



LUNDS UNIVERSITET
Ekonomihögskolan

SCRUM i Stora Projekt

Hur gör man?

Kandidatuppsats, 15 HP, SYSK01 i informatik

Framlagd: Juni, 2010

Författare: Djevdet Iséni

Kawan Jiawook

Senan Rasoul

Handledare: Agneta Olerup

Examinatorer: Claus Persson

Magnus Wärja

Titel	SCRUM i Stora Projekt – Hur gör man?
Författare	Djevdet Iséni Kawan Jiawook Senan Rasoul
Utgivare	Institutionen för Informatik
Handledare	Agneta Olerup
Examinatorer	Claus Persson Magnus Wärja
Publiceringsår	2010
Typ av Uppsats	Kandidatuppsats
Språk	Svenska
Nyckelord	Scrum, agile, kommunikation, projektledare, projektadministration, systemutveckling, stora projekt, arbetsprocess

Abstrakt

De agila systemutvecklingsmetoderna uppstod eftersom det fanns krav på en mer lätttrörlig systemutvecklingsmetod. En av de mest tillämpade agila metoder är Scrum som är anpassad för små projekt. Populariteten av Scrum har ökat konstant tack vare flera framgångar och detta har medfört att organisationer på senare år blivit intresserade av att tillämpa Scrum i stora projekt. Detta har dock visat sig vara svårt eftersom det inte finns ett allmänt ramverk som organisationer kan använda sig av. Denna uppsats har fokus på tre områden: arbetsprocess, kommunikation och projektadministration, och ämnar klargöra de faktorer som påverkas olika beroende på projektstorlek när en organisation väljer att införa Scrum. Intervjuer har genomförts med projektledare och litteratur har granskats. I slutsatsen presenterar vi ett ramverk i som består av rekommendationer på tillvägagångssätt. Vi kom fram till att det finns möjligheter för att använda Scrum i stora projekt. Detta kräver dock en del förändringar på process och struktur i de områden vi har undersökt.

Tack

- Till våra fruar som har haft stort tålamod och varit ett stöd under skrivprocessen.
- Till vår handledare för handledning och litteraturförslag.
- Till projektledarna som ställde upp på intervjuer.
- Till Alf Wickman och Saranda Iséni som hjälpte till med korrekturläsning och rättning av uppsatsen.

Innehåll

Definition	1
1. INTRODUKTION.....	2
1.1. Bakgrund.....	2
1.2. Problem.....	3
1.3. Forskningsfråga & Syfte.....	4
1.4. Avgränsningar	4
2. TEORI	5
2.1. Systemutveckling	5
2.1.1. Systemutvecklingsmetoder	5
2.1.2. Agil systemutveckling	8
2.1.3. Agil kontra Sekventiell.....	9
2.1.4. Scrum.....	11
2.2. Kommunikation.....	15
2.2.1. Synkron och asynkron kommunikation.....	15
2.3. Projektadministration	16
2.3.1. Projektstyrning	17
2.3.2. Befogenheter.....	18
2.3.3. Projektmöten.....	19
2.4. Scrum i Stora Systemutvecklingsprojekt.....	20
2.4.1. Systemutveckling i stora projekt.....	20
2.4.2. Kommunikation i stora projekt	20
2.4.3. Projektadministration i Stora Projekt	21
2.5. Ramverk.....	22
3. METOD.....	24
3.1. Undersökningsstrategi	24
3.2. Intervjuguide	24
3.3. Datainsamling	24
3.4. Kategorisering av data.....	25
3.5. Val av informanter.....	25
3.6. Validitet & reliabilitet.....	25

3.7. Etik	26
4. EMPIRI & ANALYS	27
4.1. Arbetsprocess	27
4.1.1. Roller	27
4.1.2. Artefakter	29
4.1.3. Ceremonier	30
4.2. Kommunikation	31
4.2.1. Kommunikation grupp	31
4.2.2. Kommunikation kund (Product Owner)	32
4.2.3. Kommunikation med företagsledning	32
4.2.4. Kommunikation mellan grupper	32
4.3. Projektadministration	33
4.3.1. Möten	33
4.3.2. Styrning	34
4.3.3. Befogenhet	35
5. SLUTSATSER	37
5.1. Undersökningsresultat	37
5.1.1. Rekommendationer inom temat Arbetsprocess	38
5.1.2. Rekommendationer inom temat Kommunikation	39
5.1.3. Rekommendationer inom temat Projektadministration	40
5.2. Kritisk diskussion	41
5.3. Slutord	41
Bilaga 1 – Agile Manifesto	42
Bilaga 2 – Intervjufrågorna	43
Bilaga 3 – Intervju A – IPA	45
Bilaga 4 – Intervju B – IPB	53
Bilaga 5 – Intervju C – IPC	61
Bilaga 6 – Tabeller och figurer	70
REFERENSER	71

Definition

Scrum	En agil arbetsmetod inom systemutvecklingsprojekt
Scrum Master	En person som fungerar som en coach för Scrum Teamet och säkerställer att metodens ramverk följs av teamet.
Scrum Team	En tvärfunktionell grupp av utvecklare som utvecklar delar av produkten
Product Owner	Beställare som styr projektet ur ett affärsperspektiv och administrerar Product Backlog.
Ceremonier	Avstämningsmöten
Artefakter	Dokumentations- och planeringsobjekt (Sprint Backlog, Product Backlog och Burndown Chart)
Daily Scrum	Ett dagligt avstämningsmöte mellan Scrum Master och Scrumteam
Sprint Review Meeting	Ett presentationsmöte av fungerande programvara som hålls av Scrum Teamet för kund, användare och representanter från företagsledningen.
Sprint Planning Meeting	Ett planeringsmöte för nästkommande sprint i projektet mellan Scrum Master, Scrum Team och Product Owner.
Product Backlog	En lista med krav på funktioner för produkten
Sprint Backlog	En lista med prioriterade uppgifter som ska genomföras under sprinten
Burndown Chart	Dagliga progressen för en sprint i jämförelse med hela sprinten

1. INTRODUKTION

1.1. Bakgrund

Många metoder som används vid systemutveckling utgår ifrån livscykelmodeller. Dessa metoder har använts sedan 70-talet i flera systemutvecklingsprojekt. Livscykeln finns i flera varianter men har en del grundläggande principer som genomförbarhetsstudie, systemundersökning, systemanalys, systemdesign, implementering samt drift och underhåll (Avison & Fitzgerald, 2006). Livscykeln kallas även för konventionell systemutveckling, traditionell systemutveckling eller vattenfallsmodellen, det senare är mest använd. Livscykelmodellen utgår från att det finns ett antal sammanhängande faser och modellen förutsätter att varje fas i systemutvecklingsprocessen ska vara genomförd och klar innan nästa fas i livscykeln påbörjas (Avison & Fitzgerald, 2006).

Livscykelmodellen har fått stark kritik av systemutvecklare som anser att metoden är för rigid (Cronholm, 2008). Avison & Fitzgerald (2006) menar att de krav som finns från beställaren ska specificeras i början av utvecklingsprocessen, innan analys och design påbörjats. Kraven fastställs i ett tidigt skede, därefter bearbetas de olika faserna steg för steg och arbetar utifrån kravspecifikationerna. För att gå vidare till nästa fas i processen måste fasen innan vara helt avslutad. Detta får till följd att nya eventuella krav som uppstår blir svåra att behandla. Avison & Fitzgerald (2006)

Royce (1970) hävdar att det är mycket viktigt med omfattande dokumentation vid utveckling enligt de sekventiella modellerna. Många parter är inblandade och för att designers, utvecklare, företagsledning och kunder ska kunna få en klar bild över produkten och projektet behövs det flera tusentals sidor med dokumentation (Royce, 1970).

För att i huvudsak komma ifrån omfattande dokumentation har det utvecklats agila arbetsmetoder. 2001 bildades Agile Alliance, sjutton utvecklare kom överens om vilka värderingar och mål det agila manifestet ska innehålla. De kom fram till tolv principer och det som utmärker dessa principer är att de värdesätter individen, kunden, den fungerande produkten för att nämna några (Agile manifestet, 2001). Dessa metoder bygger på livscykelmodellen där de positiva egenskaperna i modellen har bevarats och de negativa har ersatts med andra delar som anses passa bättre (Cronholm, 2008). Agila metoder har blivit populära och används av flera utvecklare i projekt. Men bara för att dessa metoder blivit populära och anses ha enbart goda egenskaper betyder det inte att agila metoder ska undgå granskning. Cronholm (2008) anser att de agila metoderna bör granskas på samma sätt som andra metoder har blivit granskade.

