



**LUND UNIVERSITY**  
School of Economics and Management

Department of Informatics

## **SCRUM & sprint-retrospektiv**

- användandet av sprint-retrospektiv, dess utformning och relevans för kontinuerlig förbättring.

Kandidatuppsats, 15 högskolepoäng, SYSK01 i informatik

*Framlagd:* Juni, 2011

*Författare:* Christian Andersson  
Johan Bergqvist

*Handledare:* Umberto Fiaccadori

*Examinator:* Lars Fernebro  
Magnus Wärja

## Abstrakt

<b>Titel</b>	SCRUM & sprint-retrospektiv - användandet av sprint-retrospektiv, dess utformning och relevans för kontinuerlig förbättring.
<b>Författare</b>	Christian Andersson Johan Bergqvist
<b>Utgivare</b>	Institutionen för informatik
<b>Handledare</b>	Umberto Fiaccadori
<b>Examinator</b>	Lars Fernebro Magnus Wärja
<b>Publicerad</b>	2011
<b>Nyckelord</b>	SCRUM, agil, sprint-retrospektiv, retrospektiv, reflektion, systemutveckling

## Sammanfattning

Undersökningen behandlar sprint-retrospektiv som är ett centralt moment inom SCRUM men även inom andra agila systemutvecklingsmetoder. Inom SCRUM delas utvecklingsprojekt upp i flera mindre iterationer, så kallade sprintar. Sprint-retrospektiv är en reflekterande del inom SCRUM som ska hållas mellan varje sprint. Detta är ett tillfälle där teamet ska reflektera över föregående sprint, ta upp saker som gått bra, som gått mindre bra och saker som behöver förbättras till kommande sprint. I studien söker vi svar på hur detta moment kan gå till och om det faktiskt bidrar till att teamet förbättras. I litteraturen har vi funnit att sprint-retrospektiv är ett moment som ibland prioriteras ner. Även detta har vi undersökt i studien. Till undersökningen har vi valt en kvalitativ ansats där vi håller ett mindre antal intervjuer med personer som har praktisk erfarenhet av SCRUM-projekt med rollen som SCRUM-master. Vi har i undersökningen inte funnit något som tyder på att sprint-retrospektivet nedprioriteras i våra intervjupersoners SCRUM-projekt. Samtliga tillfrågade såg retrospektivet som ett viktigt moment och var noga med att hålla ett retrospektiv efter varje avslutad sprint. Vidare var samtliga eniga om att det bidrog till att teamet och arbetsprocessen kontinuerligt blev bättre. Undersökningen visar däremot att retrospektivet utfördes på olika sätt mellan de tillfrågade. Vi kan inte utifrån resultaten säga att någon utformning skulle vara bättre än någon annan, då intervjupersonerna över lag var nöjda med SCRUM-processen och sprint-retrospektivet.

## Innehållsförteckning

1	Inledning.....	1
1.1	Bakgrund .....	1
1.2	Problemdiskussion.....	3
1.3	Frågeställning .....	3
1.4	Syfte.....	3
1.5	Avgränsningar .....	4
2	Litteraturgenomgång .....	5
2.1	Kort om traditionell utveckling .....	5
2.2	Agil utveckling .....	6
2.3	Skillnader mellan agil och traditionell utveckling.....	7
2.4	SCRUM .....	8
2.4.1	Roller i SCRUM.....	9
2.4.2	SCRUM-Processen.....	10
2.5	Retrospektivet.....	12
2.6	Kunskap och lärande .....	14
2.6.1	Kunskap.....	14
2.6.2	Lärande.....	15
2.7	Kontinuerlig förbättring.....	17
2.7.1	Efteranalys inom projekt .....	17
2.7.2	Reflektionsarbete inom agil utveckling.....	18
2.7.3	Kunskapsöverföring och retrospektivet .....	18
2.8	Sammanfattning av litteraturstudie & studiens undersökningsmodell .....	19
3	Metod .....	22
3.1	Tillvägagångssätt .....	22
3.2	Intervju.....	23
3.2.1	Semi-strukturerad intervju.....	23
3.2.2	Intervjuguide .....	24

3.2.3	Urval & Informanter.....	24
3.3	Datainsamling & analysmetod.....	27
3.4	Validitet & Reliabilitet .....	27
3.5	Etiska aspekter.....	28
3.6	Kritik mot tillvägagångsätt .....	29
4	<b>Presentation av empirisk data.....</b>	<b>30</b>
4.1	Relevant för undersökningen.....	30
4.2	Empiriskt resultat.....	31
4.2.1	Process för retrospektivet .....	31
4.2.2	Retrospektivet & kontinuerlig förbättring.....	35
4.2.3	Påverkande faktorer.....	38
5	Diskussion .....	40
5.1	Process för retrospektivet .....	40
5.2	Retrospektivet och kontinuerlig förbättring .....	42
5.3	Påverkande faktorer .....	44
6	Slutsats .....	46
6.1	Sammanställning – Resultat .....	47
7	Bilagor.....	48
	Bilaga 1, The Agile Manifesto .....	48
	Bilaga 2, Intervjuguide .....	49
	Bilaga 3, Intervjuprotokoll – P1 .....	51
	Bilaga 4, Intervjuprotokoll – P2.....	63
	Bilaga 5, Intervjuprotokoll - P3.....	79
	Bilaga 6, Intervjuprotokoll – P4.....	87
	Bilaga 7, Intervjuprotokoll – P5.....	97
8	Referenser.....	105

# 1 Inledning

## 1.1 Bakgrund

Verksamhetsmiljön idag förändras ständigt och ställer stora krav på att företag och organisationer måste kunna anpassa sig till dessa förutsättningar och vara beredda på snabba omställningar (Turban et al, 2010). Den föränderliga situationen ställer även krav på att dagens IT-system snabbt ska kunna anpassas till förändringar i omvärlden.

Sedan 1970-talet, då kraven på IT-system såg annorlunda ut, har utvecklare arbetat med sekventiell systemutveckling och utvecklat system baserat på traditionella utvecklingsmetoder, bl.a. vattenfallsmetoden. Vid systemutveckling enligt vattenfallsmetoden ses hela utvecklingsprojekt som en enda lång process där utvecklarna jobbar sekventiellt med att ta fram kravspecifikationer, designar, testar och implementerar systemet, för att slutligen sätta ett färdigt system i drift. Vattenfallsmetoden och liknande sekventiella metoder präglas av mycket dokumentation och strikta fas-indelningar (Forsberg & Lindström, 2008).

Under senare år har agil utveckling och uttryck som bl.a. ”agile development”, ”Extreme Programming” och ”SCRUM” florerat allt mer inom IT-branschen. Grundtanken med agil utveckling är att komma ifrån de sekventiella utvecklingsmetoderna och arbeta mer flexibelt och iterativt för att kunna hantera föränderliga krav. Uttrycket ”agile-development” introducerades 2001 av 17 processmetodiker under ett möte där de diskuterade trender inom systemutveckling och hur utvecklingsmetoderna skulle kunna anpassas för att kunna hantera de krav som ställdes på IT-system (Awad, 2005).

När IT-system utvecklas enligt agila metoder går utvecklingen från sekventiell till iterativ, och strävar istället mot att leverera IT-system i flera, mindre delar. De agila utvecklingsmetoderna benämns ofta som lättviktsmetoder på grund av att processen inte är normativ, likt traditionella utvecklingsmetoder. Restriktioner och riktlinjer finns att följa, det finns dock utrymme för egen tolkning av dessa och huruvida dessa riktlinjer följs till punkt och pricka (Kniberg, 2007).

De agila metoderna är både omtyckta och kritiserade. Agil utveckling har bland annat blivit kritiserad av utövare på grund av bristen av planering och dokumentation (Alvestad, 2006). Andra menar att metoderna är bra på grund av det faktum att dokumentation prioriteras ner och arbetet istället inriktas på att i ett tidigt stadie under processens gång skapa värde för kunden i form av levererade resultat (Schwaber & Beedle, 2002).

Två framstående agila utvecklingsmetoder är SCRUM och Extreme Programming (XP). Metoderna är iterativa och iterationerna kan pågå från endast någon vecka upp till ett par månader (Alvestad, 2006). En iteration innebär att utvecklingsgruppen tar sig an ett antal krav eller specifikationer, utvecklar den eller de komponenter som teamet valt att arbeta med, testar

koden och slutligen levererar fungerande mjukvara. Eftersom utvecklingen sker iterativt och IT-stöd levereras i flera mindre leveranser är det viktigt att under utvecklingens gång ha en tät kontakt med kunder och andra intressenter. När en delkomponent är levererad väljs sedan, i samråd med kund, nya komponenter ut att arbeta med under kommande iteration. Detta arbetssätt medför att det finns möjligheter att göra anpassningar/förändringar i systemet, även långt in i ett utvecklingsprojekt.

SCRUM är ett processverktyg som består av ett mindre antal regler och riktlinjer. Dessa beskriver inte i detalj allt som ska göras eller hur det ska göras utan kan tolkas eller utföras på olika sätt i ett projekt. Metoden SCRUM är empirisk vilket innebär att processen inte är mogen vid projektets början utan bör anpassas och experimenteras med, för att hitta tekniker som passar för rådande situation (Schwaber & Beedle, 2002). Exempelvis kan inslag från andra Agila metoder så som XP användas i ett SCRUM projekt (Kniberg, 2007).

Två återkommande uttryck inom SCRUM är *inspektera* och *anpassa*. De olika delarna av SCRUM-processen måste frekvent inspekteras för att säkerställa att processen och produkten håller den utlovade kvaliteten. Visar inspektionen att processen inte håller utlovad kvalitet eller av någon annan anledning inte når upp till processens mål, måste den anpassas (Schwaber & Sutherland, 2010).

En av grundpelarna inom SCRUM är att utvecklingen sker i iterationer, s.k. sprintar. Hur länge varje sprint varar och hur mycket arbete som utförs är upp till teamet att bestämma.

Kniberg och Skarin (2010) beskriver SCRUM i följande mening:

*”So instead of a large group spending a long time building a big thing, we have a small team spending a short time building a small thing. But integrating regularly to see the whole”*  
(Kniberg & Skarin, 2010, s 4)

I slutet av varje sprint reflekterar teamet över den avklarade sprinten och tar med sig lärdomar till kommande sprintar. På så sätt mognar processen och empirin från tidigare sprintar hjälper teamet att förbättras till kommande iteration. Denna reflektion kallas inom SCRUM sprint-retrospektiv och är en av SCRUMs riktlinjer. Ett sprint-retrospektiv bör utföras efter varje sprint. Förutom att retrospektivet ses som ett tillfälle att förbättras på ses det av många författare som ett tillfälle för att stärka lagkänsla inom teamet (Shore & Warden, 2007; Myllyaho et al, 2004).

Retrospektiv är en allmän term som generellt betyder att kolla tillbaka på händelser som redan har hänt. Shore & Warden (2007) beskriver tre olika typer av retrospektiv, *Release retrospective*, *Project retrospective* och *Iteration retrospective*. *Release* och *Project retrospective* är reflektionsmöten som hålls efter större milstolpar inom utvecklingsprojekt eller vid ett projekts slut. *Iteration retrospective* är ofta förknippat med agil utveckling och är till för att reflektera över hur arbetet gått under föregående iteration. Iterations retrospektiv är det samma som sprint-retrospektiv, bara en mer generell term för detta.

## 1.2 Problemdiskussion

Som vi tidigare nämnt är SCRUM en empirisk metod som utgår ifrån att metoden anpassas för varje enskild situation eller projekt. Inom litteraturen förespråkas det därför att SCRUM-teamet experimenterar under projektets gång tills de hittar en modell som passar för rådande situation (Kniberg, 2007). Ett viktigt medel för detta är SCRUMs sprint-retrospektiv som ska hållas efter varje sprint. Målet med retrospektivet är att inspektera hur arbetet under föregående sprint gått med avseende på människor, förhållanden, processer och verktyg (Schwaber & Sutherland, 2010). Teamet drar sedan lärdom av föregående sprints erfarenheter, positiva och negativa (inspektion), för att på så vis bli bättre och effektivare till kommande sprint. Till kommande sprint genomförs sedan de förändringar som anses vara viktiga för ett effektivare arbete (anpassning).

Här har vi tagit bort stycket som beskriver hur olika författare menar att retrospektivet är ett moment som nedprioriteras.

## 1.3 Frågeställning

Med utgångspunkt i vår problemdiskussion ämnar vi att undersöka hur retrospektivet kan vara utformade inom SCRUM, om det används i den utsträckning som förespråkas och om retrospektivet anses leda till en kontinuerlig förbättring av SCRUM-processen.

Detta leder oss till följande frågeställning:

*Hur används sprint-retrospektiv som ett verktyg för kontinuerlig förbättring avseende SCRUM-processen?*

## 1.4 Syfte

Vi vill få en bättre förståelse för hur reflektionsarbetet inom SCRUM går till och om det upplevs användas i tillräckligt stor utsträckning. Framst vill vi undersöka hur retrospektivet upplevs bidra till kontinuerlig förbättring i avseende på SCRUM-processen. Då problemen och flaskhalsar inom ett projekt kan vara många vill vi vidare se över om prioriteringar görs över vad som måste förbättras. Framförallt är vi intresserade av att kunna lyfta fram samband mellan hur sprint-retrospektiv används i respektive kontext och vilka effekter det får på förbättringsarbetet.

## 1.5 Avgränsningar

Uppsatsen kommer att behandla och undersöka hur SCRUM-mastern inom ett SCRUM-team upplever förbättringsarbete och retrospektiv inom utvecklingsmetoden SCRUM. Därav har vi valt att göra följande avgränsningar.

Uppsatsen kommer inte att belysa frågan från andra SCRUM-medlemmars perspektiv som t.ex. utvecklarna eller produktägaren.

Vi kommer enbart att behandla sprint/iterations-retrospektiv i denna uppsats och inte inkludera de övriga typerna av retrospektiv som tidigare nämnts (se avsnitt 1.1). Vi jämför vidare inte två utvecklingsmetoder i denna studie. Däremot innehåller vår litteraturgenomgång avsnitt rörande traditionell och agil systemutveckling, samt skillnader mellan dessa. Detta är enbart för att ge läsaren en grundläggande förståelse för den agila utvecklingen och för hur den har vuxit fram (se avsnitt 2.1 - 2.3).

Vi kommer inte heller att undersöka andra delar av en typisk SCRUM-process där reflektionsarbete kan förekomma.



## 2 Litteraturgenomgång

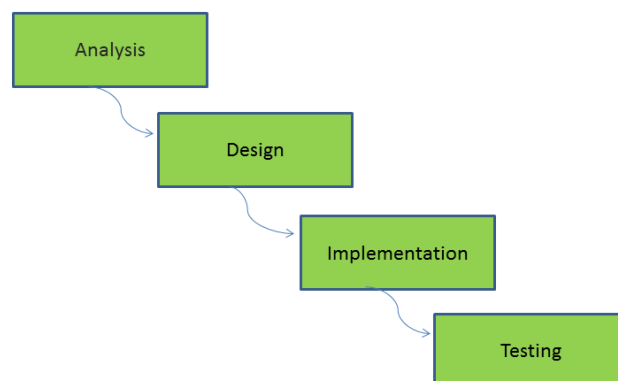
---

*För att ge läsaren en uppfattning om agil systemutveckling och hur det skiljer sig från mer traditionell utveckling kommer vi att inleda litteraturgenomgången med att kort beskriva agil- och traditionell systemutveckling. Därefter kommer vi att gå igenom SCRUM-processen för att ge läsaren en förståelse för hur arbetet inom SCRUM går till. Vi går sedan vidare och beskriver hur reflektion, kunskap och lärande hänger samman. Detta eftersom studien behandlar kontinuerlig förbättring av SCRUM-processen med hjälp av en reflekterande del. Slutligen kommer vi att koppla dessa teorier till vårt problemområde, sprint-retrospektiv.*

---

### 2.1 Kort om traditionell utveckling

Traditionella utvecklingsmetoder är plandrivna tillvägagångssätt att arbeta på där arbetet börjar med att framställa detaljerade kravspecifikationer och dokumentation som sedan följs upp med övergripande planeringsarbete för arkitektur och design. Dessa traditionella metoder kallas ofta för "heavy-weight methods" eller tungviktsmetoder på grund av dess omfattande planering, detaljerade dokumentation och design (Awad, 2005).



**Figur 2.1 - Vattenfallsmodell - process**

Vattenfallsmodellen (figur 2.1) var den första allmänt accepterade utvecklingsmodellen som fick fäste inom mjukvaruutveckling och ligger till grund för många liknande modeller som utvecklats sedan dess. I vattenfallsmodellen sker alla steg i utvecklingen (analys, design,

implementering och testning) sekventiellt, inget nytt steg påbörjas innan det tidigare är avslutat (Preece et al, 2007).

Larman (2004) beskriver vattenfallsmodellen som en process som noga planerar utvecklingsarbetet och definierar samtliga system- och kravspecifikationer i detalj innan praktiskt arbete och utveckling påbörjas.

## 2.2 Agil utveckling

Uttrycket *agil* syftar på förmågan att vara flexibel och att vara redo för förändringar. Inom området system- och mjukvaruutveckling har uttrycket *Agile Software Development* och agila utvecklingsmetoder vuxit fram för att kunna bemöta kraven från en snabbare och mer föränderlig omvärld då näringslivet efterfrågar lättare processer. Lättare processer syftar på mer flexibla och snabbare utvecklingsprocesser (Abrahamsson et al, 2002).

Awad (2005) tar i sin rapport upp exempel på flertalet agila metoder; bl.a. Adaptive Software Development (ASD), Agile Modeling, Crystal Methods, Dynamic System Development, Lean Development och SCRUM. Dessa har vuxit fram utifrån tanken att mjukvara med hög kvalitet och större andel nöjda kunder endast kan uppnås genom att göra dessa utvecklingsprocesser mer flexibla. Agila metoder beskrivs som ”light-weight methods” eller lättviktsmetoder och är enligt Awad mottagliga för förändring under utvecklingens gång.

Larman (2004) beskriver utveckling med agila systemutvecklingsmetoder som tidsbegränsad, iterativ och evolutionär utveckling. Evolutionär utveckling innebär att planering, krav och lösningar hela tiden utvecklas och definieras om mellan varje iteration. Samtidigt är varje iteration tidsbegränsad så till vida att en iteration utförs under en kortare tidsperiod, ofta under 2-4 veckor. Agila utvecklingsmetoder kan samtidigt vara svåra att definiera men delar egenskaper som nämnts ovan; kortare iterationer och förmåga att hantera förändringar under projektens gång.

Dessa egenskaper överensstämmer med resultatet från en studie genomförd av Abrahamsson et al (2002) som pekar på följande karaktäristiska drag i agila metoder:

- Inkrementell utveckling där mindre delar av system levereras i kortare iterationer.
- Nära samarbete mellan kund och utvecklare.
- Metoden är lätt att förändra och anpassa, samtidigt som den är lätt att lära sig.
- Förmåga att klara av förändringar under utvecklingens gång.

I början av år 2001 samlades sjutton representanter för agil systemutveckling för att diskutera trender inom systemutveckling och för att tillsammans komma fram till en överensstämmande bild över vad agil systemutveckling representerar (Awad, 2005). Resultatet av denna sammankomst och allians blev ett manifest kallat ”Manifesto for Agile Software

Development” (se bilaga 1) som består av fyra huvudpunkter och tolv principer som lägger fokus och beskriver de grundläggande tankarna med agil systemutveckling (Larman, 2004; Forsberg & Lindström, 2008).

Shore & Warden (2007) beskriver agil utveckling som en filosofi eller ett tankesätt inom mjukvaruutveckling baserat på det agila manifestet. Agila metoder som t.ex. SCRUM beskrivs följaktligen som metoder som stödjer manifestets huvudvärderingar. Manifestet har enligt Larman (2004) följande fyra huvudvärderingar:

#### **Individuals and interactions** *over processes and tools*

I stället för att jobba efter strikta metoder och regler har de s.k. lättviktsmetoderna mindre restriktioner och jobbar mer med dynamiken och kontakten mellan teammedlemmar och andra intressenter (Shore & Warden, 2007). Den täta interaktionen mellan människor ska säkerställa att produkten som utvecklas stämmer överens med kundernas önskemål (Schwaber & Beedle, 2002).

#### **Working software** *over comprehensive documentation*

I stället för att lägga ner väldigt mycket tid på dokumentation och specifikationer jobbar agila team med täta iterationer för att kontinuerligt leverera fungerande mjukvara till kunden. På så sätt blir kunden delaktig under hela utvecklingstiden och kan följa utvecklingsprocessen och säkerställa att produkten stämmer överens med deras mentala bild av det färdiga systemet (Schwaber & Beedle, 2002)

#### **Customer collaboration** *over contract negotiation*

Inom agil utveckling ska teamen hela tiden ha en tät kontakt med kunden och användarna för att säkerställa att systemet lever upp till kundens önskemål. Teamet arbetar med fokus på tät kundkontakt snarare än att spendera mycket tid och resurser på att förhandla om kontrakt och systemspecifikationer som sannolikt kommer att förändras under utvecklingens gång (Schwaber & Beedle, 2002).

#### **Responding to change** *over following a plan*

Den sista huvudpunkten enligt ”The Agile Manifesto” är att arbetssätt och processer ska kunna anpassas efter förändringar som kan tillkomma under projektets gång. Möjligheten att anpassa sig till snabba förändringar främjas av att teamet jobbar med täta iterationer och nära kontakt med intressenter. I stället för att jobba enligt en statisk planering som är svår att ändra jobbar teamet med dynamisk planering som har möjlighet att justeras efter varje iteration (Larman, 2004).

## **2.3 Skillnader mellan agil och traditionell utveckling**

En stor skillnad mellan traditionell och agil systemutveckling är att de traditionella metoderna ser utvecklingen som en lång process som inkluderar kravanalys, design, testning och

implementation. Även inom agil utveckling finns alla faser från analys till implementering med i processen. Däremot delas dessa faser upp och arbetas igenom flera gånger om, i så kallade iterationer (Larman, 2004). Båda tillvägagångssätten har sina styrkor och svagheter. Awad hävdar att faktorer som risk och osäkerhet, projektstorlek och människor är avgörande för vilken metod som passar bäst i det enskilda fallet (Awad, 2005).

Boehm (2002) skriver att traditionell utveckling fungerar bäst när systemkrav och specifikationer är väl kända i förväg, samtidigt som riskerna för förändrade krav för ett utvecklingsprojekt är små. Vidare beskrivs agil utveckling som ett bra alternativ i snabbt förändrande och omväxlande miljöer.

Enligt Preece et al (2007) är en av de största bristerna vid utveckling enligt den traditionella modellen ”vattenfallsmodellen” att den inte tar hänsyn till att systemets kravspecifikationer kan ändras under tiden ett system utvecklas. Det innebär att eventuella kundkrav som uppkommer under utvecklingens gång får vänta till hela cyklen är färdig och det ursprungliga systemet är utvecklat innan ändringar eller tillägg kan göras.

Det faktum att agila metoder prioriterar leveranser och kundnytta framför dokumentation och planering ses vidare av många utvecklare som en svaghet eftersom de upplever att de då saknar en överskådlig bild av utvecklingsprojektet Alvestad (2006).

Boehm (2002) menar att projektets storlek spelar roll vid valet av utvecklingsmetod. Vid mindre projekt där teamet är mindre än 15-20 utvecklare kan agila metoder vara lämpliga att använda. Vid större projekt kan det vara fördelaktigt att jobba utifrån plandrivna utvecklingsmetoder. Vidare påpekas även att plandrivna metoder inte lämpar sig för små projekt då de kan bli allt för tungrodda. Plandrivna metoder bör användas för verksamhetskritiska projekt där Boehm anser att agil utveckling kan innebära risker för utfallet.

## 2.4 SCRUM

Precis som många andra agila utvecklingsmetoder ses SCRUM som en lättviktsmetod på grund av att det bara finns ett mindre antal regler och riktlinjer som ska följas när utveckling sker enligt SCRUM. Hur teamet väljer att arbeta och vilka metoder som används för varje SCRUM-projekt bestäms utifrån rådande situation och bestäms av det egna teamet, med andra ord, hur produkten ska utvecklas (Schwaber & Beedle, 2002). Schwaber och Beedle (2002) menar vidare att SCRUM kan liknas vid ett omslagspapper till godis, dvs. SCRUM är en process, ett sätt att arbeta, som tillämpar de utvecklingsmetoder och standards som redan finns i organisationen.

SCRUM är enligt Schwaber & Beedle (2002) en empirisk process som präglas av att alla moment ska vara tidsbestämda. Att processen är empirisk innebär att teamet i varje SCRUM-

projekt kontinuerligt bör reflektera över vad som gått bra och vad som gått mindre bra, dra lärdom av detta och se till att göra bättre ifrån sig i kommande faser. Denna empiriska ansats beskrivs enligt följande: "*Management and control is exercised through frequent inspection and adaptive response*" (Schwaber & Beedle, 2002, s.94). Det innebär också att ett projekts arbetsprocess inte är huggen i sten utan kan komma att ändras under projektets gång. Efter varje sprint kan tidsplanen behöva ses över och antingen utöka eller minska den estimerade tiden.

Varje SCRUM-projekt består av en SCRUM-master, en produktägare och ett tvärfunktionellt team som jobbar med analys, utveckling och test under sprintens gång. SCRUM-mastern har det övergripande ansvaret för projektet och skall se till att teamet hela tiden tillhandahålls med de medel och förutsättningar som krävs för att uppnå ett gott resultat. Produktägaren jobbar också inom teamet och denna ska vara en person med verksamhetskompetens. Produktägaren har ansvaret att sköta kommunikation mellan team och kund, därav måste produktägaren ha kunskap om kundens behov. Det är också produktägaren som sköter jobbet med att prioritera de uppgifter som skall genomföras inom varje sprint. Teamet ska vara tvärfunktionellt, det innebär att all nödvändig kompetens ska finnas inom teamet (Schwaber & Beedle, 2002).

Karaktäristiskt för SCRUM är att teamet ska jobba självförvaltande. Det vill säga att teamet själv bestämmer vad och hur mycket arbete de kommer att hinna utföra under varje sprint. Till skillnad från en kontrollbaserad utvecklingsmetod, som t.ex. vattenfallsmetoden, där utvecklare blir tilldelande uppgifter s.k. "*pushing*", väljer utvecklarna inom SCRUM själva de uppgifter som kommer att utföras under sprinten, s.k. "*pulling*" (Moe et al, 2010).

#### 2.4.1 Roller i SCRUM

Nedan beskrivs rollerna *SCRUM-master*, *produktägare* och *teamet* som förekommer inom ett SCRUM-projekt.

*SCRUM-mastern* har en ledande roll inom SCRUM projekt. Det är *SCRUM-masterns* uppgift, att tillsammans med ledningen inom organisationen, sätta ihop ett väl fungerande team som kommer att arbeta ihop under projektets gång. Tillsammans med kunden hjälper denna också till att utse en person som kommer att få rollen som produktägare.

SCRUM-mastern ska se till att projektet följer den praxis och de regler som finns inom ramen för SCRUM. Han eller hon ansvarar också för att teamet har alla de resurser som krävs för att nå sina målsättningar under varje sprint. Utöver att bistå teamet med materiella ting som arbetsplats, mjukvara och ev. kapital är denna person också ansvarig för att teamet ska få jobba ostört under sprintarnas gång. Om personer utanför SCRUM-teamet behöver hjälp eller har andra krav och önskemål rörande produkten (eller andra ärenden), ska detta gå via SCRUM-mastern så att teamet under hela processen kan fokusera på att nå målen för varje sprint och producera så bra resultat så möjligt (Schwaber & Sutherland, 2010).

SCRUM-mastern har också ansvaret att utbilda teamet i SCRUM så att samtliga får en förståelse för och ser fördelarna med självförvaltande och tvärfunktionella team (Schwaber & Sutherland, 2010).

Inom varje SCRUM-team ska det dessutom finnas en och bara en, *produktägare*. Anledningen till att det enligt SCRUM inte accepteras att t.ex. en hel ledningsgrupp agerar produktägare är att denna person är ansvarig för ”product backlog” (se avsnitt 2.4.2). För att denna ska hållas klar och koncis finns det bara en person som har rätt att prioritera vad som ska göras i den (Schwaber & Beedle, 2002).

Produktägaren ska vara en person med verksamhetskompetens. Denna person har som redan nämnts det yttersta ansvaret för ”product backlog” där man listar kundens krav och prioriterar vad som behöver göras. Produktägaren representerar kunden och dess önskemål i ett SCRUM-projekt. För att produktägaren ska lyckas med sitt arbete krävs det att alla i organisationen respekterar och litar på besluten han fattar (Schwaber & Sutherland, 2010). Teamet jobbar utefter de prioriteringar som produktägaren gjort i ”product backlog” och ska under inga omständigheter lyssna på någon utomståendes direktiv. Produktägaren ansvarar också för att ”product backlog” är synlig för samtliga inom organisationen, inklusive kunden. På detta sätt kan utomstående hållas uppdaterade om projektets fortskridande (Schwaber & Beedle, 2002).

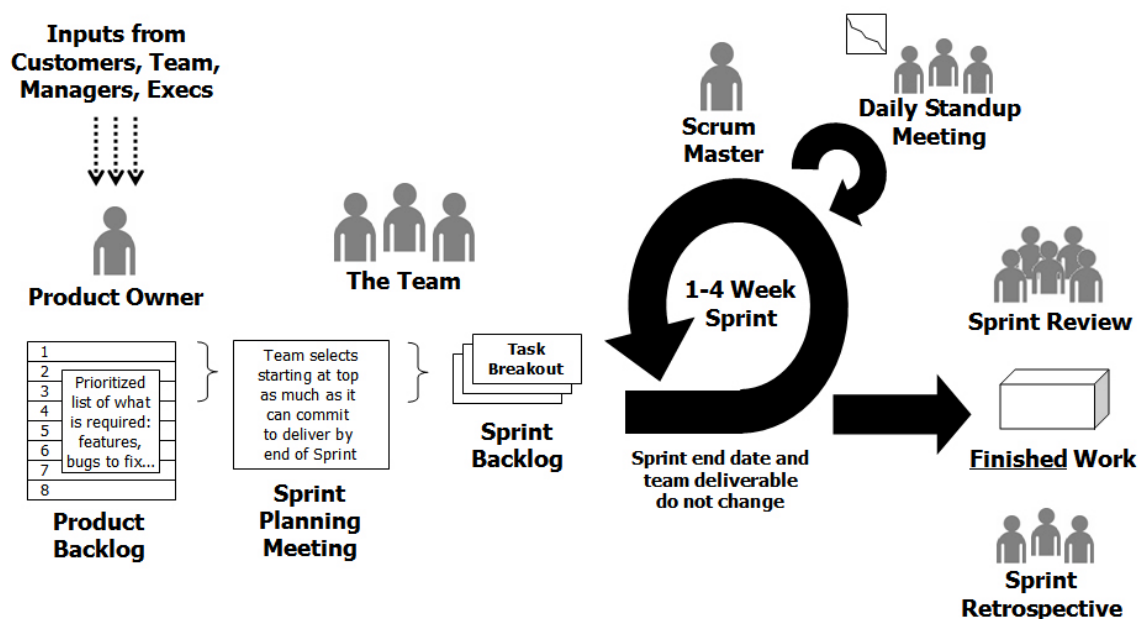
Den tredje rollen inom SCRUM är själva *teamet*. Det är teamet som gör det faktiska arbetet under sprintens gång. Teamet har enligt SCRUM full befogenhet att göra det som krävs för att nå sprintens mål. Schwaber och Beedle (2002) beskriver teamets uppgift som följande:

*”A team commits to achieving a Sprint goal. The team is accorded full authority to do whatever it decides is necessary to achieve the goal.”*  
(Schwaber & Beedle, 2002, s.35)

Teamen inom SCRUM ska vara tvärfunktionella, dvs. all den kompetens som behövs för att nå uppsatta mål ska finnas inom teamet. Det kan handla om utvecklare, designers, arkitekter m.m. Varje team bör enligt Schwaber & Sutherland (2010) utgöras av mellan 5-9 personer exklusive SCRUM-master och produktägare. Förutom att utföra det faktiska arbetet hjälper teamet produktägaren att tidsestimera arbetsuppgifter i ”*product backlog*”. Innan varje sprint väljer de också ut hur många uppgifter som kommer tas med i kommande sprint.

## 2.4.2 SCRUM-Processen

I följande stycke beskriver vi kort hur arbetsprocessen inom SCRUM ser ut med avseende på viktiga moment och artefakter, vilka är *Product backlog*, *Sprint planning meeting*, *Sprint*, *Sprint review* och *Sprint-retrospective*. Figur 2.2 illustrerar SCRUM-processen i sin helhet, de tidigare nämnda rollerna samt viktiga artefakter inom processen.



Figur 2.2 - Illustration av SCRUM-processen (Schwaber & Sutherland, 2007)

*Product Backlog* är hjärtat i SCRUM (se figur 2.2). Det är här som allting börjar och det är här kunden specificerar krav som de har på systemet. Det är inte bara kunden som får lov att lägga till objekt i backlogen, samtliga intressenter (kunden, teamet, produktägaren m.fl.) kan lägga till specifikationer i backlogen (Schwaber & Beedle, 2002). Till skillnad från traditionell utveckling är backlogen inte statisk utan specifikationer kan förändras under tidens gång. Specifikationerna eller historier som de kallas inom SCRUM ska vara korta, koncisa och beskrivas med en terminologi som kunden förstår, dvs. inga tekniska uttryck ska användas (Kniberg, 2007). Schwaber & Beedle (2002) menar dock att backlogen består av både affärsspecifika historier men även historier med rent teknisk härkomst.

Varje historia i backlogen bör ha ett antal beskrivande attribut så att utvecklarna förstår vad som ska göras och kan föra en dialog med produktägaren. Vilka attribut som ska användas är inte definierat. Det är produktägaren som har ansvar för backlogen och det är enbart han som har rätt att prioritera historierna (Awad, 2005). Men det är teamet, som tillsammans med produktägaren, estimerar tidsåtgång för varje funktion.

Historierna i backlogen behöver inte från början vara helt specificerade men om det finns chans att en historia kommer att lyftas in inom någon av de nästkommande sprintarna bör den beskrivas, estimeras och prioriteras (Schwaber & Beedle, 2002).

*Sprint Planning Meeting* är ett planeringsmöte som brukar delas upp i två faser. Först träffas teamet, kunden, produktägare och användare för att diskutera vilka historier som ska lyftas upp i kommande sprint. De historier som tas upp och ska tas med till kommande sprint flyttas därefter till en Sprint-Backlog (se figur 2.2). Denna backlog innehåller bara det som ska implementeras under kommande sprint. Under andra delen av mötet bör bara teamet och SCRUM-mastern närvara. Här diskuterar teamet hur arbetet ska läggas upp under kommande sprint för att implementera hela sprint-backlogen och nå uppsatta mål (Schwaber & Beedle,

2002). Precis som alla andra processer inom SCRUM så ska även detta möte tidsbestämmas (Schwaber & Sutherland, 2010).

