viktigt för att få ett lyckat projekt görs.

Resultatet redovisas genom att studera de aspekterna för små projekt, som presenteras i kapitel 5.1. Orsaken är att det visade sig likheter mellan de aspekterna för små projekt och de flesta karaktäristiska drag, som gör de två dynamiska metoderna lämpliga för små projekt. Med andra ord anses de aspekterna för små projekt vara egenskaper som metoderna har för att de ska vara lämpliga för ett litet projekt. Resultatet, som presenteras nedan, visar i vilken utsträckning de två metoderna kännetecknas av dessa egenskaper. Aspekterna för små projekt, som omfattar färre än

## 7. Resultat

tio deltagare och genomförs på mindre än ett år, och resultat av hur de två dynamiska metoderna förhåller sig till dem presenteras nedan i form av en punktlista. A står för den intervjuperson som telefonintervjuades för XP och B för den intervjuperson som telefonintervjuades för en anpassad RUP. Resultatet är:

- Ett litet RUP-projekt saknar i praktiken systematiska kontinuerliga tester på systemet, vilket leder till brister på en del feedback. När den anpassade RUP sker inga dagliga möten, utan möten sker varje vecka. Däremot i små XP-projekt fås dagligen feedback från kunder dagligen för att möten sker varje dag. Feedback ger möjlighet att upptäcka och lösa problemen tidigt och ger möjlighet till iterativ utveckling, vilket skapar effektivitet i ett litet projekt (Eckstein och Josuttis, 2002). XP ger möjlighet till deltagarna i projektet att kommunicera bättre genom kontinuerliga tester på systemet, vilket försäkras systemets kvalitet och ger viktig feedback från användarna (Avison och Fitzgerald, 2003; Jeffries, 2001). Genom feedback fås information från användarna, vilket hjälper projektdeltagarna att utreda var de befinner sig i utvecklingsarbetet. Mindre integrationer och tester leder till svårigheter att reda ut tidigt när felet uppstår var felet är i projektet (Jeffries, 2001). Utifrån praktiska erfarenheter om XP påpekade A under telefonintervjun att feedback fås för ofta under ett litet XP-projekt eftersom många delsystem levereras till användarna. Vidare påpekar han att feedback är viktiga för att de styr projektet i rätt riktning. Metoddokumentationen för den specifikt dokumenterade anpassade RUP syftar till att det ska existeras kontinuerliga tester på systemet under projektets gång, vilket ger feedback och leder till att upptäcka misstag tidigt (Hirsch, 2003). Ur praktiska erfarenheter om små RUP-projekt påpekar Hirsch (2000) och B att feedback är avgörande för ett lyckat litet RUP-projekt men en del feedbacks saknas.
- Både dynamiska metoderna anses vara för iterativ utveckling för att de syftar till att leverera system ofta till användarna för bland annat test. Leveranserna leder till feedback från användarna. Feedback är en vägledning för projektdeltagarna. Små projekt som genomförs på korta tidsperioder kännetecknas med tidiga och fleromgångar leveranser av systemet till användarna (The Standish Group, 1994). Fitzgerald m.fl. (2002) anser att i ett XP-projekt levereras delsystemen ofta till användarna. A ansåg under telefonintervjun att feedback fås tack vare att delsystemen levereras ofta till användarna. Systemet levereras ofta också när den specifikt dokumenterade anpassade RUP används (Hirsch, 2000). Även B ansåg att systemet levereras ofta under ett litet RUP-projekt.
- Båda dynamiska metoderna betonar behovet av en kund i projektet för att minska missförstånd av kraven. Erfarenheter kring de två dynamiska metoderna betonar behovet av en kund i projektet. En kund ska involveras i ett litet projekt hela tiden för att presentera organisationen och bestämma vilka krav som ska behandlas (Eckstein och Josuttis, 2002). Det är bara en kund, som presenterar

## 7. Resultat

kraven, deltar i ett litet XP-projekt (Jeffries, 2001). Det är samma för den specifikt dokumenterade anpassade RUP (Hirsch, 2003). Karlström (2002) påpekar utifrån hans erfarenhet att deltagandet av en kund i XP-projekt är viktigt för att modifiera och lägga till de delsystemen som överlämnades till den och användarna. Utifrån Hirsch:s (2000) erfarenheter kring små RUP-projekt konstateras att om kunden inte deltar aktivt projektet misslyckas projektet.

- Båda dynamiska metoderna betonar behovet av god kommunikation mellan deltagarna i ett litet projekt. Det är viktigt när projektet är litet med få antal deltagare att det finns möjlighet för projektdeltagarna att ha god kommunikation emellan (Cockburn och Highsmith, 2002). Erfarenheter kring de två dynamiska metoderna visar att öga mot öga kommunikation är viktigt. God kommunikation fås genom öga mot öga kommunikation, vilket leder till att projektdeltagarna kommunicerar och samarbetar i ett rum. I Hirsch:s dokumentation om den specifikt dokumenterade anpassade RUP nämns inget om öga mot öga kommunikation men han påpekar att god kommunikation är viktigt. Projektgruppen i ett litet XP-projekt ska sitta i samma rum (Pollice, 2001b). Ur praktiska erfarenheter om XP påpekade A under telefonintervjun att projektgruppen för de genomförda XP-projekt, där han involverades, satt i ett arbetsrum. B påpekade under telefonintervjun att vid små RUP-projekt sitter projektgruppen i ett och samma rum.
- Metoddokumentationen för de två dynamiska metoderna och erfarenheter kring dem betonar vikten av att deltagarna ska ha flera roller, vilket är anledning till mindre behov av stora projektgrupper. En deltagare i små projekt ska ha flera roller för att uppfylla projektets behov (Craig och Jassim, 1995). I ett XP-projekt får en projektdeltagare ha flera roller (Jeffries, 2001). A påpekade under telefonintervjun ur praktiska erfarenheter om XP att arbetet i ett litet XP-projekt är effektivt tack vare att en projektdeltagare har flera roller. För den specifikt dokumenterade anpassade RUP får varje projektdeltagare också ha flera roller. Ur praktiska erfarenheter om den anpassade RUP påpekade B vid telefonintervjun att varje projektdeltagare i ett litet RUP-projekt måste ha flera roller för att kunna klara projektets olika behov.
- Båda metoderna betonar vikten av att projektet ska vara enkelt och att bara det som måste göras görs. Varje input och output i ett litet projekt ska definieras tydligt och bara det som måste göras görs (Craig och Jassim, 1995). XP bygger på enkelhet som syftar på att bara det som behövs görs (Fowler, 2002). Även Jeffries (2001) anser att designen och aktiviteter för ett litet XP-projekt ska vara enkla. Input för projektet anses vara kraven som ställs. Kraven, som ställs av användarna, i ett XP-projekt presenteras i så kallade kundens berättelser (Beck, 1999). I praktiken sparar XP mycket tid och kraft tack vare enkelhet. A påpekade under telefonintervjun ur praktiska erfarenheter om XP att kunden definierar vad som ska göras vid ett litet XP-projekt. För den specifikt dokumenterade

## 7. Resultat

anpassade RUP är det arbetsflöden som leder projektet och beskriver vad som ska göras (Hirsch, 2003). B ansåg under telefonintervjun att bearbetning av användningsfallet, som definierar kraven och vad som ska göras, och arbetet genom arbetsflöden är karaktäristiska drag som gör den anpassade RUP lämplig för små projekt. B ansåg vidare att enkelheten för ett litet RUP-projekt är förknippad med att göra färre aktiviteter och inte med att göra enkla aktiviteter. Det förutnämnd tycks vara ett bevis på att de två dynamiska metoderna skiljer sig ifrån varandra för att XP syftar till att bara göra enkla aktiviteter.