Agila metoder fokuserar på hur ett projektteam kan organiseras i en ständigt föränderlig omgivning och bygger på projektmedlemmarnas kompetens, att de har ett gemensamt mål, att de har ett ömsesidigt förtroende och respekt för varandra (Cockburn & Highsmith, 2001). Alla i teamet ska kunna byta uppgifter med varandra och projektledaren i teamet bör vara

erfaren för att öka chanserna till att lyckas med projektet. Projektledarens uppgift är att hantera teamet utan att blanda sig i för mycket i deras arbete, att sköta kontakten med företagsledningen och skapa en god relation med kunden. (Cockburn & Highsmith, 2001)

1.2. Problem

De agila projekten har visat sig vara framgångsrika. I en undersökning svarar informanterna att av de projekt som utförts enligt de agila metoderna och som bedrivits samlokaliserat lyckades 82,7 procent (Agile Adoption Survey, 2008). Samma undersökning visar också på betydande förbättringar på bland annat produktivitet, drift samt kundnöjdhet efter att agila utvecklingsmetoder införts.

Av de agila metoderna är Scrum den största och är vanligaste förekommande i agila systemutvecklingsprojekt (State of Agile Survey, 2009). Scrum är en arbetsmetod som är anpassad för små projekt där projektgruppen som kallas *Scrum Team* vanligtvis består av fem till nio personer (Schwaber, 2004).

Scrum's popularitet har på senare år fått fler organisationer intresserade av att införa metoden i stora projekt. I ett antal artiklar från Computer Sweden (2010) beskriver företag att de vill införa agila metoder i stora systemutvecklingsprojekt. Men då dessa metoder oftast är anpassade för små projekt blir det mer komplicerat när organisationer försöker anamma agila metoder till stora projekt.

Vår definition av stora projekt är när ett projekt består av två eller flera Scrum Teams som arbetar i samma projekt. Detta för att ett stort projekt går utanför Scrum och ökar komplexiteten. Varje Scrum Team i sig är redan komplex som det är och kräver sin egna unika lösning (Schwaber, 2004). Turk et al., (2002) anser att kommunikationen ansikte mot ansikte i teamet kan skapa problem i stora projekt eftersom det finns för många inblandade vilket är tidskrävande och kundsamarbetet, som är en viktig del i Scrum, blir lidande eftersom kunden kanske inte har tid att vara med på alla möten. Coram & Bohner (2005) hävdar att projektledaren ska inta en mentors roll och coacha projektmedlemmarna till att vilja ta initiativ, den här typen av ledarskap bygger på samarbete snarare än kontroll.

Hanakawa & Okura (2004) hävdar att traditionella dokumentbaserade projektledartekniker inte fungerar på agila utvecklingsmetoder, vilket ställer höga krav på att projektledaren förstår produktkvalité och utvecklingsframstegen. Således, ligger den stora utmaningen för projektledaren i att kunna hantera produktkvalitén och utvecklingsframstegen i projekten med lite dokumentation.

1.3. Forskningsfråga & Syfte

Det huvudsakliga syftet med uppsatsen är att identifiera vilka konsekvenser det blir vid införandet av agila arbetsmetoden Scrum i stora projekt. Sedermera föreslå ett ramverk som kan användas av organisationer när de vill tillämpa Scrum i stora projekt och av forskare som vill studera fenomenet Scrum i stora projekt. Följande blir vår frågeställning:

Införandet av Scrum i stora projekt – Hur gör man?

1.4. Avgränsningar

I den här uppsatsen kommer vi att undersöka hur arbetsmetoden Scrum kan användas i stora systemutvecklingsprojekt. Projekt som blir för komplexa för att hanteras av enbart ett Scrum Team och behöver delas upp i delprojekt har vi valt att definiera som stora projekt. Vi ämnar undersöka kommunikation, befogenhet, möte, styrning och arbetsprocess. Det finns två aspekter i projektledning som vi inte kommer att beröra, dessa är de psykologiska och ekonomiska aspekterna. Detta på grund av att uppsatsen skulle bli för omfattande.

2. TEORI

2.1. Systemutveckling

Vi vill i detta avsnitt först ge en bakgrundsbild till systemutveckling för att sedan presentera olika systemutvecklingsteorier som ligger till grund för agil systemutveckling. Därefter gör vi en jämförelse mellan de agila och de sekventiella metoderna. Vi avslutar med att presentera arbetsmetoden Scrum.

2.1.1. Systemutvecklingsmetoder

Vi inleder med att beskriva sekventiell systemutveckling som är en av de tidigare metoderna att arbeta efter. Därefter redogör vi för evolutionär systemutveckling, prototyping samt iterativ systemutveckling. Detta för att agila metoder grundar sig på dessa metoder.

Sekventiell systemutveckling

Livscykelmodellen bygger på ett antal sammanhängande faser och modellen är utformad så att varje fas i systemutvecklingsprocessen ska vara genomförd och klar innan nästa fas i livscykeln påbörjas. Första fasen i livscykeln är en genomförbarhetsstudie. I denna fas utvärderas eventuella äldre system som finns i en organisation, vilka krav som ställs på systemet samt vilka alternativ som finns. Det är även viktigt att beakta vilka ekonomiska, organisatoriska, tekniska och mänskliga fördelar ett nytt system kan innebära. (Avison & Fitzgerald, 2006)

Nästa fas är en systemundersökning. Användare intervjuas och observeras, enkäter skickas ut och dokumentation granskas för att undersöka vilka hinder och önskemål som finns. När all data är insamlad är nästa fas systemanalys. Med hjälp av den tidigare undersökningen är det lättare att förstå de olika aspekterna och utforska hur en organisation ska förbättras med hjälp av ett nytt system. Därefter påbörjas systemdesignen. I denna fas utformas först olika processer som ska behandlas i systemet för att sedan gå vidare till design av databas, programmering, testning samt backup på systemet. Manualer tas även fram och därefter börjar planeringen för implementering. (Avison & Fitzgerald, 2006)

Inför implementeringen är det viktigt att göra en kvalitetskontroll för att säkerställa att rätt hårdvara används att mjukvaran är testad samt att användarna är utbildade. Säkerheten ska vara testad och vara fungerande. När alla dessa punkter är avklarade kan systemet föras in i organisationen. När systemet är på plats och används kommer sista fasen, drift och underhåll. Eventuellt kan nya uppdateringar behövas, personalen få mer utbildning, anpassningar behöver utföras. Men det är även viktigt att kontrollera att alla krav är uppfyllda samt att användarna är nöjda med interaktionen. (Avison & Fitzgerald, 2006)

Livscykelmodellen har använts en längre tid och är väl beprövad och testad, vilket enligt Avison & Fitzgerald (2006) är en fördel. Modellen inbegriper omfattande dokumentation som inte utesluter något. Kravspecifikationen är väldokumenterad och den når ut till alla från

utvecklingsteamet till användarna och företagsledningen. Att systemets processer definieras i ett tidigt skede under utvecklingen kan det innebära en trygghet för utvecklarna att i förväg veta vad som ska göras. En annan fördel är att modellen säkerställer att berörda personer är utbildade på det nya systemet. (Avison & Fitzgerald, 2006)

Dock har modellen fått en hel del kritik. Problemet enligt Avison & Fitzgerald (2006) är att de krav som finns från beställaren specificeras i början av utvecklingsprocessen, innan analys och design påbörjas. Kraven fryses i ett tidigt skede. Därefter fortsätter det vidare till de olika faserna steg för steg och arbetet sker utifrån kravspecifikationerna. Kravet innan nästa fas påbörjas i processen är att fasen innan måste vara helt avslutad. Detta får till följd att nya eventuella krav som uppstår blir svåra att behandla. Vid användning av livscykelmodellen låses oftast systemet till den process som organisationen använde vid inköp av systemet vilket kan leda till negativa konsekvenser då processer ändras konstant. Likadant är det med företagets data. Det är inte säkert att de data som organisationen använde sig av vid början av systemutvecklingen är den samma som används senare. Detta menar Avison & Fitzgerald (2006) kan leda till stort missnöje bland beställarna och användarna eftersom de inte får det system de behöver.

Evolutionär systemutveckling

På grund av de många problemen med livscykelmodellerna utvecklades de evolutionära systemutvecklingsmodellerna (Gulliksen & Göransson, 2002). Evolutionär systemutveckling bygger på inkrementell och iterativ utveckling vilket innebär att arbetet sker stegvis för att leverera ett system som successivt byggs på med fungerande systemtillägg över en period (Avison & Fitzgerald, 2006; Gulliksen & Göransson, 2002).

Till skillnad från traditionella utvecklingsmodeller anses inte den första implementeringen som en färdig slutgiltig produkt utan snarare som början på ett evolutionärt system som gradvis förbättras och optimeras tills organisationens problem mer eller mindre är lösta och alla kraven är uppfyllda (Avison & Fitzgerald, 2006). Den lärdom som dras från tidigare inkrement används för att definiera nästa inkrement. Dessa utvecklas sekventiellt men inom varje inkrementell livscykel sker utvecklingen iterativt genom analys, design, test samt implementering och detta upprepas fram tills leverans och underhåll (Gulliksen & Göransson, 2002).