Efter att *sprintens* planeringsmöte är avklarad kan det faktiska arbetet för teamet börja. Enligt Schwaber & Beedle (2002) bör varje sprint pågå under 30 dagar. Andra författare som Abrahamsson et al (2002) skriver att sprintarna kan variera i längd, allt från en vecka till en månad (se figur 2.2). Under sprintens gång har teamet själva full befogenhet att utföra uppdraget på det sätt som de anser är bäst för att nå sprintens mål. Det kan röra sig om allt från att utbilda sig till att ta in externa konsulter. Det viktiga är att det är teamet som beslutar om vad som krävs och att de har fullt förtroende av ledningen (Schwaber & Beedle, 2002).

*Sprint Review* är ett möte som utförs för att granska det utförda arbetet under sprinten. Mer utförligt inleds mötet med att produktägaren identifierar det som gjorts under sprinten, det som möjligtvis inte hunnits med och om teamet faktiskt har nått de mål som har varit uppsatta för sprinten (se figur 2.2). Efter detta presenteras resultatet i en så kallad ”sprint-demo” för kunder och andra intressenter (Schwaber & Sutherland, 2010). Resultatet diskuteras och tillsammans utifrån detta tar deltagarna ett beslut om hur de ska gå vidare och vad som skall lyftas in i kommande sprint (Schwaber & Beedle, 2002).

Efter att *Sprint Review* mötet är avslutat håller SCRUM-teamet ett reflektionsmöte, ett s.k. *Sprint Retrospective* (se figur 2.2). Retrospektivet är ett tillfälle för teamet att lära sig av sina misstag och förbättra sig till kommande sprinter Shore & Warden (2007). Enligt Schwaber & Sutherland (2010) ställs tre frågor till varje teammedlem: Vad har gått bra? Vad har gått mindre bra? Vad kan vi göra bättre till kommande sprint?

Kniberg (2007) menar att detta är en av de viktigaste delarna av SCRUM eftersom teamet här har en chans att verkligen diskutera hur de ska bli bättre. Retrospektivet är den sista delen av iterationen och efter denna session kan nästa iteration påbörjas. Retrospektivet leds av en diskussionsledare som kan vara teamets SCRUM-master, det kan även ledas av en extern facilitator (Schwaber & Sutherland, 2007; Schwaber & Sutherland, 2010).

## 2.5 Retrospektivet

Sprint-retrospektiv är ett medel för att hjälpa teamet att inspektera och anpassa SCRUM-processen och produkten som utvecklas (Derby & Larsen, 2006). Det är även ett tillfälle för teammedlemmarna att dela med sig av sina idéer och komma närmare varandra (Shore & Warden, 2007).

Schwaber och Sutherland (2010) beskriver vidare retrospektivet som ett tillfälle där teamet får möjlighet att inspektera hur arbetet under föregående sprint gått med avseende på människor, förhållanden, processer och verktyg. Teamet drar sedan lärdom av föregående sprints erfarenheter, positiva och negativa (inspektion), för att på så vis bli bättre och effektivare till



kommande sprint. Till kommande sprint genomför teamet de förändringar som anses vara viktiga för ett effektivare arbete (anpassning).

Då SCRUM som tidigare nämnts endast består av riktlinjer finns det inte någon uttalad ”best-practice”-metod eller ett tillvägagångssätt för att facilitera och utföra ett retrospektiv. Då det kan finnas många olika sätt och metoder att facilitera retrospektiven på har vi valt att redogöra för en metod framarbetad av Derby & Larsen (2006) för att ge exempel på hur ett retrospektiv kan gå till.

Derby & Larsen (2006) fem moment som ett iterations-retrospektiv bör behandla.

- Skapa stämning för diskussion
- Datainsamling
- Skapa insikt
- Bestäm vad som ska göras
- Avsluta retrospektivet

Varje retrospektiv bör inledas med ett moment som kan översättas som att skapa stämning för diskussion. Detta moment är till för att skapa en atmosfär som gör att deltagarna blir bekväma med att diskutera och reflektera över föregående sprint. Under detta moment ska även målet med retrospektivet belysas. Deltagarna kommer tillsammans överens om målet, tidsbestämmer mötet och väljer vilken ansats de ska använda för just detta möte.

Nästa moment är att göra någon form av datainsamling. Både hård data och mjuk data ska beaktas. Hård data kan gälla arbetet under sprinten, mätetal, avklarade historier etc. Mjuka data handlar om teamets känslor under föregående sprints gång. Diskussionsledaren kan använda sig av olika metoder vid detta moment. Shore & Warden (2007) beskriver en teknik för datainsamling där varje deltagare får ett antal lappar där de skriver ner både bra och dåliga saker från sprintens gång.

Nästa moment ger deltagarna en möjlighet att skapa sig insikter i de problemområden som kommer fram under datainsamlingen. Momentet är till för att deltagarna ska tolka och analysera data som kommit fram. Teamet diskuterar och undersöker vart problemen faktiskt kommer ifrån och vilken påverkan det har på teamets arbete. På så vis skapar det inom teamet en djupare förståelse om problemen.

När teamet gemensamt förstått innebörden av problem i föregående moment ska de i detta steg välja vilka förändringar som ska göras och hur de ska gå tillväga. Teamet bör välja ut en eller två förändringar som ska implementeras under kommande sprint. För många förändringar kan vara svårhanterligt och kan göra det svårt att få problem åtgärdade. En teknik för att bestämma vad som ska tas med till kommande sprint kan vara att rösta inom teamet (Shore & Warden, 2007). Detta kan göras genom att ge tre röster till samtliga deltagare. Varje deltagare får sedan lägga sina röster på den förändring de anser är viktigast.

I den sista fasen ska retrospektivet stängas och avslutas. Teamet bestämmer hur förändringarna ska följas upp och dokumenteras. För att dra lärdom av det som kommit upp under mötet bör det på något sätt dokumenteras. Ett sätt kan vara att ta ett foto av whiteboarden. Slutligen ska facilitatorn eller diskussionsledaren stänga mötet och uttrycka sin uppskattning till teamet. Mötet kan t.ex. avslutas genom att köra ett kort retrospektiv på retrospektivet och fundera igenom hur retrospektivet har gått och om det eventuellt kan bli bättre till kommande sammanträden (Derby & Larsen, 2006).

## 2.6 Kunskap och lärande

### 2.6.1 Kunskap

Kunskap kan enligt bl.a. Nonaka & Takeuchi (1995) delas in i *tyst* och *explicit* kunskap. Tyst kunskap är kunskap som finns inom en individ, ett team eller en organisation. Denna typ av kunskap kan vara svår och ibland omöjlig att överföra till andra. Explicit kunskap kan vara dokumenterad kunskap i form av t.ex. modeller eller kunskap som finns i böcker eller databaser m.m. Kunskap är inte konstant utan byter hela tiden form mellan tyst och explicit kunskap (Nonaka & Takeuchi, 1995).



Figur 2.3 - Modell över olika typer av kunskapsöverföring (Dingsøy, 2004)

Enligt Nonaka & Takeuchi (1995) finns det fyra olika typer av kunskapsöverföring, *Socialization*, *Combination*, *Internalization* och *Externalization* (se figur 2.3). Dessa olika typer av kunskapsöverföring beskrivs enligt följande:

*Socialization* innebär att en persons tysta kunskap överförs till någon annans tysta kunskap genom t.ex. observationer, samtal och imitation. Ett effektivt sätt att överföra tyst kunskap kan exempelvis vara att titta och lära.

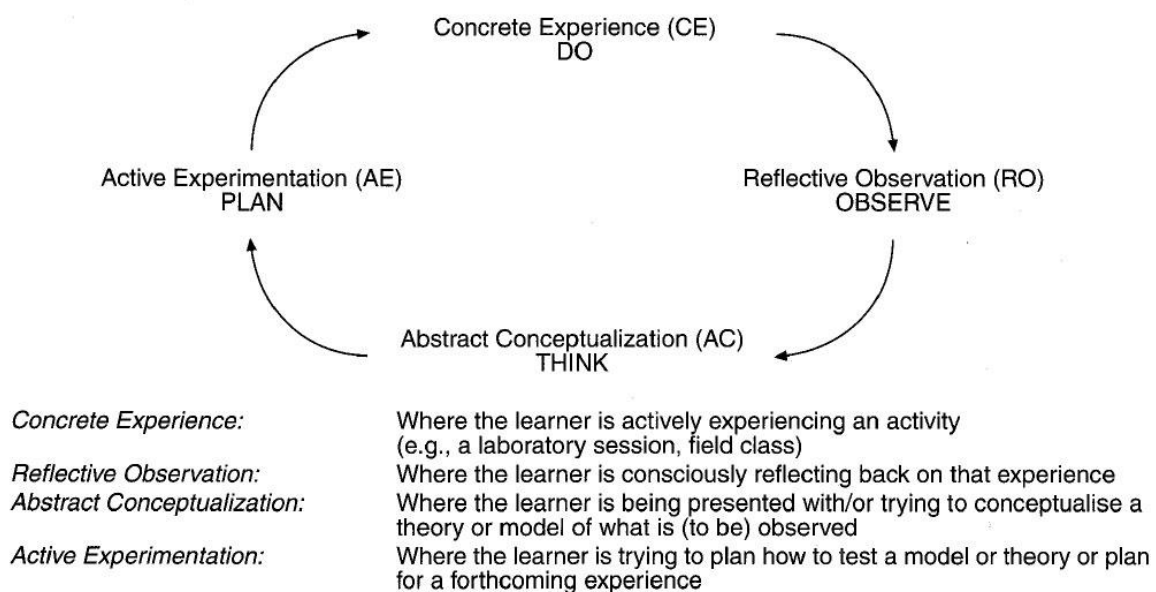
*Combination* innebär att explicit kunskap överförs till andras annan explicit kunskap. Detta kan ske genom att kombinera olika explicita kunskapskällor som t.ex. dokument eller kunskapsdatabaser.

*Externalization* handlar om att transformera tyst kunskap till explicit kunskap genom att försöka dokumentera den och göra den tillgänglig för andra som explicit.

*Internalization* är en process för att transformera explicit kunskap till individuell tyst kunskap. Dokument och manualer används ofta för att individer ska kunna läsa sig till den explicita kunskapen.

## 2.6.2 Lärande

Lärande kan ses både från den enskilda individens perspektiv men också från ett kollektivt perspektiv, t.ex. lärande inom en organisation. Kolb (1984) beskriver lärandeprocessen som en loop där början av denna loop ofta är något konkret, kanske en händelse. Individen reflekterar över händelsen, generaliserar och testar nya lösningar i olika sammanhang. Detta leder i sin tur till nya generaliseringar och i längden skapas lärande och kunskap (se figur 2.4).



**Figur 2.4 - The Experiential Learning Cycle (Healey & Jenkins, 2000)**

Många författare hävdar att reflektion är av stor betydelse för lärande (Schön, 1983; Argyris & Schön, 1978). Schön (1983) skiljer på två olika typer av reflektioner, *Reflection-in-action* och *Reflection-on-Action*. Schön menar att utövarna inom många typer av yrken behandlar varje case/arbete som unikt och att det därför är svårt att applicera standardiserade teorier för att lösa problem. För att hitta lösningar på problem och samtidigt öka sin förståelse eller kunskap måste utövaren under arbetets gång rama in problemet som hon ser det och reflektera över det. Reflektioner som sker på detta sätt benämner Schön som *Reflection-in-action*. Schön

beskriver processen med reflektion som ett medel för att skapa sig djupare förståelse om ett unikt problem som följer:

*”The process spirals through stages of appreciation, action and re-appreciation. The unique and uncertain situation comes to be understood through the attempt to change it, and changed through the attempt to understand it.”* (Schön, 1983, s 132)

Den andra typen av reflektion, *Reflection-on-action*, är när en utövare reflekterar över och analyserar resultatet av sina handlingar i efterhand. Istället för att under arbetets gång reflektera över och hitta snabba lösningar på problemen, reflekterar denne i efterhand över de konsekvenser som tidigare val medfört. Enligt Schön dokumenteras denna typ av reflektionsarbete i stor utsträckning (Schön, 1983).

Argyris (1992) hävdar att folk ofta missförstår vad lärande är och hur individen eller gruppen på ett effektivt sätt jobbar med lärande. Många förknippar lärande med problemlösning men tänker sällan på lärande i ett längre perspektiv. Att lösa problem är en viktig del för att lära sig och skaffa ny kunskap men om kunskapen ska fortleva måste djupare reflektioner göras över vad man faktiskt lärt sig. Om det krävs måste individen eller gruppen ändra och anpassa sig för att tidigare problem inte skall dyka upp igen.

1978 myntade Chris Argyris och Donald Schön uttrycken *Single-loop learning* och *Double-loop learning*. Det första uttrycket innebär att individen eller gruppen reflekterar över ett problem som dyker upp för att sedan hitta en lösning. När *single-loop learning* utövas missar individen eller gruppen att faktiskt anpassa och förändra organisationen eller processerna för att samma problem inte skall dyka upp igen. Det andra uttrycket *Double-loop learning* kan till skillnad från *Single-loop learning* beskrivas som att lära av sitt eget lärande. Mer utförligt innebär detta att det genomförs nödvändiga förändringar för att inte liknande problem ska uppstå igen, om det krävs får processer inklusive dess mål förändras. Ofta är professionella utövare väldigt duktiga på *Single-loop learning* och det finns en tillfredsställelse i att lösa problem när de uppstår. När problemen återkommer blir de ofta defensiva och lägger gärna skulden på någon annan (Argyris, 1992).

*”In short, their ability to learn shuts down precisely at the moment they need it the most.”* (Argyris, 1992, s127)

Argyris och Schöns beskrivning av lärande, single & double – loop learning, har stora likheter med begrepp som myntades av Gregory Bateson på 1940-talet. Bateson delade in lärandeprocessen i fyra olika nivåer: *Zero-Learning*, *Proto-Learning*, *Deutero-Learning* och *Trito-Learning* (Visser, 2003).

Den lägsta nivån av lärande, *Zero-Learning*, innebär att tillvägagångssätt och beteende inte förändras när ett problem uppstår. Samma problem kommer således återkomma om och om igen. Ett exempel kan vara att en person håller sin hand över elden och denna bränns. På denna nivå kommer personen att pröva att lägga handen över elden på nytt. Nästa nivå, *Proto-learning*, innebär att personen vidare drar lärdom om vad som tidigare hänt och upprepar därför inte misstaget. *Deutero-learning* är en typ av lärande som har likheter med Argyris och

Schön's double-loop learning där individen eller gruppen lär sig av sitt lärande. Mer utförligt reflekterar individen utifrån situationens kontext och omprövar sina beslut för att hitta en bättre lösning. Den sista nivån av lärande, trito-learning, sker nästan uteslutande på en omedveten nivå, där individen kontextualiserar lärandet från föregående nivå (Visser, 2003).

## 2.7 Kontinuerlig förbättring

### 2.7.1 Efteranalys inom projekt

En viktig del i alla typer av projekt är att reflektera över resultatet och dra lärdomar från tidigare erfarenheter. Att reflektera över och ta med sig både positiva och negativa erfarenheter bidrar till kompetensutveckling. Att jobba med reflekterande faser inom mjukvaruutveckling är inget nytt fenomen. Reflektionsarbetet ses som ett effektivt verktyg för lärande både på individ- och organisationsnivå (Salo et al, 2004). Myllyaho et al (2004) beskriver efteranalysarbetet som ett sätt att lära sig från processer, produkter och resurser och använda sig av lärdomarna i kommande och pågående projekt. Vidare beskriver Myllyaho et al (2004), precis som Salo et al (2004), att det är ett sätt att transformera individuell kunskap till kunskap på team och organisationsnivå. I projektledningslitteraturen ska detta reflekterande arbete göras i en så kallad reflektionsfas i projektets slutskede, innan resultatet överlämnas till kunden (Macheridis, 2005).

Det finns många sätt att utföra reflektionsarbete på under och efter projekt, valet av metod bör anpassas efter faktorer som projektets storlek och omfattning (Myllyaho et al, 2004). Myllyaho et al (2004) listar upp sex stycken fördelar som reflektionsarbete kan innebära.

1. *Det hjälper teammedlemmarna att dela och förstå varandras perspektiv*
2. *Det integrerar lärande på individuell nivå med lärande på teamnivå*
3. *Det identifierar gömda problem*
4. *Det dokumenterar best-practises och problem*
5. *Det ökar tillfredsställelsen av arbetet genom att ge personer feedback på deras arbete*
6. *I somliga fall kan det förbättra möjligheten att korrekt estimerar projektets kostnader*

Dessa punkter är generella och gäller reflektionsarbete inom både stora och små stora projekt (Myllyaho et al, 2004).

### 2.7.2 Reflektionsarbete inom agil utveckling

Inom agil utveckling är inte utvecklingsprojektet uppdelat i strikta faser utan systemutvecklingen sker istället simultant med samtliga faser som t.ex. design och analys i korta iterationer. Samtidigt bör utvecklingsteamet enligt det agila manifestets tolfte punkt (se bilaga 1) regelbundet stanna upp och reflektera över hur de ska bli mer effektiv och förändra processer och arbetssätt efter detta. På grund av den agila utvecklingens iterativa natur måste också reflektionsarbetet anpassas till det sättet att arbeta. Istället för att lägga ner mycket tid på att reflektera över stora delar av projektet bör tillfällen för reflektion utföras mer regelbundet och efter varje iteration inom projektet. En stor skillnad mellan reflektionsarbete inom stora och små projekt är att utvecklingsgruppen inom stora projekt ofta reflekterar och försöker dra lärdom av det slutliga projektresultatet. I mindre projekt och agil systemutveckling försöker utvecklingsgruppen istället dra lärdom av sitt arbete från tidigare iterationer och förbättra sig till kommande iterationer (Myllyaho et al, 2004).

Inom agil utveckling kallas ofta den reflekterande delen för ”*iteration retrospective*”, förutom att detta ses som ett tillfälle att förbättras ses det av många författare som ett tillfälle för att stärka lagkänsla inom teamet (Shore & Warden, 2007; Myllyaho et al, 2004).

Många författare med fokus på agil utveckling betonar vikten av att hålla ett retrospektiv efter varje sprint eller efter större milstolpar när det gäller större projekt (Myllyaho et al, 2004; Kniberg, 2007; Schwaber & Sutherland, 2007). Trots studier som visat positiva effekter av att regelbundet reflektera över sitt arbete hävdar flera författare att detta är ett moment som ibland prioriteras bort eller glöms bort (Schwaber & Sutherland, 2007; Kniberg, 2007; Myllyaho et al, 2004).

### 2.7.3 Kunskapsöverföring och retrospektivet

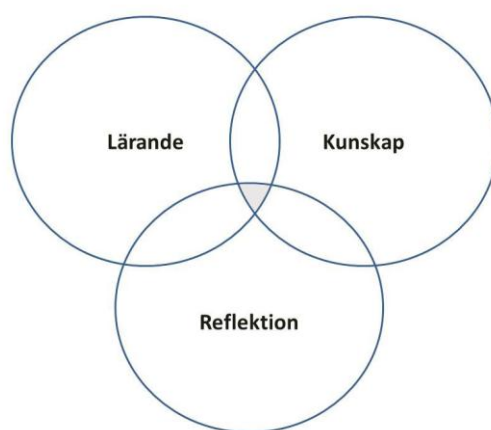
Som nämnts tidigare är reflektionsarbete inom projekt en metod för att överföra individuell kunskap till teamet och organisationen och på så vis förbättra det framtida arbetet. Dingsøy (2004) beskriver processen enligt följande:

”... *The main motivation is to reflect on what happened in the project in order to improve future practice- for the individuals that have participated in the project and the organization as a whole.*” (Dingsøy, 2004, s. 295)

Retrospektivet ses som ett tillfälle för kunskapsöverföring. Dingsøy skriver att det under ett retrospektiv överförs kunskap genom att använda en kombination av både ”socialization” och ”externalization” (se avsnitt 2.6.1). Genom att diskutera och lyssna på varandra överförs kunskap mellan individerna i en grupp. På så vis uppnås en delad förståelse för hur gruppen bör arbeta, vad som görs bra och vilka problem som existerar. Vilket i sin tur främjar att kunskap på individuell nivå kan övergå till kunskap på team-/organisationsnivå Dingsøy (2004).

## 2.8 Sammanfattning av litteraturstudie & studiens undersökningsmodell

SCRUMs sprint-retrospektiv är ett viktigt moment rörande arbetet med *The agile Manifesto*'s tolfte punkt för agil systemutveckling: "At regular intervals, the team reflects on how to become more effective, then tunes and adjust its behavior accordingly." (se bilaga 1). Då denna process till stora delar handlar om att reflektera över föregående iteration (*inspektion*) för att på så vis bli bättre till kommande sprintar (*anpassning*), har mycket av vår teorigenomgång haft fokus på områden som reflektion, lärande och kunskap. Figur 2.5 illustrerar den täta kopplingen mellan dessa områden som vi har funnit är svåra att åtskilja.



**Figur 2.5 - Illustrerar den starka kopplingen mellan reflektion, kunskap och lärande.**

Vår forskningsfråga behandlar hur och i vilken utsträckning SCRUMs sprint-retrospektiv används som ett verktyg för kontinuerlig förbättring avseende SCRUM-processen. Därav återger litteraturstudien teorier om reflektion, lärande, kunskap i kombination med agila metoden SCRUM och dess processer, med fokus på sprint-retrospektiv. Då begreppen är svåra att åtskilja behandlar studien det skuggade området (se figur 2.5).

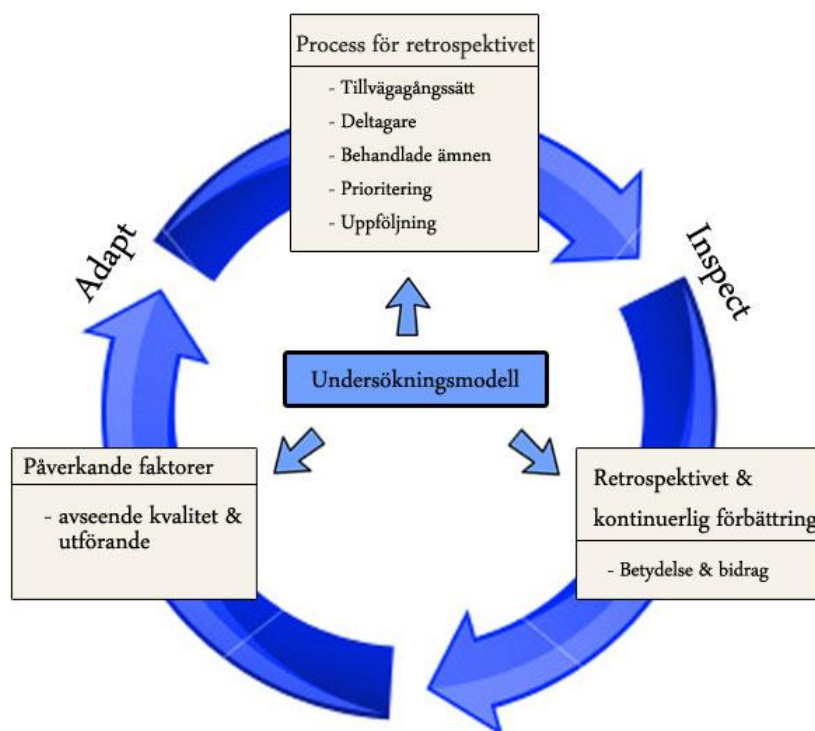
För att presentera teori som vi har funnit relevant för studien, har vi sammanställt en översikt över litteraturstudien (se tabell 2.1). Vi har själva delat in teorin inom fyra identifierade ämnesområden: *Lärande genom reflektion; Vikten av att utföra retrospektiv; Retrospektiv nedprioriteras; Sprint-retrospektivets process*". Vidare visar tabellen kopplingen mellan denna teori och vår framtagna intervjuguide.

Tabell 2.1 - Översikt över litteraturstudie med koppling till intervjuguide.

<b>Översikt – Litteraturstudie</b>		
<b>Ämnesområde</b>	<b>Litteraturstudie</b>	<b>Koppling till intervjuguide</b>
Lärande genom reflektion	Proto and Deutro learning, Visser (2003) Single and double loop learning, Schön & Argyris (1978) Reflection-in-action, Reflection-on-action, Schön (1983) Kolb (1994) Dingsøy (2004) Salo et al (2004) Mylyaho et al (2004)	<b>Fråga:</b> 5, 5.1, 6, 7, 8, 11
Vikten av att utföra retrospektiv	Mylyaho et al (2004); Shore & Warden (2007); Sutherland & Schwaber (2007)	<b>Fråga:</b> 5, 5.1, 11
Retrospektiv nedprioriteras	Schwaber & Sutherland (2007); Mylyaho et al (2004); Kniberg (2007)	<b>Fråga:</b> 3, 6
Sprint-Retrospektiv & Process	Derby & Larsen (2006); Kniberg (2007); Shore & Warden (2007)	<b>Fråga:</b> 4,4.1, 6, 7, 8, 9, 9.1, 10, 10.1, 11

För att ytterligare förtydliga vilka områden och aspekter vår undersökning kommer att beakta har vi tagit fram en undersökningsmodell (se figur 2.6). Modellen är framtagen med hänsyn till vår forskningsfråga och litteraturgenomgång.





**Figur 2.6 - Undersökningsmodell**

Vår forskningsfråga berör SCRUMs sprint-retrospektiv och dess bidrag till kontinuerlig förbättring. Följaktligen är det en förutsättning att undersöka hur dessa retrospektiv används och är utformade samt hur det faktiskt bidrar till att SCRUM-processen blir bättre.

Litteraturstudien behandlar uttrycken inspektion och anpassning som i korthet innebär att ett SCRUM-team ska stanna upp och reflektera (inspektera) över sprinten för att sedan förädla och justera (anpassa) SCRUM-processen i ett SCRUM-projekt. Ett medel för att inspektera och anpassa processen inom SCRUM är sprint-retrospektivet. Modellen (se figur 2.6) illustrerar det faktum att SCRUM-processen ständigt bör inspekteras och anpassas, vidare beskriver de tre olika boxarna tre olika aspekter som behandlas i vår studie:

#### *Process för retrospektivet*

Aspekten belyser tillvägagångssätt och hur retrospektivet faktiskt går till och är utformat.

#### *Retrospektivet & kontinuerlig förbättring*

Aspekten belyser retrospektivets betydelse och bidrag till kontinuerlig förbättring av SCRUM-processen.

#### *Påverkande faktorer*

Aspekten belyser eventuella faktorer som kan ha påverkan på utförandet och kvaliteten av sprint-retrospektivet.

Undersökningsmodellen kommer att användas för att presentera vår empiri och diskussion på ett systematiserat sätt.

## 3 Metod

Vår forskningsfråga rör kontinuerlig förbättring inom utvecklingsmetoden SCRUM och huruvida reflektionsarbete inom metoden bidrar till förbättring, och hur detta utförs. Då metoden inte innehåller tydliga riktlinjer för hur detta arbete ska se ut förväntar vi oss därför att verkligheten är väldigt nyanserad och subjektiv i förhållande till varje projekt eller organisation. Således ligger det i vårt intresse att skapa oss en bättre förståelse för ett oklart ämne genom att få fram en nyanserad beskrivning av problemområdet, sprint-retrospektiv. Därav förbehåller vi oss möjligheten att vara öppna till vad undersökningen kommer att visa och inte på förhand till fullo bestämma vad vi letar efter. En kvalitativ ansats har därför valts för insamlandet av empiri. Enligt Jacobsen (2002) sätter ett kvalitativt förhållningssätt för metod få begränsningar för vilken information som intervjupersoner kan ge. Vidare beskrivs en kvalitativ ansats lämplig i de fall då undersökaren vill belysa samband mellan individ och kontext.

### 3.1 Tillvägagångssätt

Innan arbetet med att skriva denna uppsats påbörjats hade vi för avsikt att skriva om agila systemutvecklingsmetoder. Eftersom vi insåg att ämnet var för brett och svårt att hantera i en kandidatuppsats beslutade vi oss för att inrikta oss för en metod inom agil utveckling, SCRUM. Efter möte och diskussioner med vår handledare förstod vi att även detta blir var ett för stort ämne att behandla. Vi skaffade oss en bättre förståelse för ämnet genom att gå igenom mycket litteratur relaterat till agil utveckling och framförallt SCRUM. Vidare uppmärksammades vårt nuvarande problemområde, sprint-retrospektivet. Detta moment inom metoden SCRUM beskrevs som en fundamental del av SCRUM-processen, samtidigt ansåg vi att det var ett intressant och relativt utforskat ämne. Detta gjorde att vi till slut valde att avgränsa studien till just retrospektivet inom SCRUM. Utifrån detta växte sedan vår frågeställning fram (se avsnitt 1.3).

Vi insåg snabbt att litteraturen som behandlade metoden SCRUM var otillräcklig för att kunna besvara vår forskningsfråga. Vi ansåg att en del av denna litteratur var något vinklad och ytlig. Då sprint-retrospektivet är ett tillfälle inom metoden för att reflektera, inspektera och sedan anpassa SCRUM-processen valde vi att komplettera ovanstående litteratur. Denna litteratur behandlade ämnen så som kunskap, lärande, reflektion och efteranalys inom projekt.

Utifrån vår litteraturgenomgång tog vi fram en undersökningsmodell (se figur 2.6) som beaktar de aspekter vi anser är viktiga att belysa för att kunna besvara vår frågeställning. Undersökningsmodellen har sedan använts för att på ett systematiskt sätt presentera och diskutera empirin.

## 3.2 Intervju

### 3.2.1 Semi-strukturerad intervju

Som redan nämnts berör forskningsfrågan hur förbättringsarbete genomförs med hjälp av reflektionsarbete i ett så kallat sprint-retrospektiv. Eftersom detta arbete mycket väl kan utföras på olika sätt i olika kontext anser vi att intervjuer bör genomföras med intervjupersoner som har god erfarenhet av systemutveckling och färdigheter inom agil utveckling. Detta för att få fram en nyanserad beskrivning lämplig för ändamålet.

Då vi anser att en kvalitativ ansats är mest lämplig har vi valt att utföra ett mindre antal intervjuer med möjligheten att skapa oss en bättre förståelse och beskrivning över intervjupersonens verklighet och synsätt. Jacobsen (2002) beskriver en kvalitativ ansats som tidskrävande både för datainsamling och för analysfasen. Därför är ett mindre antal intervjuer passande för denna typ av undersökning.

Intervjuerna har vi valt att utföra med en semi-strukturerad ansats, på så vis har vi möjlighet att få en öppen och utförlig dialog med våra intervjupersoner. De frågor vi tagit fram är både öppna och slutna i sin karaktär, de öppna frågorna förväntas inte kunna generera exakta svar, frågorna kan tolkas på olika sätt och kan därför behöva följas upp av spontana följdfrågor. Jacobsen (2002) skiljer på strukturerade och ostrukturerade intervjuer, där ostrukturerade intervjuer använder öppna frågor och strukturerade intervjuer använder slutna frågor. En intervjuguide har dessutom tagits fram som styrmedel för samtliga intervjuer.

Jacobsen (2002) skriver att en kvalitativ intervju inte bör vara helt ostrukturerad. Undersökaren bör i regel använda en intervjuguide som en översikt över de ämnen intervjun kommer beröra. Detta för att inte missa viktiga aspekter som kan vara av stor vikt i undersökningen.

Vi har valt att utföra samtliga intervjuer ansikte mot ansikte. Enligt Jacobsen (2002) är dessa intervjuer att föredra framför distansintervjuer, så som mail eller telefon, då det är lättare att skapa en förtrolig stämning med intervjupersonen. På så sätt kan det bli lättare att ta upp känsliga ämnen samtidigt som vi får en känsla för hur personen agerar under intervjun. Om undersökaren märker att intervjupersonen blir obekvämd kan han eller hon helt enkelt byta ansats eller fråga. Då minskar risken att intervjupersonen sluter sig och blir sämre som informationskälla.

Undersökningar har enligt Jacobsen (2002) också gjorts som visar att det är lättare för intervjupersonen att tala osanning vid en intervju på distans. Risken med en intervju via mail, där intervjupersonen har lång tid på sig att formulera sina svar, är att han eller hon formulerar svaren på ett sådant sätt som låter bra, men kanske inte alltid speglar sanningen. En anledning till att en distansintervju kan vara att föredra är att det kan reducera kostnaderna för undersökningen (Jacobsen, 2002). Vi valde därför att söka kontakt med personer och företag som låg i samma stad, eller nära anslutning till vårt lärosäte.

### 3.2.2 Intervjuguide

Intervjuguiden består av 11 huvudfrågor (se bilaga 2). Frågorna baseras utifrån vår litteraturstudie och undersökningsmodell. Vi valde att ställa allmänna frågor tidigt i intervjuerna. Enligt Jacobsen (2002) är det viktigt att inleda en intervju på detta sätt för att låta intervjupersonerna själva få berätta det dem tycker är viktigt, istället för att ställa precisa frågor för att få svar på vad vi tycker är viktigt. Allt eftersom intervjun fortgick ställdes mer precisa frågor. Intervjuerna avslutades sedan med en lättare flervalsfråga, där intervjupersonen fick möjlighet att utveckla sina svar. Detta för att intervjupersonen skulle ha en bra känsla över intervjun när intervjun avslutats.

Intervjuguiden innehåller även två stycken flervalsfrågor. Här fick intervjupersonen gradera ett antal påståenden på en fyrgradig skala. Enligt Jacobsen (2002) finns det forskning som både talar för och emot att ge respondenten en mitt-kategori. Vi valde dock att inte ha en mittkategori i vår undersökning för att på så vis låta informanterna välja sida. Detta för att undvika att personen väljer mittkategorin bara för att han eller hon inte vill svara på frågan. I samband med dessa frågor lät vi intervjupersonen utveckla sina svar om de kände att de ville ge en djupare förklaring till frågan.

### 3.2.3 Urval & Informanter

Utifrån vår strävan att få nyanserade svar och få fram ett resultat med en överensstämmande verklighetsbild har de informanter som deltagit i vår studie valts ut utifrån nedan beskrivna aspekter.

Samtliga medverkande i undersökningen har god erfarenhet av systemutveckling. Däremot skiljer sig erfarenheten åt genom olika stort användande av agila och traditionella systemutvecklingsmetoder, varav vi har försökt att hitta informanter med kunskap och erfarenheter från båda delar. Vi tror att många SCRUM-masters har god erfarenhet från systemutveckling, både från traditionella och från agila systemutvecklingsmetoder, därför tror vi att valda informanter kan ge oss en rättvis bild av hur det ser ut i verkligheten.

Samtliga informanter har dessutom valts ut utifrån att de i dagsläget agerar som SCRUM-master i ett eller flera SCRUM-projekt eller att de sedan tidigare har erfarenhet av att leda SCRUM-team i egenskap av SCRUM-master. Dessa i sin tur arbetar eller har arbetat i olika typer av organisationer, allt från mindre konsultbolag till större organisationer med SCRUM som utvecklingsmetod. Vidare har vi försökt att hitta både manliga och kvinnliga informanter. Detta är inget vi har valt att göra för att analysera skillnader i utföranden mellan kön, utan för att bibehålla ett brett spektrum av informanter.

Eftersom samtliga uppgiftslämnare har olika bakgrund och erfarenheter av SCRUM samt arbetat i olika typer av organisationer och projekt så kommer vi att vara försiktiga med att dra några generella slutsatser och beakta varje informants kontext kritiskt. I tabell 3.1 har vi sammanställt en presentation över studiens informanter.