- Förändringarna, som dyker upp under projektets gång, hanteras i de två dynamiska metoderna men på två olika sätt. Innehållet för ett litet projekt måste kunna omstruktureras för att kunna hantera förändringar som dyker upp under projektets gång (Armano och Marchesi, 2000). XP omfattar en princip som syftar till att omstrukturera projektets innehåll och designen när förändringar blir aktuella (Martin, 2002b). Ur praktiska erfarenheter om små XP-projekt påpekade A under telefonintervjun att projektets innehåll omstruktureras efter förändringar när projektledaren och kunden bestämmer att förändringarna ska hanteras. I den specifikt dokumenterade anpassningen av RUP hanteras förändringar genom att sätta rutiner som ska följas (Hirsch, 2003). Med andra ord nämns inte ordet omstrukturering i metoddokumentationen för den specifikt dokumenterade anpassade RUP. Det förutnämnd anses en ytterligare skillnad mellan de två dynamiska metoderna.
- XP har mer fokus på effektivt än vad den specifika dokumenterade anpassade RUP gör. Effektivitet är viktigt under ett tidsbegränsat projekt. Ett litet projekt ska omfatta verktyg som hjälper projektdeltagarna att skapa effektivitet i projektet för att kunna hinna med det som skall göras (Armano och Marchesi, 2000). XP omfattar en princip som syftar till att kodningen sker med ett programmeringsspråk för alla projektdeltagare och en annan princip som syftar till att koden ska vara tillgänglig för alla projektdeltagarna (Jeffries, 2001). De principerna leder till effektivitet. Även Smith (1999) påstår att deltagarna i små projekt ska koda standard med ett programmeringsspråk. A ansåg ur praktiska erfarenheter om XP att om flera lösningsförslag söks i ett litet XP-projekt kan flera programmeringsspråk användas. Inget anges i den specifikt dokumenterade anpassade RUP:s dokumentation om att det ska vara ett programmeringsspråk som skall användas i ett litet RUP-projekt. B påpekade att han använde bara ett programmeringsspråk för små projekt men inte i syftning på att skapa effektivitet. Det måste inte vara, som metoddokumentationen säger, ett programmeringsspråk för ett XP-projekt. Praktiska erfarenheter kring XP visar möjlighet till att ha flera programmeringsspråk.
- Metoddokumentationen för både metoderna betonar vikten av att förutsäga resursbehovet under ett litet projekt ofta och snabbt. I ett litet projekt resurserna, som kommer att utnyttjas under projektets

## 7. Resultat

gång, ska lätt kunna förutsägas (Smith, 1999). Det viktigt i ett litet XP-projekt att förutsäga tidigt resursförbrukningen genom principen planeringsknep, som syftar till att snabbt besluta omfattningen och resurserna för nästa delsystem (Fitzgerald m.fl., 2002). A påpekade ur praktiska erfarenheter om XP att i ett litet XP-projekt utvärderas arbetet som är gjort varje vecka och utifrån utvärderingen förutsägs resursbehovet för nästa vecka snabbt. Hirsch (2003) i hans dokumentation, om den specifikt dokumenterade anpassade RUP, fokuserar på att förutsäga resursförbrukningen under ett litet RUP-projekt. Däremot vid telefonintervjun anser B ur praktiska erfarenheter att det inte finns skillnad på hur resurserna förutsägs för ett litet RUP-projekt eller för ett annat projekt. Han menar att de förutsägs bara en gång. Det förutnämnd anses vara ett bevis på att vad som sägs i metoddokumentation för den specifika dokumenterade anpassade RUP är inte detsamma som vad som sägs ur praktiska erfarenheter om ett litet RUP-projekt.

Det verkar att olika källor om anpassningar av RUP<sup>®</sup> för små projekt inte säger samma på grund av det är olika åsikter om hur RUP<sup>®</sup> anpassas för små projekt. Med andra ord är de olika anpassningar av RUP<sup>®</sup> inte exakt lika oavsett hur mycket de liknar sig i botten. Därför bör resultatet för detta examensarbete beaktas med hänsyn till att det kan finnas skillnader mellan den specifika dokumenterade anpassade RUP, som Hirsch (2003) presenterar, och den anpassade RUP, som intervjupersonen utgår ifrån.

Utifrån de som presenteras ovan framställs två tabeller (se nedan tabell 1 och 2). Tabellerna presenterar hur de två dynamiska metoder förhåller sig både teoretiskt och praktiskt till de aspekterna som avser för små projekt. Tabellerna visar vilka aspekter som anses vara karaktäristiska drag för de två metoderna och vilka som inte anses vara det. Tabellerna visar också hur aspekterna för små projekt följs av metoderna praktiskt. Kommentarer till varför tabellen ser ut så som den gör presenteras också i tabellerna. ”Stämmer”, i tabellen nedan, betyder att aspekten, som anses vara för små projekt, följs av metoden och anses vara en egenskap hos metoden som gör att den är lämplig för små projekt. ”Stämmer inte” betyder att aspekten, som anses vara för små projekt, följs inte av metoden och anses inte vara en egenskap hos metoden som utgör dess lämplighet för små projekt. ”(Stämmer)” betyder att aspekten, som anses vara för små projekt, följs delvis av metoden och anses delvis vara en egenskap hos metoden som gör att den är lämplig för små projekt. Tabell 1 presenteras först nedan.

## 7. Resultat

Aspekter för små projekt	XP	Erfarenheter kring XP i praktiken	Kommentarer
Tester och feedback för iterativ utveckling	Stämmer	Stämmer	XP, enligt dess dokumentation och erfarenheter kring den är en iterativ metod.
Ofta leverans av delsystem till användarna	Stämmer	Stämmer	Både metoddokumentationen och praktiska erfarenheter betonar att leveranserna av systemet ger feedback.
En kund deltar aktivt i projektet	Stämmer	Stämmer	I små XP-projekt följs denna principen utan tvekan.
God kommunikation är viktigt och är bäst genom öga mot öga kommunikation	Stämmer	Stämmer	Öga mot öga kommunikation nämns tydligt både i dokumentationen och i intervjun.
En projektdeltagare har flera roller	Stämmer	Stämmer	Ett XP-projekt omfattar flera roller som ska täckas praktiskt av en begränsad antal deltagare.
Enkelt att reda ut vem som gör vad och att bara det som måste göras görs.	Stämmer	Stämmer	Både metoddokumentationen och praktiska erfarenheter betonar att ett XP-projekt omfattar bara de som måste göras.
Omstruktureringen av projektets innehåll sker när förändringar äger rum.	Stämmer	Stämmer	Omstruktureringen nämns tydligt både i metoddokumentationen och i intervjun.
Verktyg eller principer som leder till effektivitet	Stämmer	(Stämmer)	Bara ett programmeringsspråk används för att skapa effektivitet. Ur praktiska erfarenheter kan XP-projekt omfatta två eller flera programmeringsspråk i samma syfte.
Resursbehovet förutsägas ofta och snabbt	Stämmer	Stämmer	Både metoddokumentationen och insamlade fakta från intervjuerna betonar att denna aspekten är viktig i ett XP-projekt

Tabell 1. En tabell som presenterar resultatet för XP.

Utifrån tabellen som presenteras ovan konstateras att alla aspekterna för små projekt, som bör omfattas av en metod för små projekt, följs av XP. Tabellen visar att det är bara ett fall där vad som står i metoddokumentationen för XP inte stämde med de samlade fakta ur praktiska erfarenheter om XP-projekt. Tabell 2 presenteras nedan.