Vid användning av den evolutionära utvecklingsmodellen ligger inte fokus på att få ett idealiskt, komplett, fullt fungerande system baserade på fördefinierade kravspecifikationer. Istället implementeras ett system i ett tidigt skede, ett system som har en del brister, inte uppfyller alla krav och som design och interaktionsmässigt inte alls tilltalar användarna, men som fungerar tillräckligt bra för att kunna användas. Anledningen till att arbetet utförs på detta sätt är för att det ska bli enklare att ta del av fel, brister och komplikationer som användarna upplever med systemet. Därefter sker utveckling och förbättring av systemet baserade på de synpunkter som dyker upp från första inkrementet.(Avison & Fitzgerald, 2006). Detta är enligt Avison & Fitzgerald (2006) en av styrkorna med de evolutionära modellerna.

En fördel är att till skillnad från många andra utvecklingsmetoder är förändrade krav något som förväntas och som ses som en del av den stegvisa inkrementella utvecklingen. När en ändring på kraven uppstår byggs det helt enkelt in i nästa utvecklingsiteration. Evolutionär utveckling lämpar sig bäst i de projekt där kraven på systemet är svåra att definiera i förväg, kanske till och med omöjliga, eller i de fall där systemet är oerhört komplext. (Avison & Fitzgerald, 2006)

Prototyping

Att använda sig av prototyping vid systemutveckling har blivit allt vanligare och Avison & Fitzgerald (2006) hävdar att metoden är här för att stanna. Ett problem med traditionella systemutvecklingsmetoder är att användarna inte kommer i kontakt med informationssystemet förrän det är färdigt och när det är försent att göra några större förändringar i systemet. Den första och enda versionen av systemet är det som implementeras (Avison & Fitzgerald, 2006). Earl (1978) menar att ett informationssystem är något som organisationer investerat mycket tid och pengar på och detta leder till att de känner sig pressade att få användarna att acceptera systemet. I många fall uppstår missnöje bland slutanvändarna när slutprodukten är implementerad och de förändringar som behöver genomföras är dyrt att utföra i efterhand. Krav fanns på att system skulle utvecklas med betoning på design, att användarnas åsikter skulle uppmärksammas vid design av systemet samt att system först skulle utvecklas som en testversion eller prototyp, ett system som redan var pålitlig, testad innan den implementeras. (Earl, 1978)

Det finns en rad olika synsätt på prototyping (Gulliksen & Göransson, 2002; Avison & Fitzgerald, 2006) men grundidén är att utvecklare snabbt designar en förenklad fungerande version av systemet som de experimenterar med och sen demonstrerar för användarna för att därefter få återkoppling på systemets användbarhet och effektivitet (Earl, 1978). Denna cykel återupprepas tills både utvecklarna och kund är nöjda med prototypen. Vad som sker härnäst skiljer sig åt enligt Avison & Fitzgerald (2006). Antingen utvecklas ett nytt system som baseras på prototypen för att därefter slänga prototypen eller byggs det vidare på prototypen tills den utvecklats till det system som ska implementeras och användas (Avison & Fitzgerald, 2006). Earl (1978) menar att prototypen ska efter att har tjänat sitt experimentella ändamål ersättas med ett riktigt system. Gulliksen & Göransson (2002) menar att prototyping ger en rad möjligheter som till exempel att utforska nya lösningar, prova funktionalitet, hitta krav, träna kreativitet och hitta svagheter.

Trots de många fördelarna med prototyping har det riktats en hel del kritik mot metoden. Då prototypen, och sedermera systemet, uppstår och utvecklas efter hand som återkoppling erhålls menar kritiker att systemet ofta är otillräcklig och inadekvat. Vissa system är alldeles för komplexa för att designas på detta sätt. Det behövs mer formella och välplanerade strategier för att kunna utveckla dessa. En annan kritik är att testningen inte är fullständig då det oftast behövs detaljerade specifikationer att testa mot, vilket inte alltid finns vid prototyping. Detta kan bland annat leda till försämrad säkerhet på systemet. (Avison & Fitzgerald, 2006)

Iterativ systemutveckling

Iteration innebär upprepning (Gulliksen & Göransson, 2002). Vid iterativ systemutveckling är problemen och kraven inte lika väl-specifierade som vid sekventiell systemutveckling. Därför är det svårt att göra rätt från början. Syftet är att förfinas och förbättra efter hand som arbetet med systemet fortskrider. Iterationer ger en bättre bild av problemet och således är det lättare att finna lösningar på problemet. Vid iteration behandlas analys, design, utvärdering samt återkoppling och detta repeteras tills de mål och krav som ställts upp är uppnådda (Gulliksen & Göransson, 2002). Enligt Gould et al., (1997) finns det tre krav för iterativt arbete:

- Identifiering av *nödvändiga* förändringar.
- En *möjlighet* att göra förändringar
- En *vilja* att göra förändringar

Iterativt arbete innebär att den framtida lösningens design är oklar, det som gjorts och tagits fram kan behöva förändras flertalet gånger och de som arbetar med utvecklingen måste vara beredda på detta (Gulliksen & Göransson, 2002). För att ett arbete ska kunna rubriceras som iterativt fastslår Gulliksen & Göransson (2002) att varje iteration måste innehålla en grundlig analys av kundens krav och användningssammanhang, en fas till prototyputförning samt en utvärdering av prototypen som leder till ett beslut om förändringar som kan påverka hur prototypen utformas i fortsättningen.

2.1.2. Agil systemutveckling

Att utveckla system efter omfattande kravspecifikationer är inte enkelt enligt förespråkare för de agila metoder, därför fanns det ett behov av att utforma en ny metod (Avison & Fitzgerald, 2006). Agil systemutveckling bygger på iterativ och evolutionär systemutveckling samt prototyping. Gemensamt för alla dessa metoder är att arbetet sker snabbt och evolutionärt i iterationer för att tidigt få fram en prototyp eller ett koncept. Likt de andra metoderna har de samma filosofi om att förändringar är något som ska välkomnas, inte föraktas. De agila metoderna är designade för att agera snabbt på förändrade situationer för att styra i rätt riktning utan att förlora balansen och fokuset, vilket står i stark kontrast med de mer traditionella systemutvecklingsmetoderna (Kosh, 2005).

I agila systemutvecklingsprojekt ses förändringar som något positivt och som en del av projektet då dessa kan komma att förbättra projektet genom att belysa delar som inte tagits upp eller genom att tillföra ny information (Kosh, 2005). Och likadant som i de tidigare metoderna arbetas det i de agila metoderna med att leverera delar av resultatet istället för ett komplett system i slutet av utvecklingsprocessen. Men det finns något som skiljer metoden åt från de andra enligt Highsmith & Cockburn (2001):

”what is new about agile methods is not the practices they use, but their recognition of people as the primary drivers of project success, coupled with an intense focus on effectiveness and manoeuvrability”

Begreppet ”Agile Method” myntades i Utah, USA 2001 (Kosh, 2005). Personer som representerade en rad olika agila metoder träffades under ett möte för att diskutera alternativ

till de traditionella och dokumentstyrda utvecklingsmetoderna. Det som framkom var The Agile Development Manifesto, en deklARATION som undertecknades av alla deltagare (Agile manifesto, 2001). Följande ståndpunkter kom de fram till under mötet:

- *Individer och interaktioner* framför processer och verktyg
- *Fungerande programvara* framför omfattande dokumentation
- *Kundsamarbete* framför kontraktsförhandling
- *Anpassning till förändring* framför att följa en plan

Så länge det finns värde i punkterna till höger, värdesätts punkterna till vänster mer (Agile manifesto, 2001).

De agila metoderna som finns representerade i manifestet har olika namn och arbetssätt men de bygger alla på samma grunder. Generella egenskaper hos agila metoder är att de är lättviktiga och använder sig av korta iterationer där kunden aktivt deltar för att upprätta, prioritera och verifiera krav. I agila projekt förlitas det på "tyst kunskap" som finns inom projektgruppen istället för dokumentation (Boehm & Turner, 2003). Ett verkligt agilt projekt måste inkludera följande utmärkande egenskaper: iterativ, inkrementell, självorganiserande samt framåtväxande. Kosh (2005) anser att de gemensamma aspekterna för agila metoder är rörlighet, förändring, planering, kommunikation, och lärande. Gustavsson (2007) listar tillfällen då agila metoder passar bra respektive mindre bra:

Passar bra när:

- man behöver få ett användbart resultat snabbt
- projektet inte har en tydlig kravbild eller där kraven inte finns specificerade i början
- projektet genomförs i en dynamisk situation där förutsättningarna för projektet kan ändras radikalt
- projektet är så pass komplext att projektbeställare och projektgruppen har svårt att se en tydlig bild av hur slutresultatet kommer att se ut

Passar mindre bra när:

- ett fast kontrakt med alla specifikationer har upprättats
- kostnaderna för att förändra i redan levererade delresultat är mycket stora
- kundens styrning minskar möjligheten att göra förändringar och arbeta flexibelt

2.1.3. Agil kontra Sekventiell

Boehm & Turner (2003) menar att komplexiteten med mjukvaruutveckling och mängden av olika utvecklingsmetoder leder till att en jämförelse mellan agila och sekventiella tillvägagångssätt är svår att göra. Men samtidigt finns det en mängd olika egenskaper med de olika metoderna som skiljer de åt betydligt (Boehm & Turner, 2003). Följande presenteras en tabell (tabell 2.1) av Boehm & Turner (2003) där de förklarar vilka egenskaper de agila och de sekventiella metoderna har. Vi har valt denna tabell eftersom den tydliggör teorin om de olika metoderna som beskrivs ovan och klargör dess egenskaper.