**Tabell 3.1 – Sammanställning över informanter.**

Informant P1	<p><b>Konsult - Systemutvecklare</b></p> <p><i>Kön: Man</i>  <i>Utbildning: Civilingenjör, Datateknik</i>  <i>Certifierad SCRUM-Master: Nej</i></p> <p><i>Erfarenhet av traditionell respektive agil utveckling:</i></p> <p>Jobbat sedan 2006 som systemutvecklare.  Erfarenhet av traditionell systemutveckling i form av vattenfallsmetodik.  Erfarenhet av agil utveckling med SCRUM som metod, med inslag av XP-relaterade aktiviteter som par-programmering.</p> <p><i>Relevant för undersökningen:</i></p> <p>Arbetar just nu i ett SCRUM-projekt med rollen som både utvecklare och SCRUM-master med en stor organisation inom tillverkningsindustrin som kund.</p> <p><i>Eventuell påverkan av resultat - viktigt att beakta</i>  Informanten arbetar i dagsläget som SCRUM-master över ett distribuerat SCRUM-team, team-medlemmarna arbetar således inte på samma plats.  Informanten har som ovan beskrivits inte heller någon utbildning som SCRUM-master.</p>
Informant P2	<p><b>Konsult - Systemutvecklare, Agil evangelist, Agil Coach</b></p> <p><i>Kön: Man</i>  <i>Utbildning: Systemvetenskap, Datalogi</i>  <i>Certifierad SCRUM-Master: Ja</i></p> <p><i>Erfarenhet av traditionell respektive agil utveckling:</i></p> <p>Jobbat tio år som systemutvecklare.  – Erfarenhet av traditionell systemutveckling i form av vattenfallsmetodik.  – 6-7 års erfarenhet av agil utveckling, XP och SCRUM.</p> <p><i>Relevant för undersökningen:</i></p> <p>Har arbetat inom flera SCRUM-projekt. Arbetar då och då som inhyrd facilitator över sprint-retrospektiv i andras agila projekt. Är även aktiv inom olika agila intressegrupper.</p>

Informant P3	<p><b>Konsult, Systemutvecklare</b></p> <p><i>Kön: Man</i>  <i>Utbildning: Programvaruteknik</i>  <i>Certifierad SCRUM-Master: Ja</i></p> <p><i>Erfarenhet av traditionell respektive agil utveckling:</i></p> <p>Jobbat 6 år som systemutvecklare.  – Större erfarenhet av traditionell systemutveckling i form av vattenfallsmetodik.  – Erfarenhet av agil utveckling sedan 2010, SCRUM.</p> <p><i>Relevant för undersökningen:</i></p> <p>Har arbetat som SCRUM-master över 2 olika team i 2 olika projekt. Har inte sttit med som utvecklare i ett SCRUM – projekt.</p>
Informant P4	<p><b>Systemutvecklare</b></p> <p><i>Kön: Man</i>  <i>Utbildning: Dataingenjör</i>  <i>Certifierad SCRUM-Master: Nej</i></p> <p><i>Erfarenhet av traditionell respektive agil utveckling:</i></p> <p>Jobbat 6 år som systemutvecklare.  – Större erfarenhet av traditionell systemutveckling i form av vattenfallsmetodik.  – Erfarenhet av agil utveckling sedan 2010, SCRUM.</p> <p><i>Relevant för undersökningen:</i></p> <p>Har arbetat som SCRUM-master över 2 olika team i 2 olika projekt. Har inte sttit med som utvecklare i ett SCRUM – projekt.</p> <p><i>Eventuell påverkan av resultat - viktigt att beakta</i>  Informanten har som ovan beskrivits ingen utbildning som SCRUM-master.</p>
Informant P5	<p><b>Projektledare</b></p> <p><i>Kön: Kvinna</i>  <i>Utbildning: Mikrobiolog, Matte/NO lärare</i>  <i>Certifierad SCRUM-Master: Ja</i></p> <p><i>Erfarenhet av traditionell respektive agil utveckling:</i></p> <p>Jobbat med systemutveckling i 15 år på diverse olika bolag, främst inom telekom. 2007 jobbade hon för första gången med agil systemutveckling och SCRUM.</p> <p><i>Relevant för undersökningen:</i></p> <p>Har jobbat i diverse SCRUM-projekt, både som SCRUM-master och produktägare.</p>

### 3.3 Datainsamling & analysmetod

Data till studien har samlats in genom att hålla semistrukturerade intervjuer med intervjupersoner som har relevant erfarenhet av agil systemutveckling. Intervjuerna spelades vidare in, med samtycke från intervjupersonerna, för att sedan transkriberas och användas vid analysen. Vi har strävat efter att hålla intervjuernas längd till ca 45 minuter för att inte ta upp för mycket tid för våra informanter. Intervjuerna har hållits på intervjupersonernas egen arbetsplats med undantag för en intervju som vi höll på institutionen för Informatik, Lunds Universitet, dock i nära anslutning till personens arbetsplats.

Efter varje intervju gick vi igenom det inspelade materialet och transkriberade samtalen. Under tiden vi transkriberade materialet förde vi även anteckningar och noterade intressanta uttalanden av värde för studien. Transkripten skickades sedan till informanterna för godkännande. När transkriberingen var färdig och godkänd av informanterna började vi analysera materialet och söka relevant data. Relevant data valdes ut och kategoriserades utifrån vår undersökningsmodell genom att se varje intervju i sin helhet och inte bara söka svar utifrån frågor i vår intervjuguide. Eftersom många av våra frågor var av öppen karaktär och vi således kunde få relevanta svar under olika frågor och delar av intervjuerna ansåg vi att det var ett lämpligt tillvägagångssätt. För att säkerställa att vi inte missat eller tolkat viktig data fel har vi tillsammans diskuterat informanternas svar och gått igenom transkripten noggrant. Undersökningsmodellen ligger även till grund för hur vi presenterar vår analyserade empiri.

Vår intervjuguide innehöll även två stycken flervalfrågor där informanterna fick möjlighet att utveckla sina svar (se avsnitt 3.2.2). Dessa flervalfrågor presenterades med hjälp av tabeller i analyskapitlet. I dessa tabeller återfinns informanternas svar, vilket är en siffra på en fyrgradig-skala. I den mån informanten har valt att utveckla svaret har vi även kommenterat i tabellen.

### 3.4 Validitet & Reliabilitet

Enligt Jacobsen (2002) är det viktigt att kritiskt granska undersökningar för att minimera problem med giltighet och tillförlitlighet, d.v.s. validitet och reliabilitet. Detta innebär att undersökaren bör vara kritisk till huruvida data är tillförlitlig och om kvaliteten på den data som samlats in är tillräcklig.

Vidare skiljer författaren på intern och extern giltighet. Intern giltighet syftar på om data och resultat kritiskt granskats och beskriver ett fenomen på ett korrekt sätt. Den externa giltigheten syftar på om studiens resultat kan generaliseras. En kvalitativ metod har i regel inte för avsikt att generalisera utifrån ett urval av enheter till en större grupp. Syftet är istället att få grepp om och försöka fastslå ett fenomenets omfång eller frekvens (Jacobsen, 2002).

Studiens syfte är att undersöka hur arbetet med reflektionsarbete och sprint-retrospektiv bidrar till kontinuerlig förbättring i praktiken, detta genom en kvalitativ ansats. Därav har vi valt ut intervjupersoner med praktisk erfarenhet av agil systemutveckling och med erfarenhet av rollen som SCRUM-master. Antalet intervjupersoner är få och djupgående intervjuer utförs för att få en så korrekt insamlad data som möjligt. För att ytterligare säkerställa detta har vi lagt stor vikt på att uppgiftslämnare ska vara oberoende av varandra, ha god kunskap om ämnet och att de ska vara förstahandskällor utan klara motiv att ljuga. Detta för att kunna bibehålla en intern giltighet (Jacobsen, 2002).

Resultaten av denna studie kommer endast att återspegla hur kontinuerlig förbättring med hjälp av sprint-retrospektiv utförs och fungerar i våra uppgiftslämnarens kontext. Därav kommer resultaten från vår empiri inte kunna generaliseras på systemutveckling med SCRUM i allmänhet. På grund av den kvalitativa ansatsen kommer resultatet inte vara representativt i ett externt kontext. Däremot kommer vi kunna jämföra och generalisera från vår empiri till vår teoribas, med andra ord belysa likheter och olikheter (Jacobsen, 2002).

I vår litteraturstudie använder vi oss av litteratur som behandlar ämnet agil utveckling. Denna litteratur kan ses som vinklad då författarna i stor utsträckning är förespråkare för agil systemutveckling. Dessa artiklar och böcker har därför fått läsas med stor försiktighet på grund av dess subjektiva karaktär. För att nyansera vår bild av reflektion och kontinuerlig förbättring inom SCRUM har vi därför tittat på mer allmänna teorier om hur lärande, kunskap och reflektion sammanhänger.

### 3.5 Etiska aspekter

Jacobsen (2002) skriver om etiska aspekter angående förhållandet mellan forskare och undersökt. Författaren tar upp tre olika grundkrav som bör beaktas gällande detta.

#### *Informerat samtycke*

Här syftar Jacobsen på att uppgiftslämnaren frivilligt ska delta i undersökningen samt att denna har fått full information och har full förståelse för vad undersökningen innebär. Detta har vi gjort genom att tydligt beskriva vad vår undersökning kommer att behandla och vilka ämnen som kommer att diskuteras. Således har vi inte haft dolda avsikter med undersökningen.

#### *Rätt till ett privatliv*

Detta grundkrav handlar om möjligheten att identifiera uppgiftslämnare utifrån en undersökning. Undersökaren bör således beakta att viss information kan vara känslig för uppgiftslämnaren. Samtliga uppgiftslämnare har anonymiserats i denna studie och har fått information att så ska ske.



### *Krav på riktig presentation av data*

Detta krav syftar på att den information som delges inte ska tas ur sitt sammanhang. För att uppfylla detta krav ställde vi följdfrågor och bad om utförligare förklaringar i de fall något var oklart. För att säkerställa att ingen information missförstods eller togs ur sitt sammanhang gick vi tillsammans noggrant igenom analysen. Vidare delgav vi samtliga uppgiftslämnare en kopia av transkriberat material.

## **3.6 Kritik mot tillvägagångsätt**

Vår undersökning med en kvalitativ ansats innebar att vår intervjuguide blev utformad med en semistrukturerad karaktär. Vi visste följaktligen heller inte hur våra frågor skulle tolkas utan fick istället lägga in följdfrågor under intervjuerna när vi kände att svaren inte var tillräckliga. I efterhand ser vi att våra frågor skulle kunnat vara mindre öppna och mer precisa, vissa frågor borde vart ställda mer direkta utifrån vår undersökningsmodell. Exempelvis kunde en fråga som ”Vad bidrar SCRUMs retrospektiv till?” gett oss empiri som kunde vart lättare att sammanställa och tolka. Samtidigt har våra öppna frågor lett oss in på intressanta diskussioner och ämnen vi på förhand inte hade funderat på. Vilket har gett oss en bredare syn på forskningsområdet.

Vidare togs en flervalsfråga fram för att kunna få en överblick över vad SCRUMs retrospektiv faktiskt ger och bidrar till. Denna flervalsfråga hittas i intervjuguiden (se bilaga 2, fråga 11). Flervalsfrågan var baserad på 6 olika påståenden om bidragande faktorer som reflektionsarbete inom projekt i allmänhet kan innebära. Enligt författaren Myllyaho et al (2004) är dessa påståenden generella och kan gälla både stora och små projekt. I efterhand ser vi att påståendena var tolkningsbenägna och kan ha gjort att våra informanter kan ha tolkat påståendena lite olika. En förklaring kan vara att dessa påståenden är framtagna utifrån reflektionsarbete och efteranalys inom projekt i allmänhet och inte riktat mot just systemutvecklingsmetoden SCRUM. Metoden SCRUM kan vara implementerad på olika sätt och vara utformat olika i varje kontext vilket gör att våra svar är svårare att generalisera. Informanternas svarade väldigt lika på en del påståenden, medans andra svar skilde sig.

Utvalda informanter har gett oss bra och relevant data under intervjuerna. De har tack vare olika bakgrund gett oss en bred bild av vårt forskningsområde. Däremot hade vi i efterhand gärna velat ha intervjuer med fler personer med gedigen erfarenhet av agil utveckling, här syftar vi på t.ex. agila coacher eller evangelister. Vidare anser vi att antalet intervjuer var begränsad, för att ge resultatet större tillförlitlighet kunde både fler informanter valts ut och flera intervjuer utförts.

Vår litteraturstudie och val av teorier har valts ut utifrån vårt valda forskningsområde. Vi har försökt att täcka upp områden som lärande, kunskap och reflektion i kombination med systemutvecklingsteori och agila metoder. Det är möjligt att litteraturstudien hade kunnat kompletteras med ytterligare områden eller teorier. Vi anser dock att vi behandlat den teori som är nödvändig för att kunna genomföra studien och besvara vår forskningsfråga.

## 4 Presentation av empirisk data

### 4.1 Relevant för undersökningen

För att få en uppfattning om hur strikt de olika informanterna arbetat med metoden SCRUM enligt de regler och riktlinjer som vi funnit i relevant litteratur tog vi fram en flervalsfråga som ställdes tidigt i intervjuerna (se bilaga 2). Vi frågade informanterna:

*”SCRUMs ramverk består av ett antal regler och riktlinjer. I egenskap av SCRUM-master vill vi att du lyssnar på ett antal påståenden och säger en siffra, på en skala mellan 1-4 beroende på om det är något som du följer eller inte.”*

Vi gav informanterna följande svarsalternativ:

1. *Stämmer inte*
2. *Stämmer ibland*
3. *stämmer oftast*
4. *stämmer alltid*

Svaren från flervalsfrågan presenteras i tabell 4.1.

**Tabell 4.1 - Överblick över informanternas användande av SCRUM.**

Sammanställning av fråga 3 (se bilaga 2)	P1	P2	P3	P4	P5
<b>Artefakter</b>					
Vi har alltid en Product Backlog	4	4	4	4	3
Vi har alltid en Sprint Backlog	4	4	4	4	3
Vi har alltid en Sprint-Burndown Chart	4	4	4	4	4
<b>Process</b>					
Vi har alltid ett Sprint Planning Meeting	4	4	4	4	4
Vi har alltid en Sprint	4	4	4	4	4
Vi har alltid en Daily SCRUM/ Stand-up	4	4	4	4	4
Vi har alltid ett Sprint-Review Meeting	4	4	4	4	4
<i>Vi har alltid ett Sprint Retrospective</i>	4	4	4	4	4
Time-boxar ni alla ovan nämnda processer?	4	4	4	4	2-3
<b>SCRUM-TEAM</b>					
Vi har alltid en SCRUM-master	4	4	4	4	3
Vi har alltid en Produktägare	4	4	4	4	3

Samtliga uppgiftslämnare uppger att de i stor utsträckning inkluderar viktiga artefakter, processer och roller i sina SCRUM-projekt. Informant P5 förklarade att hon i sitt första SCRUM-projekt inte hade någon utsatt Product-/Sprint-backlog, SCRUM-Master eller

produktägare. Hon sa dock att detta upprättades efterhand och betonade samtidigt vikten av att dessa roller och artefakter fanns med. Samtliga informanter har arbetat som SCRUM-masters, P5 har däremot under ett projekt arbetat både som SCRUM-master och produktägare.

## 4.2 Empiriskt resultat

### 4.2.1 Process för retrospektivet

#### *Tillvägagångssätt*

Samtliga informanter var eniga om att retrospektivet var till för att ta upp problem som dykt upp under föregående sprint och sedan tillsammans försöka hitta lösningar på de problem som kom upp.

Intervjuperson P2 och P3 inledde mötet med ett moment som de kallade ”setting the stage” för att skapa sig en uppfattning om personerna i teamet och helt enkelt komma in i rätt stämning. Därefter ägnade de sig åt någon form av datainsamling. De övriga intervjupersonerna öppnade istället mötet med ett moment för datainsamling.

P1 och P5 samlade i första hand in data från deltagarna med hjälp av en öppen diskussion. P5 valde dock att ibland skriva ned problemen på gula post-it lappar och redovisa dessa anonymt. Hon motiverade detta med att retrospektivet i somliga fall kunde ta upp känsliga saker.

P2 och P3 valde ofta att dela ut post-it lappar till alla i teamet och låta de skriva ner vad de ansåg har gått bra och det som gått mindre bra under sprinten. Även P4 använde tekniken med att låta teamet skriva ned bra och dåliga saker, även om redovisningen av lapparna skiljde sig från P2 och P3s sätt att redovisa det på.

När all data var insamlad från teamet redovisades detta på något sätt. P1 och P5 valde att skriva ned alla de sakerna som kom upp under retrospektivet på papper eller i någon form av dokument. De övriga intervjupersonerna hade efter datainsamlingen ett moment där de grupperade den data som hade samlats in och gav alla grupperna ett namn. P2 och P3 grupperade data genom att låta teamet sätta upp sina post-it lappar på en white-board tavla. Därefter var det fritt fram för alla i rummet att flytta runt lapparna på tavlan och placera dem i grupper där de tyckte att problemen hörde hemma. När alla lappar hade hamnat i en grupp namngavs grupperna.

P4 lät alla i teamet läsa upp sina lappar och de områden som kom upp delades in i olika grupper.

P1 och P5 valde att inte prioritera de olika problemområden som kom upp under retrospektivet. Allt som kom upp tog de med sig och försökte förbättra till kommande sprint.

P5 menade att det ofta räckte med att problem kom upp och diskuterades på mötena och att de därefter automatiskt löstes av teamet.

Efter datainsamlingen och datagrupperingen valde P2, P3 och P4 att inom teamet prioritera det eller de problem som de ansåg vara viktigast att jobba med under kommande sprint. P2 och P3 valde att prioritera problemområden med hjälp av röstning inom teamet. Ett exempel kunde vara att samtliga i teamet fick tre magneter var att sätta upp på det problem de ansåg var viktigast. De fick sätta samtliga magneter på ett problem eller sprida ut dem på olika problem. P4 valde att under sina retrospektiv prioritera de områden som skulle tas upp genom öppen diskussion.

Antalet problem som informanterna valde att prioritera till kommande sprint skiljde sig åt. P3 och P4 valde att ta med sig max tre problemområden att jobba med. P2 tog aldrig med sig fler än ett problem att jobba med under kommande sprint.

Exempel på tekniker som P2 och P3 använde sig av för att hitta roten till problemen var ”fish-bone/cause-and-effect diagram” eller ”Five why’s”. Se Derby & Larsen (2006) för utförlig beskrivning av metoderna.

### *Deltagare*

P1 valde att endast låta SCRUM-mastern och teamet delta under retrospektivet. Han hade tidigare testat att även låta produktägaren delta men har under tiden som SCRUM-master kommer fram till att detta inte alltid var lämpligt. P1 ansåg inte att det gav teamet någonting när produktägaren satt med under retrospektivet utan snarare försvårade processen.

P2 lät ibland, om teamet tyckte att det var ok, andra intressenter förutom SCRUM-master och teamet delta på retrospektivet. Anledningen var att externt deltagande kan vara ett sätt att utbilda andra intressenter i hur metoden fungerar. Men han brukar föreslå att inte låta externa personer med auktoritär roll vara med allt för ofta.

*”[...] det blir väldigt intimt, gör man ett bra faciliterat retrospektiv så blir det väldigt intimt och väldigt känsloladdat och du vill inte ha auktoritetspersoner där för att det kan lätt förstöra.” P2.*

P2 sa även att det i vissa fall kan vara lämpligt att ta in en extern facilitator till retrospektiven för att teamet ska få en annan synvinkel på problemen och ”röra om i grytan”.

Retrospektivets deltagare under P3s projekt var SCRUM-mastern och teamet, inte produktägaren. Dock så valde P3 att ibland använda sig av en extern facilitator eller att låta teammedlemmarna själva facilitera retrospektivet för att få lite variation på det.

P4 valde precis som P1 att inte låta produktägaren vara med under retrospektivet. När retrospektivet var färdigt och det hade summerats så lät de dock produktägaren ta del av det som kommit upp under mötet.

P5s inställning till retrospektiv såg något annorlunda ut i förhållande till resten av intervjupersonerna. Hon tyckte att det var viktigt att inkludera SCRUM-master, team och produktägaren i retrospektiven.

*”Vi ville få ihop det så att den personen också kände sig delaktig. Det var ju inte alltid de skötte sig heller. Vi hade ett retrospektiv på allt när vi körde. Vi tyckte att det var schystast...”*  
P5

P5 hade själv även jobbat som produktägare och tyckte att deltagande under retrospektivet var nödvändigt för att få en inblick i hur teamet fungerade.

### *Behandlade ämnen*

Samtliga informanter uppger att de tar upp saker som gått bra, saker som gått dåligt och saker som behöver förbättras till kommande arbete.

P1 brukar inleda sina retrospektiv med att alla i teamet får säga två saker som gått bra för att på så vis få igång diskussionen. Under mötet är det fritt att ta upp allt från implementationsdetaljer till kommunikation inom teamet. Han säger dock att det som brukar komma upp under retrospektiven gäller kodningen och implementationsdetaljer. De diskuterar ofta koden som skrivits under föregående sprint och om den behöver refaktoreras.

Även P2 sa att alla typer av problem kan lyftas upp på retrospektivet. Det kan t.ex. gälla människor, teknik eller kommunikation. Han lägger mycket tyngd på de mjuka värdena inom teamet och menar att en förutsättning för att arbetet ska gå bra är att människorna mår bra. Det kommer ofta upp ämnen som inte är tekniska. Det kan t.ex. vara externa eller privata problem inom teamet.

P3 var noga med att påpeka vikten av att inte bara diskutera problem utan även lyfta fram de sakerna som gått bra under föregående sprint. Han förklarade detta under intervjun:

*”[...] det är bra om man inte bara fokuserar på det negativa. Utan även försöker lyfta upp lite av det positiva under arbetets gång för att det blir gärna fokus enbart på det negativa. Man behöver se vad man gjort bra också. Det tror jag är viktigt att lyfta fram.”* P3

P4 har också tre frågor på sin agenda inför varje retrospektiv. Han frågar teamet vad som gått bra, vad som gått dåligt och vad de kan förbättra. Det kan enligt honom handla om allt från kod och teknik till hur väl teamet jobbar med processen.

P5 säger också att retrospektivet inte är begränsat till några specifika ämnen. Enligt henne kan teamet ta upp allt från tekniken i minsta detalj till andra problem som t.ex. att SCRUM-mastern varit sjuk.

### *Prioritering av problem*

Informanterna besvarade frågor rörande om de prioriterade de problem som kom upp under retrospektivet, dvs. välja ut ett mindre antal problem att förbättra under kommande sprint.

P1 och P5 väljer i sina SCRUM-projekt att göra en lista över alla problem som dykt upp under föregående sprint och arbeta med att förbättra samtliga områden till kommande sprint. P1 poängterar dock att de endast lyfter fram problem som är relevanta för kommande sprint.

Att välja ett problem och inte mer var något som P2 var väldigt noga med. Han drog paralleller till *Lean* och förklarade det som följer:

*"[...] hittar ni den största flaskhalsen och tar bort den. För att efter att ni har ändrat systemet, det är ju då "System thinking". Systemet ser ut på ett visst sätt och löser man då ett problem, då förändras hela systemet" P2*

Vidare sa han:

*"[...] spelar ingen roll om man skapar en lista med 1,2,3,4,5 problem, för när vi har löst problem nr 1 så kommer allt vara omkastat igen. Så man vet inte var flaskhals nr 2 kommer sitta." P2*

P3 och P4 väljer inte att enbart lyfta fram ett problem men begränsar det ändå till tre stycken per retrospektiv. Båda personerna hade tidigare låtit bli att begränsa antal problem som tas med till kommande sprint men har efter erfarenhet kommit fram till att det blir bäst om de begränsar antalet problem. P3 sa: *"Det kändes som vi fick mer gjort då."* när han förklarade varför de valt att begränsa antal problem som tas med till kommande sprint.

### *Uppföljning*

Informanterna fick besvara frågor rörande om de gjorde någon slags uppföljning av arbetet med retrospektivet för att kontrollera att problem faktiskt blir lösta.

Den enda av våra intervjupersoner som hade en fastställd uppföljning på problemen var P4. Inom hans organisation väljer de att skicka problemen vidare till en *Process Improvement* grupp som ska se till att problemen blir åtgärdade. Denna grupp ligger dock utanför själva SCRUM processen.

De övriga intervjupersonerna hade ingen strikt uppföljning av problemen som kommit upp under retrospektivet.

P1 ansåg att uppföljningen och det som teamet lärt sig fanns i koden. Den enda uppföljningen som de använde sig av var när de granskade koden som producerades.

Både P2 och P5 sa att det positiva med retrospektiv var att det var självkontrollerande på så vis att om ett problem kommer upp på nästa retrospektiv så har det bevisligen inte blivit löst.

P2 var noga med att poängtera att det är teamets ansvar att se till att problemen löses, behöver de däremot hjälp av SCRUM-mastern så är det bara att säga till.

P4 menar att ansvaret för problemen kunde hamna på både SCRUM-master eller teamet. I slutändan hamnade dock ansvaret oftast på SCRUM-mastern.

#### 4.2.2 Retrospektivet & kontinuerlig förbättring

##### *Betydelse och bidrag*

Informant P1 anser att sprint-retrospektivet är det som ger mest inom metoden SCRUM. Han motiverar det med att retrospektivet är ett tillfälle för att involvera hela teamet och där de får ta beslut gemensamt. Genom att göra detta kan teamet utan att peka på enskilda individer belysa vad de ska undvika och vilken väg de ska gå. Vidare menar han att retrospektivet hjälper teamet att bli mer effektivt samtidigt som det förbättrar systemet och systemutvecklarnas kodning.

P3 ser retrospektivet och reflektion som en viktig del för att bli bättre. Han menar att det inte enbart gäller för metodiken SCRUM men att retrospektivet lyfts fram som en mer självklar del i den. Genom att stanna upp och reflektera över vad teamet håller på med kan de förebygga att arbetet skenar iväg åt det ena eller det andra hållet. P4 delar denna syn och tillägger att det är positivt att teamet får avsatt tid för just detta.

P2 beskriver retrospektivet som ett verktyg för att förbättra processen och säger samtidigt att retrospektivet är elementärt inom agil utveckling. Genom att utföra ett retrospektiv menar även han att teamet blir mer effektivt.

P2 säger:

*”Vad jag tycker om det är att det är den absolut viktigaste byggstenen inom alla agila metodiker som finns. Det är helt ohållbart att kalla någonting för agile om man inte gör någon form av retrospektiv. Om jag går in i ett team som inte kör agile idag och som är sugna på att testa så är det här alltid det jag börjar med. Det är alltid första steget. Du behöver inte ens ha iterationer för att köra retrospektiv, du kan bara säga att vi kör ett retrospektiv varje månad.”*

P5 delar P2s syn om att retrospektivet är en viktig och positiv del av agila metoder. Hon beskriver vidare hur det i en mer traditionell vattenfallsmetodik inte lyfts fram på samma sätt men att det kan bero på vilken typ av projektledare projektet har. Hon menar vidare att man kan eller bör lägga in ett retrospektiv även inom vattenfallsmetodiken men att det inom agila metoder är mer uppenbart att det blir gjort. Hon säger bland annat:

*”I ett vattenfallsprojekt så kan det ju tendera att bli som så att man kör igång projektet och tänker: ”Oh nej, nu skulle vi ju slutrapportera det här. Nu kan vi fråga alla som varit med*

*vad de tyckte!". Det kan tendera till att bli så. Här är det lite mer självklart att man gör det. Det ingår i processen på ett mer naturligt sätt. Så det tycker jag är bra, absolut." P5*

På frågor rörande vad retrospektivet faktiskt bidrar till för att SCRUM-teamet ska bli mer effektivt fick vi följande svar.

P1 menar att retrospektivet hjälper teamet att bli mer effektivt samtidigt som det förbättrar systemet och systemutvecklarnas kodning.

Informant P2 ser det som en process som till slut leder till att teamet kommer fram till något som ska göra teamet bättre eller fixa processen på något sätt. Vidare beskriver P2 i anslutning till svar rörande om hur det går att mäta om teamet faktiskt blir bättre, att resultatet av förbättringsarbetet inte bara behöver röra sig om synliga förbättringar.

*"En förbättring i team behöver nödvändigtvis inte synas utan kommer i förlängningen vara bättre kod, att man rör sig snabbare, att man planerar bättre och blir effektivare. Men vad det handlar om egentligen är att skapa en bra stämning och en schyst tillvaro, det i sin tur ger i förlängningen de här effekterna." P2*

P2 har en inställning till agil utveckling om att det handlar om att skapa en human arbetsmiljö där teamet kan jobba i en schyst takt och vara produktivt, detta i sin tur förklarar han leder till effektivisering och en lyckad implementation av agila metoder. Genom att skapa en bra stämning och en schyst tillvaro menar han att det till slut leder till effektivisering.

P4 beskriver likt övriga informanter retrospektivet som en möjlighet att reflektera över vad som hänt under sprinten och att det i sin tur kan leda till att teamet blir mer effektivt. Detta i sin tur kan leda till att teamet sparar in tid och möjligen reducerar frustration i teamet.

För att få en större inblick i och se skillnader i hur våra uppgiftslämnare ser på SCRUMs retrospektiv avslutades våra intervjuer med en flervalsfråga med möjlighet att motivera sina svar. Frågan behandlar huruvida retrospektivet inom SCRUM främjar/bidrar till sex stycken påståenden hämtat från vår litteraturstudie (se bilaga 2). Svar och påståenden presenteras i tabell 4.2. Informanterna har fått möjlighet att svara på en fyrgradig skala och fick möjligheten att utveckla sina svar. Frågan vi ställde till informanterna löd:

*Hur väl anser du att SCRUM-retrospective främjar följande påståenden?*



Tabell 4.2 - Sex påståenden om vad retrospektivet bidrar till.

	1. Stämmer inte 2. Stämmer ibland 3. Stämmer oftast 4. Stämmer alltid				
Retrospektivet bidrar till	P1	P2	P3	P4	P5
Det hjälper teammedlemmarna att dela och förstå varandras perspektiv	4	4	3 Det är alltid någon som inte är med eller vill vara med etc. Det har inte 100 % teckning. Ambitionen: 4	3 Teammedlemmarna är ofta redan medvetna om detta, jobbar tätt ihop och diskuterar problem i vardagliga arbetet.	4
Det integrerar lärande på individuell nivå med lärande på teamnivå	3	2 Görs mest i det vardagliga arbetet.	2 Fokus ligger på problem och andra saker under retrospektivet.	1 Fokus ligger inte på att öka förståelsen eller att lära sig nya saker under retrospektivet.	4 Ja, det får man med sig på något vis, genom diskussion
Det identifierar gömda problem	2 Det behöver det inte alltid göra.	4 Väldigt mycket och väldigt ofta	4	2 Ger oftast ingen "aha" upplevelse, till viss del kända sen tidigare.	3 Det är inte alltid alla problem kommer upp.
Det dokumenterar best practises och problem	2-3 Dokumenterar för stunden och för nästa sprint, inte för att kunna gå tillbaka och se hur man har arbetat.	1 Dokumenterar ingenting förutom för eget bruk.	4	3-4 Skapar "improvement" listor.	2 Viss form av dokumentation men inte jättemycket.
Det ökar tillfredsställelsen av arbetet genom att ge personer feedback på deras arbete	4	3	4 Speciellt om man lyfter upp positiva aspekter också.	2 Detta passar mer in i andra processer i verksamheten.	4
I somliga fall kan det förbättra möjligheten att korrekt estimera projektets kostnader	1 Retrospektivet ses inte som ett verktyg för detta.	2	4 Genom en bättre samförståelse så gör man definitivt bättre estimat.	3 Det är ingen jättestor bidragare men det hjälper.	3 Det kan det nog bidra till men det finns andra aspekter man jobbar med där.

### 4.2.3 Påverkande faktorer

På frågor rörande eventuella faktorer som kunde påverka utförandet och kvaliteten på ett sprint-retrospektiv gav informanterna följande svar.

P1 menade att en viktig faktor som kunde påverka kvaliteten på retrospektivet var att han arbetade med ett team som faktiskt ville förbättra sig. Han påpekade att, om teamet inte diskuterade problem under mötet, så var det SCRUM-masterns ansvar att se till att det faktiskt blev en diskussion. I slutändan så menade han dock att det var upp till teamet att vilja delta på retrospektivet. Om inte teamet vill jobba enligt SCRUM så ska man inte heller göra det.

Andra faktorer som kunde påverka retrospektivet enligt P1 var externa intressenter. *”Det är inte populärt med utvecklare som sitter i möten, och det gäller retro-mötet också”*. Han förklarade vidare; om det är så att t.ex. en produktägare inte vill att teamet ska sitta i långa möten, så gäller det att utbilda denna person i SCRUM så att denne får en insikt i varför dessa möten är så viktiga.

P2 tycker att deltagarnas vilja att vara med på retrospektivet är mycket viktigt. Han kan i inledningen av retrospektiven säga till teamet att de som inte vill inte behöver vara med på mötet. Detta för att ge alla deltagare en väg ut från retrospektivet. Att kunden skulle vara motvillig till retrospektivet är inget som P2 anser vara en ursäkt, han menar att det är SCRUM-masterns ansvar att alltid se till att det blir ett retrospektiv. Men att kunden skulle vara negativt inställd tycker P2 skulle vara konstigt:

*”Så att en kund skulle vara skeptisk till det tycker jag är konstigt, för det vi egentligen pratar om är att om vi inte reflekterar och förbättrar processen så kommer vi inte kunna gå snabbare”P2*

En faktor som enligt P2 kan påverka kvaliteten på retrospektivet är att det ibland kan utföras på fel sätt. SCRUM-mastern eller diskussionsledaren vet helt enkelt inte hur han eller hon faciliterar ett retrospektiv. P2 betonar vikten av att förstå hur mycket psykologi det finns i att vara facilitator av ett retrospektiv. Diskussionsledaren måste ha kunskap om gruppdynamik, veta hur människor fungerar och snabbt kunna kartlägga personligheter i ett team. Om teamet har dåliga erfarenheter av retrospektiv sedan tidigare kan även detta vara en källa till motvilja.

Faktorer som ointresse från teamet eller svårigheter att få teamet motiverade såg även P3 som faktorer som kunde påverka utförande och kvalitet av retrospektivet. Viktigt enligt honom var även att retrospektivet skulle vara välplanerat samt att diskussionsledaren har god kunskap om gruppen sedan tidigare. Enligt P3 skulle det kunna vara så att kunden tycker att retrospektivet är en dålig idé och inte vill att retrospektiven utförs. Om det är på detta vis menar P3 att valet att arbeta enligt metoden SCRUM bör ifrågasättas. Vidare sa han att anledningen till att kunden skulle kunna vara kritiskt inställd mot retrospektivmöten kunde vara att kunden har dålig kunskap om SCRUM.