## 7. Resultat

Aspekter för små projekt	Den specifikt dokumenterade anpassade RUP	Erfarenheter kring en anpassad RUP i praktiken	Kommentarer
Tester och feedback för iterativ utveckling	Stämmer	(Stämmer)	I ett litet RUP-projekt saknas en del viktig feedback och systematiska kontinuerliga tester på systemet.
Ofta leverans av delsystem till användarna	Stämmer	Stämmer	Både metoddokumentationen och praktiska erfarenheter betonar att leveranserna av systemet ger feedback.
En kund deltar aktivt i projektet	Stämmer	Stämmer	I små RUP-projekt följs denna principen utan tvekan.
God kommunikation är viktigt och är bäst genom öga mot öga kommunikation	(Stämmer)	Stämmer	Öga mot öga kommunikation nämns inte i metod-dokumentationen för den specifikt dokumenterade anpassning av RUP. Däremot nämns att det är viktigt med god kommunikation under projektets gång.
En projektdeltagare har flera roller	Stämmer	Stämmer	Ett RUP-projekt omfattar flera roller som ska täckas praktiskt av en begränsad antal deltagare.
Enkelt att reda ut vem som gör vad och att bara det som måste göras görs.	Stämmer	Stämmer	Både metoddokumentationen och praktiska erfarenheter betonar att ett RUP-projekt omfattar bara de som måste göras.
Omstruktureringen av projektets innehåll sker när förändringar äger rum.	Stämmer inte	Stämmer inte	Ett litet RUP-projekt saknar en princip som syftar till att omstrukturera projektets innehåll vid hantering av förändringar.
Verktyg eller principer som leder till effektivitet	Stämmer inte	Stämmer inte	Den specifika anpassade RUP saknar specifika verktyg eller principer som skapar effektivitet. Den anpassade RUP har inte fokus på att producera kod effektivt. Därför är det mindre fokus på effektivitet i ett litet RUP-projekt.
Resursbehovet förutsägs ofta och snabbt	Stämmer	Stämmer inte	Ur praktiska erfarenheter förutsägs resurserna i ett litet RUP-projekt på samma sätt som i andra projekt, bara en gång.

Tabell 2. En tabell som presenterar resultatet för den anpassade RUP.



## 7. Resultat

Utifrån de som presentas ovan anses att resultatet på första delen av frågeställningen är att både XP och den specifika dokumenterade anpassade RUP har karaktäristiska drag som gör de lämpliga för små projekt. De karaktäristiska dragen liknar de mesta som presenteras ovan i tabellkolumnen ”Aspekter för små projekt”. Utifrån tabellerna konstateras brister av vissa karaktäristiska drag, som XP tar upp, för den specifikt dokumenterade anpassade RUP för små projekt. Tabellerna visar också hur XP passar bäst för små projekt.

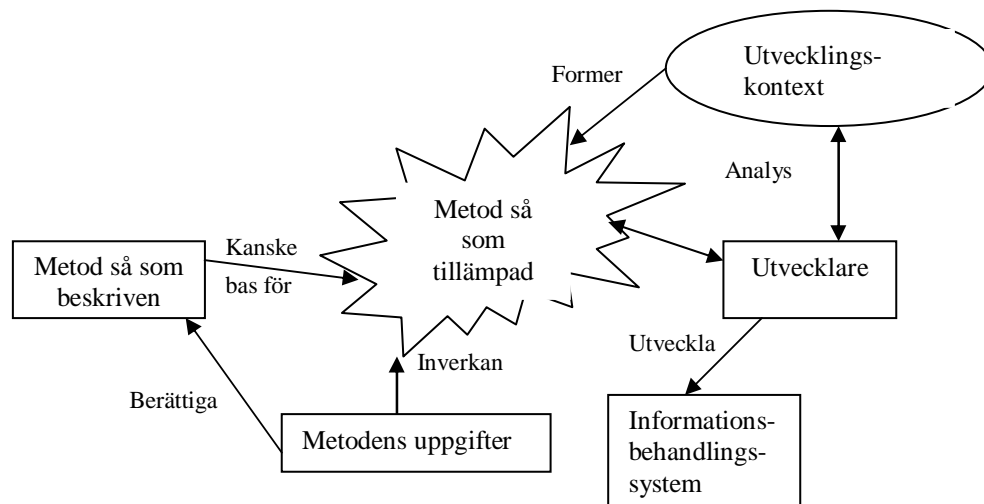
Utifrån de som presentas ovan anses att resultatet på andra delen av frågeställningen för problempreciseringen skiljer sig lite från resultatet på första delen av frågeställningen. Vissa aspekter för små projekt, som nämns i form av karaktäristiska drag i metoddokumentationen för de två dynamiska metoderna, saknades bland erfarenheter kring de två dynamiska metoderna. Andra aspekter för små projekt, som nämns i form av karaktäristiska drag i metoddokumentationen för de två dynamiska metoderna, nämns bland erfarenheter kring de dynamiska metoderna i praktiken. Det finns två fall för den specifika anpassade RUP, där två aspekter nämndes i dess metoddokumentation men nämndes inte bland erfarenheter kring denna metod (se tabell 2). Ett ytterligare fall är vad som sägs i metoddokumentationen om att det ska vara bara ett programmeringsspråk som används i ett XP-projekt för att skapa effektivitet. Praktiska erfarenheter visade att två eller flera programmeringsspråk kan användas i ett XP-projekt i samma syfte.

## 8 Analys

### 8.1 Analys av resultatet

I detta avsnitt kommer resultatet för detta examensarbete att analyseras. Syftet är att sätta in resultatet i ett större teoretiskt sammanhang.

Utifrån resultatet, som presenteras i kapitel 7, konstateras att det finns skillnad mellan vad som står i metoddokumentationen och hur metoden används. Orsaken är att vissa aspekter för små projekt, som nämns i metoddokumentationen för den specifika dokumenterade anpassade RUP och XP, existerades inte i små RUP och XP-projekt (se tabell 1 och 2). Ett fall till är vad som sägs om att det ska vara bara ett programmeringsspråk som används för XP-projekt, där praktiska erfarenheter visade att två eller flera kan användas. Med andra ord skiljer det sig mellan vad som skrivs om systemutvecklingsmetoden och hur den fungerar i praktiken. Skillnaden mellan metoddokumentationen och hur systemutvecklingsmetoden används i praktiken diskuteras av Fitzgerald m.fl. (2002), som presenterar ett ramverk för metodanvändningen i ett systemutvecklingsprojekt. Ramverket visar hur systemutvecklingsmetoden kan skilja sig ifrån dess dokumentation när den används i ett systemutvecklingsprojekt. Strukturen visas i figur 4 nedan.



Figur 4. En struktur för systemutvecklings metodanvändning (efter Fitzgerald m.fl., 2002, s. 12)

Komponenten ”Metod så som beskriven” syftar till att en systemutvecklingsmetod ska vara beskriven och vara dokumenterad för att denna teori ska gälla för den. Komponent ”Metod så som tillämpad” syftar till att visa hur metoden används i praktiken (Fitzgerald m.fl., 2002). Fitzgerald m.fl. (2002) påpekar att komponenten ”Metodens uppgift” förklarar och redogör vad metoden har för uppgift i systemutvecklingsprojektet, exempelvis att säkerställa god projektstyrning och kontroll över utvecklingsprocessen. Vidare påpekar de att komponenten ”Utvecklingskontext” förklarar komplexiteten och dynamiken i organisationen, där systemet ska utvecklas. Faktorer som bör tas hänsyn till vid beskrivning av utvecklingskontexten är teknologin, förändringar i strategin, filosofiska och praktiska gränser. Komponent

## 8. Analys

”Utvecklare” är för att betona människans roll. Det är människan och inte metoder som utvecklar systemen (Fitzgerald m.fl., 2002). ”Utvecklare” komponenten beskriver utvecklarnas roll, motivation och kunskap. Fitzgerald m.fl. (2002) förklarar vidare att olika utvecklare inte kan tillämpa och använda samma metod på samma sätt. Med andra ord spelar utvecklarna en stor roll vid avgörandet av hur metoden ska tillämpas. Komponenten ”Informationsbehandlingssystem” fokuserar på output från systemutvecklingsprojektet, med andra ord informationssystem. Frågor som diskuteras för denna komponent är de olika sätt att klassificeras, beskriva och förstå informationssystemet (Fitzgerald m.fl., 2002).