Tabell 2.1 Agile and Plan-Driven Method Home Grounds (Boehm & Turner, 2003)

Agile and Plan-Driven Method Home Grounds		
Characteristics	Agile	Plan-driven
<i>Application</i>		
Primary Goals	Rapid value; responding to change	Predictability, stability, high
Size	Smaller teams and projects	Larger teams and projects
Environment	Turbulent; high change; project focused	Stable; low change; project/organization focused
<i>Management</i>		
Customer Relations	Dedicated on-site customers; focused on prioritized increments	As-needed customer interactions; focused on contract provisions
Planning and Control	Internalized plans; qualitative control	Documented plans, quantitative control
Communications	Tacit interpersonal knowledge	Explicit documented knowledge
<i>Technical</i>		
Requirements	Prioritized informal stories and test cases; undergoing unforeseeable change	Formalized project, capability, interface, quality, foreseeable evolution requirements
Development	Simple design; short increments; refactoring assumed inexpensive	Extensive design; longer increments; Refactoring assumed expensive
Test	Executable test cases define requirements, testing	Documented test plans and procedures
<i>Personnel</i>		
Customers	Dedicated, collocated CRACK* performers	CRACK* performers, not always collocated
Developers	Able to revise a method (break its rules) to fit an unprecedented new situation. Able to tailor a method to fit a precededented new situation	Able to revise a method (break its rules) to fit an unprecedented new situation. With training, able to perform procedural method steps like. coding a simple method, simple refactoring, following coding standards.
Culture	Comfort and empowerment via many degrees of freedom (thriving on chaos)	Comfort and empowerment via framework of policies and procedures (thriving on order)
*Collaborative, Representative, Authorized, Committed, Knowledgeable		

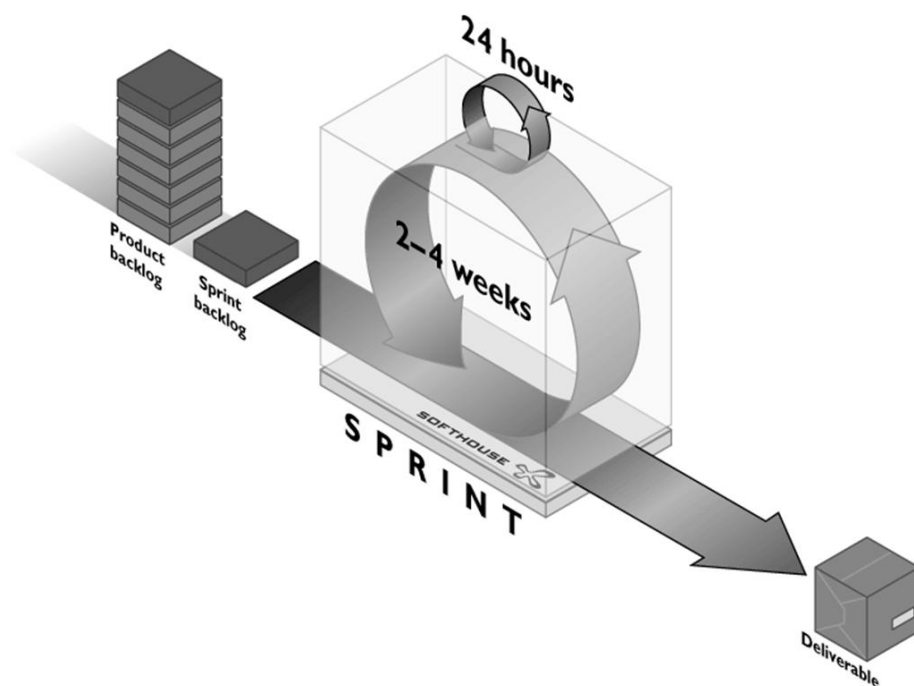
Boehm & Turner (2003) presenterar även fem kritiska faktorer (tabell 2.2) som är viktiga för att avgöra vilken av de två metoderna som är lämpligast att använda i ett specifikt projekt. Om ett projekt har en blandning av kritiska faktorer från både de agila och de sekventiella metoderna bör en sammansättning av bägge metoderna användas för att nå bästa resultat (Boehm & Turner, 2003).

Tabell 2.2 The Five Critical Agility/Plan-Driven Factors (Boehm & Turner, 2003)

The five critical factors		
Factor	Agility Considerations	Plan-Driven Considerations
Size	Well-matched to small products and teams. Reliance on tacit knowledge limits scalability.	Methods evolved to handle large products and teams. Hard to tailor down to small projects.
Criticality	Untested on safety-critical products. Potential difficulties with simple design and lack of documentation.	Methods evolved to handle highly critical products. Hard to tailor down to low-criticality products.
Dynamism	Simple design and continuous refactoring are excellent for highly dynamic environments, but a source of potentially expensive rework for highly stable environments.	Detailed plans and Big Design Up Front excellent for highly stable environment, but a source of expensive rework for highly dynamic environments.
Personnel	Requires continuous presence of a critical mass of experts that are able to revise a method to fit an unprecedented new situation and able to tailor a method to fit a precededent new situation	Needs a critical mass of experts that are able to revise a method to fit an unprecedented new situation and able to tailor a method to fit a precededent new situation during project definition, but can work with fewer later in the project—unless the environment is highly dynamic.
Culture	Thrives in a culture where people feel comfortable and empowered by having many degrees of freedom.	Thrives in a culture where people feel comfortable and empowered by having their roles defined by clear policies and procedures.

2.1.4. Scrum

Scrum är en iterativ och inkrementell arbetsmetod som är anpassad för att kunna hantera komplexa projekt där det är omöjligt att förutspå allt som kommer inträffa. Scrum som arbetsmetod är enkel. Processen, metoder, artefakter, reglerna är få och lätta att lära sig. Scrum synliggör problemen i ett tidigt skede av ett projekt. Detta gör att projektmedlemmar vet vad som händer och kan följaktligen rätta till problemen och därigenom styra projektet mot önskade mål. (Schwaber 2004) Således är Scrum inte enbart en mjukvaruutvecklingsmetod (Kosh, 2005). Fördelarna med Scrum enligt Schwaber (2004) är att metoden effektiviserar systemutvecklingen och dess grupp samt utökar affärsnyttan. Scrum består av tre roller, tre artefakter och ceremonier.



Figur 2.1 Scrum uppbyggnad (Softhouse AB)

Roller

Inom ett Scrum Team finns det inga fasta roller som till exempel, testare, utvecklare eller analytiker. Meningen är att alla i teamet ska kunna byta uppgift med varandra och att tillräckligt med kompetens finns i teamet för att utföra arbetet. Det är viktigt att ingen medlem i teamet byts ut under projektets gång, de bör få tid att arbeta ihop teamet. Det är teamet tillsammans med Scrum Master som bär hela projektet. Projektets tre roller behandlar definition och sammanställning av produktens funktioner, kundmöte, coaching samt utveckling. Inga andra aktörer får blanda sig i projektets arbete. När teamet erhållit Sprint Backloggen från Product Owner ska de lämnas ostörda fram tills nästa sprintmöte. (Schwaber, 2004)

Product Owner är den som representerar intressenterna i projektet. Det kan vara antingen en beställare från kundsidan eller en representant från det egna företaget. Product Owner bör vara en person som har erfarenhet inom marknaden liksom produkten, detta för att kunna prioritera aktiviteter och förändringar efter marknadsvärdet. Således har denne person som uppgift att säkerställa att produkten hela tiden utvecklas utifrån en aspekt på affärsnyttan. Product Ownern sammanställer alla behov och krav som utgör grunden för förändringar av produkten. Detta kräver att Product Ownern är ständigt närvarande under projektets gång för att snabbt kunna agera vid uppstående situationer. När målsättningen med produkten är klar delas projektet upp i delprojekt och varje del ska dels skapa affärsnytta men också gå att delleverera. Product Owner gör samtidigt en prioriteringslista över vilka förändringar som ska göras och levereras. När detta är klart och en ny sprint kan sättas igång kallas Scrum Teamet till möte. (Schwaber, 2004)

Scrum Master fungerar som en coach för teamet. Den viktigaste uppgiften är att säkerställa att Scrum följs. Scrum Mastern träffar sitt team varje dag i korta möten (*Daily Scrums*). Här följer Scrum Mastern upp arbetet och lyssnar på sitt team, vägleder dem och kommer med tips och råd om hur arbetet ska fortsätta. Scrum Master ansvarar för att Scrum Team och Product Owner är rätt utbildade inför projektet och att alla beröra parter får rätt information. De gånger utomstående vill ta del av projektet får de ta det med Scrum Mastern eftersom denne ska se till att teamet arbetar effektivt och ostört. Fokus ligger alltid på att ge teamet bästa möjliga förutsättningar för att uppnå det uppsatta målet för sprinten. (Schwaber, 2004)