Eftersom P4 alltid i förhand har avsatt tid till sina retrospektiv redan i början av varje sprint anser han inte att det finns några faktorer som skulle kunna påverka att retrospektivet faktiskt hålls som planerat. Han har dock testat lite olika tekniker för problemdiskussion och funnit att somliga metoder fungerar bättre än andra. Att låta teamet fundera och skriva ner problem på lappar och låta alla presentera sina lappar fungerar enligt P4 bättre än en öppen diskussion. Om deltagarna har en öppen diskussion finns det risk att teamet "bara håller med" det någon annan i teamet har sagt och inte tar upp sina egna punkter, på så vis fastnar teamet i ett och samma spår under mötet.

P5 betonar vikten att ta upp alla "mjuka" faktorer under retrospektivet.

*"Alla dessa mjuka faktorer är viktiga att ha med. Säger inte alla vad alla tycker så blir det ju inte bra. Då blir man ju inte bättre." P5*

En annan viktig faktor är att teamet får tillåtelse att hålla sina möten. Hon säger att det finns vissa företag som säger att man inte ska lägga tid på detta.

## 5 Diskussion

### 5.1 Process för retrospektivet

Analysen av empiri i föregående kapitel visade att processen för sprint-retrospektiv varierade och kunde se olika ut mellan våra informanter. Samtliga informanter uppgav att retrospektivet är ett forum för att diskutera och reflektera över SCRUM-processen och för att kunna lösa problem och göra nödvändiga förändringar för att bli bättre. De intervjupersoner vi valt att inkludera i studien driver således sina SCRUM-projekt i linje med ”The Agile Manifestos” tolfte punkt (se bilaga 1). Att med jämna mellan stannar upp och reflekterar över processen, för att därefter göra nödvändiga justeringar.

Däremot kan vi se att retrospektivet skiljer sig informanterna emellan gällande strukturen och val av metoder. Här syftar vi på hur mötets agenda kan se ut och huruvida olika övningar för att starta diskussion, samla in data, analysera uppkomna problem och positiva erfarenheter, urval och uppföljning är implementerade. Samtliga informanter är eniga om att retrospektivet bidrar till att förbättra deras arbetsprocess. Utifrån empirin har vi sett att samtliga informanter anser att sprint-retrospektiv bidrar till en förbättring och effektivisering av SCRUM-processen, oavsett om de väljer att driva mötet efter dokumenterade metoder och övningar eller öppen diskussion. Vi har således inte funnit något som tyder på att den upplevda betydelsen ser annorlunda ut beroende på hur retrospektivet utförs.

Det råder skiljaktigheter mellan informanterna rörande om retrospektivet alltid identifierar gömda problem. Vi kan se ett samband mellan att de informanter som uppgav att retrospektivet alltid identifierar gömda problem också är de informanter som använder sig av dokumenterade tekniker när de faciliterar sina retrospektiv. Med dokumenterade tekniker syftar vi på tekniker beskrivna av Derby & Larsen (2006).

När det gäller vad som diskuterades under retrospektivet var informanterna eniga om att de behandlade saker som gått bra, dåligt och som måste förbättras, detta stämmer väl överens med den agila litteratur vi har behandlat i litteraturstudien (Schwaber & Sutherland, 2007; Derby & Larsen, 2006). Anledningen till att just dessa saker behandlas av samtliga tillfrågade tros vara att dessa återkommer inom nästan all litteratur som vi läst om SCRUM och sprint-retrospektiv.

Empirin visade även att informanterna var eniga om att alla typer av ämnen kunde diskuteras under retrospektiven. Det framgick dock att somliga av de tillfrågade hade fokus på mer tekniskt lagda problem medans andra fokuserade på mer mjuka faktorer som människor och förhållanden. Att retrospektivet är ett forum där teamet kan diskutera alla möjliga typer av problem stämmer väl överens med vad som framkom i litteraturgenomgången. Det vill säga att retrospektivet är ett tillfälle där teamet ska inspektera processen med anseende på människor, förhållanden, processer och verktyg (Schwaber & Sutherland, 2010).

Analysen visade att de tillfrågade har olika ståndpunkter när det gäller vem som ska vara med på retrospektiven. Samtliga var eniga om att SCUM-mastern och teamet skulle delta på mötet. Analysen visade sig även att externa intressenter ibland fick delta, i syfte att ge dem en bättre förståelse om hur metoden fungerade. Vi tolkade detta som att retrospektivet i första hand är en intern angelägenhet för SCRUM-teamet, men kan även vara ett medel för att utbilda organisationen och andra intressenter i metoden SCRUM.

Diskussionerna under retrospektivet leder till ett antal definierade problem som bör åtgärdas under kommande sprint för att processen ska förbättras. Analysen visade att antalet problem som läggs fram för åtgärd inför kommande sprint varierade i antal. Somliga av informanterna uppgav att de hade svårt att åtgärda mer än ett fåtal problem medans andra försökte åtgärda allt som kom upp under kommande sprint. Att välja ut ett fåtal problem ligger i linje med Derby & Larsens (2006) resonemang om prioritering. De menar att risken med att välja ut för många problem är att det kan leda till att ingenting blir gjort. Vi tror att antalet problem som kan hanteras under kommande sprint till stor del beror på problemens karaktär. Ett SCRUM team som bara tar upp problem med teknisk karaktär har rimligtvis större möjligheter att hantera fler problem än ett team som hanterar större problem som konflikter inom gruppen. Vilken strategi SCRUM-mastern väljer tror vi även kan bero på hur väl inläst denna person är på teorier som berör just facilitering av retrospektiv.

Utifrån empirin noterade vi att en av våra informanter verkade ha en mer teknisk inriktning på sina retrospektiv än övriga tillfrågade, trots att han var öppen för att diskutera problem med annan karaktär. Att deras möten verkade ha fokus på teknik kan bero på att de för tillfället arbetar distribuerat med SCRUM. Vi tror att detta kan bidra till att de mjukare faktorerna är svårare att diskutera. Detta skulle ligga i linje med Jacobsens (2002) resonemang om att möte, ansikte mot ansikte, gör att det blir lättare att ta upp känsligare ämnen. Författaren syftar dock på intervjuer vid en kvalitativ datainsamling.

Vi undersökte även om informanterna gjorde någon uppföljning för att säkerställa att problemen faktiskt blev åtgärdade. Analysen visade att enbart en av de tillfrågade hade någon uttalad uppföljning av detta. Övriga ansåg att retrospektivet var självkontrollerande på så vis att prioriterade problem som inte var lösta skulle dyka upp i något av de kommande retrospektivmötena.

Av det empiriska materialet kan vi inte med säkerhet säga att retrospektivet är ett medel som implementerar *double-loop learning* (Argyris & Schön, 1978). SCRUM-teamen identifierar problem och hittar lösningar som de försöker implementera under kommande sprint. Empirin visar dock inte i detalj hur de hanterar de problem som kommer upp under retrospektivet och om teamet i längden drar ”lärdom av lärandet”, som Argyris & Schön definierar *double-loop learning*. Men att retrospektivet bidrar till någon form av lärande och åtminstone till *single-loop learning* kan vi se utifrån vår insamlade empiri.

## 5.2 Retrospektivet och kontinuerlig förbättring

Vår analys av empirin visar att våra informanter alltid utför ett retrospektiv efter varje sprint i sina SCRUM-projekt. Det är klart och tydligt att de personer som vi intervjuat inte nedprioriterar detta moment. Flera författare (Schwaber & Sutherland, 2007; Kniberg, 2007; Myllyaho et al, 2004) skriver att retrospektivet ibland glöms eller prioriteras bort vilket vi i vår studie inte sett några indikationer på. Tvärtom betonar flera av informanterna vikten av att inkludera detta moment i metoden SCRUM. Att retrospektivet är en elementär del av SCRUM styrks även av många författare, bl.a. (Schwaber & Sutherland, 2007; Salo et al., 2004; Myllyaho et al, 2004).

Retrospektivet inom SCRUM beskrivs av informanterna som en positiv och viktig del i deras utvecklingsprojekt. Samtidigt som det inom SCRUM känns mer naturligt att genomföra möten avsatta för reflektion anser flera av informanterna att ett retrospektiv skulle kunna vara fördelaktigt att applicera även inom mer traditionella utvecklingsmetoder. Informanternas påpekande om att retrospektiv även bör användas vid traditionell utveckling tror vi beror på att majoriteten av våra informanter jobbat med traditionell utveckling innan de började med agil utveckling. På grund av att de har erfarenhet av båda delar har de uppmärksammat positiva och negativa aspekter som skulle kunna appliceras på den andra.

Att teamet faktiskt tar sig tid mellan varje sprint för att reflektera över föregående sprints arbete är något som samtliga intervjupersoner beskriver som elementärt. Detta stämmer väl överens med vad vi funnit i teorin avseende SCRUM och agil utveckling. Det stämmer också väl överens med det vi funnit i teorin om lärande. Många författare beskriver hur viktigt det är med reflektion för att lära (Kolb, 1994; Schön, 1983). Reflektionsarbetet under retrospektivet kan jämföras med Schön's (1983) beskrivning av reflection-on-action, dvs. att man stannar upp och reflekterar över arbetets gång.

Även om informanterna inte explicit talade om lärande när de talade om att förbättra sin process kan vi se tydliga likheter mellan retrospektivet och hur Kolb (1984) beskriver lärandeprocessen. I likhet med Kolbs lärandecykel får deltagarna till att börja med ta upp och komma fram till konkreta problem eller positiva erfarenheter de upplevt från föregående sprint. De reflekterar sedan över detta, conceptualiserar problem(en) och experimenterar och testar alternativa lösningar. På så vis lär de sig att göra kommande arbete bättre och effektivare. När vi frågade informanterna om de ansåg att retrospektivet var ett moment där lärande på individuell nivå integrerades med lärande på teamnivå fick vi spridda svar från de tillfrågade. Samtidigt svarade samtliga informanter att retrospektivet bidrog till att processen förbättrades. Vi tror att informanternas tolkade frågan om integrerat lärande olika, gällande vilken typ av kunskap som skulle kunna integreras inom gruppen. En av informanterna tog upp teknisk kunskap som ett exempel och menade att retrospektivet inte var ett forum för att dela med sig av sådan kunskap. Vilket inte till fullo överensstämmer med teorin om att ett retrospektiv delar kunskap mellan individ- och organisationsnivå (Dingsøy, 2004). Däremot

kan vi se att teamet lär sig att tillsammans göra processen bättre och detta tyder på att retrospektivet faktiskt är ett forum där lärande sprids inom teamet. Detta styrks även av att informanterna var eniga om att retrospektivet hjälper teamet att dela och förstå varandras perspektiv vilket också ligger i linje med vad Dingsøy (2004) skriver om vad ett retrospektiv kan bidra till.

När det gäller vad retrospektivet faktiskt bidrar till tycker samtliga informanter att det bidrar till en effektivisering av SCRUM-teamets arbetsprocess. Detta stämmer väl överens med Schwaber och Sutherlands (2010) syn på retrospektivets betydelse. Vi har sett från empirin, att retrospektivet bidrar till en bättre arbetsprocess. Däremot så tror vi inte att retrospektivet är det enda momentet inom SCRUM som bidrar till förbättring och effektivisering. Generellt sätt jobbar utvecklingsteamerna inom agil utveckling med ett tätt samarbete med kunden och arbetar med korta iterationer. På så sätt kan teamet tidigt uppmärksamma förändringar som behöver göras eller förbättringar som måste implementeras. Det finns flera moment där dessa förändringar kan uppmärksammas inom SCRUM. Det kan vara under det dagliga SCRUM-mötet, under sprintens utvärderingsmöte eller under retrospektivet. Vi har funnit att våra informanter anser att retrospektivet är ett väldigt viktigt moment för kontinuerlig förbättring av arbetsprocessen, men inte det enda.

Vilka delar av processen som förbättras har däremot informanterna beskrivit på olika sätt, förmodligen för att informanterna lägger fokus på och behandlar ämnen med lite olika karaktär. En av informanterna beskrev retrospektivets bidrag ur ett tekniskt perspektiv och påpekade att retrospektivet leder till att koden och i slutändan den färdiga produkten blir bättre. Andra beskrev att målet med retrospektivet var att fånga upp problem med olika karaktär, exempelvis problem med gruppdynamik eller dålig kommunikation. Det vi kunde se var att samtliga informanter som hade en SCRUM-master certifiering betonade vikten av att diskutera de ”mjuka” faktorerna när de höll sina retrospektiv, inte bara problem med teknisk karaktär. Vi tror att karaktären på de problem som dyker upp under reflektionsmöten varierar mycket från projekt till projekt och från team till team, vilket också gör det svårt att identifiera och peka på generella punkter som retrospektivet alltid bidrar till. Eftersom metoden ses som empirisk och att teamet sällan på förhand vet vilka problem som kommer att hanteras under retrospektivet.

I teorin fann vi att retrospektivet bidrar till att identifiera gömda problem inom teamet (Dingsøy, 2004). Vår analys visar att informanterna inte alltid instämmer med detta påstående. Några av informanterna ansåg att retrospektivet alltid identifierar gömda problem medan andra svarade att gömda problem inte alltid kommer upp till ytan under retrospektivet. Att teamet inte lyckas att identifiera gömda problem under retrospektivet tror vi beror på hur diskussionsledaren väljer att facilitera mötet. Det kan även bero på att diskussionsledaren inte använder sig av tekniker för att hitta roten till de upplevda problem som kommer upp. Vi fann att de informanter som ansåg att retrospektivet alltid identifierade gömda problem använde sig av olika tekniker för att hitta roten till problemen. Exempel på tekniker som nämndes var ”fish-bone diagram” eller ”five why’s”, vilka också beskrivs av Derby & Larsen(2006) som bra metoder för att hitta bakomliggande orsaker till problem.

### 5.3 Påverkande faktorer

Informanterna beskrev sina svar lite olika angående om det fanns faktorer som skulle kunna påverka själva utförandet och kvaliteten på ett sprint-retrospektiv.

Då vi i litteraturstudien funnit att retrospektivet inom SCRUM ibland blir nedprioriterat eller bortglömt (Schwaber & Sutherland, 2007; Salo et al, 2004; Myllyaho et al, 2004) ställdes frågor rörande svårigheter att faktiskt utföra retrospektivet. Informanterna ansåg att en viktig del som kunde påverka kvaliteten på retrospektivet var att de medverkande faktiskt ville ha ett retrospektiv och att de ville bli bättre som team. Ett ointresse från deltagarnas sida kan innebära att viktiga problem inte ventileras inom teamet, flera av informanterna menade att om det inte finns ett intresse borde teamet fundera på om de ska arbeta enligt SCRUM över huvud taget. Några av informanterna använde sig av tekniker beskrivna av Derby & Larsen (2007) i inledningen av retrospektivet. Detta för att skapa rätt stämning så att teamet känner trygghet och för att främja diskussion. Av empirin fann vi att det var just de informanter som använde sig av tekniker beskrivna av Derby & Larsen som svarade att retrospektivet alltid identifierar gömda problem (se tabell 4.2). Detta tyder på att om diskussionsledaren använder sig av olika beprövade tekniker för att främja diskussion är möjligheterna större att identifiera gömda problem.

Vi undersökte även om det fanns externa faktorer som kunde påverka möjligheterna att hålla ett retrospektiv efter varje sprint inom SCRUM-projekt. De tillfrågade höll alltid ett retrospektiv mellan varje sprint och uppfattade således inte att det fanns några faktorer som skulle påverkat möjligheten till detta i deras projekt. Dock svarade några av informanterna att de trodde att externa intressenter, så som kund och ledning skulle kunna vara kritiska till att lägga alltför mycket tid på retrospektiv. Samtidigt kommenterade flera av de tillfrågade att om detta är ett problem så är det viktigt att samtliga involverade får utbildning inom SCRUM, detta för att de ska förstå de fördelar som regelbundna retrospektiv kan innebära. Alla intressenter måste se arbetet med SCRUM och retrospektiv ur ett långsiktigt perspektiv och faktiskt förstå att det i längden gör att teamet blir effektivare, även om det kräver mycket resurser. Vi tolkar detta som att informanterna tycker det är viktigt att ha fullt förtroende från den egna organisationen och externa intressenter. Vi kan se ett samband mellan detta och teorin om vikten att låta teamet bestämma och erhålla fullt förtroende ifrån organisationen (Schwaber & Beedle, 2002).

Utöver möjligheten att utföra ett retrospektiv och uppnå en förståelse för varför teamet utför det diskuterades även retrospektivets kvalitet och bidrag. Vi kan se ett samband mellan kvaliteten på retrospektivet och SCRUM-masterns kunskap i hur dessa retrospektiv utförs på bästa sätt. Flera av informanterna uppgav att de hade prövat olika tekniker och metoder när de utför sina retrospektiv. Utifrån empirin kan vi se att majoriteten av de tillfrågade har prövat olika metoder både när det gäller att främja diskussioner, skapa rätt stämning och prioritera och välja ut kritiska problem.



Det vi kan konstatera är att samtliga informanter arbetar på olika sätt med olika metoder samt att dessa metoder väljs ut för att passa just deras kontext. Detta ligger i linjer med Schwaber & Beedle (2002) som ser SCRUM som en empirisk process där teamet experimenterar, lär sig och utvecklas efter hand. Därav går det inte att säga att ett SCRUM-team arbetar med retrospektiv på ett bättre sätt än ett annat. Vi kan dock anta att utbildning och variation av olika metoder kan leda till ett bättre tillvägagångssätt som passar det enskilda teamet. Vi kan även se att flera av informanterna ansåg att det var viktigt att diskussionsledaren hade god kunskap om gruppen och dess gruppdynamik.

## 6 Slutsats

Avsikten med denna uppsats var att besvara vår forskningsfråga: *Hur och i vilken utsträckning används sprint-retrospektiv som ett verktyg för kontinuerlig förbättring avseende SCRUM-processen?*

För att kunna besvara denna fråga utgick vi ifrån vår undersökningsmodell som behandlade tre delområden med dess tillhörande aspekter: *Processen för retrospektivet; Retrospektivet & kontinuerlig förbättring; Påverkande faktorer*. Se figur 2.6.

Gällande hur sprint-retrospektivet används och är utformat har vi kommit fram till att momentet utförs på olika sätt hos samtliga informanter. Det vi sett är generellt för samtliga är att de håller ett tidsbestämt retrospektivmöte efter varje sprint. Alla informanter har även beskrivit att de har ett moment för att samla in data från föregående sprint. Här tar teamet upp saker som gått bra, saker som gått mindre bra och saker som behöver förbättras. Därefter väljer de ut vilka problem som bör åtgärdas under kommande sprint.

Strukturen på mötena ser olika ut mellan våra intervjupersoner. Somliga väljer att använda sig av strukturer som finns beskrivna i vår litteraturstudie (se avsnitt 2.5) där retrospektivet delas in i 5 faser, från att skapa stämning för diskussion till att avsluta själva retrospektivet. Andra väljer att hålla sitt retrospektiv genom att hålla en öppen diskussion och helt enkelt gemensamt komma fram till vad som måste förbättras, utan att använda några speciella metoder för detta. Vi kan se ett samband mellan strukturen och val av metoder applicerat på retrospektivet och SCRUM-masterns kunskap i hur dessa retrospektiv kan utföras. Däremot kan vi inte konstatera att tillvägagångssätt och metoder i ett specifikt retrospektiv skulle vara bättre än i ett annat då informanterna i denna studie överlag är nöjda med hur det utförs.

Under vår litteraturstudie fick vi klart för oss att SCRUM-teamet med fördel kan göra ett urval och prioritera uppkomna problem under retrospektivet. Detta kan vara effektivare för teamet då de inte försöker lösa samtliga problem på en gång, vilket kan underlätta och leda till att problemen mer sannolikt faktiskt blir åtgärdade. Inte heller när det gäller hur teamet prioriterar de problem som behöver åtgärdas under ett retrospektiv kan vi dra några generella slutsatser om hur ett SCRUM-team bör gå till väga. Några av informanterna väljer att prioritera ett, två eller tre problem, medan andra försöker åtgärda samtliga problem inför nästa sprint. I denna studie har vi inte sett att det ena skulle vara att föredra framför det andra, eftersom samtliga tillfrågade var nöjda med deras sätt att arbeta. Däremot påpekade två av informanterna att de funnit att SCRUM-teamet fick fler problem åtgärdade genom att välja ut ett mindre antal kritiska problem.

Majoriteten av informanterna ansåg inte att någon speciell uppföljning på de problem som ska åtgärdas efter ett retrospektiv var nödvändig. Undantaget var en av de tillfrågade som arbetade i en organisation där teamet lämnade över ansvaret till en annan instans som låg utanför SCRUM-processen. Generellt kan vi säga att de flesta tillfrågade ansåg att någon stor uppföljning inte var nödvändig eftersom retrospektivet var självkontrollerande så till vida att problem som inte var åtgärdade dök upp som problem även i framtida retrospektiv.

Vidare fann vi vår litteraturstudie att retrospektivet är ett moment som ibland prioriteras ner av olika anledningar. I vår undersökning fann vi inget belägg för detta. Samtliga av våra tillfrågade höll alltid ett sprint-retrospektiv efter varje iteration, huvudsakligen för att de ansåg att momentet är ett viktigt verktyg för kontinuerlig förbättring inom SCRUM. Dock nämnde någon av de tillfrågade att de vet med sig att det finns team och organisationer som prioriterar ned eller bort retrospektiven. Vilket tyder på att det kan finnas problem med nedprioritering av retrospektiv i andra kontext än de som undersökts i denna studie.

För att ta reda på varför det eventuellt nedprioriteras undersökte vi även om det kunde finnas motstånd mot att retrospektiven utförs. Det vi kan konstatera utifrån informanternas svar är att det är viktigt att SCRUM-teamet faktiskt vill förbättra sig och delta under retrospektiven, ansvaret för att så ska ske ligger huvudsakligen på SCRUM-mastern. Samtidigt beskrev informanterna att det kan vara så i vissa fall att externa intressenter är motvilliga till att SCRUM-teamet utför retrospektiv, i dessa fall trodde informanterna att det rörde sig om att externa intressenter har för dålig kunskap och utbildning inom SCRUM och vad retrospektivet faktiskt kan bidra till i längden. Det är alltså viktigt att teamet får fullt förtroende från organisationen.

Studien tyder på att retrospektivet bidrar till att effektivisera SCRUM-teamets arbetsprocess. Vi kan dock inte utifrån studien peka på exakt vad som blir bättre eftersom SCRUM-teamen tar upp olika problem och möjligheter vid varje iteration. Vi kan däremot se att arbetet med retrospektiven kan bidra till förbättringar inom aspekter med både mjuk och hård karaktär, t.ex. kommunikation inom gruppen och tekniska lösningar. Vidare kan vi konstatera att samtliga informanter var eniga om att retrospektivet hjälper gruppen att dela och förstå varandras perspektiv och att problem diskuterades som annars kanske inte hade tagits upp.

## 6.1 Sammanställning – Resultat

- Retrospektivet bidrar till att SCRUM-teamet kontinuerligt förbättras.
- Informanterna i denna undersökning prioriterar aldrig ner retrospektivet.
- Hur retrospektivet används varierar i form av utformning, struktur och applicerade metoder.
- Retrospektivet är självkontrollerande och uppföljning är inte alltid nödvändig.
- För att uppnå kontinuerlig förbättring av SCRUM-processen är det viktigt att SCRUM-teamet faktiskt vill förbättra sig och delta på retrospektivet.
- Det är viktigt att SCRUM-teamet har fullt förtroende från organisationen och andra intressenter.

I

## 7 Bilagor

---

Detta avsnitt innehåller samtliga bilagor tillhörande uppsatsen, både bilagor som refererats i vår litteraturstudie men även bilagor rörande intervjuer och intervjuprotokoll.

---

### Bilaga 1, The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

*Individuals and interactions* over processes and tools  
*Working software* over comprehensive documentation  
*Customer collaboration* over contract negotiation  
*Responding to change* over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

#### Principles behind the Agile Manifesto

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

[Online] [www.agilemanifesto.org](http://www.agilemanifesto.org) [Läst 15 april 2011]

## Bilaga 2, Intervjuguide

Intervjuguide	
Personens bakgrund	
Ålder, Utbildning, Anställd sedan, Roll	
<b>1. Vad har du för erfarenheter av traditionell respektive agil systemutveckling?</b>	
<b>2. Kan du nämna någon utbildning/certifiering inom SCRUM/agila metoder du har fått/genomfört?</b>	
<p><b>3. SCRUMs ramverk består av ett antal regler och guidelines. I egenskap av SCRUM-master vill vi att du lyssnar på ett antal påståenden och säger en siffra, på en skala mellan 1-4 beroende på om det är något som du följer eller inte. Utveckla gärna dina svar.</b></p> <p><b>Artefakter</b></p> <ul style="list-style-type: none"> <li>- Vi har alltid en Product Backlog</li> <li>- Vi har alltid en Sprint Backlog</li> <li>- Vi har alltid en Sprint Burn-down</li> </ul> <p><b>Process</b></p> <ul style="list-style-type: none"> <li>- Vi har alltid ett Sprint Planning Meeting</li> <li>- Vi har alltid en Sprint</li> <li>- Vi har alltid en Daily SCRUM/Standup</li> <li>- Vi har alltid ett Sprint Review Meeting</li> <li>- Vi har alltid ett Sprint Retrospective</li> </ul> <hr/> <p><b>SCRUM-Team</b></p> <ul style="list-style-type: none"> <li>- Vi har alltid en SCRUM-Master</li> <li>- Vi har alltid en Product owner</li> </ul>	<p>På en skala från 1-4</p> <ol style="list-style-type: none"> <li>1. Stämmer inte</li> <li>2. Stämmer ibland</li> <li>3. stämmer oftast</li> <li>4. stämmer alltid</li> </ol>

<p><b>3.1 Time-boxar ni alla ovan nämnda processer?</b></p>	
<p><b>4. Hur ser processen ut i detalj i ditt arbete med retrospektiv, vad avhandlas och vilka instrument används?</b>  <b>4.1 Vem deltar i detta möte?</b></p>	
<p><b>5. Vad anser du om sprint-retrospektiv och reflektionsarbete inom SCRUM?</b>  <b>5.1 Vad tycker du kan bli bättre?</b></p>	
<p><b>6. Vilka faktorer påverkar möjligheterna att <u>utföra</u> ett retrospektiv som ett verktyg för förbättringsarbete?</b></p>	
<p><b>7. Vad krävs för att arbete med sprint - retrospektiv ska resultera i förbättring?</b></p>	
<p><b>8. Vilka faktorer påverkar <u>kvaliteten</u> av arbetet med retrospektiv som verktyg för förbättringsarbete?</b></p>	
<p><b>9. Gör ni någon form av prioritering av de saker som kommer upp under ert sprint-retrospektiv?</b>  <b>9.1 Hur prioriteras områden som behöver förbättras inför kommande sprint?</b></p>	
<p><b>10. Gör ni någon uppföljning av retrospektivet som visar att det faktiskt leder till förbättring?</b>  <b>10.1 Hur ser denna uppföljning ut?</b></p>	
<p><b>11. Hur väl anser du att SCRUMs sprint-retrospective främjar följande påståenden? Utveckla gärna dina svar.</b></p> <ul style="list-style-type: none"> <li>- Det hjälper teammedlemmarna att dela och förstå varandras perspektiv</li> <li>- Det integrerar lärande på individuell nivå med lärande på teamnivå</li> <li>- Det identifierar gömda problem</li> </ul>	<p>På en skala från 1-4</p> <ol style="list-style-type: none"> <li>1. Stämmer inte</li> <li>2. Stämmer ibland</li> <li>3. Stämmer oftast</li> <li>4. Stämmer alltid</li> </ol>

<ul style="list-style-type: none"> <li>- Det dokumenterar best practises och problem</li> <li>- Det ökar tillfredsställelsen av arbetet genom att ge personer feedback på deras arbete</li> <li>- I somliga fall kan det förbättra möjligheten att korrekt estimerar projektets kostnader</li> </ul>	
--	--

### Bilaga 3, Intervjuprotokoll – P1

Intervjuperson 1 – P1		
Talande (J, P1, C)		
1.	J	Vi kan ju börja lite kort med att du berättar lite om dig själv; namn, ålder, akademisk utbildning, nuvarande anställning och anställningsroll.
2.	P1	Mitt namn är NN och är 32 år gammal och har pluggat till civilingenjör inom datateknik i Luleå.
3.	J	Hur länge har du arbetat för XXXX?
4.	P1	Jag är anställd hos XXXX , ett konsultbolag i Stockholm där jag varit anställd sedan 2009. Jag har självklart arbetat på andra ställen tidigare också. Jag började jobba inom systemutveckling 2006.
5.	J	Vad är din roll på XXXX?
6.	P1	Jag jobbar som konsult som systemutvecklare helt enkelt och arbetar just nu i ett projekt i rollen som både utvecklare och SCRUM-master.
7.	J	Första frågan då. Vad har du för erfarenheter av traditionell respektive agil utveckling?
8.	P1	Menar du positiva eller negativa erfarenheter?
9.	J	Snarare mer om vilken bakgrund du har inom utveckling både agilt och traditionellt.
10.	P1	<p>På min första anställning så var det ju traditionell vattenfallsmetodik, det var ju för att vi var inte fler än 4-5 personer som satt jämte varandra och i stort sett utvecklade allt själv som vi höll på med. Så jag hade väldigt snabbt kontakt med bossen och han visste inte hur man skulle sköta projekt och så vidare. Han visste bara vad han ville ha och sen sa han det till mig och sen fick jag lägga upp arbetet själv. Det fanns ingen direkt bas utan man får kalla det ett vattenfall helt enkelt.</p> <p>Sen när jag började arbeta som konsult, har jag nog i stort sett suttit med agila projekt hela tiden. Det är väl oftast inte kunden som har valt det utan det är snarare teamet som har valt det.</p>

11.	C	Är det någon speciell typ av agila projekt, då tänker jag på XP, SCRUM m.m.
12.	P1	Nej jag har bara suttit i SCRUM, men givetvis har det skett en del 2 och 2 programmering och sådana saker. Men SCRUM är ju det som jag har kört helt och hållet.
13.	J	Kan du nämna någon utbildning/Certifiering inom SCRUM/Agila Metoder som du har fått/genomfört?
14.	P1	Nej
15.	J	Så du är i princip självlärd då?
16.	P1	Nej alltså, vi hade väl nå kurs på universitetet som rörde på XP men däremot inget SCRUM, det var liksom ingenting som universiteten kände till på den tiden. Nu låter det som det var för tjugo år sen. Men det var något som det stod väldigt lite om i böcker men det var ingen som undervisade om det eller någonting. Utan det är något jag har lärt mig i arbetslivet helt enkelt.
17.	C	Men ingen utbildning ”inhouse” så att säga?
18.	P1	Nej, Det är mer att man har kastats in i ett projekt och har dom använt SCRUM så har dom fått lärt en helt enkelt.
19.	J	Nu arbetar du som SCRUM-master , det stämmer?
20.	P1	Det är korrekt
21.	J	<p>SCRUM som ramverk består av ett antal regler och guidelines. I egenskap av SCRUM-master vill vi att du lyssnar på ett antal påståenden och säger en siffra, på en skala mellan 1-4 beroende på om det är något som du följer eller inte.</p> <p>Där siffrorna står för:</p> <ol style="list-style-type: none"> <li>1. Finns/Används ej</li> <li>2. Finns/Används I begränsad utsträckning</li> <li>3. Finns/Används för det mesta</li> <li>4. Finns och används i samtliga av dina SCRUM project</li> </ol> <p style="margin-left: 40px;">Artefakter</p> <ul style="list-style-type: none"> <li>- Product Backlog</li> <li>- Sprint Backlog</li>   <li>- Sprint Burndown</li> </ul> <p style="margin-left: 40px;">-----</p> <p style="margin-left: 40px;">Process</p> <ul style="list-style-type: none"> <li>- Sprint Planning Meeting</li> <li>- Sprint</li> <li>- Daily SCRUM</li> <li>- Release Planning Meeting</li> </ul>



		<ul style="list-style-type: none"> <li>- Sprint Review Meeting</li> <li>- Sprint Retrospective</li> <li>- Timeboxing</li> </ul> <hr/> SCRUM-Team <ul style="list-style-type: none"> <li>- SCRUM-Master</li> <li>- Product owner</li> <li>- Team</li> <li>- Release Burndown</li> </ul>
22.	C	Om vi börjar att titta på lite artefakter inom SCRUM, till att börja med Product Backlog?
23.	P1	I stort sett alla SCRUM-projekt jag har varit i är product backlog i stort sett det viktigaste, nästan i alla fall, den har alltid varit med. Så det är en 4:a
24.	J	Sprint Backlog? Sprint Burndown?
25.	P1	Samma där... en 4:a
26.	J	Sen har vi dom här bitarna med planering och möten i själva SCRUM-processen. Enligt samma skala gradera förekomsten av "Sprint Planning Meeting", "Sprint", "Daily SCRUM", Release Planning Meeting, "Sprint review meeting", "Sprint Retrospective"
27.	P1	Samtliga av dessa nämnda möten och planering förekommer i de SCRUM projekt jag för. Det enda undantaget är "Release Planning meeting" som inte brukar utföras alls. Alltså en 4:a på samtliga förutom Release Planning meeting som får en 1:a
28.	C	Brukar dessa sammankomster och möten "TimeBoxas"
29.	P1	När ni säger "timeboxing". Menar ni estimaten på de olika tasks/item i ett projekt?
30.	C	Frågan är om du "timeboxar", då det är en stor del av SCRUM att allt ska tidsbestämmas. Som tex . Sprint planeringsmötet, daily SCRUM och retrospektivet m.m. Alltså om ni tidsbestämmer hur långt ett möte ska vara.
31.	P1	Ja det enda undantaget där, I nuvarande projektet och det är under stand-uppen(daily SCRUM). Där vi i det nuvarande SCRUM teamet och projektet sitter på olika platser. Där jag sitter här i Lund och de övriga utvecklarna i Stockholm. Alltså, vi har inte med stand-uppen i själva tidsestimeringen men däremot har vi den var dag osv, men, den är lite för lång kanske. Den kan gå upp emot 20 minuter ibland men det är för att vi faktiskt inte sitter fysiskt bredvid varandra. Men jag skulle säga att det är en fyra, då man inte brukar timeboxa standup (som i sin tur ska vara runt 5-15 minuter).
32.	C	Om vi börjar med de här begreppen från början kopplat till "timeboxing". Om vi börjar med "Sprint planning meeting". Timeboxar ni det och håller er inom uppsatta begränsningar?