Ramverket fokuserar på att ge möjlighet till projektdeltagarna och forskarna i ämnet att förstå komplexiteten i systemutvecklingsprojektet (Fitzgerald m.fl., 2002, s. 18). Varje komponent av ramverket sprider sig över hela systemet. Utifrån kopplingarna mellan komponenterna i figur 4 konstateras att komponenterna samverkar, påverkar och beskriver varandra. Sammanfattningsvis syftar ramverket på att förklara skillnaden mellan metoden i praktiken och vad metoddokumentation säger. Skillnaden beror på utvecklingskontexten, metodens uppgifter för systemutvecklingsprojektet, hur metoden är beskriven och slutligen utvecklarna. Denna skillnad betonas av Argyris och Schon (1974) som förklarar distinktionen mellan teori och teori-i-praktiken.

### 8.1.1 Ramverkets lämplighet för att analysera resultatet

Denna del av ramverket (se figur 4), som blir lämplig att analysera genom resultatet för detta examensarbete, syftar till att betona skillnaden mellan hur systemutvecklingsmetoden ska användas och hur den används i praktiken med avseende på metodens uppgifter i systemutvecklingsprojektet. Denna del av strukturen består av tre komponenter, som är ”Metoden så som beskriven”, ”Metod så som tillämpad” och ”Metodens uppgifter”. Kopplingarna mellan de komponenterna visar hur de påverkar varandra.

Ramverket syftar till att definiera metodens uppgifter för systemutvecklingsprojektet. Fitzgerald m.fl. (2002) delar upp metodernas uppgifter till två grupper, rationella och politiska. De rationella nämns i metoddokumentationen och de politiska nämns inte i metoddokumentation. Metodens uppgifter i ett litet projekt är bland annat att minska komplexiteten i systemutveckling, vilket behandlas i kapitel 2.3.1. Denna uppgift nämns av Fitzgerald m.fl. (2002, s. 90) som förklarar att för att minska komplexiteten i systemutvecklingen är det viktigt att reda ut vad informationssystemet måste göra och hur ska det göras. Fitzgerald m.fl. (2002) anser att denna uppgift är rationell. Han förklarar vidare att det finns olika sätt för olika projekt att behandla komplexiteten i systemutvecklingen. Därför kan den tillämpade metoden skilja sig ifrån dess metoddokumentation, vilket stämmer med vad resultatet för detta examensarbete visar. Under genomförandet av detta examensarbete kunde inte hittas ett generellt ramverk för att minska komplexiteten i ett projekt trots att de dynamiska metodernas grundidéer är förknippad med att minska komplexiteten i systemutvecklingsprojekten. De politiska uppgifterna nämns inte i metoddokumentationen och är olika för olika projekt. Därför påverkar de sätten att tillämpa systemutvecklingsmetoder, vilket kan leda till att den tillämpade metoden skilja sig ifrån metoddokumentationen. De politiska uppgifterna påverkar troligtvis sättet att tillämpa en metod. Därför kan konstateras att tillämpningar av den anpassade RUP och XP skilde sig ifrån deras metoddokumentation. Med andra ord kan teorin, om att de politiska uppgifterna påverkar metodtillämpningar, kan stödjas av resultatet för detta examensarbete.

## 8. Analys

Ramverket syftar till att en systemutvecklingsmetod ska vara dokumenterad och ska vara för att lösa problemen för ett systemutvecklingsprojekt, som kännetecknar med förseningar och med att inte möta kraven som ställs från användarna (Fitzgerald m.fl., 2002, s. 87). Fitzgerald m.fl. (2002) påpekar att metoden ska fokusera på de sociala och organisatoriska aspekterna för ett informationssystem. XP och den specifikt dokumenterade anpassade RUP anses vara alternativ till traditionella systemutvecklingsmetoder och vara för att möta kraven samt lösa problemen för ett systemutvecklingsprojekt (diskuterad i kapitel 2.2.4 och 2.3). Resultatet för detta examensarbete visar hur de både metoderna tar hänsyn till de sociala och organisatoriska aspekter, där fokuserar metoderna på god kommunikation och på att möta kraven som ställs av användarna. Med andra ord kan teorin, om att en metod ska vara för lösning av problemen i systemutvecklingsprojekten, stödjas av resultatet för detta examensarbete.

Sammanfattningsvis förklarar ramverket att utifrån behoven av att minska komplexiteten i systemutvecklingen, hur metoden är beskriven och vad den har för uppgift bestäms hur metoden blir tillämpad. Det förutnämnd kan vara orsaken till att resultatet för detta examensarbete ser ut som den gör.

### 8.1.2 Djupare analys av resultatet

Resultaten som presenteras i tabell 1 (se kapitel 7) kan analyseras med detta ramverk. Där de både tabellkolumnen, XP och erfarenheter kring XP i praktiken, visar samma resultat betyder det att kopplingen mellan ”metod så som beskriven” och ”metod så som tillämpad” i ramverket stämmer inte. Exempelvis resultatet som visar XP, enligt dess dokumentation och erfarenheter kring den är en iterativ metod, som omfattar tester av systemet och feedback från användarna. Överenskommelsen mellan metoddokumentationen och erfarenheter visar att vad ramverket säger, specifikt kopplingen mellan ”metod så som beskriven” och ”metod så som tillämpad”, inte stämmer. Där de både tabellkolumnen, XP och erfarenheter kring XP i praktiken, visar olika resultat betyder det att kopplingen mellan ”metod så som beskriven” och ”metod så som tillämpad” i ramverket stämmer. Exempelvis resultatet som visar att metoddokumentationen för XP säger att det är bara ett programmeringsspråk som används i projektet för att skapa effektivitet. Ur praktiska erfarenheter konstateras att XP-projekt kan omfatta två eller flera programmeringsspråk i samma syfte. Skillnaden mellan metoddokumentationen för XP och vad som sägs om den ur praktiska erfarenheter från tillämpningar av den visar att vad ramverket säger, specifikt kopplingen mellan ”metod så som beskriven” och ”metod så som tillämpad”, stämmer.

Det är det samma för resultatet som presenteras i tabell 2 (se kapitel 7). Där de både tabellkolumnen, den specifika dokumenterade anpassade RUP och erfarenheter kring anpassade RUP i praktiken, visar samma resultat betyder det att kopplingen mellan ”metod så som beskriven” och ”metod så som tillämpad” i ramverket stämmer inte. Exempelvis resultatet som visar att ett litet RUP-projekt, enligt metoddokumentationen och erfarenheter kring den, saknar en princip som syftar till att omstrukturera projektets innehåll vid hantering av förändringar. Överenskommelsen mellan metoddokumentationen och erfarenheter visar att vad ramverket säger, specifikt kopplingen mellan ”metod så som beskriven” och ”metod så som tillämpad”, inte stämmer. Där de både tabellkolumnen, den specifika dokumenterade anpassade RUP och erfarenheter kring anpassade RUP i praktiken, visar olika resultat betyder det att kopplingen mellan

## 8. Analys

”metod så som beskriven” och ”metod så som tillämpad” i ramverket stämmer. Exempelvis öga mot öga kommunikation nämns inte i metoddokumentationen för den specifikt dokumenterade anpassning av RUP. Däremot nämns ur praktiska erfarenheter kring små RUP-projekt att det är viktigt med öga mot öga kommunikationen under projektets gång. Skillnaden mellan metoddokumentationen för XP och vad som sägs om den ur praktiska erfarenheter från tillämpningar av den visar att vad ramverket säger, specifikt kopplingen mellan ”metod så som beskriven” och ”metod så som tillämpad”, stämmer.