Scrum Teamet består av fem till nio personer med skilda kompetenser och med odefinierade roller. Alla i teamet ska kunna göra allt inom teamets arbetsuppgifter. Detta kräver att teamet får arbeta ihop under en längre period, sammansättningen bör inte ändras. Specialister på vissa områden kan förekomma bland de enskilda individerna men huvudsaken är att teamet ska innehålla den kompetens som fodras för att kunna genomföra projektet och lösa sprintens uppgift. Det är teamet som bestämmer hur arbetet ska läggas upp, hur de ska lösa uppgifterna och hur uppgifterna ska fördelas mellan team medlemmarna. Således är de ansvariga för att dessa uppgifter utförs. Teamet fungerar självorganiserande och tar beslut om hur utvecklingen ska fortskrida. (Schwaber, 2004)

Artefakter

Ett Scrum projekt innehåller tre artefakter *Produkt Backlog*, *Sprint Backlog* och *Burndown Chart*. Samtliga är viktiga inom projektet för att säkerställa krav på produkten, få en överskådlig bild och driva projektet framåt. (Schwaber, 2004)

Product Backlogen är en förteckning med krav på vilka funktioner som produkten ska ha samt vilka förändringar som ska göras. Således ska den behandla både affärsmässiga och tekniska perspektiv. Listan liknar en kravspecifikation i traditionella systemutvecklingsprojekt. Product Owner tar fram Product Backlogen. Det är Product Owner som prioriterar de olika kraven, baserade på vad som enligt Product Ownern är mest centralt utifrån det affärsmässiga. Hur de olika kraven ska hanteras rent tekniskt är det inte Product Ownern som bestämmer utan det är upp till teamet att själv fastställa utvecklingen av de tekniska bitarna och hur de ska fördelas inom teamet. Product Backlogen innehåller även ett estimat på tidsåtgång för utvecklingen, som tagits fram av teamet. Listan är inget som binder kraven och tidsåtgång utan ska ses som ett levande dokument som kan ändras under projektets gång beroende på hur omvärlden ser ut och om kraven på produkten ändrats. Dessa förändringar får enbart göras av Product Ownern. (Schwaber, 2004)

När Product Ownern fastställt vad som ska prioriteras skapar han en mer detaljerad lista över dessa punkter och denna lista benämns *Sprintbacklog*. Det som listas i Sprint Backlogen är det som teamet ska arbeta med under aktuell sprint. Punkterna i listan kallas för *Tasks* och är mer tekniskt detaljerade än Product Backlogen. Varje Task specificerar vilken aktivitet som ska genomföras. Dessa bör ta mellan 4-16 timmar att utföra, tar det längre tid ska en Task brytas ner i mindre punkter. De tillfällen när ändringar måste göras i dokumentet är det endast

teamet som får göra det. När en teammedlem har genomfört en Task, uppdaterar han sprintbacklogen. Således har teamet en bra översikt på sprintens tillstånd. (Schwaber, 2004)

För att stämma av återstående arbete och arbetstid i sprinten mot kvarvarande dagar i sprinten används en *Burndown Chart*. Varje dag markerar teamet hur mycket som återstår av det tidsplanerade arbetet och ser då om sprinten är i fas med den tidsplanen. Medlemmarna i teamet rapporterar resterande arbete på sin uppgift och de räknar därefter ner tiden mot noll. En Burndown Chart för Product Backlog kan även framställas för att få fram ett diagram som ger en bild av sambanden mellan återstående arbete givet ett visst tillfälle och projektets utveckling i avslutade uppgifter. På detta sätt kan projektdeltagare se hur den planerade tiden för projektet påverkas om de tillför eller tar bort uppgifter och funktioner. (Schwaber, 2004)

Ceremonier

Ett Scrum projekt har tre ceremonier; *Daily Scrum*, *Sprint Planning Meeting* och *Sprint Review Meeting*. Ceremonierna finns för att tidsplanen ska kunna hållas samt att alla intressenter, projektdeltagare såväl som utomstående blir informerade om projektets tillstånd och utveckling. (Schwaber, 2004)

Daily Scrum är ett dagligt möte som hålls av Scrum Mastern med Scrum Teamet (Schwaber, 2004). Mötet hålls vid samma tidpunkt varje dag och ska pågå i max 15 minuter med alla i teamet närvarande. Syftet med mötet är att dela information mellan medlemmarna för att alla inom teamet ska vara medvetna om läget i sprinten och projektet. Scrum Mastern är den som ansvarar för att mötet genomförs på ett korrekt sätt och att alla i gruppen får komma till tals. Under mötet ska alla i teamet svara på följande tre frågor:

- Vad har du gjort sedan förra mötet?
- Vad ska du göra fram till nästa möte?
- Vad har du för hinder i ditt arbete?

Fokus på mötet ska främst riktas på Sprint Backloggen och utvecklingen av projektet. (Schwaber, 2004)

Sprint Planning Meeting är ett möte mellan Scrum Master, Scrum Team och Product Owner och ibland även andra intressenter som dock inte får anföra något under mötet. Syftet med mötet är att planera den kommande sprinten i projektet och ge teamet den information som krävs för att kunna arbeta obesvärat under hela sprinten. Inför mötet uppdaterar Product Ownern prioriteringarna i Product Backloggen och strukturerar upp kraven. Dessa prioriteringar beskriver Product Ownern sedan för teamet under mötets första del. Därefter är det upp till teamet att fastställa vilka uppgifter de kommer att arbeta med under kommande sprint. Under mötets andra del arbetar teamet med att dela upp uppgifterna till mindre Tasks som de ska arbeta under nästkommande sprint. (Schwaber, 2004)

När sprinten har avslutats hålls en *Sprint Review Meeting*. Deltagare på dessa möten är Product Owner, Scrum Master, Scrum Team och andra intressenter som till exempel slutanvändare. Här presenterar Scrum Team för övriga deltagare vad som har uträttats under sprinten. På mötet får deltagare insikt i hur projektet utvecklats och får tillfälle att komma med återkoppling och synpunkter. Product Ownern diskuterar också med Scrum Team under mötet om hur funktionerna som arbetats fram under sprinten ska fungera i verkligheten. Nästa fas i mötet är att diskutera hur de ska gå vidare med nästa sprint. Deltagarna resonerar om vad som ska prioriteras och vad som är viktigast att åstadkomma under den kommande sprinten. I slutet på mötet utvärderar deltagarna den gångna sprinten, där teamet får tillfälle att berätta om vad de ansåg var bra respektive mindre bra under arbetets gång. Genom att besvara dessa frågor och diskutera de kan teamet resonera om hur de ska förändra och förbättra inför nästa sprint. (Schwaber, 2004).

2.2. Kommunikation

Kommunikation avser två eller flera personer som samtalar om en specifik situation och skapar en gemensam bild av situationen genom diskussion. Kommunikation kan vara *symmetrisk*, vilket innebär att en relation mellan två individer är i jämvikt eller kan relationen vara *komplementär*, vilket innebär att det finns en dominans hos den ena partnern exempelvis chefen som innehar en formell makt och utnyttjar den för att ha ett övertag i kommunikationen det i sin tur utlöser undergivenhet hos den anställda. Engquist (2002) menar att det blir en ond spiral och det effektivaste sättet att bryta spiralen är att chefen i det här fallet ändrar sitt beteende. Engquist (2002) diskuterar vidare att ett samtals resultat beror på yttre faktorer som kultur, arbetsmiljö, regler och personliga faktorer som begåvning, personlighet och kunskap.

2.2.1. Synkron och asynkron kommunikation

När människor utväxlar information eller kunskap mellan varandra inom projektet kan de göra detta antingen asynkront eller synkront. Synkron kommunikation innebär att människor interagerar i realtid, medan asynkron innebär att människor interagerar vid olika tidpunkter med varandra. (Kock, 2007)

- Synkron kommunikation: ansikte mot ansikte, telefon, videokonferens.
- Asynkron kommunikation: E-post, fax, brev, SMS.