33.	P1	Jaja, det är ju så att om vi har ett planeringsmöte som ska ta 2 timmar som exempel, så oavsett om vi inte är klara så stannar vi efter 2 timmar. Och sen får vi planera in nästa möte och ta det då istället.
34.	C	Och sprinten , den timeboxas också?
35.	P1	Ja det är ju så att ibland är det helgdagar osv. Men vi säger att en sprint är i 2 veckors intervall. Om det nu skulle smyga sig in en helgdag under sprinten så ändrar vi inte på vår planering. Torsdag är kod-stopp och fredagar är testdag liksom. Det är fortfarande 2 veckors som gäller.
36.	C	Då kommer vi till Daily SCRUM, eller stand up, det finns ju lite olika benämningar på det där.
37.	P1	Jo precis, som jag sa lite tidigare så är den lite för lång nu. Däremot så är den inte med i vårt estimat. Det är något som görs var dag så den finns inte med i vårt tids-estimat. Den ska bara göras liksom.
38.	C	Kör ni det varje dag?
39.	P1	Mm, Vi kör det varje dag. Men som sagt den är lite för lång. Vi är inte mer än 3 personer i teamet/projektet just nu iaf. Och om vi hade suttit bredvid varandra så hade den varat i 5 minuter istället för nuvarande 15-20. Så det bryter lite tidsgränsen där ibland.
40.	C	Ok det är då en 4:a? Daily SCRUM förekommer alltid?
41.	P1	Ja den förekommer alltid.
42.	C	Release-Planning Meeting då?
43.	P1	Vad menar du med release planning meeting?
44.	C	Ja om ni har något möte för att planera inför att ni ska släppa någonting, alltså en release. Det kanske inte är en "äkta" release vid varje sprint?
45.	P1	Nja, det beror ju på hur man definierar det. För oss är det ju en "äkta" release var gång. Det är helt enkelt så vi har lagt upp det att efter 2 veckor ska vi ha något som funkar till 100%. Rent praktiskt är inte det alltid sant, men efter det vi har gjort i alla fall så ska det vara en riktig release. Sedan så ligger ju inte all funktionalitet med så den kommer aldrig användas i riktig produktion. Men för oss är det en riktig release. Och vi skiljer inte på en stor eller en liten release utan det är en release varannan vecka.
46.	C	Ni har ett möte innan då för att planera en sådan release?
47.	P1	Nja, vi har ett möte för att estimerade kommande och så har vi själva demonstrationsmöte som är själva releasen men vi har liksom inget, jag tror du eftersöker ett möte där vi faktiskt diskuterar hur det presenteras.
48.	J	Ja precis.
49.	P1	I så fall, nej det har vi inte utan det är 2 veckors sprint. På den tiden hinner man inte implementera så mycket om man ska vara helt ärlig. Så då gör vi så att vi har ett demonstrationsmöte med vår kund efter 2 veckor och då går vi igenom

		hela demonstrationen och visar det vi har gjort, får feedback och så vidare. Och det är det vi gör. Så det beror helt och hållet på hur man definierar det. Om man menar att demonstrationsmötet är ett sådant möte så följer vi det, annars inte.
50.	C	Nej, det mötet kommer vi till sen, det heter då Sprint review meeting.
51.	P1	Ja då är det egentligen det vi kör.
52.	J	Så att förtydliga, "Release-Planning meeting" används inte?
53.	P1	Nej.
54.	J	Då blir det en 1:a . Då kommer vi till den punkten som vår uppsats egentligen behandlar, vilket är "Sprint-Retrospective" och reflektionen över sprinten. Används det i ditt nuvarande projekt eller har det använts i övriga SCRUM projekt du varit involverad i?
55.	P1	Det har alltid använts i föregående projekt, men nu kör vi det. Vi kör 2-3 timmar varje vecka t o m. Så det följs helt och hållet i stort sett.
56.	J	En 4:a då. Det här var bara en liten snabb genomgång för att vi ska få en förståelse för vad som görs och inte görs i dina projekt.
57.	P1	Absolut.
58.	C	Vi hade några begrepp till i samma fråga som vi också vill ha rangordnade. Och det rör rollerna i projektet. SCRUM-master , Produkt ägare och teamet i sig, förekommer alla tre i alla projekt?
59.	P1	Ja det gör det. Däremot så kan produktägaren variera. Det behöver ju inte vara produktägaren på riktigt, utan det kan vara en mellanhand. I mitt förra projekt så var det egentligen bara en projektledare som var produktägaren fast det var givetvis inte hon som finansierade projektet. Hon hade bra kontakt med produktägaren och så fick hon motsvara vår produktägare helt enkelt. Och det är ganska vanligt att man gör på det sättet.
60.	C	Men just rollen som produktägare finns alltid?
61.	P1	Ja den finns alltid.
62.	C	Det är då någon med stor kunskap om verksamheten och har kontakt med kunden helt enkelt.
63.	P1	Ja precis.
64.	C	Då sätter vi en 4:a på rollerna.
65.	J	Då går vi in lite mer på detalj nu då. Hur ser processen ut i detalj i ditt arbete med sprint-retrospektiv, vad avhandlas och vilka instrument används?
66.	P1	Då ska vi börja med instrument kanske då. Det är väl i stort sett notepad. Ingenting annat. Möjligtvis några lappar på skrivbordet också men vi använder absolut ingenting som vi följer upp mer än att vi skriver ner allt som vi kommer överens om att vi behöver förbättra. Vi kommer fram tillsammans om vad som

		gick dåligt helt enkelt. Sedan självklart också vad vi beslutar och vad vi ska göra åt saken. Sedan mötet efter det ser vi till att försöka följa de punkterna som vi faktiskt skrev ner. Men dom skrivs bara in i notepad och sedan gör jag en sammanställning i ett mail och mailar ut det till hela teamet.
67.	C	När du säger mötet efter. Är det nästa Daily SCRUM då?
68.	P1	Nej, nästa Sprint-retrospektiv möte.
69.	J	Själva processen då, i det här fallet, nu jobbar ni lite distribuerat på olika platser osv men hur sker själva mötet och vad brukar hända?
70.	P1	Ja eftersom det är distribuerat så sker det då över telefon, telefonmöte. Och då går vi igenom, framförallt refaktoreringen av kod är en stor sak. Det kommer alltid upp på de här mötena. Man sitter och kodar i 1,5-2 veckor och då märker man att vissa practices inte är best practice. Men i ett SCRUM-projekt så är inte det relevant många gånger. Visst du kan skriva världens finaste kod, men det kommer att ta ett halvår att göra det. Produktägaren i sig bryr sig inte om koden är fin, utan han/hon vill ha en färdig produkt helt enkelt. Efter 2 veckor så går vi igenom vissa saker som att här kände vi att det inte var den mest snygga koden men om inte det då spelar någon roll för nästa sprint, Ja då tar vi inte upp det på ett sådant möte. Utan just nu fungerar det liksom. Men när det väl kommer till att vi faktiskt måste göra något åt det, vad skulle man kunna säga, jo tex. Om man vill via en kodad funktion vilja kunna skicka ett mail till någon, vilket är väldigt lätt, då man bara kopplar en sådan funktion till en send-button på en hemsida tex. I sprint 1 är detta kanske relevant. Sprint 1 kanske inte behöver mer. Men sedan i sprint 10 så ska det helt plötsligt in en kö där ett mail ska kunna skickas om 2 timmar osv. Och då måste vi göra om och refaktorerar det vi gjorde i sprint 1. Men vi gör ingenting åt det förrän vi faktiskt kommer till sprint 10. Då det blir ett problem eller då det blir relevant.
71.	J	Så när vi talar om vad som avhandlas och diskuteras på sprint-retrospektiv mötena så är det i princip kodning?
72.	P1	Nej, inte bara. Det är ju vad som har fungerat i själva arbetet och vad som inte har fungerat. Det kan ju vara allt i fråga för långa stand-ups, dålig kommunikation osv. Vi hade ett exempel här nu senast. Där en av våra teammedlemmar misstolkade ett use-case. Use-caset i sig var inte jätte väl utformat utan var uppdelat i olika filer. Missförståndet upptäcktes alldeles för sent. Vi skulle ha demovisning på måndagen och vi märkte det på fredagen innan, då var det helt omöjligt att klara av detta tills dess. Då tog vi upp det på retro-mötet att vi måste köra en review på koden när den är klar. Inte när vi börjar närma oss en release utan när den är klar sätter vi in det i "sprint backloggen" där vi skriver ner det att vi sätter en timma på review och sedan när reviewn är klar så kan man sätta "task/item" som "klar/done" helt enkelt. Det är typiskt en sådan sak som man tar upp på retro mötena. Och man följer upp det och ser till att det faktiskt blir gjort, så att man inte stöter på problemet igen.
73.	J	Under själva retro-mötet då, hur går diskussionen till, gör du något speciellt för att få igång snacket?
74.	P1	Ja, det är ju så att dessa retro-möten är till för att vi ska bli bättre och effektivare

		i vårt arbete. Mötet är inte till för att peka ut någon utan det är att få fram problem och "pinpointa" problem. Det är också viktigt att se över vad vi gjort bra självklart. Jag brukar börja varje möte med att fråga varje team-medlem om 2 saker de tycker har varit bra under gångna sprinten. På så vis så får man igång diskussionen.
75.	J	Nästa fråga då. Vad anser du om sprint-retrospektiv och reflektionsarbete inom SCRUM? Vad tycker du kan bli bättre?
76.	P1	Ja. Jag tycker personligen att retro-mötet är det som ger mest i SCRUM. För att man involverar hela teamet och man tar beslut gemensamt och man kan faktiskt utan att peka på enskilda personer säga vad man ska undvika och vilken väg man ska gå. Problemet med SCRUM generellt är ju det att kunden inte känner till det. Våra kunder är väldigt dåligt utbildade inom SCRUM, så det är väldigt viktigt när man går in i ett projekt att som SCRUM-master faktiskt se till att kunden faktiskt blir utbildad i det. Alltså jag säger inte att man måste skicka iväg kunden på en veckas kurs utan som SCRUM-master så måste du få dom att förstå hela processen, förstå alla fördelar och alla möten osv. Sen är det självklart så att man inte behöver informera kunden om alla mötena, då kan dom anse att "har ni möte nu igen" när ni kan sitta och koda istället. Men i och med att det faktiskt förbättrar hela systemet och kodningen samtidigt som man blir mer effektiv även om man då har 2 timmars möte så är det ju positivt. Det enda negativa är enligt mig att kunden sällan är utbildad i det helt enkelt. Oftast inte, det händer ju att dom är det så klart.
77.	J	Det leder lite in på nästa fråga, och det är då, vilka faktorer påverkar utförandet samt möjligheten att utföra ett retrospektiv och förbättringsarbete enligt dig. Alltså, vad krävs för att arbetet med retrospektivet ska resultera i förbättring?
78.	P1	hmmm...ja.
79.	J	Om du fokusera på det första där, vilka faktorer är det som påverkar?
80.	P1	För att mötet ska bli av eller?
81.	J	Ja, dels för att kunna utföra mötet samtidigt som det skall ge någonting.
82.	P1	Ja, det är ju att teamet faktiskt vill förbättra sig. Faktiskt att man...om inte teamet vill förbättra sig, om de inte bryr sig, eller om de inte känner något för projektet över huvudtaget så är det inte lika intressant heller att gå på ett retromöte. Utan då sitter de bara där... Man har ju varit på möten som folk egentligen...de säger inte ett skit liksom, de bara sitter där och halvsomnar. Men då gäller det ju också för SCRUM-mastern att se till att de faktiskt är med i diskussionen. Men det är ju ett måste egentligen att teammedlemmarna vill göra detta. Om inte teammedlemmarna vill köra SCRUM så ska man ju inte göra det. Och sen är det ju det där med produktägaren också... de är ju inte... Nu har jag ju inte explicit fått tillsagt mig att det där får ni inte göra. Men... Det är inte populärt med utvecklare som sitter i möten, och det gäller retro-mötet också. Det är...trots allt så är det de som betalar din lön...i stort sätt...som konsult i alla fall. Då betalar de ju ett timpris för var konsult. Det är kanske lite skillnad

		<p>om de betalar en klumpsumma för hela projektet men nu betalar de timpris och då kanske de inte vill sitta och spendera de pengarna på att utvecklarna sitter i möten hela dagarna. Nu är det ju inte så många möten, men det är klart att det kan vara en nål i ögat för dem. Det kostar ju pengar. Nu är ju vissa...nu är ju vissa kunder lite mer känsliga än andra. Det finns ju vissa som inte bryr sig alls och säger att, så länge vi får ett bra resultat så spelar det ingen roll liksom. Medans andra då är väldigt priskänsliga...de vill liksom skära ned på priserna så mycket så möjligt. Och det kan man ju förstå visserligen men... Men då gäller det ju faktiskt att utbilda produktägaren om SCRUM, om varför dessa möten är så viktiga och varför hela processen är viktig, så att de inte faktiskt vill försöka skära ned på de där sakerna. Och det är väldigt svårt faktiskt, får då har du ju alltid någon som tänker ekonomi och någon annan som tänker system, och det kan ju alltid bli en "clash" där emellan.</p>
83.	C	Okej, så... om vi sammanfattar det lite så är det teamets egen benägenhet att faktiskt förbättras under projektets gång...
84.	P1	Absolut
85.	C	Sen är det andra faktorer som pengar och kunden
86.	P1	Ja, jo precis.
87.	J	Sammansättningen av gruppen och så vidare, tror du att det påverkar?
88.	P1	Sammansättning?
89.	J	Ja...om ni är fem utvecklare i ett projekt till exempel och beroende på deras erfarenhet med att jobba enligt SCRUM. Hur tror du att det påverkar, just utifrån att vara utvecklare?
90.	P1	Ne alltså...SCRUM är jävligt lätt på det sättet att man...det tar inte mer än två möten så är man igång med det liksom. Så även om du har väldigt lite praktisk erfarenhet av det så räcker det med att du läser på Wikipedia liksom, vad grunderna är. Sen när du väl sitter där i praktiken så är det inga problem, så jag ser inte erfarenheten som något större problem utan det är någonting som man bara lär sig med gången, helt enkel.
91.	J	Då tar vi nästa fråga här nu. Vilka faktorer påverkar kvaliteten av arbetet med retrospektivet som verktyg för förbättringsarbete? Det kom vi in på lite tidigare. Till exempel att ni sitter på olika ställen.
92.	P1	Ne..alltså, givetvis så är det alltid bättre med fysisk kontakt men jag tycker inte att det påverkar resultatet i sig. Däremot så kan diskussionerna bli lite längre än normal och det kan bli missförstånd och så vidare men då gäller det ju bara att vara klar, alltså klar och tydlig, som SCRUM-master att har vi kommit överens om detta...om man sitter och diskuterar i 20 minuter så måste man ju avsluta med en summering. Stämmer det att vi har kommit överens om detta? Och sedan har man ju den uppföljningen som man skriver ner allting man kommer överens om, mailar det, och sen får de svara igen om det kanske inte stämmer. Och den biten brukar man ju inte ha när man sitter fysiskt, men resultatet blir ju det samma, alltså så länge det inte har blivit något missförstånd och det har inte skett hitintills. Så funkar det liksom, det är inga problem. Däremot känner jag

		väl inte att det finns något verktyg som man faktiskt måste ha utan det är snarare att skriva ner det en gång och se till att alla har fattat det, behåll det till nästa retromöte och sen kan du slänga det. Alltså, det finns ingen anledning till att spara massa dokument bara för sakens skull utan meningen är ju att förbättra arbetet och sen om den...om nu det som man kommer överrens om eller det felet man gjorde i sprint 1 återkommer i sprint 10 då spelar det ingen roll egentligen om man gjorde det felet i sprint 1. Meningen är ju ändå att man ska följa upp det och se till att det ändras. Det är ju alltid lätt att vara efterklok och säga att "Ja men va fan, det där gjorde vi men det skulle vi inte ha gjort men nu gjorde man det". Så det spelar ingen roll ändå, ja som sagt, det är bara att betona att man behöver inte sitta och spara de dokumenten mer än till nästa sprint, eller retromöte. Då kan man gå igenom det och se att man följt upp det, efter det så är det bara att slänga skiten.
93.	C	Då kommer vi till nästa fråga här...vi pratade lite om detta tidigare. Det kommer upp saker som kunde gått bättre på de här retromötena.
94.	P1	Mm
95.	C	Gör ni någon prioritering på de saker som kommer upp på ert retromöte?
96.	J	Och hur prioriterar ni de område som behöver förbättras inför nästa sprint?
97.	P1	Ja, det är ju så att... innan retro-mötet. Eller vi har ju retro-mötet efter vår demonstration, vi har det ju inte när vi kodat klart och innan demonstrationen utan vi har det verkligen efter. Men innan demo-mötet så sätter vi oss ner, detta ska man kanske egentligen göra på demo-mötet...Men då sätter vi oss ned och kollar i produkt backloggen vad som ska göras, så sitter vi och poängbestämmer det helt enkelt. Men då vet vi även... vid det tillfället så vet alla vad som ska göras. Vi går ju även igenom implementationsdetaljer. Vi går inte ner på alldeles för låg nivå men vi diskuterar ändå implementationen och alla kommer överens om hur en "task" ska lösas. När vi vet det, då vet vi även vad som är viktigt i retron, vad vi gjorde fel, vilket är det viktigast att prioritera för nästkommande sprint. Om vi tyckte att en lösning har ful implementation, vi kanske skulle göra någonting åt det. Om inte det är med i vårt kommande arbete så är det inte relevant. Alltså, som sagt, du kan sitta i två år och bara förfina koden, du kan få hur snygg kod som helst men du får ingenting gjort liksom. Så därför gäller det att prioritera det som faktiskt är relevant för kommande sprint. Sedan om det...nu har jag inte stött på det...så jag vet ju inte vad som hade hänt om det hade varit så att det varit alldeles för mycket som skulle refaktorerats inför nästa sprint. Man hade helt enkelt fått skippa något.
98.	C	Men tar ni dessa beslut tillsammans som team då?
99.	P1	Ja, de tar vi tillsammans. Absolut, absolut. Jag tar i stort sätt inga beslut själv. Alltså, enstaka beslut som produktägaren kommer med. "Kan ni göra detta?", då kan jag ta det beslutet själv men ofta så diskuterar vi saker i teamet. Jag ska ju inte vara någon beslutsfattare, jag ska bara vara där och se till att reglerna följs och att hela processen blir positiv för alla medverkande helt enkelt.
100.	J	Sen handlar nästa fråga lite om uppföljningen. Hur säkerställer ni att arbetet med retrospektiv faktiskt leder till förbättring? Och hur följer ni upp att

		resultatet har lett till förbättring? Du har ju tagit upp uppföljningen under en lite kortare tidsperiod. Men hur säkerställer ni att något viktigt ni lärt er fortfarande finns kvar efter en månad eller två?
101.	P1	Ja...det är en lite halvklurig fråga. Men det är...vad ska man säga... Arvet är ju i koden. Alltså arvet av det vi lärt oss finns och existerar i koden. Och när du väl kommer dit så vet du vilket sorts beteende du ska utveckla, eller vilka sorts regler som gäller när du utvecklar. Om vi väljer att refaktorera någonting så kan det visserligen vara så att allting inte är refaktorerat, du kanske inte har tid att refaktorerat allting. Men då gäller det alltså att påminna, för vi har ju det där reviewarbetet. När du väl har gjort en sak klar då ska någon köra en review på det, alltså inte två, tre timmar och ingen speciell kodgranskning, bara se till att de nya reglerna följs, liksom. Men det är väl det enda då i så fall, att man ser det i koden och ser det i review'n helt enkelt. Men det är inte så att vi skriver ner det i något längre perspektiv och står med pekpinnen och säger att "där gjorde ni fel" liksom. Jag vet inte om det är svar på er fråga?
102.	C	Ni har alltså ingen uppföljning på så sätt att man kanske räknar antalet saker som kommer upp under första retro-mötet och sedan säkerställer att det antalet problem minskar. Att problemen blir färre ju fler sprintar in i processen man kommer utan ni skriver ned och hoppas att det blir bättre och tar det inom grupper?
103.	P1	Jo. Fast det kanske går, men det är inte helt sant...men det kanske går in lite på det att vi estimerar oftast inte i timmar. Alltså vi estimerar i poäng och då vet vi hur många poäng vi kan göra i en sprint. Och ju fler sprintar som går desto fler poäng kommer vi att kunna göra. Alltså vi märker ju det att vi blir bättre och bättre i systemet för att vi tar fler och fler poäng hela tiden. Det är förmodligen inte svar på din fråga men det blir lite på det sättet att vi märker att vi arbetar mer effektivt hela tiden. Ne, det är nog fan inte svar på din fråga.
104.	J	Om man vänder och vrider lite på frågan och säger "tror du att det skulle vara positivt för gruppen att ha ett större retrospektiv eller reflektionsmöte där man gör en mer djupgående dykning på alla problem som har dykt upp?"
105.	P1	Ne, jag tror inte det för grejen är den att det är fortfarande sprintar vi arbetar med. Alltså under en två veckors period så kan en refaktorering varit nödvändig för sprinten, men hoppar du till fjärde eller femte sprinten så kanske du måste gå tillbaka till sättet du arbetade på innan. Så då... om man följer upp det så kan det bli väldigt komplicerat för då kan man säga kanske att "Oh den refaktoreringen var bra innan men nu är den dålig, men den kommer att bli bra igen". För det är ju så att kunden kan ändra sig på (knäpp med fingrarna) direkt liksom. Du kan ha suttit i två veckor och implementerat någonting men sen efter dem två veckorna säger de att det inte ska vara med releasen, vi ska ta bort det. Och det sker ju hela tiden att kunden ändrar sig och därför att ha en sådan uppföljning på det sättet då är inte relevant för att det behöver inte vara relevant om sex månader. Om ni förstår vad jag menar.
106.	J	Mm absolut. Till sist så har vi ett antal påståenden som vi vill att du ska ta ställning till och sätta på en skala från 1 till 4. Där ett betyder stämmer inte, två stämmer ibland, tre stämmer oftast och fyra stämmer alltid. Frågan lyder hur väl



		anser du att sprint-retrospektiv främjar följande påståenden. Det första påståendet är: Det hjälper teammedlemmarna att dela och förstå varandras perspektiv.
107.	P1	Ja, en fyra.
108.	J	Påstående två. Det integrerar lärande lärande på individuell nivå med lärande på teamnivå.
109.	P1	Ja, antingen en trea eller fyra där också. Vad var trean?
110.	J	Det stämmer oftast.
111.	P1	Ja, trea då.
112.	J	Påstående tre. Det identifierar gömda problem.
113.	P1	Det behöver det ju inte alltid göra. Ta tvåan, det stämmer ibland.
114.	J	Påstående fyra. Det dokumenterar best practices och problem.
115.	P1	Nja... som sagt vi väljer att inte dokumentera det på det sättet så det är inte så att vi har någon dokumentation som vi kan gå tillbaka till och faktiskt säga att "detta är best practice". För som sagt, det är ju bara relevant för sprinten i sig och nästkommande men det har ju inte ett längre liv än så, enligt mig då. Fast nu har jag kanske lite väl mycket fokus på dokumentera. Men visst, det tar ju upp best practices och problemlösningar så det är ju en fyra på det men däremot så dokumenterar vi inte det på ett sånt sätt att vi kan gå tillbaka till det. Så jag vet inte alls hur ni vill betydsätta det då.
116.	C	Ni har ju någon form av dokumentation. Ni dokumenterar i notepad som du sa och dokumentet lever till kommande sprint. På så sätt har ni ju viss dokumentation. En två eller trea kanske?
117.	P1	Ja.
118.	J	Påstående fem.
119.	P1	Vänta, jag ska bara hämta snus.
120.	J	Påstående fem. Det ökar tillfredsställelsen av arbetet genom att ge personer feedback på deras arbete.
121.	P1	Jo, men det är en fyra på det. Med plus i kanten!
122.	J	Och påstående sex. I somliga fall kan det förbättra möjligheterna att korrekt estimerar projektets kostnader.
123.	P1	Nja... Jag skulle vilja säga att det är en etta. Alltså det är mycket möjligt att det finns någon annan som skulle säga något annat men för mig så har det aldrig varit ett verktyg för det liksom.
124.	J	Ok, precis. Möjligtvis är det så att era erfaren heter från sprint till sprint säker kostnaderna.
125.	P1	Jo, exakt exakt. Det är vi viktigt att...om man säger som så att produktägaren,

		han får stå för alla use-case som ska implementeras. Och sedan när alla use-casen...det kommer ju aldrig att bli alla. Men i alla fall de inom version ett releasen, om man säger så. När alla use-cases är uppsatta kan vi som utvecklare, eller hela teamet då sätta oss ner och estimerar hur många poäng det är. Vi vet hur mycket vi kan göra under en sprint. Alltså vi säger 80 poäng. Och hela produkt backloggen kanske ligger på 640, då kan vi ju estimerar att det tar ungefär 8 veckor... eller det tar åtta sprintar för oss att göra klart allting. Och det är väl enda gången som vi faktiskt kan veta den ungefärliga kostnaden.
126.	J	I början av projektet då?
127.	P1	Nja, efter två till tre sprintar. Första sprinten är alltid off liksom. Folk har ingen koll alls på sin estimering. Den andra sprinten börjar det blir lite bättre och tredje sprinten är väl det värdet man faktiskt kan börja använda sig av. Sen kommer det ju att förbättras hela tiden... men tidigast efter tredje sprinten skulle jag vilja säga.
128.	C	Du sa alltså en etta på det sista påståendet?
129.	P1	Ja, för oss har det inte med retrospektiv-mötet att göra utan detta sker under vårt estimerings-möte. Det är när vi sitter ner och estimerar vad de olika sakerna göra, hur mycket det kostar i poäng. Det kan ju omvandlas till det men det gör vi inte, men det skulle kunna göras. Man skulle kunna sagt att det var i tid istället.
130.	C	En typisk sak som borde kunna komma upp på ett retro-möte. Det är ju under de tidigare sprintarna, precis som du sa, att man inte lyckas implementera så många poäng som man bestämt.
131.	P1	Jo precis, absolut. Det kan ju bero på många faktorer. Oftast är det ju i början att folk är alldeles för optimistiska. De ser en sak och säger att det inte kommer ta mer än fyra timmar. Men lägger man på unit-test, review med mera så kanske det ligger på tolv timmar. Även en simpel sak liksom. Men visst, det tas upp på retro-mötena att man missar målet. Vad ska vi göra åt detta? Men själva estimeringen och hur mycket själva kostnaden kommer att bli det diskuterar vi aldrig i retromötena. Utan det är ju någonting som vi snarare tar upp under demo-mötena då vi presenterar vårt arbete. Då säger vi att nu ligger vi i den här farten, vi kan avverka 120 poäng. Kollar vi produkt backlogen så ser vi hur mycket vi kan ta i kommande sprint och vi ser även hur mycket mycket vi har kvar helt enkelt. Då brukar vi, eller jag brukar då säga, att om vi håller denna farten så är projektet klart om X antal veckor, eller sprintar. Men det är ju alltid så att produktägaren lägger till saker. Alltså, det bara poppar upp ideer för dem hela tiden och det är det som är fördelen med SCRUM. De ska ju faktiskt sitta och se sitt eget system. Hur de bildas hela tiden. Så de kan ha feedback hela tiden på de små delarna som vi arbetar på. Så den tiden kommer alltid att utökas, tyvärr är det så. Om man har en deadline är det viktigt som SCRUM-master att säga ”Ja du kan få det men om du väljer att kasta in det i sprinten nu så kan det bli så att vi faktiskt inte blir klara med de andra funktionaliteterna när vi når deadline så att säga”.
132.	J	Har du något att tillägga Christian?

133.	C	Ja. En fråga som vi egentligen skulle tagit upp tidigare när vi diskuterade detaljerna kring retrospektivet. Vem är det som deltar på retromötet?
134.	P1	Det är bara utvecklarna. Vi tar inte med produktägaren på det.
135.	C	Det vill säga SCRUM-masters och teamet?
136.	P1	Mm. Hela teamet förutom produktägaren då. Det bör nämnas att det gäller mitt nuvarande projekt. Vi har ju haft andra retrospektivmöten där faktiskt produktägaren satt med. Och det är kanske den erfarenheten som gjorde att jag inte ville att vi skulle göra det. Än så länge har det inte gett dem någonting och det har bara försvårat vår process. Det kommer en massa frågor om varför ni gjorde så eller så. Och så står han där och så fort de hör "bla..., bla... fel" så noterar dem det liksom om någon gjorde fel och då blir det helt plötsligt negativ stämning. Retrospektivet ska ju inte straffa de som gjort fel utan ska snarare se till att folk gör rätt istället. Det har ingenting med bestraffning att göra utan det har bara med förändringsarbete att göra. Det blir lite galet när produktägaren faktiskt är med på dessa möten.
137.	J,C	Kanon, tack för din hjälp med denna intervju, det var mycket intressant!

## Bilaga 4, Intervjuprotokoll – P2

Intervjuperson 2 – P2		
Talande (J, P2, C)		
1.	J	Vad har du för bakgrund, vem är du och vad har du för utbildning?
2.	P2	NNNN, jag är 28 år och jag är systemutvecklare, agil evangelist och agil coach. Det är liksom... jag håller på med systemutveckling får man väl säga i alla aspekter då. Jag har hållit på med systemutveckling professionellt i 10 år ungefär. Jag började programmera när jag var 12. Utbildning på gymnasienivå, naturvetenskap i botten och sen har jag läst systemvetenskap här i Lund.
3.	J	När examinerades du?
4.	P2	Du menar när jag fick min examen?
5.	J	När du började jobba professionellt då...

6.	P2	Jag började jobba professionellt efter min första termin på systemvetenskap och sen jobbade jag parallellt med mina studier. Heltidsjobb och studier samtidigt. Sen plockade jag dessutom en massa poäng på datalogi. Jag har ingen examen. Jag har en uppsats kvar som inte är färdigskriven. Men utöver det så har jag gått hela programmet. Det blev för mycket med jobbet där... och nu har jag jobbat så länge så att känns inte som att... Så det är väl det. Som jag sa tidigare är jag certifierad SCRUM-master och hållit på med agil-metodik och agil-utveckling i 6-7 år ungefär. Jag började få upp ögonen för det på de tekniska detaljerna, mycket med testdriven utveckling och den typen av saker. Jag tyckte dessutom att så som man jobbade då fungerade inte.
7.	J	Traditionell utveckling då?
8.	P2	Ja, traditionell utveckling. Jag satt med i en del sådana projekt och märkte att det finns ju någonting fel eftersom man sitter och jobbar i två år på ett projekt och man kommer aldrig till skott liksom, det släpps ingenting, det kommer inte ut någon mjukvara. För mig, så som jag hade jobbat tidigare under mina hobbyprojekt så handlade det ju om att släppa hela tiden. Så när jag väl började titta på XP, för det var XP jag började titta på, så märkte jag att mycket av det som jag gjort tidigare, i liksom hobbyprojekten med polare och sånt, det är ju faktiskt XP. Så om jag tar mig igenom den teorin och lär mig det här så kanske det finns ett och annat att hämta ur de här sakerna. Sen har jag faktiskt haft turen att hamna i massor av agila projekt och driva egna agila projekt och är med i många grupper, agile-Skåne och sånt. Vi träffas varje månad och pratar just om agile och hur man kan ta det vidare till nästa steg och sånt.
9.	J	Och SCRUM då. Om vi inriktar oss lite på det...Det har du också jobbat med mycket i olika projekt?
10.	P2	Jag har jobbat med SCRUM vid två tillfälle får man väl säga egentligen. Sen beror det ju på...för att... Jag kan ju tycka att det inte var SCRUM, men de som implementerade tyckte att det var det, och det kan man ju bråka om. Om man går enligt SCRUM definitionen, enligt hur jag tycker att man skulle köra SCRUM så har jag aldrig lett en fungerande implementation av SCRUM.
11.	J	Du nämnde ju det lite tidigare här... Kan du nämna någon utbildning/certifiering inom SCRUM/agila metoder du har fått eller genomfört?
12.	P2	Ja, alltså jag är SCRUM-Master certifierad, och det är den jag har.
13.	J	Det är den du har?
14.	P2	Ja.
15.	C	Vad innebär en sådan utbildning?

16.	P2	När jag tog den så innebar det att du sitter tillsammans med examinatorer. Jag hade Jeff Sutherland och Christer Östergard och Jim Coplin. Du går en tvådagars utbildning med agile-gurus och sen så beror det helt på vad du gör det till själv. Jag tyckte ändå att jag fick ut väldigt mycket av det för att ha de här människorna som har den erfarenheten av SCRUM och agile och kunna prata med dem och ställa direkta frågor. Jag hade ändå jobbat med agile och SCRUM innan jag gick utbildningen och fråga liksom så här och så här och så här. Om mitt team gör det här... hur ska jag reagera om på det här? Men jag tror det fanns många andra där som bara gled med. Efter två dagars kurs så får du certifieringen oavsett om du lärt dig någonting eller inte. Så att för mig så gav det ändå en hel del för det var ändå två dagar med "hot shots". Men jag kan ju tänka mig att generellt så är det en grej man bara kan glida igenom. Däremot har jag hört nu att de har infört ett prov eller något sånt, jag vet inte riktigt hur det funkar. Men det var det inte när jag tog den.
17.	C	Som tredje fråga lyder så här: SCRUM's ramverk består av ett antal regler och guidelines. I egenskap av SCRUM-master vill vi att du lyssnar på ett antal påståenden och säger en siffra på en skala mellan 1-4. Om det är något som ni har följt, eller som du har följt i dina SCRUM projekt. Ett innebär att det här momentet inte funnits eller inte har använts. Två innebär att momentet finns och används i begränsad utsträckning. Tre betyder att det används för det mesta och fyran innebär att det använts i samtliga SCRUM projekt.
18.	P2	Är det då projekt som jag har varit ansvarig för eller projekt som jag har suttit med i?
19.	C	Som du har varit ansvarig för. Då börjar vi med artefakterna... Product Backlog.
20.	P2	Det är samtliga.
21.	C	Sprint Backlog.
22.	P2	Samtliga.
23.	C	Sprint Burndown Chart
24.	P2	Samtliga.
25.	C	Detta är för att vi vill få ett grepp om hur strik man följer SCRUM. Processerna går vi vidare til sen. Sprint Planning Meeting.
26.	P2	Samtliga.
27.	C	Sprint.

28.	P2	Samtliga.
29.	C	Jo det säger ju sig självt. Daily SCRUM eller Stand-up.
30.	P2	Samtliga
31.	C	Release planning meeting.
32.	P2	Vad var tvåan och trean igen?
33.	C	Finns/ används i begränsad utsträckning är tvåa. Trean är finns och används för det mesta.
34.	P2	En trea.
35.	C	Sen har vi sprint-review meeting.
36.	P2	Review meeting. Syftar ni på retrospektivet då eller?
37.	C	Nej, vi tänker mer på review i form av demon och utvärdering med kunden.
38.	P2	Ok, samtliga.
39.	C	Sen till slut har vi Sprint-retrospective
40.	P2	Mm, samtliga.
41.	C	Slutligen lite om SCRUM teamet och rollerna. SCRUM-master, har det funnits i samtliga projekt?
42.	P2	Mm, de gångerna vi har sagt att vi har kört SCRUM så har vi haft en SCRUM-master. De gångerna vi inte haft en SCRUM-master har vi sagt öppet att vi kör en SCRUM liknande agil metod för att det är per definition inte SCRUM om du inte har en SCRUM-master.
43.	C	Produktägare.
44.	P2	Ja, det är samtliga projekt.
45.	C	Och slutligen själv teamet...
46.	P2	Ja...Du kommer ju inte så långt utan ett team!