### 8.2 Relaterat arbete

I detta avsnitt presenteras andra studier och undersökningar som är relaterade till arbetets resultat. Syftet är att studera resultatet från detta examensarbete och relatera resultat till andra studier.

Resultatet från detta examensarbete visar att det finns skillnad på hur en anpassad metod är dokumenterad och hur den används i praktiken. Insamlade fakta från metoddokumentationen om den specifika dokumenterade anpassade RUP och XP skiljer sig lite ifrån insamlade fakta om små RUP- och XP-projekt. Det finns många studier som visar att metoderna inte används på samma sätt som de är beskrivna i deras metoddokumentation. Några av de studierna är utförda av Fitzgerald (1998), Russo m.fl. (1996), Westrup (1993) samt Wynekoop och Russo (1995). Fitzgerald (1998, s. 326) i en studie om utvecklarnas roll vid tillämpning av en systemutvecklingsmetod påpekar att olika utvecklare tillämpar metoder på olika sätt, vilket resulterar i en unik tillämpning av metoden för varje projekt. Orsaken är behoven av att förbättra användning av metoderna för olika systemutvecklingsprojekt. Även Russo m.fl. (1996, s. 393) påpekar att systemutvecklingsmetoderna tillämpas på olika sätt för att kunna passa olika behov för olika organisationer. Studien som är utförd av Russo m.fl. (1996) behandlar relationen mellan utvecklingskontext och tillämpningar av systemutvecklingsmetoder. Studien som är utförd av Wynekoop och Russo (1995) behandlar systemutvecklingsmetoder och varför de i praktiken inte tillämpas så som de är. I studien förklaras orsaken till varför systemutvecklingsmetoden tillämpas på olika sätt. Resultaten för denna studie visar att orsaken är mindre förståelse för hur metoderna valts, anpassas och hur de bör väljas.

Resultaten av alla de studierna, som presenteras ovan, är relaterade till resultatet för detta examensarbete. Orsaken är att de behandlar i princip samma ämne som detta examensarbete behandlar. En ytterligare orsak är att resultatet för de studierna liknar resultatet för detta examensarbete.

## 9 Slutsatser och diskussion

### 9.1 Arbetets slutsatser

I detta avsnitt kommer slutsatser för detta examensarbete presenteras. Syftet är att utifrån resultatet från genomförandet och analysen identifiera slutsatserna för denna studie. De två delfrågorna som studerades i detta examensarbete presenteras först följt med de slutsatserna som är relaterade till dem.

#### **Vad är de dynamiska metodernas karaktäristiska drag som gör att de är lämpliga för små projekt?**

De karaktäristiska dragen som gör XP lämplig för små projekt presenteras nedan. De karaktäristiska dragen konstateras utifrån genomförandet av denna studie. De är:

- Kontinuerlig leverans av systemet till användarna för testning, vilket ger feedback.
- XP är en iterativ utvecklingsmetod.
- En kund som deltar aktivt i projektet.
- Det finns möjlighet att tidigt och snabbt förutsäga resurserna som ska förbrukas under projektets gång.
- Projektets innehåll omstruktureras snabbt om något förändras.
- Koden är tillgänglig för alla projektdeltagarna, vilket anses vara en princip som leder till effektivt under ett litet projekt.
- Bara ett programmeringsspråk som används under ett litet XP-projekt för att säkerställa god kommunikation och effektivitet under ett litet projekt.
- Fokus på att projektet ska kännetecknas med god kommunikation mellan alla projektdeltagarna genom öga mot öga kommunikation.
- En projektdeltagare får ha flera roller.

De karaktäristiska dragen som gör den specifikt dokumenterade anpassade RUP lämplig för små projekt presenteras nedan. De karaktäristiska dragen konstateras utifrån genomförandet av denna studie. De är:

- Kontinuerlig leverans av systemet till användarna för testning, vilket ger feedback.
- Den specifikt dokumenterade anpassade RUP ger möjlighet till god projektstyrning i ett litet projekt, vilket leder till att alla projektdeltagarna vet vad de ska göra.
- Den specifikt dokumenterade anpassade RUP ger möjlighet till att hantera förändringar genom att klarlägga vad som förändrar vad, när och hur.
- En kund involveras aktivt för att projektdeltagarna ska kunna hantera kraven aktivt.
- En projektdeltagare kan ha flera roller.
- Diskussioner och samarbete mellan deltagarna och kunden, vilket leder till god kommunikation.

## 9. Slutsatser och diskussion

- Fokus på att förutsäga resurserna som ska förbrukas under projektet gång.
- Väl definierade riktlinjer och information om de olika delarna i projektet genom artefakter, vilket minskar missförstånd mellan projektdeltagarna.
- Bara det som anses viktigt för att få ett lyckat projekt görs.

De karaktäristiska dragen för både metoderna, som presenteras ovan, kan jämföras med aspekterna för små projekt, som presenteras i kapitel 5.1. Utifrån jämförelsen konstateras att de karaktäristiska dragen följer en stor del av aspekterna för små projekt.

Utifrån litteraturstudie som omfattade metoddokumentationen för XP konstaterades att alla de aspekterna för små projekt följs av XP (se tabell 1). Alla de aspekterna, som följs av XP, liknar de karaktäristiska dragen som gör att XP är lämpliga för små projekt. Med andra ord är de karaktäristiska dragen och de aspekterna för små projekt är lika, vilket gör att XP anses lämplig för små projekt. För den specifikt dokumenterade anpassade RUP konstateras att den sjunde och åttonde aspekten inte följs av den specifikt dokumenterade anpassade RUP (se tabell 2). Den specifikt dokumenterade anpassade RUP har färre karaktäristiska drag, som gör att den lämplig för små projekt, än XP. De slutsatserna är rimliga eftersom ett litet RUP-projekt hanterar inte förändringarna genom omstrukturering av projektet. En annan orsak är att den specifikt dokumenterade anpassade RUP fokuserar inte på effektivitet. En ytterligare orsak är att det finns en aspekt som inte nämns i metoddokumentationen för den specifikt dokumenterade anpassade RUP. Denna aspekt är öga mot öga kommunikationen.

Sammanfattningsvis finns de många karaktäristiska drag som de två metoder omfattar som gör att de är lämpliga för små projekt men i olika grad. XP anses vara mer lämplig för små projekt än den specifikt dokumenterade anpassade RUP. Orsaken är att XP följer all de aspekterna för små projekt och har mera karaktäristiska drag som gör den lämplig för små projekt.

### **Vilka erfarenheter finns från metodtillämpningar med de två metoderna som styrker argumenten från metodlitteraturen avseende dess lämplighet för små projekt?**

Utifrån insamlade erfarenheter från både litteraturstudie och telefonintervjuer konstateras att alla argumenten, som är från metodlitteraturen avseende de två metoders lämplighet för små projekt, styrkas inte. Litteraturstudie styrker en del argument, en annan del styrkas av telefonintervjuer och en tredje del av både litteraturstudie och telefonintervjuer. Sammanfattningsvis styrker resultatet av erfarenheter kring XP nästan alla argument som är från metodlitteraturen avseende dess lämplighet för små projekt. Denna slutsats är rimlig eftersom alla de aspekterna för små projekt följs av XP i praktiken och anses viktiga för ett lyckat litet XP-projekt förutom en aspekt (se tabell 1). Den aspekten som inte styrkas av de insamlade fakta ur praktiska erfarenheter om XP-projekt är planeringsknepen, som är för att skapa effektivitet i projektet. Praktiska erfarenheter visar att flera programmeringsspråk kan användas i ett XP-projekt för att skapa effektivitet. Erfarenheter kring den anpassade RUP visar brister i den att passa små projekt. Denna slutsats är rimlig eftersom den anpassade RUP följer inte fyra aspekter, som avses avgörande för små projekt (se tabell 2). Två av de aspekterna är redan

## 9. Slutsatser och diskussion

konstaterad sedan tidigare vid första frågeställningen besvarades att de inte följs av den anpassade RUP. De två aspekterna är:

1. Omstruktureringen av projektet sker snabbt när förändringar äger rum
2. Det måste finnas verktyg eller principer som leder till effektivitet.