Asynkron kommunikation anses vara effektivare när det gäller att förmedla allmän information eftersom det är mindre kostsamt och vanligtvis går det snabbare än synkron kommunikation. Ett exempel på detta kan vara när projektledaren behöver komma ut med information till samtliga i projektet, då lönar det sig att skicka E-post istället för att samla alla och förlora värdefull tid som kan utnyttjas till att utveckla produkten. (Kock, 2007)

Synkron kommunikation används när det handlar om att förmedla kunskap till någon eller några i projektet vilket kan vara när projektledaren vill diskutera med utvecklarna om hur systemets arkitektur ska se ut. Då är det viktigt att ta det synkront för det ger de inblandade möjlighet att fråga om de inte har förstått informationen, att försöka förmedla detta asynkront

kommer att ta längre tid med större risk för missuppfattning och feltolkning av meddelandet. (Kock, 2007)

Det finns två andra situationer när det är bättre att använda synkron kommunikation istället för asynkron kommunikation. Det första är när projektledaren ska diskutera eller överlämna personlig information som är känslig till en annan projektanställd. Att skicka informationen asynkront kan få personen som tar emot information att känna sig mindre värd eller oviktig. Det andra är när information är känslig av etiska eller lagliga skäl då ska informationen framföras synkront för att minska risken att informationen kommer ut till obehöriga. (Kock, 2007)

	Fördel	Nackdel
Synkron	Snabbare feedback, hantering av känslig information.	Kostsamt, tidzon beroende, tidskrävande.
Asynkron	Allmän informationsspridning blir effektivare och billigare.	Större risk för missuppfattning av information.

Figur 2.2 För- och nackdelar i kommunikation

2.3. Projektadministration

Det finns ett antal teoretiska aspekter och tillvägagångssätt i relation till projektledning inom agila mjukvaruutvecklingsmiljöer. Svensson & von Otters (2003) teoretiska aspekter om projektledning där fokus på tajming, planering, uppgifter, lokalisering och andra projekt förvaltningspraxis är centrala punkter i deras undersökning som har projektledning i fokus vilket är relevant för vår undersökning gällande projektledning inom agila och iterativa mjukvaruutvecklingsmiljöer. Vi kommer att använda oss av Ken Schwaber (2004) som beskriver tillvägagångssättet av projektadministrationen i arbetsmetoden Scrum, vilket är en viktig del när vi undersöker införandet av arbetsmetoden Scrum i stora systemutvecklingsprojekt. Inom projektadministration har Kerzner (2009) präglat ämnet med sin bok som anses av många experter vara ”projektledarens bibel”. Vi har valt några av Kerzners beskrivningar, tillvägagångssätt och metoder som berör vårt undersökningsämne. Projektledarens visioner, färdigheter och arbetssätt beskrivs bäst i (Christensen & Kreiner, 1997). Vi har valt att använda oss av deras synpunkter och råd om projektadministration i vår undersökning. Med dessa aspekter, tillvägagångssätt och metoder har vi kunnat skapa en förståelse om projektadministration och vilka grundläggande tillvägagångssätt och teoretiska utgångspunkter en projektadministratör bör utgå ifrån.

Svensson & von Otter (2003) förklarar att projektledaren inte är ensam om att göra arbetet med planering, schemaläggning, uppskattning och lokalisering. I flexibla utvecklingsmiljöer är det mer av en lagsport. Schwaber (2004) valde namnet ”Scrum Master” istället för det traditionella namnet ”projektledare”. Scrum Masterns arbetsuppgifter skiljer sig från en traditionell projektledares uppgifter till en viss utsträckning. Denna skillnad i terminologi är symbolisk och reflekterar en drastisk förändring i ledarens arbetssätt, arbetsuppgifter och ledarskapskaraktär. (Schwaber, 2004)

Scrum Master är en av de tre roller som medverkar i ett Scrumprojekt. En Scrum Master innehar den platsen som normalt innehas av projektledaren. En traditionell projektledare är ansvarig för definition och hantering av arbetet medan en Scrum Master ansvarar för processerna inom Scrum. Scrum Mastern bör ha en klar uppfattning om hur Scrum fungerar och effektivt hantera Scrumprojekt. En Scrum Masters befogenheter är till stor del indirekta, i huvudsak härstammar de från Scrum Masterns kunskaper om Scrum regler, praxis och arbetet för att säkerställa planuppföljningen av Scrum projektets aktiva medlemmar. (Schwaber, 2004)

Scrum Mastern ansvarar för projektets framgång och bidrar till att öka sannolikheten för framgången, detta genom att hjälpa Product Owner, vilket är den andra rollen i ett Scrum projekt. Att välja det mest värdefulla i Product Backlog och genom att hjälpa tredje rollen, Scrum Team, att effektivt omvandla backlogen till funktioner. Att lära sig Scrum Masterns grundläggande och praktiska uppgifter är lätt för de flesta, men vissa människor har svårt att lära sig konsten av att vara en Scrum Master. Scrum är en enkel och okomplicerad uppsättning av metoder, regler och roller. Däremot är filosofin bakom Scrum något mindre enkel och kan ibland vara svår att förstå. (Schwaber, 2004)

2.3.1. Projektstyrning

Christensen & Kreiner, (1997) hävdar att projektstyrning används när projektet är i *genomförandefasen*. Det är i genomförandefasen olika aktiviteter utförs och förverkligar målen i praktiken. Det handlar om att hålla projektets medlemmar på spåret och att aktiviteterna utförs i överensstämmelse med tidsplan, budget och specifikationer.

En nyckelfaktor för projektledarens framgång i denna fas är befogenhet som bör tilldelas projektledaren av projektorganisationen. Detta för att säkerställa att produkten utvecklas enligt projektplanerna (Christensen & Kreiner, 1997).

Svensson & von Otter (2001) medger att det är svårt att beskriva en bra projektledare på några få ord. Ett antal analogier har prövats som ett försök till en bättre beskrivning där projektledarrollen har jämförts med regissörens, lagledarens, dirigentens för att nämna några.

Det finns en uppsjö av litteratur som tar upp projektledarens egenskaper, färdigheter och förmågor. Svensson & von Otter (2001) har valt att använda Lewis (1998) som menar att en projektledare ska vara en god lyssnare för att kunna vara en god rådgivare. Projektledaren ska kunna fatta beslut. Beslutfattandet är en vardaglig händelse i en projektledares vardag och

ingår i de flesta procedurerna i ett projekt. Att kunna delegera korrekt är en annan viktig färdighet som bör finnas hos projektledaren.

Att styra ett projekt, är att genom andra människor genomföra och utföra planer och uppdrag som är nödvändiga för att uppnå eller överträffa ett mål. Projektstyrning innebär åtgärder som (Kerzner, 2009):

Bemanning: ser till att en kvalificerad person väljs ut för varje position, att färdighet och förmåga anpassas till uppgifterna en position innehåller.

Utbildning: att lära individer och grupper hur de ska fullgöra sina uppgifter och ansvar.

Handledning: att dagligen ge medarbetarna instruktioner, vägledning och disciplin vid behov, detta för att de ska kunna fullgöra sina uppgifter och ansvar.

Delegering: att tilldela arbete, ansvar och befogenheter för att andra medarbetare ska kunna utföra sina arbetsuppgifter fullt ut.

Motivering: att uppmuntra medarbetarna till att arbeta effektivt, detta genom att uppfylla eller vädja deras behov.

Rådgivning: att hålla privata diskussioner med medarbetarna om hur han eller hon kan åstadkomma högre arbetseffektivitet, lösa ett personligt problem eller förverkliga sina ambitioner.

Samordning: ser till att aktiviteter och uppdrag bedrivs i förhållande till deras innebörd, betydelse och/eller värde, detta ska ske med minsta möjliga komplikation eller konflikt.

2.3.2. Befogenheter

En projektledares befogenhet kan definieras som den rättsliga eller rättmätiga makten att befalla, agera eller styra andras aktiviteter. Befogenhet kan delegeras av en överordnad. Makt, å andra sidan, ges till en individ av de underordnade och kan anses som ett mått på deras respekt för honom eller henne. En projektledarens befogenhet är en kombination av hans eller hennes makt och inflytande där underordnade, kamrater och kollegor gärna accepterar projektledarens sätt att leda. (Kerzner, 2009)

Befogenhet är nyckeln till projektledningsprocessen. Projektledaren måste hantera både funktionsrelaterade och organisatoriska förfrågningar samt uppdrag. Detta genom att samordna de nödvändiga aktiviteterna för att överträffa målet av ett specifikt projekt. Projektets befogenhet stadgar ett nytt tankesätt som är nödvändigt för att kunna förena de organisatoriska krafterna och styra de mot ett gemensamt mål. Det innebär att utföra projektet oavsett var krafterna befinner sig inom organisationen. En projektledare som misslyckas med att bygga eller behålla sina allianser kommer att finna motstånd i form av likgiltighet på de krav och behov som är nödvändiga för att lyckas med projektet. (Kerzner, 2009)

Hur stora befogenheter som tilldelas projektledaren varierar beroende på projektets storlek och den potentiella konfliktmängd med organisationens chefer. Den sorten av konflikt har alltid förekommit, dock måste vissa fundamentala rättigheter tilldelas projektledaren, detta för att projektledaren ska inneha kontrollen över projektet.