47.	C	Nej precis. Men de är dessa tre roller som dyker upp i litteraturen så vi ville ta upp samtliga!
48.	C	Slutligen...Det här med time-boxing. Om man går tillbaks till de här processerna som sprint-planning, sprint, daily SCRUM, release planning, sprint-review och retrospektiv. Har du alltid time-boxat de här momenten?
49.	P2	Ja.
50.	C	Ja. Då är det en fyra också.
51.	J	Så även t.ex. en daily-SCRUM. Då har ni sagt att det får inte ta längre än 10 minuter eller?
52.	P2	Ja. Oftast är man ju färdig snabbare men...ja minns inte ens hur långt det var liksom, 5 eller 10 eller nåt sånt. För det är egentligen...så som vi körde det när vi körde SCRUM. Nu gör vi det annorlunda men i SCRUM har vi alltid kört tre frågor. Vad gjorde jag igår, vad tänker jag göra idag och har jag några problem. Man ska svara på dessa tre frågor väldigt kort och koncist. Börjar folk liksom stå och prata mellan så slår man dem också får de straff.
53.	C	Det gäller stand-up då ja?
54.	P2	Ja.
55.	C	Eller daily SCRUM. Det heter lite olika beroende på vart man läser någonstans.
56.	P2	Ja. Jag kan ju säga så att jag kommer att använda lite...ni får säga till om ni inte fattar terminologin. Men jag har aktivt försökt löskoppla mig själv från SCRUM terminologin för att...även om jag gillar SCRUM för att det hjälper en att implementera agile så tycker jag i mångt och mycket det är rätt mycket som är fel. Så jag använder saker som stand-up, och iteration istället för sprint. Så ni kommer ju märka att jag använder lite andra ord så är det något ni inte förstår så får ni bara säga till.
57.	J	En fråga som du var inne på där...när det gäller SCRUM-mastern. I de fall man inte använder SCRUM-mastern så var det ett gemensamt beslut i teamet att inte ha en SCRUM-Master...
58.	P2	Ne, det var inte det jag menade. Jag menade bara att om det har funnits ställen där vi sagt att vi har inte denna rollen...för att i min bok, eller i SCRUM's bok, skulle jag vilja påstå att SCRUM masterns roll är att skydda teamet från externt crap som kommer. Och är det så att man inte har en sådan dedikerad person, utan om man säger att man är tre stycken som sitter och jobbar här liksom...om det är någon som gör "burn down charts" och rapporterar utåt... fine, det kan han göra men om kunden fortfarande kan kuta in i rummet och ställa sig och skrika på vilken utvecklare han vill och det inte finns en dedikerad person för att ta hand om den här skrikande kunden, då har du ingen SCRUM master och då kör du inte SCRUM. Och då

		ska du inte säga att du gör det heller. För att det är bara fel. SÅ i de fall det inte har funnits utrymme för att ha en person som verkligen är väktare för teamet och skyddar teamet från problem och externa saker. Där har vi helt enkelt sagt att det räknas inte som SCRUM. För mig är det så i alla fall. Du kan inte ta delar av SCRUM och säga att du kör SCRUM, antingen gör du det eller så gör du det inte.
59.	C	Är det det som blir "SCRUM-ish"
60.	P2	Ja som sagt. SCRUM-butt och SCRUM-Whatever och SCRUM-like. Det finns inte. Antingen kör du SCRUM eller så kör du inte SCRUM. Kör du inte SCRUM så säg då hellre att du kör agile för att det är mer korrekt att göra det, även om du har 99% SCRUM.
61.	J	Då kan vi gå in på nästa här då. Vi kanske får formulera om frågan lite. Men det här är utifrån SCRUM-mastern också. Hur ser processen ut i detalj i ditt arbete med retrospektivet, vad avhandlas och vilka instrument används? Du får jättegärna gå in djupare om det är så att du har sett hur andra har gjort den här biten.
62.	P2	Just retrospektiv har jag väldigt bra koll på för att... för mig är det alltså...det är inte... Jag har gjort väldigt mycket facilitering av retrospektiv. Bara...hos andra liksom. För ibland så är det ju bra med en extern facilitator som kommer dit och kör retrot. Så som jag brukar lägga upp det är ju enligt agile retrospectives boken, jag vet inte om ni sett den? En tunn gul bok. Fantastiskt bra. Då är det liksom... först har du "setting the stage". Det handlar ju mycket om att få in folk i rätt stämning för att det handlar om att när man är där så ska vi tillsammans reflektera över hur det har gått och hitta lösningar på de problem som kan finnas. Vi är inte där för att anklaga varandra, vi är inte där för att peta på varandra eller hitta vems fel det var. Det bryr vi oss inte om. Vi är där för att prata igenom problemen och försöka hitta några vettiga lösningar på dem. Sen efter det så kör man någon form av datainsamling. Jag kör alltid olika övningar vid varje retrospektiv. För att om man kör samma övningar, vilket jag märkt att väldigt många gör, vilket är skillnaden med att ta in en extern facilitator, så kan man fastna i ett visst mönster och gör om samma sak om och om och om igen. Det är aldrig bra. Du vill hela tiden röra om i grytan. Så för dessa steg så brukar jag köra olika övningar. Du har "setting the stage" och så har man datainsamlingen, så att man på något sätt samlar in data från deltagarna. Sen har du datagrupperingen där man på något sätt grupperar data. Sen väljer man ut den data, någon viss delmängd av information, sen bearbetar man den informationen, för att så småningom komma fram till något som ska göra det bättre eller fixa processen på något sätt. Ett exempel för "setting the stage" är då...det finns en som heter "prisoners, vacationers, shopper och explorers". Då tar man helt enkelt en skål och så sätter man upp de här och så säger man att: "De som känner sig som "prisoners" är dem som absolut inte vill vara här och tror att de hade gjort mer nytta om de varit någon annanstans. Ni som är "vacationers" tycker det är schysst att slippa ifrån datorn och få sitta inne på ett möte och kaka kakor. Ni som är "explorers", ni tittar bara runt och vill ta in så mycket så möjligt av allting och ni som är "shoppers" vill ta med en



		<p>grej härifrån”. Sen får alla skriva på en lapp vad de tycker att de är och lägga det i en skål så att det är helt anonymt, sen blandar man skålen och sätter upp det på en vägg. Du skriver så här många av den typen, så många av den typen och så många av den typen. Det hjälper för att se var man...det här brukar jag göra med nya grupper som jag inte känner för det hjälper en att se vad folk har för inställning till retrospektivet. Jag har alltid backup övningar om jag skulle behöva byta någonting på grund av att gruppen inte fungerar och sånt för det kan också hända... Vill ni att jag går igenom ett typsikt retrospektiv?</p>
63.	J	Ja det är intressant, det tycker jag.
64.	P2	<p>Så ska jag se till att det inte tar alldeles för lång tid. Sen ser man ungefär vad man har att göra med för typ av grupp och då vet man vilken typ av övningar man ska ha. Man kan även hitta vissa personligheter för det är också det mycket som facilitatorn i ett retrospektiv... Finns det någon som pratar väldigt mycket? Hur får jag då tyst på honom så att killen i hörnet säger någonting? För att när man faciliterar ett retrospektiv så är det viktigt att se till att alla på något sätt och för sin talan, för det är gruppens talan du vill ha fram, inte den skrikigaste individen. Sen efter det kör man någon form av översikt... dataintagning. Man kan sätta upp alla ”burndown charts” från iterationen, man kan sätta upp andra information radiators, man kan rulla in ”SCRUM-boarden”, man kan säga till folk att gå ut om man sitter i ett mötesrum nära där de arbetar och sen gå runt på alla arbetsplatser och kolla vad folk har på skrivborden, vad de har på skärmen för tillfället och sånt. Försök komma ihåg den föregående iterationen så som det har varit. Om man då har kört det här och man har fått några ”prisoners” i gruppen så säger man att: ”Efter ni gått runt en runda så är de som vill det välkomna tillbaks. Är det någon som verkligen inte vill vara här så får ni lov att gå härifrån. Men tänk på att om ni kommer tillbaks så är ni här frivilligt”. Då släpper...det är en psykologisk grej... då släpper man lite på den här mentala spärren att jag är här för att jag måste. Då har de fått en väg ut. Om de valt att inte ta den så har de aktivt valt att vara med på retrospektivet. På så sätt kan man liksom...man får hela tiden hitta dessa hinder och personligheter som kan dyka upp och försöka lösa de problemen. Sen när man kört denna dataöversikten så kan man ju köra mad, sad, glad. Det är en jätte enkel övning som vi brukar köra. Det går snabbt. Man sätter upp whiteboarden, ”mad”, ”sad” och glad. Sen får alla post-it lappar och så ska man skriva någonting som gjorde en arg, ledsen eller glad på en post-it, läsa upp det högt inför hela gruppen och sätta upp det på tavlan. ”Jag blev jätteglad när vi fick fika i tisdags”, ”Jag är jätteledsen för att byggservern kraschar hela tiden”. Sen kör man så och då kommer andra att få inspiration om de hör dig säga något och tänkta ”just det, det var ju det. Det var skitroligt!” och då kommer jag att tänka på det här... Sen låter man detta gå en ”time-boxad” tid. Sen har man fått sin datainsamling. Efter detta ska man gruppera datan. Ett sätt att göra detta på efter ”mad”, ”sad”, glad är att man gör ”silent mapping”. Då får alla gå fram till white-boarden, vara tysta och flytta lappar som har med varandra att göra. Då flyttar man liksom ihop lappar och om någon inte tycker att en lapp hör hemma någonstans får de flytta den. Men de får inte lov att prata. Sen får man hålla på så tills att det har stabiliserat sig på tavlan. Sen som facilitator så ringar man in lapparna</p>

		<p>och låter gruppen skrika ut namnet på de olika grupperna. T.ex. ”Team work”, ”Byggserver”, ”Teknik”. Då får man olika klickar av information. Sen för att välja vad man ska diskutera eller prata om så brukar man då, brukar jag, kanske använda röstning i ett sånt här scenario. Då får alla tre stycken magneter. Man får sätta hur många magneter man vill på samma punkt eller en magnet på tre punkter. Efter man gjort detta så ser man vilken punkt som har mest röster. Alla de andra lapparna tar jag bara och slänger i papperskorgen och suddar ut det. För att det är som så att vi har bara en begränsad tid på ett retrospektiv och ska vi diskutera alla problem som funnits så hinner vi inte med. Så ta ett av de viktigaste problemen och diskutera detta. Om de andra sakerna fortfarande är ett problem vid nästa retrospektiv tillfälle kommer de att komma upp på bordet igen. Så det är inte lönt att spara lappar, det är inte lönt att föra anteckningar, det är inte lönt att göra de här grejerna. Bara släng alla andra lappar som inte ska diskuteras. Sen efter detta så kör man någon diskussion kring det här och då finns det olika versioner som ”5 why's” eller att man kör en ren öppen diskussion, vilket fungerar jävligt dåligt generellt sett. Då blir det ju att den skrikigaste pratar mest. Man kan köra ”+delta” om vi nu ska prata lösningar på det här. Alltså, vilka lösningar ser du? vilka förändringar ser du? Det finns jättemånga olika övningar man kan göra. Sen finns det ett steg som jag glömde nämna innan och det är när allt väl är färdigt. Då brukar jag köra ett ”closing”. Det är som ett litet 10 minuters retrospektiv på retrospektivet. Så att jag som facilitator, för det är bara för min egen skull, kan ta till mig. Okej, vilka övningar tycker den här gruppen var bra och vilka var dåliga, var retrospektivet värdefullt för dem, kom vi fram till någonting vettigt? Så att jag kan förbättra mig. För att agile handlar ju liksom hela tiden om förbättring. Även jag som facilitator måste ju hitta var jag är svag och kan förbättra mig.</p>
65.	C	Just det. Det var väl lite det som vi fastnade för, ”inspect and adapt”, som ofta dyker upp.
66.	P2	Jo men det är genomgående liksom. Det finns jättegamla teorier man kan titta på som projektteorier och arbetsteorier som bara pratar om det här. ”Do, act, inspect, adapt”. Sen är det ju den här cirkeln... du gör någonting, du tittar på utfallet, du ändrar, sen gör du någonting och så fortsätter det.
67.	J	När du faciliterar retrospektivet. Är det flera grupper, SCRUM team, som du gör detta för eller är det bara ett
68.	P2	Nej. Det är för ett team.
69.	J	Hur stora brukar teamen vara?
70.	P2	Jag har faciliterat allt från ”tremanna” team till 10...12 är nog den största gruppen jag varit med om och då hade vi två stycken intressenter. För det är också en intressant sak man kan göra... Om man har många retrospektiv så ibland har man bara med teamet. Men ibland kan man ta med andra intressenter, om teamet tycker det är okej. Då får även dessa se hur metoden fungerar och också tycka till. Så du kan ta med produktägare och du kan ta

		med andra ”stake holders”. Som coach brukar jag föreslå att inte göra det alltid. För det finns...alltså det blir väldigt intimt, gör man ett bra faciliterat retrospektiv så blir det väldigt intimt och väldigt känsloladdat och du vill inte ha auktoritetspersoner där för att det kan lätt förstöra. Det är därför man ibland, om man har en dålig SCRUM master som agerat som projektledare mycket, så sparkar man ut denna och tar in en extern facilitator för då kan han få ut de riktiga problemen i teamet på ett annat sätt än vad en dålig SCRUM-master kan. En bra SCRUM-master är ju kompis med teamet och han är egentligen teamets skydd. Om teamet säger ”Gå och hämta kaffe till mig” så ska ju SCRUM-master gå och hämta kaffe. Det är inget fint jobb att vara SCRUM-master. Det är tvärt om. Du får ta all skit och får inget av äran.
71.	J	Vi kan hoppa vidare här för att vi kommer nog glida in på de här bitarna igen. Vad anser du om sprint retrospektiv och reflektionsarbete inom SCRUM och vad tycker du kan bli bättre? Brister som du sett/brukar se.
72.	P2	Vad jag tycker om det är att det är den absolut viktigaste byggstenen inom alla agila metodiker som finns. Det är helt ohållbart att kalla någonting för agile om man inte gör någon form av retrospektiv. Om jag går in i ett team som inte köra agile idag och som är sugna på att testa så är det här alltid det jag börjar med. Det är alltid första steget. Du behöver inte ens ha iterationer för att köra retrospektiv, du kan bara säga att vi kör ett retrospektiv varje månad. Och bristerna...bristerna är ju att man ibland ser det göras helt fel. Folk har inte läst litteraturen. Folk vet inte hur man faciliterar. Folk vet inte hur mycket psykologi det finns i att vara en facilitator, hur mycket man måste kunna om gruppdynamik, hur mycket man måste kunna om hur människor fungerar, hur mycket man snabbt måste kunna kartlägga olika personligheter i ett team, speciellt om det är ett team som man inte känner. Jag brukar...om någon frågar mig om att facilitera så brukar jag fråga om jag får komma dit någon vecka innan och bara hänga med teamet en arbetsdag. För om du bara slängs in i ett rum
73.	C	Vilka faktorer påverkar utförandet samt möjligheten att utföra ett retrospektiv och förbättringsarbete enligt dig? Och, vad krävs för att arbete med sprint-retrospektiv ska resultera i förbättring?  Om vi fokuserar på den sista biten. Vad krävs för att det verkligen ska bli bättre, ok, man tar upp en hel del grejer...
74.	P2	Mm, det som krävs är att man utför projektet korrekt, alltså om man tittar på min beskrivning så betyder det att man kommer att prata om ett problemområde och försöker hitta roten till det problemområdet. Säg tex att: det kan finnas saker som att vi har kommunikationsproblem. Varför har vi kommunikationsproblem? Jo för att det är 2 i teamet som aldrig dyker upp på daily stand-ups. Varför dyker 2 i teamet inte upp på stand-upsen? Då får man svar som: Jo men de är för att de måste lämna barn på dagis. Ok, varför måste dom lämna sina barn på dagis? För att de är deras föräldrar. Ok, hur kan vi lösa det här problemet. Kan någon annan lämna deras barn på dagis?

		<p>Kan företaget köpa in någon som gör det åt dom?          Nej det finns inga pengar till det.          Ok, men vad står i vår makt till att lösa problemet .          Att få upp de här sakerna till diskussion och sen när man är färdig med det ha en konkret förändring , en. För att har du tio förändringar i ett retrospektiv så kommer dom aldrig att inträffa. För att det blir för mycket, välj en konkret förändring och ta med den därifrån och sätt någon på att implementera den. Och se till att den händer, alltså, sätt ett ansvarigt namn på denna förändringen, den personen behöver inte göra det själv, men du är väktare för att se till att det här händer. Att man fixar en kille som kan köra barnen till dagis eller att vi flyttar ”stand-up” mötet en halvtimme så att dom i lugn och ro kan lämna sina barn på dagis eller vad det nu kan vara för en lösning teamet kommer fram till. Som facilitator ska du liksom bara peka folk i rätt riktning, sen får teamet själv välja den riktningen. För det måste vara ett beslut som teamet känner sig bekväma med. Det kan inte vara någonting som någon annan har tryckt på dom.</p> <p>Så för att ett retrospektiv ska ge förändring så krävs just det här att man på något sätt staplar ner dom förändringar man kommer fram till att genomföra och sedan sätter någon på detta för att genomföra dom sakerna. Om man då pratar ur ett SCRUM-perspektiv så brukar det allt som oftast falla ut på SCRUM-mastern. Det behöver inte vara det, för det kan vara en teknisk detalj som att ”Byggservern fungerar dåligt” . Det kan leda till att man i nästa iteration klipper 5 poäng. Så vi sätter en lägre målsättning med 5 poäng för att sätta upp en”task” för att fixa Byggservern. För nu har det här skapat så mycket frustration, så nu är det dags. Och då är det någon i teamet som får fixa byggservern men det är SCRUM-mastern som får ta fighten med produktägaren om att vi kommer att leverera 5 poäng mindre.</p>
75.	J	Den andra frågan i den här större frågan är ju då vad som påverkar att du ens ska kunna köra det här retrospektivet och kvalitén i det. Exempelvis Storlek på grupper/team , påverkar det?
76.	P2	Nej, Alltså du kan alltid ha ett retrospektiv. Det finns ingenting som begränsar möjligheten att utföra ett retrospektiv. Det är så fall din egen... Jag vet inte vad som skulle kunna påverka.
77.	C	Kan kunden påverka? Jag tänker pengamässigt och tid.
78.	P2	<p>Nej, det är ju lite så att, ja någon skulle kunna säga det men är du då en duktig coach så ska du se till att det händer ändå. Det är ditt ansvar att lösa det här problemet liksom. Jag ser ingenting som skulle kunna hindra mig att utföra ett retrospektiv, även om gruppen är motvillig så gäller det att för dig att köpa gruppen. Säg ”Vi testar en gång, gillar ni inte det så behöver ni aldrig mer göra det”. Eller om kunden är motvillig: Ta ut teamet och säg ”Grabbar jag bjuder er på middag någon gång i veckan ur egen ficka och kör ett retrospektiv ute istället.”.</p> <p>Det finns alltid lösningar på dom här problemen, de är bara det att folk inte är villiga att sträcka sig tillräckligt långt för att hitta dom. Så för min egen del finns det ingen ursäkt till att inte ha något retrospektiv. Och det är också</p>

		<p>det här att om man väl har haft ett bra retrospektiv där det har fallit ut något bra ur det så kan du ju dels köpa teamet så att dom kommer gilla det och även kunna bevisa för kunder som kan vara skeptiska, var skillnaden, eller hur mycket skillnad vi faktiskt uppnår.</p> <p>Så att en kund skulle vara skeptisk till det tycker jag är konstigt, för det vi egentligen pratar om är att om vi inte reflekterar och förbättrar processen så kommer vi inte kunna gå snabbare. Kan vi inte gå snabbare så kommer vi behålla den här takten hela projektet igenom. En fråga som "Vill du verkligen att vi ska hålla den här takten hela projektet rakt igenom?" och "Vill du att vi ska jobba snabbare?" Det är jättesvårt att svara nej på den frågan.</p> <p>Så det gäller ju att som coach och SCRUM-master att utbilda och visa på varför det är viktigt och om det finns skeptiker se till att omvända dom. Det låter väldigt "sektigt" men det är lite så.</p>
79.	C	<p>Det kan vara så då att kunden har det här kortsiktiga perspektivet, dom räknar bara poäng som man ska klara av för varje sprint. Om man då plockar bort 5 poäng som exempel för att hinna med det här, att dom inte ser det här långsiktiga att ni kommer bli 5 poäng snabbare till sprint 3-4.</p>
80.	P2	<p>Ja precis, det är det som är att vara SCRUM-master. Det är den här typen av problem som SCRUM-mastern är till för att lösa. Det är den här typen av saker, som när teamet säger att vi vill göra det här så är det SCRUM-masterns ansvar att se till att det händer. Därför är det oftast väldigt tråkigt när SCRUM-mastern är underordnad chefen. För att det betyder att det kan utmynna i situationer där SCRUM-mastern säger att någonting är jätteviktigt och chefen säger att antingen gör du detta eller får du sparken.</p> <p>Det finns historier om detta, jag tror det var Christer Östergårdh som sa det att en gång så var det en kund som kom in och störde teamet hela tiden och för att få stopp på det så lyfte han ut sitt skrivbord och satte det framför dörren in till utvecklingsrummet och satt sedan där som en slags väktare. Så varje gång kunden kom så vägrade han flytta sig och sa att "Nej, du måste gå igenom mig." Och är då kunden hans chef kan man kanske inte bete sig på detta viset. Men då var han en extern konsult som var där för att implementera SCRUM på det stället och då har man ändå andra möjligheter. Visserligen kan man riva ditt kontrakt, men då kan man säga "jag gjorde mitt bästa, men uppenbarligen ville dom inte tillräckligt hårt."</p>
81.	J	<p>Då går vi över till nästa fråga. Det kanske vi redan har fått svar på här men det handlar om vilka faktorer som påverkar kvalitén och arbetet med retrospektiv som verktyg för förbättring.</p>
82.	P2	<p>Ja alltså, det är allt. Viktigt är att ha en bra facilitator och viktigt är att alla förstår vad det handlar om, m.a.o viktigt är att de som inte förstår lär sig för att det handlar ofta om att de har fått dålig information eller suttit...</p> <p>Mitt största problem när jag faciliterar retrospektiv är att du hamnar i en</p>

		<p>grupp som redan har lite stigma av dåliga implementerade retrospektiv, så de har en bild av vad de kommer att utsättas för och har då en motvilja av att vilja göra det. Det finns en amerikansk psykolog Virginia Satir som har en sån här "Satir change curve" och det betyder att varje gång något ska förändras så måste det bli sämre innan det blir bättre. Så utifrån en baseline/status quo går kurvan alltså ner en bit innan du kommer upp och problemet med det är att när man är längst nere i den här gropen så är det väldigt bekvämt att säga: "Men nu är det ju sämre än vad det var tidigare" och då görs något som kallas "revert". Som betyder att du går tillbaka till ditt ursprungsläge och "status quo". Och det innebär helt enkelt att man anser att bara vi går tillbaka till det vi hade tidigare så kommer vi vara lika produktiva som vi var då. Detta gör folk, man testar SCRUM, tycker inte att det gick så bra, och sen tillbaks, Testar SCRUM på nytt, och det gick inte heller bra. Sen tar man in en coach, coachen säger "Nu ska vi testa SCRUM". Gensvaret är då "Nej nej, vi har redan gjort det tre gånger och det funkade inte".</p> <p>Detta uppstår på grund av att de som försöker implementera det har inte den här kunskapen och det är jättetråkigt. För sen är det ju då SCRUM och AGILE som får dålig press när det egentligen bara är dåliga implementationer.</p>
83.	C	Vi talade tidigare om prioriteringar och du sa ju att ni faktiskt prioriterar det som kommer upp på sprint-retrospektiv.
84.	P2	Ja ett sätt att göra det på i alla fall.
85.	C	Så nästa fråga är: Hur prioriterar ni områden som måste förbättras? Det gick du in på också.
86.	P2	Ja nu drog ju jag bara ett exempel från min "playbook". Som SCRUM-master eller retrospective-facilitator så skapar man ju en "playbook" .
87.	C	Men som du sa tidigare, ni hittar ett område som ni ska jobba på som prioriteras, jag har en underfråga där. Är det beroende på sprintens längd då, om det är 1 veckors sprint eller 3 veckors sprint?
88.	P2	Det spelar ingen roll, jag tycker ju som så att retrospektiven är korta möten där man inte kan lösa all världens problem, det går inte på ett kort möte. Så att försöka hitta... Känner ni till Lean?
89.	J,C	Ja
90.	P2	Ja, så då hittar ni den största flaskhalsen och tar bort den. För att efter att ni har ändrat systemet, det är ju då "System thinking". Systemet ser ut på ett visst sätt och löser man då ett problem, då förändras hela systemet. Så det spelar ingen roll om man skapar en lista med 1,2,3,4,5 problem, för när vi har löst problem nr 1 så kommer allt vara omkastat igen. Så man vet inte var flaskhals nr 2 kommer sitta, det kanske inte alls var en flaskhals som helt plötsligt blir ett problem på grund av lösningen från problem 1. För allt i systemet är ihopkopplat med varandra.

		Därför menar jag att hittar man den största flaskhalsen där skon skaver mest och sätter ett plåster på det. Och sedan frågar, ”Hur känns det nu? gå en runda i det här”. För det är så att ändrar man för mycket saker på en gång och allt blir jättemycket sämre, då vet du inte vilken eller vilka av dom här sakerna som gjorde det sämre. Men gör du bara en förändring så kan man alltid säga att det var just den förändringen som gjorde att det blev sämre och kan då ändra det på nytt och har då bara förlorat en iteration. Så det är väldigt viktigt, i alla fall för mig att se det som en tratt, vi börjar väldigt brett och sedan slimmar slimmar för att hitta finare och större problem och sedan löser dom problemen.
91.	C	Det här problemet, som ni då hittar under retrospektivet. Hur ser uppföljningen ut då i kommande sprint? Gör ni någon uppföljning för att säkerställa att ni faktiskt jobbar på det här problemet?
92.	P2	Jag skulle vilja säga att det beror väldigt mycket på. Det är teamets ansvar att lösa det. Så enkelt är det och vill teamet ha SCRUM-masterns hjälp så ska teamet komma till SCRUM-mastern men annars är det här helt och hållet teamets ansvar. Sen brukar jag faktiskt börja retrospektivet efter att man har kört ”Setting the stage” så brukar jag ta upp punkter från föregående retrospektiv och se över, ”hur det har gått med det här? blev det bättre? blev det sämre ” bara för att pejla läget, inte för att på något sätt kontrollera men bara för att teamet själv ska fundera.
93.	C	Ok, Slutligen så har vi ett antal påståenden här som vi har hittat utifrån litteratur rörande agil utveckling och retrospektiv. Vi vill att du ska placera in dessa på en skala på 1-4, där <ul style="list-style-type: none"> <li>- Stämmer inte</li> <li>- Stämmer ibland</li> <li>- stämmer oftast</li> <li>- stämmer alltid</li> </ul> <p>1: <i>Det hjälper teammedlemmarna att dela och förstå varandras perspektiv</i></p>
94.	P2	Ja det gör det kanske, det är många faktorer som spelar in, det är inte en flervalsfråga ☺. Men visst, det gör det.
95.	J	En 4:a då alltså?
96.	P2	Ja, men med en kommentar.
97.	C	2: <i>Det integrerar lärande på individuell nivå med lärande på teamnivå</i>
98.	P2	Vad menar ni med det?
99.	C, J	Vi menar det att kunskap som finns hos individen delas ut till teamet just under retrospektivet. För tanken med retrospektivet är väl också att ta upp positiva saker, det är ju inte bara negativa saker som ska tas upp utan det gäller att hålla fast vid det man gör bra.

100	P2	Nja, jag tycker att retrospektivet är en ganska dålig form för att sprida kunskap och lärande, jag ser det som att det är till för att förbättra processen. I alla fall så som jag kör det. Sen är det ju en viss nivå av teambuilding men jag skulle inte se det som kunskapsspridning utan det gör man framförallt när man jobbar liksom. Så om det är kunskapsspridning ni är ute efter så kanske en 2:a, men inte jättemycket.
101	C	3: <i>Det identifierar gömda problem</i>
102	P2	Mm, väldigt mycket och väldigt ofta. Så 4:a
103	C	4: <i>Det dokumenterar best practises och problem</i> , vi tänker på dokumentation...
104	P2	Nej, det kan jag ju inte påstå att det gör. Jag dokumenterar ingenting från mina retrospektiv förutom mina egna anteckningar och de är bara mina egna privata som ingen annan får ta del av, så en 1:a
105	C	5: <i>Det ökar tillfredsställelsen av arbetet genom att ge personer feedback på deras arbete</i>
106	P2	Kanske, en 3:a
107	C	6: <i>I somliga fall kan det förbättra möjligheten att korrekt estimera projektets kostnader</i>
108	P2	Nej det vet jag inte, det är klart, det är en del av ett system så det påverkar ju säkert men du sa i vissa fall?
109	C	Ja i somliga fall...
110	P2	Ja okej då, det kan vara en 2:a
111		Dessa påståenden är med här för att de har med team och lärande att göra och det är för att återkoppla lite för att få lite mer djup.
112	P2	Ne som sagt det handlar om att förbättra processen, folk lär ju sig alltid och det kan ju hända och det är svårt för mig som facilitator att svara på vad deltagare har fått ut av det här, förhoppningsvis bara bra saker. Men som sagt, det är där för att förbättra processen och får man "rob-off" effekt på andra grejer så är det bara en bonus som man med glädje tar emot.
113	J	Jag tänkte på det här, tidigare hade vi en fråga som handlade om hur man följde upp att det faktiskt går bra, att arbetet med retrospektiv faktiskt gynnar teamet osv och att man kan se att man blir effektivare och snabbare för varje sprint, att antalet poäng som tas under sprinten ökar m.m



114	P2	<p>Alltså människor är ju konstiga på det sättet, man kan ju inte mäta hur bra folk mår med siffror och mycket för mig tror jag handlar om att skapa en human arbetsmiljö där man kan jobba i en schyst takt och vara produktiv, lyckas man med det så har man lyckats med Agile. En förbättring i team behöver nödvändigtvis inte synas utan kommer i förlängningen vara bättre kod, att man rör sig snabbare, att man planerar bättre och blir effektivare. Men vad det handlar om egentligen är att skapa en bra stämning och en schyst tillvaro, det i sin tur ger i förlängningen de här effekterna. Det handlar ju om det här, det finns en författare som heter "Eric Grease" som säger att du får vad du mäter.</p> <p>Så hade jag gått in för att bara mäta poäng så hade jag kunnat få ut hur mycket poäng som helst från ett team, om jag hade varit en fullständig skitstövel. Ett exempel: "Knäcker vi inte 50 poäng så får ni ingen lön idag". Alltså, då får du precis vad du mäter, du vet inte om det här är hållbart i längden, du vet inte om folk mår bra och det är inget humant sätt att arbeta på.</p> <p>Så det jag gillar med Agile är att det handlar väldigt mycket om de mjuka faktorerna. Men sen ska man ju mäta men mätvärdena är inte till för att se till vad som behöver förbättras, utan det är bara för att ha koll på läget liksom. Så ofta när det går dåligt får man ju se under retrospektivet om det är något speciellt som har hänt. Ex: "Var det någon begravning, är det någon som har dött" Det kan ju vara sådana saker också som gör att det går lite sämre under någon iteration/sprint.</p> <p>Som exemplet med barnpassningen, det är ett riktigt exempel, det var faktiskt en flaskhals som faktiskt har kommit upp, det är även såna typer av saker som det gäller att lösa, det är inte alltid bara saker i själva arbetet.</p> <p>Så hur man följer upp det, det känns dumt att säga det. För mig är det mycket magkänsla.</p> <p>Mår teamet bra och det ser fint ut och vi levererar i tid och vad vi har "commitat". Det är det som är viktigt för mig i alla fall, att vi levererar vårt "commitment".</p> <p>Sen om kunden tycker att vårt "commitment" är för lågt gäller:</p> <p>"Ja Tyvärr, så är det. Det här är så snabbt vi kan gå utan att gå in i väggen, och tycker du att vi går för sakta så kanske du bör skaffa ett annat team".</p> <p>Tyvärr är det så, för man kan aldrig pressa fram hastighet, poäng eller timmar. Utan det kommer av sig själv, de här andra sakerna ger ju det. Och det är ju mycket såna småsaker som kan påverka som exempel: Det var någon hade gjort en mätning som sa att om det här teamet får kl 15 fika varje eftermiddag från Espresso House så kunde de höja "commitment" med 5 poäng. Så det kan ju va såna löjliga saker och där finns det ju ingen graf som säger att fika från Espresso House höjer effektiviteten för ett team, men för just det här teamet var kl 15 fikat jätteviktigt. För andra team kommer det aldrig funka, för där kan det vara helt andra krav.</p> <p>Så alla människor är individuella, alla team är individuella och det är därför</p>
-----	----	--

		<p>alla dom är psykologi bitarna är sjukt viktiga. För det är det som är den stora skillnaden mellan klassisk projektledning och coaching/scrum-master. Att det handlar mycket mer om människor än vad det handlar om att rapportera. En klassisk projektledare är anställd av ledningen för att hålla koll på projektet så att det inte går åt skogen, en klassisk Scrum-master är anställd av teamet för att skydda dom mot ledningen så att de kan få jobba ostört. Så det är väldigt stor skillnad. Det finns många Scrum-masters där ute som tror att de är projektledare men det får stå för dom, det är inte alls det man ska göra, det är inte alls den arbetsbeskrivningen.</p>
115	J	<p>Så när det gäller uppföljningen, när ni har haft ett sprint-retrospektiv där ni tagit fram den största flaskhalsen ni vill förbättra antar jag att ni har en uppföljning som ni tar upp på nästa möte eller?</p>
116	P2	<p>Alltså jag brukar mäta det så här, säg att vi har en flaskhals som tex att byggservern kraschar, om vi kör ett retrospektiv igen och vi fortfarande har ett problem med att byggservern kraschar så har vi uppenbarligen inte löst det. Så att för mig är ett retrospektiv självkontrollerande på det sättet att om problem inte är lösta så kommer dessa komma upp igen. Eftersom vi alltid pratar om vilka problem vi har. Så har problemlösningen fungerat så finns problemet inte kvar och kommer inte komma upp på retrospektivet igen och då vet vi att det är borta. Sen vad det gav, det vet vi inte, det vi vet är att problemet som störde teamet mest är fixat.</p>

**Bilaga 5, Intervjuprotokoll - P3**

<b>Intervjuperson 3 – P3</b>		
Talande (J, P3, C)		
1.	J	Lite bakgrund om dig, ålder, vad du har för utbildning, var du jobbar och hur länge du varit anställd samt vilken roll du har?
2.	P3	NNNNN. Jag är 31 år gammal, väldigt snart 32. Jag jobbar ju då på XXXX och där har jag jobbat sedan i augusti. Min roll där är konsultutvecklare i .NET precis som alla andra där, det är ju det vi är specialister på. Det var någon till fråga du bakade in där?
3.	J	Utbildningen?
4.	P3	Ja just det, utbildningen. Jag gick på Malmö högskola där jag läste programvaruteknik. Så det är ju en ”bachelor” man får ut där då. Utöver det så har jag gått lite strökurser, Microsoft cert och så. Och en SCRUM master utbildning som jag tog under min tid på XXXX där jag jobbade för XXXX. Min väg genom den här branchen har väl varit... först XXXX faktiskt, det är ju fastigheter egentligen men jag satt och gjorde lite grejer där. Sen blev det XXXX som ni kanske känner till, vi satt i där borta i XXXX gamla lokaler. Sen blev det XXXX och sen XXXX som det hette emellan där. Det var jag och två till. Sedan XXXX då. Så har ni hela vägen.
5.	C	När tog du din examen från Malmö Högskola?
6.	P3	2005
7.	J	Vår första fråga här... Vad har du för erfarenheter av traditionell respektive agile systemutveckling?
8.	P3	Traditionell, sådan här vattenfallsutveckling, är det väl kanske främst, som vi har sysslat med. Det har ju då drivits i de flesta projekten egentligen under hela tiden jag har sysslat med utveckling fram tills... Jag jobbade 8 månader som SCRUM master och det var väl egentligen första kontakten jag hade med att verkligen utöva det agila sättet. Detta var 2010. Sen har det faktiskt inte blivit så mycket mer än det. Jag har kört... Alltså XXXX (nuvarande arbetsplats) har ju en önskan att köra agilt. Tyvärr satt jag som ensam där och då blir det lite svårt att göra det. Men jag försökte lämna in saker i tre veckors iterationer i alla fall men det blev ju inte samma sak. Tyvärr har jag inte fått tillfälle att jobba i teamet utan bara som SCRUM-master för två team som jag skötte samtidigt under 8 månader.
9.	J	På XXXX då?
10.	P3	Nej, på XXXX.
11.	C	Du nämnde det lite tidigare angående certifikatet men nästa fråga lyder:

		”Kan du nämna någon utbildning/certifiering inom SCRUM eller agila metoder som du har fått eller genomfört?”. Du kan gärna berätta lite om utbildningen också, vad det innebär mm.
12.	P3	Ja, vad det har inneburit är väl... Jag antar att utan utbildningen hade jag inte fått chansen att jobba som SCRUM-master under tiden på xxxx. Jag har för mig att utbildningen hölls av softhouse och det var ett tag innan jag fick den rollen sen. Men det var också under xxxx regi då, de gav mig den utbildningen. Under den utbildningen så kommer jag ihåg att det var rätt så mycket fokus på hur man drar igång ett SCRUM projekt men kunder som inte vet vad det handlar om. Hur man då ska hjälpa till och coacha dem då, hur man sätter upp den första backloggen... Ja, hur man drar igång ett projekt. Det var ungefär den delen som kursen gick igenom.
13.	C	Var det en tvådagars utbildning?
14.	P3	Ja, exakt.
15.	C	Nästa fråga här är för att vi ska få en förståelse för hur strikt du följer SCRUM och de regler och guidelines som finns... SCRUM's ramverk består av ett antal regler och guidelines. I egenskap av SCRUM master vill vi att du lyssnar på ett antal påståenden och säger en siffra på en skala mellan 1-4 beroende på om det är något som du följer i dina SCRUM projekt eller inte. Ett betyder alltså att momentet inte finns eller används. Två betyder att det används i begränsad utsträckning. Tre betyder att det används för det mesta. Fyra är att det alltid används. Vi börjar med några artefakter som Product Backlog.
16.	P3	4
17.	C	Sprint Backlog
18.	P3	4
19.	C	Sprint Burn Down Chart
20.	P3	4
21.	C	Sen går vi in på processen... Sprint-planning meeting.
22.	P3	4
23.	C	Själv sprinten...Kanske självklar?
24.	P3	Absolut, klar 4:a
25.	C	Daily SCRUM eller stand-up
26.	P3	Jajamen, 4.
27.	C	Release Planning meeting
28.	P3	Det skulle jag nästan vilja sätta en ett a på.
29.	C	Okej. Sen har vi Sprint-review meeting. Det vill säga när ni kör demo mm för kund och andra intressenter.