Förutom de två aspekterna finns det ytterligare två aspekter som inte följs helt av den anpassade RUP i praktiken. De två aspekterna är: tester som ger feedback och att förutsäga resursbehovet snabbt. Orsaken är att systematiska kontinuerliga tester på systemet saknas i små RUP-projekt. En ytterligare orsak är att resursbehovet förutsägs bara en gång i små RUP-projekt.

Sammanfattningsvis visar erfarenheter kring XP att den är lämplig för små projekt utan tveksamhet. Däremot visar erfarenheter kring den anpassade RUP att den har delvis brister som påverkar dess lämplighet för små projekt. De resultaten som är utifrån metoddokumentation för de två dynamiska metoderna skilde sig lite från de insamlade fakta som är ur praktiska erfarenheter. Denna skillnad är för att metoden så som den är tillämpad skiljer sig från metoden så som den är beskriven (diskuteras i kapitel 8).

### 9.2 Reflektion

Nedan diskuteras slutsatser för detta examensarbete. Examensarbetet och resultatet utvärderas. Slutsatserna utvärderas genom att klargöra hur det förhåller sig till det förväntade resultat. Nedan diskuteras också undersökningsarbetet för detta examensarbete. Förslag på vad som kunde göras bättre under undersökningsarbetet presenteras också. Hur litteraturen valdes och presenterades diskuteras också för att en stor del av arbetet utgörs av en litteraturstudie.

Att hitta litteratur som behandlar systemutvecklingsmetoder var inte svårt. Det finns massor av böcker och artiklar som behandlar ämnet. Böcker och artiklar som användes var relevanta, vilket är orsaken till att slutsatserna är pålitliga. De flesta böckerna och artiklarna som är avsedda för detta examensarbete behandlar systemutvecklingsmetoder generellt. De dynamiska metoderna är ett nytt ämnesområde och var behandlat i flera böcker, vilket ledde till att stort antal böcker och artiklar valdes i början. Därefter valdes ut för detta examensarbete de mest relevanta med avseende på problempreciseringen. Examinatorn och handledaren var till stor hjälp eftersom de har erfarenhet som behövs för att göra ett urval av relevanta böcker och artiklar. Utifrån referenslista för detta examensarbete bör det vara möjligt att konstatera vilka författare som är stora och välkända i ämnet. Orsaken är att deras namn nämns för det mesta flera gånger i referenslista, vilket betyder att de skriver och utforskar mycket om ämnet. Eftersom författarna för litteraturer och artiklar, som är avsedda för detta examensarbete, är olika och har olika erfarenheter var det viktigt att undersökaren tar upp både för och emot påståenden om ämnet av de författarna. Undersökaren kunde förhålla sig kritisk. Det tog mycket tid att läsa genom litteraturen och översätta de men de är verkligen lärorika. Det som kunde bli bättre är att fokusera mera på små projekt vid litteraturstudie för att kunna studera deras egenskaper. Det som kunde också bli bättre är att hitta flera böcker om små systemutvecklingsprojekt för att det skulle ge mera djupare uppfattning om vad som är små projekt. Det var lite



## 9. Slutsatser och diskussion

svårt att hitta litteratur om små systemutvecklingsprojekt som behandlar vad de har för egenskaper och vad som kännetecknar dem.

Förutom litteraturstudie var det två telefonintervjuer som genomfördes. Intervjupersonerna var lätta att prata med och att ställa frågor till. De bad om att bli kontaktad om mera frågor behöver ställas. Det positiva samarbetet mellan intervjuare och intervjupersonerna ledde till mindre missuppfattningar och mera tydlighet. Slutna frågor blev relevanta för detta examensarbete för att fokusera mer på de aspekterna för små projekt och de karaktäristiska dragen för de två dynamiska metoderna, som hittades redan när första frågeställningen besvarades. Intervjuerna behandlade som mest erfarenheter som styrker de karaktäristiska dragen som står i metodlitteraturen. Orsaken är att med hjälp av aspekterna för små projekt redades ut de karaktäristiska dragen för de två dynamiska metoderna. Det som kunde bli bättre är att ha öppna frågor för att reda ut om de karaktäristiska dragen verkligen existeras eller inte. Istället bland sista frågorna ställdes en öppen fråga som reder ut eventuella andra och inte förutnämnda karaktäristiska drag. En annan sak som kunde bli bättre är att genomföra intervjuerna i god tid innan för att kunna studera svaren väl och kontakta intervjupersonerna för eventuella mera djupa följdfrågor. Intervjupersonerna hade inte mer än två års erfarenhet om små projekt, vilket ledde till att slutsatserna som drogs efter de telefonintervjuerna kan inte generaliseras helt. Ämnet är relativt nytt och det skulle vara nästan omöjligt att hitta mera erfarna intervjupersoner. En lösning borde vara att intervjua flera än bara två intervjupersoner för kunna generalisera slutsatserna utan tvekan.

Förväntade resultat var att genom undersökningsarbetet hitta mycket som styrker det som står i litteratur om de två dynamiska metodernas lämplighet för små projekt. För den ena metoden, XP, var det mycket som styrker dess lämplighet för små projekt. Däremot för den andra, den specifikt dokumenterade anpassade RUP, var det svårt att hitta mycket i litteratur som varken styrker eller ej dess lämplighet för små projekt. Det var lika svårt att hitta en intervjuperson som anpassar RUP<sup>®</sup> exakt som Hirsch, vilket ledde till tvekan om studierna kring erfarenheter om den specifikt dokumenterade anpassade RUP är heltäckande. Det kunde ha blivit bättre om jag valde en annan dynamisk metod än den specifikt dokumenterade anpassade RUP. Slutsatserna kunde då bli mera heltäckande och mer exakt utan en känsla av tveksamhet.

Det var inte lätt att arbeta ensamt, det var på gott och ont. Det är viktigt när man arbetar ensamt att ta ansvar och planera i god tid. Det var inget problem med det. Problemet var att hitta goda idéer för problempreciseringen och genomförandet av arbetet. Examinatorn och handledaren har verkligen varit till stor hjälp för att komma över detta problem. De kom alltid med nya förslag och var till stor hjälp vid utveckling av mina idéer. Undersökningsprocessen för detta examensarbete var lärorik och meningsfull.

### 9.3 Förslag på fortsatt arbete

Nedan presenteras förslag på fortsatt arbete för detta examensarbete. Förslagen kan vara ett tips vid framtida examensarbeten.

## 9. Slutsatser och diskussion

Examensarbetet behandlar två dynamiska metodernas lämplighet för små projekt. Det finns flera vägar att arbeta vidare med tillämpningar av systemutvecklingsmetoder. En väg är att undersöka varför systemutvecklingsmetoderna vid tillämpning av dem skiljer sig ifrån deras metoddokumentation. Slutsatserna för detta examensarbete visar att systemutvecklingsmetoderna så som de är tillämpad skiljer sig ifrån deras metoddokumentation. Därför kan det finnas intresse att undersöka hur de skiljer sig ifrån varandra. Frågeställningen kan vara:

- Vad är de bakomliggande faktorer som gör att systemutvecklingsmetoderna inte tillämpas så som de är beskrivna?

Det finns säkert mera frågeställningar som behöver besvaras genom detta ämnesområde. Utvecklingen genom systemutvecklingsmetoderna sker ganska snabbt, vilket leder till att nya frågeställningar dyker upp hela tiden.

## Referenser

Abrahamsson, P., Salo, O. and Ronkainen, J. (2002) Agile software development methods – Review and analysis. [online] VTT, Tillgänglig på Internet: <http://www.inf.vtt.fi/pdf/> [hämtat 2003-02-18].