Projektledarens ansvar gentemot projektets progress och framgång i sin helhet kan bli ett dilemma för projektledaren som upplever att han eller hon har ett stort ansvar men sällan den totala makten över projektet. I vanliga fall tilldelas befogenheter till projektledare som ger makt över ett objekt som de är ansvariga för. (Christensen & Kreiner, 1997)

RAM (Responsibility Assignment Matrix) är ett dokument där gruppernas medlemmar skriver under, ett RAM innehåller svar på frågor som ”Vem behåller signaturmakten?” ”Vem måste bli underrättad?” ”Vem tar besluten?” dessa frågor kan endast besvaras genom en klar definition av befogenhet, ansvar och ansvarsskyldighet (Cleland & King, 1968). Kerzner (2009) hävdar att de centrala begreppen i befogenhet är:

- *Befogenheten* är den enskildes rätt att ta de nödvändiga besluten som krävs för att uppnå sina mål eller ansvar.
- *Ansvar* är uppdraget för slutförandet av en specifik händelse eller aktivitet.
- *Ansvarsskyldigheten* är acceptansen av framgången eller misslyckandet.

2.3.3. Projektmöten

Dessa möten är nödvändiga för att klargöra och dokumentera delarna av ett projekt samt registrera framstegen därefter. Det finns tre typer av möten i ett projekt (Kerzner, 2009):

Gruppmöten: de flesta projekt har möte varje vecka, varannan månad eller varje månad, där projektledaren informerar arbetsgruppen om projektets status. Dessa möten är flexibla och hålls endast om de kommer att gynna laget eller arbetsgruppen.

Styrelsemöten: styrelsen har rätten till att kräva månatliga möten där statusgenomgång tas upp. Men om projektledaren anser att status på projektet kan identifieras vid andra mötestillfälle då är projektledaren berättigad att avslå dessa extra statusmöten.

Kundmöten: dessa möten är ofta kritiska och har fast struktur i sin schemauppläggning. Projektledaren måste vara väl förberedd inför mötet för att kunna visa kunden progressen i projektet och diskutera eventuella problem som uppstått.

2.4. Scrum i Stora Systemutvecklingsprojekt

2.4.1. Systemutveckling i stora projekt

I stora projekt som kräver större arbetsinsatser än ett Scrum Team klarar av att hantera rekommenderar Schwaber (2004) att projektet ska delas upp på flera Scrum Team som därigenom arbetar parallellt på samma projekt. Insatserna från dessa team samordnas genom ett antal mekanismer som sträcker sig från formell samordning till Ad-hoc. När det finns flera Scrum Team som arbetar på ett stort projekt samtidigt betyder det att projektet skalas och varje avskalad del av projektet blir ett delprojekt och har sin egen komplexitet som kräver en egen lösning. (Schwaber, 2004)

Grundtanken är att varje Scrum Team i ett projekt inte är större än vad som förespråkas, vilket innebär max nio personer, men själva projektet kan ha flera hundra inblandade. Då de olika teamen arbetar med utvalda delar av systemet menar Stober & Hansmann (2009) att det kan krävas ett tvärfunktionellt team som ska reflektera över det totala projektet. Ett team som inte arbetar på detaljnivå utan som arbetar övergripande med de olika delarna som delprojekten arbetar med.

Innan ett stort projekt delas upp är det viktigt att det införs en lämplig infrastruktur i projektet. Dessutom behöver det utformas en mer detaljerad produktinformation och teknisk arkitektur så att arbetet kan fördelas mellan de olika Scrum Teamen. Beroende på om teamen befinner sig i samma byggnad eller är geografiskt utspridda krävs det en utarbetad plan för hur hanteringen av kommunikation, hur möten skall ske mellan teamen, vilken struktur och vilka roller som ska finnas i projektet (Schwaber, 2004).

I Scrum finns det en Product Owner som är ansvarig för produkten. Han formulerar kraven och som säkerställer att produkten utvecklas med fokus på affärsnyttan (Schwaber, 2004). Men enligt Stober & Hansmann (2009) kan rollen Product Owner delas upp i stora projekt i en Chief Product Owner som arbetar nära företagsledningen samt ett antal ordinarie Product Owner som arbetar tillsammans med Scrum Mastern och Scrum Teamet.

Product Owner är den som tar fram Product Backlogen. Ett projekt utgår från en Product Backlog, detta för att inte öka komplexiteten (Schwaber, 2004). Dock anser Stober & Hansmann (2009) att backlogen kan behöva delas upp i mindre listor som är anpassade för det enskilda teamet i stora projekt. Med utgångspunkt från huvud Product Backlogen skapas mindre backlog listor som teamen erhåller. Backlogen ska lista det som teamet ska arbeta med och vad de ska prioritera (Stober & Hansmann, 2009).

2.4.2. Kommunikation i stora projekt

Alla Agila metoder bygger på samarbete, både inom och utanför utvecklingsgruppen. Agila metoder bygger på informell kommunikation snarare än omfattande dokumentation för att snabbt sprida information till hela teamet. För att lyckas med spridningen av information behöver projektledarens roll ändras från att styra med hela handen till att vara mer av en coach och mentor (Coram & Bohner, 2005). Detta kan skapa problem om alla team inte

befinner sig på samma geografiska plats. Det finns olika kommunikationsmedel som projektledaren kan använda såsom E-post och chatt, men på bekostnad av den synkrona kommunikationen som är en signifikativ del av arbetsmetoden Scrum (Turk et al., 2002). Kommunikationen med företagsledningen är viktig eftersom de sitter på pengarna och utan företagsledningens stöd tenderar projektet att misslyckas. Således, är det projektledarens uppgift att med lite dokumentation, men med delar av produktens funktionalitet kunna visa att projektet gör framsteg. Genom att diskutera och visa kan projektledaren få ledningen övertygad och därmed stöd från ledningen. (Coram & Bohner, 2005)

2.4.3. Projektadministration i Stora Projekt

Styrning i Stora Projekt

I projektstyrning ingår ett antal ansvarigheter som projektledare, Scrum Master och Scrum Team ska ta på sig. Det kan hända att likadana ansvarigheter tilldelas flera roller beroende på projektstorleken och vilka roller som medverkar i projektet. Gällande bemanning och utbildning ser projektledaren till att kvalificerade personer intar den mest lämpliga arbetspositionen samt ger projektledaren de medverkande i projektet möjligheten till få rätt utbildning för att kunna fullgöra sina uppgifter och ansvar på det mest effektiva sättet.Handledning är ansvaret som tilldelas Scrum Mastern, Scrum Teamet och projektledaren i stora mjukvaruutvecklingsprojekt. (Schwaber, 2004)

Delegering är uppgiften som innebär arbetstilldelning, även tilldela ansvar och befogenheter för att kunna uppnå högre arbetseffektivitet genom att utnyttja personalens förmågor och färdigheter. Delegering är projektledarens ansvar i projektet. Både Scrum Mastern och projektledaren har ansvar för motivering och rådgivning. Dessa åtgärder är väsentliga för arbetseffektiviteten och uppnås genom uppmuntring, tillmötesgående för medarbetarnas behov, önskemål och frågor inom en rimlig gräns. Vid behov har de även privata diskussioner med medarbetare. (Kerzner, 2009)

Förmågan att Samordna är en viktig egenskap hos en projektledare. I regel är det Scrum Mastern och Scrum Teamet som samordnar, men inom stora projekt kan det hända att projektledaren också deltar i koordinering och kan indirekt kontrollera de åtgärder och procedurer som samordningen innefattar. (Schwaber, 2004)

Befogenheter i Stora Projekt

I ett Scrumprojekt delas personer in i två grupper. Dessa är *engagerade* och *åskådare*, de engagerade är Product Owner, Scrum Master och Scrum Team dessa är tydliga och lätta att förstå. Alla andra i en organisation är åskådare som kan vara intresserade av projektet och som kan ha ett starkt egenintresse i projektets framgång, men som måste arbeta via de engagerade. Åskådarna har ingen direkt makt eller befogenhet över projektets genomförande och framsteg. Schwaber (2004)

Att hantera komplexa arbetsprocesser är aldrig lätt. En fördel med Scrum gällande befogenhet är att användning av Scrum förenklar frågor om ansvar och befogenhet för Scrum Mastern.

Dock i stora projekt som innehåller flera Scrum Teams kan konflikt uppstå mellan Scrum Masters som anser att de har rätt i en viss fråga som kan beröra flera team. (Schwaber, 2004)

Möten i Stora Projekt

För att kunna hantera Daily Scrum i stora projekt med många team införs Scrum of Scrums (Schwaber, 2004; Stober & Hansmann, 2009). Scrum of Scrums är ett extramöte där representanter från de olika teamen deltar och redogör för progressen i det egna projektet. De delar information, ger feedback mellan representanterna, och definierar gemensamt status för hela projektet på sätt ökar förståelsen för projektet som helhet. Om projektet rör sig över stora avstånd föreslår Paasivaara et al., (2008) att representanterna svarar på frågor i ett E-post innan mötet ska starta för att hålla mötestiden. Scrum of Scrums är bra för ökad kommunikation mellan teamen. Om teamen haft problem med sin del kan de diskutera med andra team och möjligen få ett svar på problemet och få ny kunskap som ökar kompetensen inom det egna teamet. (Schwaber, 2004)

2.5. Ramverk

Syftet med vårt Ramverk

Vi har skapat ett ramverk för att kunna sammanfatta samtliga teorier som vi använt i vår undersökning samt presentera de essentiella delarna av dessa teorier. Ramverket behandlar endast de faktorer som vi har resonerat oss fram till med stöd av tidigare presenterade teorier. Dessa faktorer kan ha styrande och/eller förändrande effekter på införandeprocessen av arbetsmetoden Scrum i stora systemutvecklingsprojekt.