30.	P3	Absolut, 4.
31.	C	Sprint-Retrospektiv
32.	P3	Absolut, 4.
33.	C	Nu går vi inte lite på rollerna inom teamet. SCRUM-Master.
34.	P3	Ja, 4.
35.	C	Produktägare?
36.	P3	Ja, 4.
37.	C	Vi tog upp ett antal processer tidigare som Sprint-planning meeting, Sprint, Daily SCRUM, release planning meeting, Sprint-review och Sprint-retrospektiv. En viktig del inom SCRUM är att dessa moment time-boxas. Görs detta?
38.	P3	Absolut, 4.
39.	J	Då går vi vidare på nästa del då. Vi går lite mer i detalj. Denna uppsatsen rör ju retrospektivet och reflektionsarbetet inom SCRUM. Hur ser processen ut i detalj i ditt arbete med retrospektiv, vad avhandlas och vilka instrument används?
40.	P3	För min del så är ett retrospektiv ett sätt att identifiera problem eller saker som har gått bra. Allting får lyftas upp. Det kan ju vara att man inte är nöjd med lokalen man sitter i eller att någon jobbar dåligt. Allt mellan himmel och jord. Det finns massor med sätt att behandla detta på. Det finns böcker som man kan ta hjälp av som jag ofta gjorde. Jag höll inte alltid dessa möten, ibland hade vi en utomstående och ibland var det någon i teamet. Vi varierade lite. Men vad hette den... ”Agile retrospectives” hette boken. Där finns de olika delarna man bör gå igenom, allt från ”setup” till... alltså informationsinsamling och hur man kommer ner till en kontenta och en lista på vad man behöver ta action på. Det som jag försökte fokusera på när jag själv höll dem var att inte identifiera alla saker man ville åtgärda utan kanske hitta en eller två jätteviktiga saker för att om man sen ska ge sig på dessa saker i kommande sprint... Är det mycket så är det svårt att ta tag i allting. Har man en sak så kan man fokusera på den och kanske istället beta av en sak åt gången, då hinner man faktiskt med ganska mycket över tiden. Istället för att det inte blir någonting eftersom man försöker göra för mycket på en gång. Detta var någonting som vi lärde oss efterhand.
41.	C	Skulle du kort kunna gå igenom ett typiskt, eller ett möjligt retrospektiv, som ni höll. Det vill säga hur processen såg ut och kanske exempel på någon metod som du använde.
42.	P3	En typisk start kunde vara att man gick inom gruppen och gav alla varin lapp där de fick skriva ner sina förväntningar på på dagens retro. Sen gick vi igenom detta tillsammans för att se vilka förväntningar som finns och kunna utgå från det. Sen gjorde vi en övning där man låter hela gruppen skriva bra eller dåliga saker på en post-it lapp och sätta upp på tavlan. Det är bara att mata på till det att man inte kommer på mer. När detta är klart så sorteras

		det in i grupper på tavlan och namnger dessa. På så sätt så hittar man områden som det varit mycket prat om. Sedan får gruppen sätta prickar till exempel, med en whiteboard penna, tre prickar var. Man ska då markera vilka av dessa grupper som man vill fördjupa sig inom. Typiskt sätt så kommer en grupp att få mer prickar. Man går då in och behandlar denna djupare och kanske gör ett "fish-bone" diagram. Man titta djupare på det och försöker finna problemet inom området och analyserar detta enligt någon metod. Antingen med denna metoden men det finns även t.ex. Five why's. Man tar ett problem och ställer fråga varför i fem olika steg. Det är ganska svår men med denna metoden kan man ibland identifiera källan till problemet. Efter detta kommer man förhoppningsvis fram till någonting som man kan ta action på. Har man då jättemycket saker som man diskuterat i gruppen så väljer man ut en av dessa "rotsakerna" att fokusera på. Om man nu väljer som vi gjorde och tar en, två eller tre saker. Det skulle kunna vara ett typiskt retrospektiv.
43.	C	Ni brukar alltså begränsa er till en, två eller tre saker?
44.	P3	Ja, det försökte vi göra mot slutet. Det kändes som vi fick mer gjort då.
45.	J	Vad är det som tas upp under mötet, kan det vara allt möjligt?
46.	P3	Mm, allt möjligt. Självt tycker jag att det är bra om man inte bara fokuserar på det negativa. Utan även försöker lyfta upp lite av det positiva under arbetets gång för att det blir gärna fokus enbart på det negativa. Man behöver se vad man gjort bra också. Det tror jag är viktigt att lyfta fram. Det blir ingenting man gräver ned sig i, men det kan ändå vara bra att få se det.
47.	J	Då går vi vidare till nästa fråga. Vad anser du om sprint retrospektiv och reflektionsarbete inom SCRUM? Vad tycker du kan bli bättre?
48.	P3	Vad jag tycker kan bli bättre än de riktlinjer som finns?
49.	J	Vi säger så här. Tycker du att retrospektivet är en viktig del för att bli bättre?
50.	P3	Ja, definitivt. Och det gäller egentligen inte bara SCRUM men det är jättebra att det lyfts fram i den metodiken. Men det tycker jag absolut är jätteviktigt. Man måste stanna upp och reflektera lite över vad man håller på med. Annars tror jag det kan skena iväg på det ena eller det andra hållet. Det är även viktigt att folk, under retrospektivet, blir pushade lite, för att säga vad de tycker. Den som leder retrot måste se till att alla får sina röster hörda. Det är viktigt att alla får ventilera och att allas röster blir hörda. Så det är definitivt något som man ska göra och se väldigt seriöst på.
51.	J	Ska vi gå vidare.
52.	C	Vi kan vänta lite. Frågan löd: vad anser du om sprint-retrospektiv och reflektionsarbete inom SCRUM. Det finns ju andra tillfälle då man kan reflektera över vad man gjort mm. T.ex. under stand-up. Om man ställer retrospektivet mot stand-up:en. De skiljer ju sig lite.
53.	P3	Ja. Under "daily" så är det... Där ska man egentligen mer berätta vad man

		gjort, vad man ska göra och om det är några problem. Så man går ju inte alls in på det djupet eller tar upp alla de ämnen man kan göra på ett retro. Dessutom är tanken med "daily" att den ska vara väldigt kort. Att man inte ska fördjupa sig i saker där... Hittar man några problem får man heller ta det efter dailyn, de som är berörda då. Så att man inte uppehåller hela teamet, de behöver ju komma igång och arbeta. Så där ser jag en skillnad.
54.	C	Okej. Men det är ju ändå en typ av reflektion. Man får igång tankeverksamheten.
55.	P3	Ja, absolut. Och den är ju jättebra att ha varje morgon för att synka gruppen.
56.	J	Retrospektivet ger lite mer utrymme helt enkelt?
57.	P3	Ja, absolut.
58.	J	Nästa fråga här. Vilka faktorer påverkar utförandet samt möjligheten att utföra ett retrospektiv enligt dig? Och en liten följetång också... Vad krävs för att ett retrospektiv ska resultera i förbättring?
59.	P3	Oj, vad som kan påverka... Det finns ju supermycket som kan påverka. Det kan ju finnas ett ointresse från dem som är med. Då blir det inte bra... om man inte kan få dem motiverade så klart. Nu snöade jag in lite på den första biten där... Just det, saker som kan... Jo, som jag var inne på. Det här med vad man måste göra för att det ska bli lyckat. Det kan ju också påverka. Leder man ett retro dåligt så är det ju inte säkert att man får ut någonting av det. Vill man ha ett riktigt bra retro så ska man gärna planera det i förväg och ha lite kunskap om gruppen sedan tidigare, hur det har fungerat och så där. Man kan styra det på rätt väg om man märker att det börjar bana iväg åt fel håll och att det inte kommer att gagna en. För övrigt så... mycket personliga faktorer och dålig planering skulle kunna påverka. Vad var det mer i följetången du hade där?
60.	J	Vad krävs för att arbetet med retrospektivet faktiskt ska resultera i förbättring?
61.	P3	Det är väl lite det som jag var inne på innan. Att man verkligen kommer ner till någonting konkret och att man ser till att ta tag i det. Man kan hantera det på olika sätt men att man identifierar vem som ska ta tag i problemet. Ofta är det SCRUM-mastern. Om det inte handlar om en relation mellan två personer i teamet eller någonting. Vilken man som SCRUM-master så klart kan hjälpa till med... NI förstår vad jag menar. Att man identifierar vem som tar tag i det hela och ser till att det blir utfört.
62.	C	Detta var ju lite interna faktorer du tog upp här, inom teamet, som kan påverka. Finns det några externa faktorer som kan påverka...
63.	P3	Just retrot?
64.	C	Ja, precis.
65.	P3	Du menar att man inte har tillgång till lokal eller någonting?
66.	C	Lokal, kunder, intressenter... jag vet inte riktigt.

67.	P3	Ja, det är klart. Att man inte har någon lokal är ju uppenbart att det skulle kunna bli problem. Men man kan i och för sig stå ute på parkeringen... Det skulle ju kunna vara så att kunden tycker att det är en dålig idé att ha retro och inte vill att man ska göra det. Då kanske man kan ifrågasätta hela SCRUM men om de nu skulle välja att vilja jobba enligt delar av SCRUM och inte köra retro så fine, då är det väl så. Då är det så klart något externt som skulle kunna påverka. Det händer säkert... Kanske oftare än man skulle vilja önska. Jag har varit ute hos kunder där det är så.
68.	J	Vad är det som har hänt då?
69.	P3	Jag har själv inte suttit i dessa uppdrag utan har bara hört det. De har väl ansett att det är tidsslöseri eller något. Men nu är detta som sagt tredjehandsinformation. Det är kanske ont om pengar eller så känner det att det inte kommer att ge någonting resultatmässigt. Det kan ju vara att man har dålig kunskap om det. Man har retro och håller dem efter bästa förmåga men det är svårt om man inte har så bra kunskap eller är så duktig på det. Så det är väl också en faktor. "Den som ska hålla i det kommer inte till jobbet för att han inte fyllt på bensin"
70.	C	Sånt kan säkert hända...
71.	P3	Är vi nöjda här?
72.	J	Ja, men vi kommer säkert in på det lite igen. Nästa fråga är lite koppla till det här också. Vilka faktorer påverkar kvaliteten av arbetet med retrospektivet.
73.	P3	Som vi var inne på innan... det här med litteratur och att få hjälp med att planera de här sakerna så man har med alla delar och sätter upp det på rätt sätt. Att man startar och avslutar det på rätt sätt tror jag är viktigt. Det är bra att ta hjälp av litteratur eller andra verktyg. Vi har ju kört mest med verktyg som post-it, A4 och pennor och jag tycker inte att man behöver mycket mer än så. Om det är planerat utefter det så att säga. Jag har inte så bra kunskap om andra verktyg inom retron själv. Jag kan inte svara dig mer utförligt än så.
74.	C	Då går vi vidare... Vi tog upp det här lite tidigare men vi tar det igen. Gör ni någon form av prioriteringar av de saker som kommer upp under ert sprint-retrospektiv?
75.	P3	Ja, definitivt.
76.	C	Hur proriteras område som behöver förbättras inför kommande sprint?
77.	P3	Gemensamt skulle jag vilja svara, i gruppen. Det har väl oftast handlat om någon form av röstning när vi haft retro. Det har varit väldigt demokratiskt.
78.	C	Vem deltar på retromötena?
79.	P3	Vi har haft SCRUM-mastern och teamet. Om jag ska vara helt ärlig så vet jag inte exakt vad best-practice där är men jag tyckte det var bra att vara med själv för att få höra vad teamet hade att säga.



80.	J	I de fallen du själv inte höll i retrospektivet?
81.	P3	Ja, NNNN, som ni träffat tidigare höll till exempel det vid något tillfälle. DÅ var jag med aktivt själv. Men ingen annan var med under tiden jag ledde, inte produktägaren eller någon annan. Det var främst för teamet.
82.	J	Du har ju sagt det här med att ni grupperar problemområden och hittar de större problemområdena. Läger man någonting på kö om det inte tas upp så att det kan tas vid kommande sprintar.
83.	P3	Jag sparade så klart sakerna som kom upp och hade det på min egen lilla to-do lista. Men är det saker som verkligen är problem så kommer dessa att komma upp på nästa retro och nästa igenom och så vidare. Kommer det upp två gånger i rad kanske det är läge att ta tag i det... Blir det inte prioriterat av teamet kanske man kan påminna om att det är ett återkommande ärende. "Nu har vi pratat om det här ett par gånger, är det säkert att den inte är så viktig". Man kan hjälpa till på traven lite om man själv ser något annat.
84.	C	Då har vi gått igenom hur retro-mötet ser ut och hur ni prioriterar. Nästa fråga handlar om uppföljning. Hur säkerställer ni att retrospektivet faktiskt leder till förbättring och hur följer nu upp detta under kommande sprints gång? Eller har ni någon uppföljning? Problemen som identifierats och valt att lyftas fram?
85.	P3	Nej, utöver min egen personliga uppföljning så hade vi ingen gemensam uppföljning. Utan ofta var ju de saker som behövdes tas itu med rätt så uppenbara saker, så när det var löst så visste alla det. Det kunde vara att det saknades en whiteboard eller någonting. Det var ju ganska tydligt när den kom upp. Var det någonting annat som inte var lika uppenbart så berättade man självklart för gruppen men vi hade inte några utsatta möten eller någonting. Det var väl på "daily" i så fall som man tog upp det.
86.	C	Jag tänkte på om ni delade ut ansvar på olika personer, vid väldigt specifika problem, som du som SCRUM-master kan lösa eller som teamet får i uppgift att lösa?
87.	P3	Jo, det kunde absolut vara så. Men det mesta hamnade på mig ändå. Eftersom jag ville att de skulle ta tag i så lite annat som möjligt. Men... nu kommer jag inte på något på rak arm, men visst fanns det något som var typiskt för teamet att ta tag i. Svaret är ja, det gjorde vi.
88.	J	Tror du att något instrument där man sammanställer hur det har gått för teamet skulle underlätta i arbetet med SCRUM retrospektiv...
89.	C	Någon form av dokumenterad uppföljning?
90.	P3	Ja...ja, varför inte? Jag vet inte riktigt hur det skulle se ut men, jo visst. Jag vet att vissa är väldigt emot det här med att dokumentera. Jag kan tänka mig att ni fått något sådant svar någon gång! Om det inte är någonting som belastar teamet, men kanske istället SCRUM-mastern som gör den här uppföljningen och kan visa hur bra man har varit. Så ja, varför inte? Det är väl jättekul att få reda på det. Det hade jag tyckt, även om det inte ligger i linje med alla andra. Jag tänkte på att utöver alla "burn-downs" mm som

		man har så har man ju sin "velocity", över hur många poäng man tar. Där får man ju en indikation på hur det går. Fast den kan man ju tolka lite som man vill. De där poängen, eller storypoängen, de varierar ju med tiden. Teamets uppfattning om vad en poäng är varierar med tiden. Så då varierar ju också resultatet i velocityn. 20 poäng för ett år sedan behöver inte vara samma sak som 20 poäng idag. Det är lite svårt av den anledningen men det kanske ändå är en indikation på något sätt. Skulle man kunna mäta det tydligare är det ju bra.
91.	C	Velocityn handlar ju om förmågan att estimeras också, vilket ofta är svårare vid tidiga sprintar. Man lär sig efter hand.
92.	P3	Jo man lär sig... men på något sätt förändras det där också, vad en poäng innebär. Det kan vara att det tillkommer någon i teamet eller att någon faller bort. Helt plötsligt så förändras det. Det kan vara marginellt men ändå. Ibland svänger det mer, speciellt om det kommer in några nya till ett befintligt team. Då ser man ofta en väldig dipp i velocityn men man märker också att det blir mycket diskussioner angående vad en poäng faktiskt är.
93.	J	Då tar vi sista frågan också... Hur väl anser du att SCRUM-retrospective främjar följande påståenden? Vi har alltså ett par påståenden är som vi vill, precis som tidigare, att du ska gradera in i en skala från 1 till 4. Ett betyder stämmer inte. Två stämmer ibland. Tre stämmer oftast och fyra stämmer alltid. Det första påståendet är: "Det hjälper teammedlemmarna att dela och förstå varandras perspektiv".
94.	P3	Jag skulle vilja svara 4... men det blir nog ändå en trea. Jag tror inte att det alltid går så väl under alla retron. Det är alltid någon som inte är med eller vill vara med eller något annat. Det är inte 100% teckning där och därför blir det ingen fyra. Ambitionen är 4.
95.	C	Du får gärna, precis som du gjorde nu, utveckla svaret om du känner att du vill det.
96.	J	Påstående två. "Det integrerar lärande på individuell nivå med lärande på teamnivå"
97.	P3	Lärande... Kan ni ta det lite utförligare?
98.	J	Det vi tänker på är att en individ i ett team sitter på kunskap och att denne under retrospektivet kan dela med sig av detta till resterande medlemmar.
99.	P3	Då skulle jag nog vilja sätta en tvåa där. Det är väl så ibland. Oftast blir det fokus på andra saker som problem och så. I retrot kan man ju mycket väl lyfta upp att det behöver gå åt det hållet istället men jag tror sällan att det gör det. Ibland.
100.	J	Påstående tre. "Det identifierar gömda problem"
101.	P3	4:a.
102.	J	"Det dokumenterar best-practices och problem".

.		
103	P3	Ja... 4.
104	J	”Det ökar tillfredsställelsen av arbetet genom att ge personer feedback på deras arbete”.
105	P3	4:a, definitivt. Speciellt om man lyfter upp lite positivt också.
106	J	Och sist men inte minst... ”I somliga fall kan det förbättra möjligheten att korrekt estimerat projektets kostnader”.
107	P3	4:a. Genom en bättre samförståelse så gör man definitivt bättre estimat.

## Bilaga 6, Intervjuprotokoll – P4

Intervjuperson 4 – P4		
Talande (J, P1, C)		
1.	C	Ja då kan vi börja med lite bakgrundsfakta om dig. Vad du har för utbildning? Hur länge du har varit på din nuvarande anställning och vad din roll är här på företaget?
2.	P4	<p>Mm, jag började 1997 att läsa till dataingenjör i Blekinge och tog min kandidatexamen där. Och Började sedan på min magister ett halvår i Ronneby och ett halvår i Australien. Fick sedan anställning i Lund och flyttade ner här. Det var i början på 2001. Bytte sedan jobb och började arbete för xxx och här har jag varit i 10 år. Jag har varit runt en del på de olika avdelningarna här, mest har det varit vårt säljsystem som jag har lagt ner mest tid på.</p> <p>När det gäller min roll här på företaget, jag var med från början när man byggde det här säljsystemet. Det är rätt roligt för att man satte ihop ett team med de bästa man kunde hitta i organisationen, då kom jag in som ny här och var ganska grön då, så jag lärde mig otroligt mycket där. Sen så har jag haft en massa olika roller, utveckling, teknisk specialist, sen har jag arbetat med krav och sånt. Men det jag gör nu är att jag är en så kallad ”Business Analyst”.</p> <p>Om det i verksamheten kommer upp att man vill göra det här, som tex. ”Vi</p>

		<p>vill göra services till våra kunder”. Så är det ganska abstrakt, då försöker man i dialogen då försöka få ner detta till krav som sedan kan implementeras i ett system. Det är ”Business Analyst” uppgift. Om man då kör SCRUM då tex och man ser på vad det mynnar ut i så är det ett gäng User-Stories som då är första steget som en ”Business Analyst” gör.</p> <p>Sen så var/är jag även single-developer, så jag sitter och utvecklar lite också, även om det inte blir så mycket. Men det jag gör är att jag kodgranskar och ser över så att all kod ser bra ut och att vi inte stöter på problem och så.</p> <p>Sen så är jag även SCRUM-master nu, så jag har en massa olika roller.</p>
3.	J	<p>Då kan vi börja med vår första fråga då som går lite in på det här med din bakgrund. Vad har du för erfarenheter av traditionell respektive Agil systemutveckling?</p>
4.	P4	<p>Om ni menar traditionell utveckling som sån här typisk vattenfalls utveckling. Så när jag kom in på xxx så körde man RUP. Vi hade ett projekt då som vi körde som var ett av de större projekten som xxx körde i Europa. Det var även en massa andra projekt då eftersom man byggde nya stora lager runt om i Europa m.m. Så då försökte man hålla ihop allt via RUP. Andra projekt körde väldigt stenhårt på RUP. Skrev use-cases, flödes-diagram, satt och analyserade och designade väldigt länge då innan man ens började koda.</p> <p>Medans vårt team då gjorde vad som behövdes för att kunna kod-generera och kunna göra rätt så mycket, så vi ritade och gjorde våra UML-diagram och satte in dom i RUP då.</p> <p>Men vi var redan då mycket mer agila, så vi gjorde fler och mindre iterationer och hade tät kontakt med vår beställare osv. Men då fanns liksom inte SCRUM på det sättet. Det fanns XP och så men det var inte så pass utbrett. Och det var tack vare det som vi levererade en massa versioner och att vi lyckades väldigt bra i jämförelse med de mer täta projekt då vid sidan om som jobbade mycket med RUP då och körde väldigt strikt.</p> <p>På xxxx har vi sedan gått från att arbeta med väldigt mycket RUP. Alla processer och så är enligt RUP, men internt jobbar vi mer med SCRUM och agile.</p> <p>De senaste åren har vi arbetat väldigt mycket med det och har fått en helt annan leveransnivå, jämfört med vad vi hade innan. Det var väldigt mycket strul då.</p>
5.	J	<p>Kan du nämna någon utbildning eller certifiering inom SCRUM/Agila metoder som du har fått eller genomfört?</p>
6.	P4	<p>Jag har inte gått någon utbildning, jag har blivit tillfrågad om jag vill bli certifierad inom SCRUM men jag har inte känt något behov av det. Det jag lärt mig är ju dess väldigt mycket praktiskt, som jag sa innan, vi jobbade väldigt mycket på det här sättet tidigare men vi hade inget namn för det. Sen är det mest böcker och bloggar på internet man läser, men det mesta kommer från praktiskt arbete.</p>

7.	C	<p>Ja, Nästa fråga då får du här för att vi ska få en förståelse för hur strikt ni följer SCRUM-processen. SCRUM's ramverk består av ett antal regler och guidelines. I egenskap av SCRUM-master vill vi att du lyssnar på ett antal påståenden och säger en siffra, på en skala mellan 1-4 beroende på om det är något som du följer eller inte.</p> <p>På en skala från 1-4</p> <ul style="list-style-type: none"> <li>- Stämmer inte</li> <li>- Stämmer ibland</li> <li>- stämmer oftast</li> <li>- stämmer alltid</li> </ul> <p>Om vi först tar lite om artefakterna så är det första påståendet då, ”vi har alltid en Product Backlog”</p>
8.	P4	Ja, en 4:a
9.	C	”Vi har alltid en Sprint-Backlog”
10.	P4	4:a
11.	C	”Vi har alltid en sprint-burndown chart”
12.	P4	4:a
13.	C	Då går vi in på processerna, ”Vi har alltid ett sprint-planning meeting”
14.	P4	4:a
15.	C	Följdfråga där, ”Vi har alltid en sprint”
16.	P4	4:a
17.	C	”Vi har alltid eller varje dag en Daily SCRUM/Standup”
18.	P4	4:a
19.	C	”Vi har alltid ett SPRINT-review meeting”. Då är det den ”reviewn” för demon vi tänker på.
20.	P4	4:a
21.	C	”Vi har alltid ett SPRINT-retrospective”
22.	P4	4:a
23.	C	Sen går vi in och kolla på själva teamet. ”Vi har alltid en SCRUM-master”
24.	P4	4:a
25.	C	”Vi har alltid en produktägare i projekten”
26.	P4	4:a
27.	C	Om vi då går tillbaka till processerna vi tog upp här tidigare, ”sprint-planning”, sprint, ”daily-SCRUM”, ”sprint-retrospective”, timeboxar ni

		dom här processerna?
28.	P4	4:a
29.	J	Ok, då går vi vidare in på retrospektivet och reflektionsarbetet inom SCRUM-projekt. Det vi vill kolla på då är: Hur ser processen ut i detalj i ditt arbete med retrospektiv, vad avhandlas och vilka instrument används?
30.	P4	<p>”Sprint-Retrospective” är en av dom bitarna som jag själv har sagt att vi är väldigt dåliga på.</p> <p>Vi har ett möte då självklart. Det kommer fram en massa bra saker och saker vi måste förbättra. Detta sammanställer vi sedan på en lista. Retrospektivet är på fredagen och måndagen därpå så har vi då en ny sprint-planering och då känns det som att vi ofta inte tar oss tillräckligt med tid att bearbeta och förbättra de ändringar vi har med. Så listan blir mer som att, den finns där men det är ingen som jobbar med den.</p> <p>Så det vi har gjort den senaste tiden är att välja ut 3 stycken av de här förbättringarna vi vill göra av kanske 10-20 st saker. Man tar 3 st som man känner ger mest utifrån att, precis som man prioriterar en backlog utifrån vilka business case dom har så prioriterar man även förbättringar, vilket IT-business case dom har, eller produkter kan man säga.</p> <p>Sen så försöker vi jobba lite mer med dom men vi har fortfarande förbättringar där att göra. Man kan säga att vi inte har så mycket resurser avsatt för att jobba med de här.</p> <p>Det vi dock har är att vi dels har nyutveckling och dels support då, men det är samma team och vi har mycket med andra produkter och sådär att göra också. Så istället för att vi ska försöka förbättra produkten så försöker vi förbättra hela teamet, så då har vi ett forum som heter något som ”improvement process”. Där kommer det in saker från allting, från supporten och från sprintarna då. Där är det mer så att man sätter en person som är ansvarig för det, deadlines och så vilken prioritering man har, så där jobbar man mycket mer med det och det är därför man inte har tillräckligt med resurser.</p>
31.	J	Men detta då, var detta något som ni gjorde under retrospektiv mötet eller var detta ett annat typ av möte?
32.	P4	Det är ett annat typ av möte, men det har en input som kommer från retrospective, från supporten och från andra som har åsikter om att ”det här måste vi förbättra”
33.	J	I frågan här frågade vi också vad det var som avhandlades, alltså vad är det som diskuteras under mötet?
34.	P4	I retrospective så är det ”hur har vår sprint gått?” ”vad kan vi förbättra i vår sprint?” och ”vad kan vi förbättra och vad gick bra.”. Man avhandlar bara sprinten i den då.
35.	J	Kan du ge exempel?

36.	P4	Det kan vara allt från kod till hur vi jobbar med businessen.
37.	C	Vi tänkte då om du kunde gå in lite mer i detalj på hur ett typiskt retrospektiv-möte kan se ut. Från det att ni sätter er ner eller hur ni nu utför det.
38.	P4	Eftersom vi är en så stor grupp, så dels har vi flera team, vi har 2 st team(utvecklingsteam). Då sätter sig varje team sig ner i varsitt rum och sen så får varje team ett gäng lappar som man under tex 15 minuter som vi timeboxar då får brainstorma under. Sen efter de här 15 minuterna eller om man är klar innan. Så kan man säga ”Ok, nu kör vi”. Sen är det fritt fram för vem som helst att börja. Det kan vara att någon säger tex.”det var bra att vi hade businessen närvarande” och sen slänger man fram sin lapp. Sen säger någon annan att det har jag också. Sen sammanställer man det i ett antal punkter då.
39.	C	Ok, vilka är det som deltar under det här mötet, du sa teamet? Är det inklusive produktägaren då eller?
40.	P4	Nej produktägaren är inte med på det då vi sitter och brainstormar då vi är flera team då. Men sen när vi har demon så är alla med. Och sen så går det över från retrospektiv och sen sammanställer vi det då för hela gruppen.
41.	J	Du sa det att ni var 2 st grupper, 2 stora team, hur många är det i varje team?
42.	P4	Vi är 8 i varje team. Sen har vi ett test team också. Vi har dedikerade testare i teamen men vi har även ett dedikerat test-team också. De jobbar inte alltid i sprinten för de har mycket annat som de måste testa av. Gamla releaser tex, eller före detta releaser som tar ett tag innan det kommer ut i produktionen.
43.	C	Ok, Vad anser du om sprintens retrospektiv och reflektionsarbete inom SCRUM och vad tycker du kan bli bättre?
44.	P4	Alltså det som är bra är att man avsätter tid och får en möjlighet att tänka och reflektera tillbaka på vad som hänt under sprinten. Det som kan bli bättre är att jag tror att många team och företag som jobbar med SCRUM tycker att retrospektiv är bra men jag är ganska säker på att dom tycker att dom saker man kommer fram till inte jobbas med. För det är väldigt bra saker som att, speciellt det här med problem som återkommer, om man märker att dom återkommer, även att det är jobbigt att fixa dom så i det långa loppet så kan man tjäna en massa tid och frustration då kanske. Vissa saker är ju också väldigt svåra att ändra. Som tex. Xxxx versionssystem. Tycker man att det är jobbigt så är kanske inte det den lättaste biten att fixa och bara byta ut. Men det är väl just det här med att man får tid till att få till ett sånt här möte men kanske inte tillräckligt med tid för att fixa de grejer som kommer upp.
45.	J	Då kommer vi in på nästa fråga. Vilka faktorer påverkar utförandet samt möjligheten att utföra ett retrospektiv och förbättringsarbete enligt dig?
46.	P4	Kan du förklara frågan lite?