Andersen, E. (1994) *Systemutveckling- principer, metoder och tekniker*. Studentlitteratur, Lund.

Argyris, C. and Schon, D. (1974) *Theory in Practice: Increasing Professional Effectiveness*. Jossey\_bass, USA.

Armano, G. and Marchesi, M. (2000) A Rapid Development Process with UML. *ACM SIGAPP Applied Computing Review*, ACM press, vol.1, pp. 4 –11.

Artim, J., Brigeford, C., Christman, L., Coplien, J., Rosson, M. B., Taylor, S. And Wirfes-Brock, R. (1998) Object-oriented practice in 1998: Does it help or hinder collaboration?. *Conference on Object Oriented Programming Systems Languages and Applications*, Canda, ACM Press, pp. 45-47.

Avison, D.E. & Fitzgerald, G. (1998) *Information Systems Development: Methodologies, Techniques and Tools*. Alfred Waller Ltd.

Avison, D.E. & Fitzgerald, G. (2003) *Information Systems Development: Methodologies, Techniques and Tools*. McGraw-Hill Publishing Company, UK.

Beck, K. (1999) *Extreme Programming Explained: Embrace change*. Addison Wesley, MA.

Beck, K. and Fowler, M. (2001) *Planning Extreme Programming*. Addison-Wesley, USA.

Bell, J. (1993) *Introduktion till forskningsmetodik*. Studentlitteratur, Lund.

Booch, G. (2001) Software Solutions. *Communications of the ACM*, Vol. 44, No. 3, pp. 119-121.

Brinkkemper, S. (1996) Method engineering: engineering of information systems development methods and tools. *Information and Software Technology* nr. 38, pp. 275-280.

Bäcklund, A.-K. (1994) *Just-in-time*. Studentlitteratur, Lund.

Cockbrun, A. and Highsmith, J. (2002) *Agile Software Development*. Pearson Education, Inc.

Coldewey, J., Eckstein, J., McBreen, P. and Schwanninger, C. (2000) Deploying lightweight processes. *Conference on Object Oriented Programming Systems Languages and Applications*, USA, ACM Press, pp. 131-132.

## Referenser

- Craige, S. and Jassim, H. (1995) *People and project management for IT*. McGRAW-HILL Book Company Europe, London.
- Crocker, R. (2002) The 5 reasons XP can't scale and what to do about them. Tillgänglig på Internet: <http://www.xp2003.org/conference/papers/Chapter15-Crocker.pdf> [hämtat 2003-02-18].
- Dahmström, K. (1991) *Från datasamling till rapport*. Studentlitteratur, Lund.
- Eckstein, J. and Josuttis, N. (2002) Scaling Agile Processes: Agile Software Development in Large Project. Tillgänglig på Internet: [http://xp2002.org/tut\\_desc/Eckstein.html](http://xp2002.org/tut_desc/Eckstein.html) [hämtat 2003-02-24].
- Fitzgerald, B. (1996) Lessons learned from the use of methodologies: time to halt the bipolar drift, *Information Systems methodologies.1996 Fourth Conference on Information Systems Methodologies in Irland*, pp. 17-21.
- Fitzgerald, B. (1998) An Empirical Investigation into the Adoption of System Development Methodologies. *Information & Management*. Vol. 34, pp. 317-328.
- Fitzgerald, B., Russo, N. L. and Stolterman, E. (2002) *Information Systems Development – Methods in Action*. The McGraw-Hill Companies, Inc., UK.
- Flower, M. (2002) The new methodology. Tillgänglig på Internet: <http://www.martinfowler.com/articles/newMethodology.html> [hämtat 2003-02-11].
- Goldkuhl, G. (1993) *Verksamhetsutveckla datasystem*. Intention AB, Affärlitteratur AB, Linköping.
- Goldkuhl, G. (2000) *Från process till praktik*. Centrum för studier av människa, teknik och organisation, Linköpings Universitet.
- Gummesson, E. (1985) *Forskare och konsult – om aktionsforskning och fallstudier i företagsekonomin*. Studentlitteratur, Lund.
- Griffin, A. (2002) A customer Experience: Implementing XP. Tillgänglig på Internet: <http://www.objectmentor.com/resources/articles> [hämtat 2003-03-20].
- Hellevik, O. (1987) *Forskningsmetoder i sociologi och statsvetenskap*. Typografen 1/Sandby Grafiska, Lund.
- Highsmith, J. (2001) History: The Agile Manifesto. Tillgänglig på Internet: <http://www.agilemanifesto.org/history.html> [hämtat 2003-02-14].
- Hirsch, M. (2002) Making RUP Agile- OOPSLA 2002 Practitioner Report. Tillgänglig på Internet: <http://oopsla.acm.org/extra/pracreports/MakingRupAgilePresentation.pdf> [hämtat 2003-02-11].

## Referenser

- Hirsch, M. (2003) The Zühlke Software Development Process. Version 1, [www.zuehlke.com](http://www.zuehlke.com).
- Jayarantna, N. (1996) Choice of a methodology for information systems development!. *Information Systems methodologies. 1996 Fourth Conference on Information Systems Methodologies in Irland*, pp. 23-28.
- Jeffries, R. (2001) What is extreme programming?. Tillgänglig på Internet: <http://www.xprogramming.com/xprmag/whatisxp.html> [hämtat 2003-03-17].
- Karlsson, F. (2002) *Meta-method for method configuration – A Rational Unified Process case*. Linköpings universitet.
- Karlsson, P. (2001) *Förbättringar av Ida Systems systemutvecklingsmodell*. Linköpings tekniska Högskola.
- Karlström, P. (2002) Introducing Extreme Programming – An experience raport. Tillgänglig på Internet: <http://www.agilealliance.com/articles/Danielkarlstrom--IntroducingExtremeProgramming.pdf> [hämtat 2003-02-18].
- Kruchten, P. (2000a) *The Rational Unified Process – An introduction*. Second edition, Addison Wesley Longman, Inc.
- Kruchten, P. (2000b) From Waterfall to Iterative Development – A Challenging Transition for Project Managers. Tillgänglig på Internet: <http://www.therationaledge.com/rosearchitect/mag/current/spring00/f8.html> [hämtat 2003-02-17].
- Kruchten, P. (2001) Agility with the RUP. *IT Journal*, Vol. 14, No. 12, pp. 27-33.
- Lindahl, E. (2003) Processer för små team. Tillgänglig på Internet: <http://www.lindysoft.com/processer/extreme.html> [hämtat 2003-04-08].
- Martin, R. C. (2002a) Why XP?. Tillgänglig på Internet: <http://www.objectmentor.com/resources/articles/why.html> [hämtat 2003-02-11].
- Martin, R. C. (2002b) Extreme Programming. Tillgängligt på Internet: <http://www.objectmentor.com/processImprovement/index.html> [hämtat 2003-02-11].
- Martin, R. C. (2002c) Agile Processes. Tillgänglig på Internet: <http://www.objectmentor.com/resources/articles/agileProcess.pdf> [hämtat 2003-02-11].
- Mathiassen, L. (1998) *Objektorienterad analys och design*. Studentlitteratur, Lund.
- Noyes, B. (2002) Define a flexible process with Rational Unified Process. Tillgänglig på Internet: <http://www.fawcette.com/resources/managingdev/methodologies/rup/> [hämtat 2003-02-18].

## Referenser

- Olle, T. W., Hagelstein, J., Macdonald, I. G., Rolland, C., Sol, H. G., Van Assche, F. J. M. and Verrijn-Stuart, A. A. (1991) *Information Systems Methodologies – a Framework for Understanding*. Second edition, Addison Wesley Longman, Inc.
- Patel, R. and Davidsson, B. (1994) *Forskningsmetodikens grunder – Att planera, genomföra och rapportera en undersökning*, Andra upplagan. Studentlitteratur, Lund.
- Patel, R. and Tebelius, U. (1987) *Grundboken i forskningsmetodik – kvalitativt och kvantitativt*. Studentlitteratur, Lund.
- Pollice, G. (2001a) Using the Rational Unified Process for small projects: Expanding Upon eXtreme Programming. Tillgänglig på Internet: <http://www.rational.com/media/products/rup/tp783.pdf> [hämtat 2003-02-11].
- Pollice, G. (2001b) RUP and XP, Part I: Finding Common Ground. Tillgänglig på Internet: [http://www.therationaledge.com/content/mar\\_01/f\\_xp\\_gp.html](http://www.therationaledge.com/content/mar_01/f_xp_gp.html) [hämtat 2003-02-11].
- Pollice, G. (2001c) RUP and XP, Part II: Valuing Differences. Tillgänglig på Internet: [http://www.therationaledge.com/content/mar\\_01/f\\_xp2\\_gp.html](http://www.therationaledge.com/content/mar_01/f_xp2_gp.html) [hämtat 2003-02-11].
- Reifer, D. J. (2002) How good are agile methods?. *IEEE SOFTWARE*, Vol. 19, No. 4, pp. 16-18.
- Russo, N., Hightower, R. & Pearson, J. M. (1996) The failure of methodologies to meet the needs of current development Environments. *Information Systems methodologies. 1996 Fourth Conference on Information Systems Methodologies in Irland*, pp. 387-391.
- Scott, K. (2002) *The Unified Process Explained*. Pearson Education, Inc.
- Smith, M. R. (1999) Home Page for Small Software Project Development Courses. Tillgänglig på Internet: <http://www.enel.ucalgary.ca/People/Smith/619.94> [hämtat 2002-04-12].
- Sommerville, I. (2001) *Software Engineering*. Addison-Wesley, England.
- Strand, L. (2002) *UML och RUP – Att lyckas med OO-projekt*. Docendo Sverige AB, Stockholm.
- The standish Group (1994) The CHAOS Report (1994). Tillgänglig på Internet: [http://www.pm2.com/sample\\_research/chaos\\_1994\\_4.php](http://www.pm2.com/sample_research/chaos_1994_4.php) [hämtat 2003-04-12].
- Wallenquist, A., Körnefors, J., Eriksson, A., Lindberg, P., Virklund, H., Karlsson, T., Bengtsson, J., Hirt, M., Starck, E., Brandberg, G., Johansson, J., Mickelsson, M., Skoogh, P., Hjelm, M., Nilsson, K., Adermark, S., Tallungs-Kentor, P., Nöteberg, S., Ellnestam, O., Holmberg, O., Lindahl, E. och Larsson, J. (2003) Agile – det moderna synsättet på mjukvaruutveckling. Tillgänglig på Internet: <http://www.agilesweden.org/doc/agile.pdf> [hämtat 2003-04-08].

## Referenser

Wells, D. (1999) *The Rules and Practices of Extreme Programming*. Tillgänglig på Internet: <http://www.extremeprogramming.org/rules.html> [hämtat 2003-02-14].

Westrup, L. (1993) Information Systems Metodologis in use. *Journal of Information Technology*, no. 8, pp. 267-275.

Wisén, J. och Lindblom, B. (1998) *Effektivt projektarbete*. Norstedts Juridik AB, Stockholm.

Wynekoop, J. and Russo, N. (1995) Systems Development methodologies: unanswered questions. *Journal of Information Technology* nr. 10, pp. 65-73.

## Intervjufrågor om XP

### Inledande frågor

1. I vilket företag jobbar Ni? Vad är företagets verksamhet?
2. Hur många års erfarenheter har Ni om XP för små projekt, som omfattar högst tio deltagare och genomförs på mindre än tolv månader?

### Huvudfrågor

3. Hur viktiga var feedback som fås av användarna i ett litet XP- projekt?
4. Vilka är för- och nackdelar med kontinuerlig feedback för ett litet XP-projekt?
5. Vilka är för- och nackdelar med för ofta leverans av systemet till användaren under ett pågående litet XP-projekt?
6. Hur viktigt är det med kunddeltagande i ett litet XP-projekt? Vilka är för- och nackdelar med det för ett litet projekt?
7. Vad spelar kommunikation för roll mellan deltagarna i projektet i ett litet XP-projekt?
8. På vilket sätt anses att öga mot öga kommunikationen är viktigt i ett litet XP-projekt? Vilka är för- och nackdelar med det för ett litet projekt?
9. Hur fördelas rollerna i ett litet XP-projekt? Vilka är för- och nackdelar med det för ett litet projekt?
10. Hur avgörs definieras omfattningen i ett litet XP-projekt?
11. Hur hanteras förändringar i kraven och projektdeltagarna i ett litet XP-projekt?
12. Kännetecknas ett litet XP-projekt med att bara ett programmeringsspråk som används? Vilka är för- och nackdelar med det för ett litet projekt?
13. Finns det möjlighet i ett litet XP-projekt att förutsäga resurserna som ska behövas under projektets gång? Hur stödjer XP projektdeltagarna att förutsäga resurserna?

### Slutfrågor

14. Vad anser Ni ur praktiska erfarenheter vara karaktäristiska drag som gör XP lämplig för små projekt?



## Bilaga 1

15. Vad är som är mest viktigast vid tillämpning av XP i ett litet projekt? Vad är det som är mest viktigast i XP som stödjer arbetet i ett litet projekt?
16. I hur stor grad lyckas XP att stödja ett litet projekt?

## Intervjufrågor om en anpassade RUP

### Inledande frågor

1. I vilket företag jobbar Ni? Vad är företagets verksamhet?
2. Hur många års erfarenheter har Ni om RUP för små projekt, som omfattar högst tio deltagare och genomförs på mindre än tolv månader?
3. Hur kan Ni jämföra den anpassade RUP för små projekt som Ni använder med Hirsch:s RUP för små projekt ? Vilka skillnader och likheter finns mellan dem?

### Huvudfrågor

4. Hur viktiga var feedback som fås av användarna i ett litet RUP- projekt?
5. Vilka är för- och nackdelar med kontinuerlig feedback för ett litet RUP-projekt?
6. Vilka är för- och nackdelar med för ofta leverans av systemet användaren under ett pågående litet RUP-projekt?
7. Hur viktigt är det med kunddeltagande i ett litet RUP-projekt? Vilka är för- och nackdelar med det för ett litet projekt?
8. Vad spelar kommunikation för roll mellan deltagarna i projektet i ett litet RUP-projekt?
9. På vilket sätt anses att öga mot öga kommunikationen är viktigt i ett litet RUP-projekt? Vilka är för- och nackdelar med det för ett litet projekt?
10. Hur fördelas rollerna i ett litet RUP-projekt? Vilka är för- och nackdelar med det för ett litet projekt?
11. Hur avgörs och definieras omfattningen i ett litet RUP-projekt?
12. Hur hanteras förändringar av kraven och projektdeltagarna i ett litet RUP-projekt?
13. Kännetecknas ett litet RUP-projekt med att bara ett programmeringsspråk som används? Vilka är för- och nackdelar med det för ett litet projekt?
14. Finns det möjlighet i ett litet RUP-projekt att förutsäga resurserna som ska behövas under projektets gång? Hur stödjer anpassade RUP projektdeltagarna att förutsäga resurserna?

## **Slutfrågor**

15. Vad anser Ni ur praktiska erfarenheter vara karaktäristiska drag som gör den anpassade RUP lämplig för små projekt?
16. Vad är det som mest viktigast vid tillämpning av den anpassade RUP i ett litet projekt?  
Vad är det som är mest viktigast i den anpassade RUP som stödjer arbetet i ett litet projekt?
17. I hur stor grad lyckas den anpassade RUP att stödja ett litet projekt?