47.	J	Alltså faktorer som påverkar möjligheten att ha det här mötet, det kan ju vara så att det är ont om tid ibland eller anser du att det alltid finns tid till det?
48.	P4	Det är ju alltid så att redan i början av en sprint så bokas det alltid in möten och det är redan bokad tid till retrospectiv. Det finns alltid tid till det.
49.	J	Finns det någon som anser när du nu har arbetat i de här SCRUM-projekten, har du någon gång känt av något motstånd för arbetet med retrospectivet.
50.	P4	Nej inte så, men det är väl mer att det vi har testat i de senaste 2 sprintarna är att vi alla är samlade i konferensrum och sen kör demo osv. Sen ska vi då ut och diskutera och köra sitt retrospective då och sen ska man fika och ha lite avbrott då. Så det vi har testat är då att man ska tänka igenom innan och har sedan en kortare sejour där man frågar snabbt om en bra och en dålig sak. Då tar man då sin viktigaste grej som man känner för. Men jag tycker inte att det är lika bra faktiskt. Jag tycker det är bättre att sitta i enskilda rum, sen gör ingen sin "läxa" innan. Så då får man istället påminna om att team medlemmarna ska komma ihåg att tänka på en bra och en dålig sak. Sen är det oftast i den sista veckan då när det dessutom är som mest att göra och då är det ingen som sitter och tänker och tar sig tid till att fixa en bra och en dålig grej till detta. Så det är inte många som tänker igenom såna här saker innan, vilket man istället gör om man får mer tid på sig och istället har avsatt tid för att göra.
51.	J	Vi hade också en liten följetång där till samma fråga. Vad krävs för att ett retrospectiv ska resultera i förbättring?
52.	P4	Det är det jag sa innan där, att ha retrospectiv, innebär att man avsätter sin tid, men man planerar inte in tiden därefter för att fixa saker tillräckligt. Utan man fokuserar istället på nästa sprint, vad som ska göras, vad som ska implementeras, sina stories och sådär. Jag tror generellt att man inte jobbar tillräckligt med dessa sakerna.
53.	J	Så man ska jobba mer med att avsätta mer tid för att lösa dom problemen man beslutar att ordna helt enkelt?
54.	P4	Ja, som man ser ett värde i att ordna.
55.	C	Då är vi inne på lite faktorer igen. Vilka faktorer tror du påverkar just kvalitén på själva retrospectivet. Om det finns yttre och inre faktorer som kan påverka, inom teamet eller externa faktorer som kan påverka själva utförandet?
56.	P4	Det som man får bäst kvalité av är samma som om ni tänker på hur man tidsestimerar. Att man själv får tänka igenom vad som var bra och dåligt i den här sprinten, samma då som vad kostar den här storyn att göra och vilka problem finns det. För att sen när man tänker igenom det och skriver ner på lappar och allihopa presenterar det, istället för att man går runt bordet då, där någon yttrar sig och säger tex: "jag tycker versionssystemet är dåligt". Fastnar man då är det inte bra, för då har helt plötsligt alla då



		runt bordet börjat tänka på samma problem. Så att man fastnar i ett och samma spår. Så alla får tänka igenom innan och sen så får dom presentera det. Det är det som gör det till ett bra retrospektiv.
57.	J	Du nämnde det innan här, att ni har börjat välja ut ett fåtal problemområden och denna fråga handlar då om. Gör ni någon prioritering av dom saker som kommer upp under era sprint-retrospektiv?
58.	P4	Ja, vi väljer ut ungefär 3 saker från båda teamen, så totalt 3 saker för båda teamen. Och sen kan det gå in som input till den här ”improvement process” då.
59.	J	Och följdfrågan var då hur de här sakerna prioriteras då.
60.	P4	Mm, det är gemensam prioritering.
61.	C	Är det någon form av röstning då eller?
62.	P4	Nej, det är bara en diskussion då helt enkelt om; det här måste vi fixa. Det blir aldrig några diskussioner om tex: ”Nej, det här ska vi inte fixa.” utan vi är ganska överens under de här mötena.
63.	C	Då går vi vidare till det som händer eller det som kan hända efter retrospektivet. Dvs. kommande sprint eller kommande sprintar, Hur ser uppföljningsarbetet ut av retrospektivet? Dvs. De här 1, 2 eller 3 punkterna som ni tar upp. Hur följer ni upp att det faktiskt blir bättre på de här punkterna?
64.	P4	Då det funkar bäst, är när dessa får komma in som input till ”process improvement”. Där sitter teamchefen, projektledare, management m.m och kollar på den här listan och där kan då vem som helst bli ansvarig för en uppgift. Sen får man då tid och sen får man göra uppföljning efter det. Sen när man är klar så stänger man den. Uppföljningen är var tredje vecka kanske, mötet är då. Men det är egentligen utanför själva ”sprint-produkten”, för det är andra saker som kan förbättras också, eftersom vi jobbar så ”tight”. Funkar det inte bra i supporten så påverkar det oss inom utveckling tex.
65.	C	Dom här punkterna då som kommer upp på retrospektivet, tar ni med dom och stämmer av det på kommande retrospektiv på något sätt?
66.	P4	Nej det är där vi är dåliga inom att hantera våra sprintar eller ”sprint-issues”. Men som sagt, vi kanske tar upp det sen på mötet och kommer det in då så... men vi kollar inte upp på varje retrospektiv hur saker var från förra retrospektiv. Vi fokuserar bara på föregående sprint.
67.	J	Den sista frågan då, som återigen är lite påstående, tänk på att du gärna får utveckla svaren om du vill det. Det är samma skala som tidigare, en skala från 1-4.  5. Stämmer inte 6. Stämmer ibland 7. stämmer oftast

		<p>8. stämmer alltid</p> <p>Hur väl anser du att SCRUM's retrospective främjar följande påståenden?</p> <p>Påstående 1: "Det hjälper teammedlemmarna att dela och förstå varandras perspektiv."</p>
68.	P4	<p>3:a</p> <p>Oftast så är vi ganska medvetna redan, det är inte bara så att det är på retrospektivet vi diskuterar vad som är bra eller dåligt. Utan vi jobbar ju så tätt ihopa, så har någon problem så tar man ju upp det.</p>
69.	J	Det är mer vardagligt då menar du?
70.	P4	Mm.
71.	J	Påstående 2: "Det integrerar lärande på individuell nivå med lärande på teamnivå."
72.	P4	Hur menar du då?
73.	J	Alltså att kunskap kan delas från en individ till hela gruppen
74.	P4	Nej alltså där är inte, inte just i retrospektivet är det något sånt. Utan det är mer att det speglar mer om vad som har hänt under sprinten. Inte att öka förståelsen eller att lära sig nya saker eller så.
75.	J	Utan bara för att kunna arbeta effektivare då?
76.	P4	Ja, det är det som är målet hela tiden. Hur bli vi bättre och hur gör vi en bättre produkt.
77.	J	Så det är en 1:a där då med andra ord?
78.	P4	Ja
79.	J	Påstående 3: "Det identifierar gömda problem."
80.	P4	Ja, jag skulle nog säga en 2:a där. Det är lite som jag sa innan, är det ett problem så då drar vi det under oss direkt, i olika grupperingar då. Vi har några stycken "BA" som jobbar tillsammans och blir det några problem inom våra businessar så är det ganska synligt där. Det blir aldrig en sån där aha upplevelse, att vi kommer på "oj har vi detta problemet här".
81.	J	Påstående 4: "Det dokumenterar best practises och problem."
82.	P4	Ja, vi dokumenterar ju problem. Då vi tar fram "improvement listor" och vad som gick bra. Vad menar du med best practice?
83.	J	Ja alltså att man kan få fram ett bästa sätt att jobba. Att det är så här vi vill jobba, eller vi bör jobba för att vara så effektiva som möjligt. "Varför gör vi inte det?". Att man ser vart man ska komma men det är massa problem i vägen...

84.	P4	Ok
85.	J	Stämmer detta alltså?
86.	P4	Ja, en 3:a-4:a
87.	J	Påstående 5: "Det ökar tillfredsställelsen av arbetet genom att ge personer feedback på deras arbete."
88.	P4	Det blir inte så mycket, det kommer en del personlig feedback, speciellt om vi får in en ny resurs, speciellt om vi önskat en ny resurs, det kan vara en testare eller nåt sånt som inte har vart med innan. Då kan det komma upp såna saker men man sitter inte och säger att "Ja den ändringen där gjorde du bra" på den detaljnivån eller så. Men är det exceptionellt kan det komma upp såna saker. Den mesta feedbacken vi ger är på varje utvecklare som löser problem, som sedan går till en annan utvecklare och som sedan gör en "kod-review" på det. Där får man då mer av det här, detta var bra och detta var inte bra kodningsmässigt. Är det något som inte funkar, mer personligt eller liknande. Nu funkar det väldigt bra personligt så det har väl inte uppkommit. Men är det något vi vill förbättra med våra relationer och hur vi jobbar så är det mer dom här grupperingarna som ex. "BA-gruppen". Hur jobbar vi bättre med våra krav, det är mer processen i själva sprinten. Även utveckling, som "hur förbättrar vi vår utveckling?". Där har vi utvecklingsmöte varje tisdag, där vi har några punkter vi tar upp där. Men det är inte sprinten utan det är ofta de här processerna som ger den här feedbacken då.
89.	J	Så vilken siffra skulle passa in där?
90.	P4	2:a
91.	J	Påstående 6: "I somliga fall kan det förbättra möjligheten att korrekt estimeras projektets kostnader."
92.	P4	Det enda, som när man beräknar kostnader för ett projekt, så bryter vi ner det i "stories" och kollar vad varje "story" kommer att kosta och vilken "effort" som krävs. Om det då har en massa dolda saker som kostar då, om inte det kommer upp, oavsett om det kommer upp i retrospektiv eller i andra forum är egentligen strunt samma. Men kommer det inte upp så kommer ju inte den kostnaden att räknas med. Så det är ju ett av verktygen att få fram detta. Ett av huvudpunkterna med SCRUM är att, om vi tar en traditionell projektledare och en SCRUM-projektledare. Det finns ju projektledare inom SCRUM också. Men som då driver den typen som de här 2 projekten. "Vattenfalls" eller vanliga projektledaren tycker allt flyter på och funkar bra för att man har specificerat allt i början. Medans om den projektledare går över till en SCRUM-metod så tycker dom att det bara är en massa problem hela tiden. Och syftet med många av dom verktygen i SCRUM är ju då att faktiskt lyfta problem, daily SCRUM, retrospective tex. Där kommer alla problem upp, skillnaden är att samma problem fanns i den andra metoden men de lyfts aldrig upp på det sättet. Så ju mer problem och risker du kan få upp desto bättre kan du ju också estimeras hur mycket ditt projekt kommer att kosta. För att redan från dag

		till dag kan du se hur vår plan ser ut. Det det här säger är att den kostnaden och den tidsplanen vi sätter för det här projektet är vad det är idag. Det vi vet är att den planen troligtvis inte kommer se likadan ut sen. Men vi har metoder för att kunna förändra den här planen eftersom vi inte har gjort den här estimeringen och analysen och design innan så är retrospektivet en del då för att kunna få koll på kostnaden. Det är ingen jättestor bidragare men det hjälper ju. Så en 3:a
93.	J	Det var egentligen alla våra frågor. Något som vi däremot kanske har missat här: Hur långa är era sprintar som ni kör?
94.	P4	Vi kör 3-veckors sprintar. Vårt system är väldigt komplext, xxxx landskap är väldigt invecklat. Vi har system som är upp till 22 år gamla och hela infrastrukturen runt omkring då och hur man skickar data mellan system då. Ovanpå det ligger det ju då en ny infrastruktur sen 10 år tillbaka. Ovanpå det sen ligger det ytterligare olika system. Om man då tar vårt säljsystem som exempel. Ska man då ha något hemkört så måste man ha data ifrån produktsystem, dels från designgruppen och information därifrån, kundsystem och leveranssystem. Så vårt system här hos oss då ligger högst upp. Så när något inte funkar där så funkar det inte hos oss. Så det är väldigt beroende, hela kedjan. Så det är en liten utmaning. Så att kunna få ut något i produktion, dels så ska det igenom alla testmiljöer, så det är inte så att man utvecklar någonting och som sen produceras på en webbsida eller så utan det ska testas och det ska utbildas. Man ska flyga över folk från olika delar av världen och sen kanske någon ska övervaka över installationer. Så det är en väldigt lång kedja därefter och som sagt var, det är mycket integrationer. Så ibland måste de ligga före oss och planera. Så 3 veckor är egentligen den kortaste perioden vi kan ha, vi har haft längre innan men vi har försökt korta ner och 3 veckor känns ganska lagom. Sen är det ju lite "Overhead" också, som nästa vecka är det kod-stopp på torsdagen och torsdagen är egentligen då vi ska köra demo och vi ska förbereda och få ut allt i olika miljöer och så. Så vi tappar alltid i alla fall 1,5 dag. Och 1,5 dag på 30 personer är en rätt stor kostnad då. Det funkar för små team de kan ha liksom nån vecka eller så...
95.	C	Så teamets definition av när något är färdigt, då är det alltså ute i produktion?
96.	P4	Nej, för då hade vi aldrig kunnat stänga våra sprintar. Innan hade vi det så att när man väl har utvecklat det och har gjort den här "code-reviewen" så sätter man att det är "implemented" som status. Men det tyckte vi då var för kort och för tidig status, vi ville då utöka det. Det var då vi tog in testare i teamet också. Dom hade vart där innan men grejen är att när det är så mycket som ska fixas i andra system och integration och så där så tar det ju väldigt mycket tid att utveckla det. Så sista dagen i princip, det var då de kunde testa grejerna. Vilket gav då att tiden innan då kunde de inte göra så mycket, men problemet är att i den här cykeln då så har det inte funnits någon röd tråd utan testarna har kommit in på slutet och skrivit sina test-case osv. Det har gjort att vi har dragit in testarna mycket tidigare i

		processen med kravarbete och de här bitarna. Så de här sakerna är numera med i sprintarna, så det funkar mycket bättre nu.
97.	J C	Tack så mycket för dina svar, bra svar på alla frågorna. Tack för din tid!

## Bilaga 7, Intervjuprotokoll – P5

Intervjuperson 5 – P5		
Talande (J, P5, C)		
1.	C	Till att börja med vill vi veta lite bakgrund om dig. Om du har någon utbildning, var du är anställd, hur länge du varit anställd och vilken roll du har.
2.	P5	Jag är anställd på XXXX i Helsingborg och jag är anställd som chef på projektkontoret här. Min utbildningsbakgrund är att jag är mikrobiolog och matte/NO 4-9 lärare. Den är lite annorlunda än de flesta andra här. Jag är inte någon civilingenjör eller liknande, jag kommer från en helt annan bakgrund. Däremot har jag jobbat inom teknik sedan 1997. Mycket inom telekom, xxxx, xxxxx, xxxx, xxxx, xxxx och allt vad det heter! Så det var lite bakgrund.
3.	C	Vad har du för erfarenheter av traditionell respektive agil systemutveckling?
4.	P5	Traditionell systemutveckling, det är egentligen där jag har min största erfarenhet. För det är inom det som jag jobbat som projektledare sedan 1997 egentligen. Så det var väl egentligen för när jag började på xxxx... nu ska vi se...när började jag där... 2007! Det var då jag började jobba med agila metoder, och just SCRUM. Men det var inte SCRUM hela tiden utan där körde vi även vattenfallsmodellen. Så min erfarenhet inom vattenfallsmetodiken är mycket, mycket större än inom SCRUM metodiken. Jag har jobbat med båda men det är övervägande vattenfall.
5.	C	Några andra agila metoder du har erfarenhet av eller det är SCRUM?
6.	P5	Det är SCRUM jag har jobbat med.
7.	J	Har du varit SCRUM-master hela tiden?
8.	P5	Jag har jobbat både som Product owner och som SCRUM-master beroende på vilket projekt det var.
9.	C	Du sa att du hade lång erfarenhet av traditionell utveckling... Det var från 1997 och framåt då?

10.	P5	Ja precis, från 1997 till november förra året. Då jobbade jag som projektledare och det var mycket traditionell projektledning. Men det inslag av agila metoder och SCRUM
11.	J	Kan du nämna någon utbildning eller certifiering du fått inom SCRUM eller andra agila metoder?
12.	P5	Det är den här SCRUM-master utbildningen som jag har, en tvådagarskurs, det är den jag har.
13.	C	Du får gärna berätta lite om utbildningen och vad det givit dig.
14.	P5	Ja... Det var en rätt så... Jag vet inte om jag ska kalla det utbildning. Det var mer att man gick igenom...det kan man i och för sig också kalla utbildning... vi gick igenom alla steg man skulle tänka på och inte tänka på. PÅ den utbildningen fick vi göra grejerna också. Man fick sitta och estimerar, man fick göra den här sprint backloggen och den vanliga backloggen. Man fick göra ett SCRUM projekt så att säga. Så det var bra på det sättet. Men jag tycker att den utbildningen hade kunnat utökas en hel del för att det känns lite som att man gick det, man fick informationen, och dag tre då skulle man jobba som SCRUM-master. Den kunde varit mer gedigen om man säger så. Så att säga att man är certifierad SCRUM-master...ja, jo, shore! Det är man, men det är inte lika med att man kan det. Då den var lite mager den utbildningen. Men jag tror det är den som finns, den här tvådagars SCRUM-master utbildningen. Jag har inte sett några andra.
15.	C	Det stämmer nog.
16.	P5	Sen finns det motsvarande för product owner vet jag. För den har jag kollegor som har gått. Sen tror jag det finns någon för deltagare i ett SCRUM projekt, men jag är inte 100 där.
17.	C	Du sa tidigare, utanför micken, att de flesta som var på utbildningen hade stor erfarenhet av traditionella metoder.
18.	P5	Ja.
19.	C	Var upplägget på så sätt att man fasades över från traditionell utveckling till SCRUM?
20.	P5	Nej. Det var det inte. Det var som att "Den bakgrunden har jag och nu ska jag lära mig detta". Diskussionen (ang om SCRUM alltid är lämplig) fördes utanför utbildningen. Vi som hade jobbat som traditionella projektledare länge var ganska ifrågasättande. "Fungerar detta verkligen?", för globala stora projekt med 1000 deltagare. Det var utanför kursen som vi började diskutera detta. Det var inte på kursen alls. Det var mer att de glorifierade SCRUM, "Det fungerar överallt". Det var då vi började fråga oss om det verkligen är så.
21.	C	Tredje frågan är för att vi ska få en uppfattning om hur strikt du som SCRUM-master följer SCRUM's regler och guidelines. SCRUM's ramverk består av ett antal regler och guidelines. I egenskap av SCRUM-master vill vi att du lyssnar på ett antal påstående och säger en siffra på en skala mellan

		1-4 beroende på om det är något som ni följt eller inte. Ett betyder att "stämmer inte", 2 är att det "stämmer ibland", 3 att det "stämmer för det mesta" och fyra "stämmer alltid". Vi börjar med några artefakter inom SCRUM. Vi har alltid en product backlog.
22.	P5	3 kan man säga där.
23.	C	Du får gärna utveckla dina svar.
24.	P5	Det var lite så där i början. Nu tänker jag tillbaks på projekten jag drivit innan. Där var det lite så här att man fundera på "Vad ska vi med den till" och "vad har vi det till". Product owner funderade lite..."Hmmm ojdå...nu måste jag skriva ner det också". Så ja, det stämmer, men inte från början. Utan det var någonting som växte fram kan man säga. Eller tvingades fram kan man nästan säga. Man behöver ju verkligen en product backlog. Jo, vi kom ju så långt att vi hade en men det var lite teksamt i uppstarten.
25.	C	Vi har alltid en sprint backlog.
26.	P5	Det är också en 3:a där, det trillade ju inte när vi fick en product backlog. Så ja, det stämmer att vi hade.
27.	C	Vi har alltid jobbat med en sprint burndown chart.
28.	P5	Japp, det är en 4:a.
29.	C	Vi går in lite på processen... Vi har alltid ett sprint-planning meeting.
30.	P5	Ja, en 4:a alltså.
31.	C	Vi har alltid en sprint.
32.	P5	Ja, en 4:a.
33.	C	Vi har alltid en daily SCRUM eller stand-up.
34.	P5	Ja, en 4:a
35.	C	Vi har alltid ett sprint review meeting.
36.	P5	Ja, en 4:a
37.	C	Vi håller alltid ett sprint-retrospektiv
38.	P5	4:a
39.	C	Vi går in på teamet... Vi har alltid haft en SCRUM-master.
40.	P5	Ja... 3 kan vi säga där för den är lite mixad mellan Product Owner och SCRUM-master under en viss tid.
41.	C	Så man hade "två hattar"?
42.	P5	Man hade två hattar... Mm en 3:a.
43.	C	Vi har alltid en produktägare

44.	P5	Det är 3 då. Men när vi väl kom på att det inte var samma person så...
45.	C	Slutligen... Time-boxing är en viktig faktor inom SCRUM. Avseende processerna som vi tog upp tidigare. Sprint planning meeting, sprint, daily SCRUM, sprint review meeting och sprint retrospektiv. Har dessa processerna alltid timeboxats?
46.	P5	Hm... 2 eller 3 kan man nog säga. Det har väl varit lite sådär.
47.	J	Då går vi vidare med hur retrospektivet ser ut i detalj. Frågan lyder: "Hur ser processen ut i detalj i ditt arbete med retrospektiv, vad avhandlas och vilka instrument används?"
48.	P5	Vi hade det ganska manuellt skulle jag vilja säga. Det var mycket diskussion egentligen. "Vad gjorde vi bra?" och "Vad gjorde vi dåligt?". Sen skrev vi ner det. Vi hade inte några verktyg direkt. Utan det var väldigt... vi skrev ner det på en whiteboard. Vad vi hade gjort bra och vad vi hade gjort dåligt och vad vi skulle ta med oss till nästa sprint. Så det var inte något mer organiserat eller strukturerat sätt att göra det på.
49.	C	Du får gärna berätta i detalj hur ett typiskt retrospektiv kule se ut efter en sprint.
50.	P5	Vi bokade naturligtvis in ett möte. Vi hade ett möte. Vi försökte faktiskt hålla det hyfsat kort. Vissa hade ju en hel dag, men det fick inte vi tillåtelse att ha. Det tycktes vara för mycket tid. Vi försökte hålla det ganska kort. Frågeställningen var egentligen: "Vad har vi gjort bra?", "Vad har vi gjort dåligt?". Sen fick var och en runt bordet berätta vad man tyckte. Någon gång körde vi med gula lappar för att få det lite mer anonymt i vissa läge. Ibland är det ju lite känsligt med vissa grejor som händer och inte händer.
51.	J	Vad var det för saker som diskuterades?
52.	P5	Det kunde var allt från ner i minska detalj teknik till att SCRUM-mastern var sjuk.
53.	C	Vem deltog på dessa möten?
54.	P5	Det gjorde SCRUM-mastern naturligtvis, den personen ledde det. Alla deltagare i teamet och produktägaren. Det var ett krav att produkt ägaren skulle vara med.
55.	C	Ett krav från?
56.	P5	Från oss som körde det. Vi tyckte att det var bra att den personen var med. Det är nog inte alla som har det.
57.	J	Vad var det som fick er att vilja detta?
58.	P5	Vi ville få ihopa det så att den personen också kände sig delaktig. Det var ju inte alltid de skötte sig heller. Vi hade ett retrospektiv på allt när vi körde. Vi tyckte att det var schystast istället för att gå till den personen sen och säga: "Hallå, kan du?", utan att den personen fick höra det från alla. Därför ville vi ha den personen med. Jag jobbade själv som produkt owner. Jag har



		jobbat som SCRUM-master men mer som product owner och jag ville gärna vara med. Annars får man ingen uppfattning om hur teamet fungerar, eller ”varför blev det så här”, eller ”Varför trillade det inte ut det jag hade förväntat mig från den här sprinten”. Det var riktigt bra att vara med.
59.	C	Då har jag en liten följetång där... Vem fick lov att tala på de dessa möten?
60.	P5	Alla. Absolut.
61.	J	Vi kan gå in på nästa fråga då. Vad anser du om sprint retrospektiv och reflektionsarbete inom SCRUM.
62.	P5	Det tycker jag är jättebra. Det är faktiskt en av de bättre grejerna inom agil metoder jämfört med vattenfall. Där lyfts inte allt fram på samma sätt. Beroende på vilken projektledare det är dock. Man kan ju eller man bör ju lägga in det även i vattenfallsmetodiken, men här är det lite mer uppenbart att man gör det. Man jobbar de här 4 veckorna, som vi gjorde, sen hade man ett retrospektiv. Så det är faktiskt något som är bra med SCRUM. Att man hela tiden jobbar med att förbättra sig,
63.	C	Retrospektiv kan man ju ha i alla former av projekt. Det vi tänker på är ju främst iterations-retrospektivet.
64.	P5	Absolut. I ett vattenfallsprojekt så kan det ju tendera att bli som så att man kör igång projektet och tänker: ”Oh nej, nu skulle vi ju slutrapportera det här. Nu kan vi fråga alla som varit med vad de tyckte!”. Det kan tendera till att bli så. Här är det lite mer självklart att man gör det. Det ingår i processen på ett mer naturligt sätt. Så det tycker jag är bra, absolut.
65.	J	Vilka faktorer påverkar utförandet samt möjligheten att utföra ett retrospektiv, och förbättringsarbete, enligt dig?
66.	P5	Hur tänker du då?
67.	J	Det kan vara allt möjligt. Du nämnde tidigare att ni skulle vilja ha retrospektivet en hel dag men ni hade bara bara en halvdag.
68.	P5	Egentligen finns det inga begränsningar alls mer än att... de runt omkring tyckte att det tog lite för mycket tid med just retrospektivet. Vi försökte begränsa oss så gott vi kunde, annars fast det inga... det ingick och skulle göras. Tiden var faktiskt en liten faktor... ”Ni ska inte hålla på i 8 timmar.” begränsa det till 4 eller gärna mindre.
69.	J	Som en liten följdfråga. Vad krävs för att arbetet med retrospektivet ska resultera i förbättring?
70.	P5	Det är ju öppenhet av alla naturligtvis. Utan ärlighet så kommer man inte vidare långt. Att man inom teamet som man jobbar i kan vara ärlig och säga vad man tycker och tänker. Annars är det meningslöst egentligen med dessa retrospektiv.
71.	C	Hade du några speciella knep för att uppnå just denna öppenheten? Jag kan tänka mig att alla i större team inte är lika glada för att tala.

72.	P5	Ne det är dem verkligen inte och det är något man får jobba med. Både som SCRUM master och som traditionell projektledare så är detta något som är jätteviktigt att jobba med. Att man hela tiden ser till att alla får möjlighet att säga något helt enkelt. Även om det inte är något jätteviktigt så ska alla få säga sitt. Det är något man får lirka fram ibland. Det är även lite hur man är som person, hur man tar folk och så där. Men jag har ju gjort... inte i SCRUM utan när jag jobbat som traditionell projektledare... så har jag gjort det skriftligt vid vissa tillfällen. Bara för att alla ska få skriva sitt och anonymt dessutom. Men det har varit i undantagsfall kan jag säga, då har det varit något som inte alls fungerat i teamet. Men inte när jag kört SCRUM, då har vi alltid haft muntliga retrospektiv.
73.	C	Den förra frågan tog upp faktorer som påverkar utförandet av retrospektivet. Nu undrar vi om det finns några faktorer som kan påverka kvaliteten på retrospektivet?
74.	P5	Ja du...
75.	C	Du var inne på öppenhet tidigare.
76.	P5	Alla dessa mjuka faktorer är viktigt att ha med. Säger inte alla vad alla tycker så blir det ju inte bra. Då blir man ju inte bättre. Så de mjuka delarna är ju viktiga, att man kan vara ärlig och öppen. Men om man titta lite mer på annat... Ja, att man ska få tillåtelse att ha dem helt enkelt. Ibland kan det vara så att vissa företag säger: "Nej, nu jobbar ni på, inget sånt här tjafs där ni sitter och diskuterar". Man måste verkligen få möjlighet att ha dem. Något annat kan jag inte komma på.
77.	J	Det som diskuteras under dessa möte, problem som dyker upp med mera. Gör ni någon form av prioritering av de saker som dyker upp under ert sprint retrospektiv?
78.	P5	Det har nog vi aldrig gjort faktiskt när jag tänker efter.
79.	J	Så ni listar upp problemen och sen...
80.	P5	... Sen har vi dem. VI har aldrig gjort någon riktig prioritering. Jag skulle tro, eller jag misstänker, att det berodde på att vi inte hade tid att göra det. Men det har vi faktiskt inte gjort.
81.	C	Följdfrågan där var egentligen hur det prioriteras. De sakerna som kommer upp under retro-mötet, tar ni med allt till kommande sprint och fokuserar på detta?
82.	P5	Nej, det vi gjorde var att... Om det var sådana riktiga bra grejer som kom upp så var det någon som blev ansvarig att se till antingen att något inträffar eller att det inte inträffar. I många fall var det kanske SCRUM-mastern som skulle se till att det t.ex. fanns resurser, att det fanns en dator eller något annat. Sådana grejer. Men sedan kunde även andra i teamet få en uppgift. Speciellt när man sitter och knappar kod eller något sånt: "Se till att någon granskar din kod innan du checkar in den", till exempel. Kom sånt upp så var det ju någon annan som fick ansvar.

83.	J	Nästa blir väl lite av en följdfråga också. Gör ni någon uppföljning av retrospektivet som visar att det faktiskt leder till förbättring?
84.	P5	Man gör ju det när man kommer till nästa retrospektiv. Då kolla man om man har gjort någonting åt det som kom upp sist.
85.	C	Så det är den uppföljningen ni har. Man tar med sig det som kom upp på föregående retro?
86.	P5	Ja precis. Så gjorde vi i alla fall. Det var ett enkelt sätt att göra det på.
87.	C	De grejorna som, om det nu är några, som inte har förbättrats. De ligger kvar till kommande retro i så fall.
88.	P5	Ja, då får vi ta upp det igen. Fördelen med detta är ju att det faktiskt kommer upp! I många lägen kommer det aldrig upp att det är problem. Bara det är en fördel med att jobba på det sättet. Att ha dessa återkommande genomgångarna.
89.	C	Ja problem kan ju faktiskt lösas genom att det bara kommer upp, man behöver inte alltid göra någonting åt det.
90.	P5	Ja precis! När det lyfts så tydliggörs det och så ser och tänker alla på det. Sen kan det vara löst. Så det finns många poänger med att ha dessa återkommande genomgångarna.
91.	C	Okej, Vi har ett antal påstående till vid nästa fråga. Hur väl anser du att SCRUM retrospektiv främjar följande påstående? Det är samma skala som tidigare. En etta betyder ”stämmer inte”, en tvåa ”stämmer ibland”, trea ”stämmer oftast” och fyra ”stämmer alltid”. Påstående ett är: ”Det hjälper teammedlemmarna att dela och förstå varandras perspektiv”.
92.	P5	4:a.
93.	C	Precis som tidigare får du gärna utveckla svaren om du vill.
94.	P5	Den är rätt så klar.
95.	C	”Det integrerar lärande på individuell nivå med lärande på teamnivå.”
96.	P5	Ja, 4:a också måste jag säga. Det får man med sig på något vis, genom diskussion. Att man tydliggör det och att det kommer upp. Jo 4:a måste jag säga där.
97.	C	”Det identifierar gömda problem”
98.	P5	4:a... eller en 3:a vill jag säga. Det är inte alltid allt kommer upp.
99.	C	”Det dokumenterar best-practises och problem”.
100.	P5	2. Viss för av dokumentation med det är inte så jättemycket, nej.
101.	C	”Det ökar tillfredsställelsen av arbetet genom att ge personerna feedback på deras arbete”
102.	P5	Ja, fyra. Det är bra på det sättet!

103.	C	”I somliga fall kan det förbättra möjligheterna att korrekt estimerat projektets kostnader”
104.	P5	3 kan det nog bidra till. Det finns andra aspekter som man jobbar med där.
105.	C	En 3:a.
106.	P5	Ja.
107.	C	Vill du utveckla det? Du säger en 3:a, att det stämmer oftast. På vilket sätt kan just retrot främja detta?
108.	P5	Det kan komma upp sådana här grejer på retrospektivet också som ”Här har vi estimerat fel”. Sådana diskussioner är rätt så ofta förekommande. ”Vi skulle faktiskt gjort detta på åtta timmar, nu tog det 16 timmar. Varför gjorde det det?”. Sådana diskussioner kan ju hjälpa till. Så det stämmer ju oftast. Om sånt kommer upp så stämmer det. Men annars så är det ju sånt man jobbar med hela tiden när man sitter och tidsestimerar sina leverabler så att säga.
109.	C	Hur många medlemmar har ni haft i teamen under din tid som SCRUM-master och product owner om det varit olika projekt?
110.	P5	Jag har haft team som varit från tre stycken + en SCRUM master, det var nog det minsta. Då var jag SCRUM master i det fallet. Sen hade vi ett av de större, där var vi minst 15 i alla fall. Men då inkluderades även SCRUM mastern. Ja, ungefär 15 stycken plus en product owner, och då var jag product owner i det fallet. Så det är väl där spannet har legat.
111.	C	Hur påverkade detta retrot? Just antalet teammedlemmar.
112.	P5	Ingenting faktiskt. Fördelen med detta var ju att dessa personerna kände varandra sedan innan, det var på samma avdelning. De hade jobbat ihop tidigare och det hade ju säkert en påverkan på det hela. Det var inga externa, vilket naturligtvis var en fördel. Nej... det var lika ”vild” diskussion i alla fallen, sen har jag kört något retro som var lite mitt emellan också.

## 8 Referenser

Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J. (2002). *Agile software development methods: Review and analysis*, VTT Publications 478, VTT, Finland

Alvestad, K. (2006). *Agile Software Development: A Qualitative Explorative Survey of a Small Organizations Effort to Professionalize*, Department of Computer and Information Science, Norwegian University of Science and Technology

Argyris, C. (1992). *On organizational learning*, 2<sup>nd</sup> ed, Blackwell Publishing, Cornwall, UK

Argyris, C., Schön, D. (1978). *Organizational learning: a theory of action perspective*, Addison-Wesley, Reading

Awad, M. A. (2005). *A Comparison between Agile and Traditional Software Development Methodologies*. University of Western Australia, School of Computer Science and software Engineering.

Boehm, B. (2002). *Get Ready for Agile Methods, with Care*, Software development, IEEE, Jan 2002, s.64-69

Derby, E., Larsen, D. (2006). *Agile Retrospectives: Making good teams great*, The pragmatic Bookshelf, Raleigh

Dingsøy, T. (2004). *Postmortem reviews: purpose and approaches in software engineering*, Information and Communication Technology, Trondheim, Norway

Forsberg, E., Lindström, M. (2008). *Scrum med XP-relaterade tekniker: Införandet av Scrum och dess påverkan på systemutvecklingsgruppen*, Institutionen för Informatik, Lunds Universitet

Healey, M., Jenkins, A. (2000) *Kolb's Experiential Learning Theory and Its Application in Geography in Higher Education*, Journal of Geography, 99: 5, s. 185 — 195

Jacobsen, D. I. (2002): *Vad, hur och varför. Om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen*. Studentlitteratur, Lund

Kniberg, H. (2007). *Scrum and XP from the Trenches: how we do Scrum*, Lulu.com, USA

Kniberg, H. Skarin, M. (2010). *Kanban and Scrum - Making the Most of Both*, Lulu.com, USA

Kolb, D. (1984). *Experimental Learning: Experience as The Source of Learning and Development*, Prentice Hall Inc., Upper Saddle River

Larman, C. (2004). *Applying UML And Patterns: An introduction to object-oriented analysis and design and iterative development*, (third edition), Pearson Education Inc., Boston, US.

Macheridis, N. (2005). *Projektaspekter: kunskapsområden för ledning och styrning av projekt*, Studentlitteratur, Lund

- Moe, N. B., Dingsøy, T., Dybå, T. (2010). A teamwork model for understanding an agile team: A case study of a Scrum project, Elsevier, Information and Software Technology 52, s 480–491
- Myllyaho, M., Salo, O., Kääriäinen, J., Hyysalo, J., Koskela, J. (2004). *A Review of Small and Large Post-Mortem Analysis Methods*, Department of Information Processing Science, University of Oulu, Finland
- Nonaka, I., Takeuchi, H. (1995). *The Knowledge-Creating Company*, Oxford University Press, New York
- Preece, J., Rogers, Y., Sharp, H. (2007): *Interaction Design – Beyond Human-Computer Interaction*, 2nd ed. John Wiley & Sons
- Salo, O., Kolehmainen, K., Kyllönen, P., Löthman, J., Salmijärvi, S., Abrahamsson, P. (2004). Self-adaptability of agile software processes: a case study on post-iteration workshops, 5th International Conference on Extreme Programming and Agile Processes in Software Engineering
- Schwaber, K., Beedle, M. (2002). *Agile Software Development with SCRUM*, Pearson Education, Upper Saddle River
- Schwaber, K., Sutherland, J. (2007). *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process, Draft 4/22/2007*, [Online]  
Tillgänglig via: <http://www.crisp.se/scrum/books/ScrumPapers20070424.pdf> [lästes 20 April 2011]
- Schwaber, K., Sutherland, J. (2010). *Scrum Guide* [Online]  
Tillgänglig via: <http://www.scrum.org/scrumguideenglish/> [lästes 20 April 2011]
- Schön, D. (1983). *The Reflective Practitioner: How professionals think in action*, Basic Books Inc., US.
- Shore, J., Warden, S. (2007). *The art of Agile Development*, O’reilly, USA
- Turban, E., Sharda, R., Delen, D. (2010) *Decision Support and Business intelligence systems*, Ninth Edition, Pearson, Upper Saddle River
- Visser, M. (2003). Gregory Bateson on deutero-learning and double bind: A brief conceptual history, *Journal of History of the Behavioral Sciences*, Vol. 39(3), s. 269–278