Ramverkets struktur

Ramverket är uppbyggt på tema, faktorer och förändring eller icke förändring av värdet i respektive faktor. Vi har kategoriserat de data som vi behandlar i vår undersökning i tre teman. Nämligen arbetsprocess, kommunikation och projektadministration. Dessa teman är centrala i vår studie när organisationer inför den agila arbetsmetoden Scrum.

Hur vi använder ramverket

Utifrån de kategorier vi har skapat har vi utformat frågor som bygger på ramverket och vår undersöknings riktlinje. Med ramverket kopplar vi samman insamlad data från intervjuerna och de teoretiska utgångspunkterna som vi använder oss av i vår undersökning. Vi kommer att använda ramverket när vi analyserar våra insamlade empiriska data och även effektivisera vårt arbete när vi håller oss inom ramverkets riktlinjer och fokuserar endast på de utvalda fenomenen.

Tabell 2.3 Ramverk

STORA SCRUM PROJEKT			
TEMA	FAKTORER	Ändrar sitt värde	Behåller sitt värde
ARBETSPROCESS	Roller	●	
	Artefakter		●
	Ceremonier		●
KOMMUNIKATION	Kund	●	
	Grupp	●	
	Grupper	●	
	Ledning	●	
PROJEKT-ADMINISTRATION	Möte	●	
	Styrning	●	
	Befogenhet		●

3. METOD

3.1. Undersökningsstrategi

I den här uppsatsen har vi valt en strategi som innebär att vi först har införskaffat förväntningar av hur verkligheten ser ut. Detta genom att läsa vetenskapliga artiklar och teorier om det problemområde vi haft för avsikt att undersöka. Utifrån vald litteratur har vi skapat ett ramverk som ligger till grund för vår studie. Med hjälp av ramverket har vi tagit fram relevanta intervjufrågor. Frågorna har vi haft som ett underlag i vår undersökning för att se om förväntningarna går ihop med verkligheten genom den insamlade empirin. (Jacobsen, 2002)

3.2. Intervjuguide

Med vårt ramverk som utgångspunkt utformade vi en intervjuguide. När vi skapade frågorna strävade vi efter att inte ha allt för abstrakta frågor men samtidigt inte för specificerade. Detta för att informanterna skulle kunna ge oss utförliga och genomtänkta svar istället för korta svar. Informanterna fick ta del av intervjuguiden innan intervjun genomfördes för att kunna skapa en bild av vilka faktorer och frågor som skulle diskuteras. Frågorna ställdes inte alltid i ordningsföljd, i vissa fall kändes en fråga irrelevant och i andra fall ställde vi följdfrågor för att därefter återgå till intervjuguiden. Guiden användes som ett hjälpmedel av både oss och informanterna.

Intervjuguiden (Bilaga 2) börjar med ett antal inledande frågor dels för att vi som författare ska kunna skapa oss en bättre bild av informanten men också för att få informanten att känna sig bekväm. Därefter övergår intervjuguiden till fem teman där vi diskuterar ett antal faktorer som finns inom respektive tema. Anledningen till att vi valt att dela upp frågorna i teman var för att vi bättre skulle kunna strukturera upp intervjun, skapa en bättre helhetsbild samt att vi skulle få en större chans att täcka in det område vi behandlade, Scrum i stora projekt, och således undgå att komplettera med ytterligare intervjuer. Vi avslutade varje tema med att fråga om informanten hade något att tillägga. Vi strävade efter att ställa frågor som gav svar som inte överlappade med varandra.

3.3. Datainsamling

Vi valde att intervjua tre projektledare, två av dem befann sig i varsitt stort projekt medan den tredje nyligen hade varit med och avslutat ett stort projekt. Vi fick tillstånd av informanterna att spela in intervjuerna. Vi transkriberade intervjuerna för att kunna utföra en noggrann dataanalys. Dessa finns i bilaga 3,4 och 5.

Det finns för- och nackdelar med vårt tillvägagångssätt i hanteringen av intervjuerna. Den största fördelen är att en transkribering garanterar att allt från intervjun kommer med (Jacobsen, 2002). Transkriberingen av intervjuerna underlättade för oss att kategorisera och tolka informationen. En nackdel är att det blir mycket information att hantera och personen

som transkriberar kan lätt tappa modet, men det senare var inga problem för oss. Vi valde att inte tolka eller skriva om meningar. Detta för att behålla informationen intakt och utifrån det tolka och analysera transkriberingsresultatet. Informanterna fick en kopia av transkriberingen för att kunna ta del av den och godkänna. (Jacobsen, 2002)

3.4. Kategorisering av data

Att kategorisera data innebär att data grupperas. Kategorisering är ett ganska fundamentalt arbetssätt som underlättar data behandlingsstruktureringen. Det bidrar till att visa hur vissa slags data som är insamlad ur intervjuer, teorier och metoder behandlar samma fenomen eller ämne, detta görs genom att skapa olika kategorier utifrån texten. Kategorisering är ett nödvändigt verktyg som eliminerar behovet av att hantera stora datamängder på ett ineffektivt och passivt sätt. Inom kvalitativa undersökningar deklarerar de viktiga faserna i två huvudpunkter (Jacobsen, 2002):

- *"Först måste det skapas några relevanta kategorier utifrån de data som finns till förfogande."*
- *"Därpå måste enheter (observationer och intervjuer) hänföras till kategorierna."*

Vi använde oss av datakategorisering, detta innebär att kategorisera insamlade intervjudata i kapitel 4 och befintlig data ur teorierna i kapitel 2. Våra kategorier är indelade i tre teman (Arbetsprocess, Kommunikation och Projektadministration) för att underlätta för läsaren, reducera misstolkning och missförstånd samt öka transparensen och förståelsen av uppsatsen.

3.5. Val av informanter

Vi valde att intervjua tre projektledare inom agila projekt som använder Scrum i stora systemutvecklingsprojekt. Projektledarna besitter djupa kunskaper om projektledning och har goda erfarenheter av att använda Scrum i stora projekt. Detta ökar trovärdigheten och giltigheten i vår uppsats och är av stor vikt för vår undersökning samt ger oss möjligheten att undvika behandling av irrelevant data.

3.6. Validitet & reliabilitet

Inom kvalitativa undersökningar är det viktigt att överväga och prioritera faktorn objektivitet, att det är extra viktigt att forskaren ska vara objektiv i sitt arbetssätt. Forskare ska undvika att dra snabba slutsatser, all fakta som kan vara avgörande ska undersökas noggrant. Detta innebär att feltolkning av fakta kan påverka undersökningens resultat, det vill säga inriktning av slutsatsernas kontext i en undersökning. Brister kan uppstå i arbetets validitet och reliabilitet (Jacobsen, 2002; Kvale, 1996).

De två faktorerna har en elementär innebörd som bidrar till att åstadkomma en giltig- och trovärdig undersökning. Detta hanterar vi genom att observera och kritisera insamlade data noggrant. För att vara objektiva i vår undersökning samt kunna producera en kvalitativ undersökningsmetod har vi valt följande faktorer (Jacobsen, 2002; Kvale, 1996):

- Giltighet: definieras som något riktigt och konkret i en utlåtande slutsats
- Tillförlighet: definieras som trovärdighet eller reliabilitet för undersökningens resultat.

3.7. Etik

Jacobsen (2002) beskriver i sin bok de etiska aspekterna i förhållande till intervjuerna som ingår i vetenskapliga studier. Vi har valt att följa Jacobsens (2002) tillvägagångssätt gällande etiska aspekten. Detta för att den etiska aspekten är en väsentlig del av intervjuprocessen.

- Att hålla intervjuerna inom informantens naturliga arbetsmiljö.
- Att ge informanten möjligheten till att gå igenom frågorna för att öka trovärdigheten genom att få intervjupersonen att ändra i frågor som kan innehålla konfidentiell information eller olämpliga frågor.
- Att presentera syftet med studien.
- Att berätta för intervjupersonen var arbetet kommer att publiceras och vem som får tillgång till arbetet.
- Att kunna erbjuda en kopia på slutliga upplagan av kandidatuppsatsen.

Kvale (1996) förslår att skydda informanter, detta genom att eliminera all information som kan anses känslig eller sekretessbelagd information som gör det möjligt att spåra informanten. Vi har erbjudit informanterna att läsa igenom intervjuprotokollet efter varje intervju.

Att presentera syftet med arbetet för informanten är en viktig faktor inom den etiska aspekten (Jacobsen, (2002). Faktorn berör i första hand intervjuriktningen, att vara medveten om syftet med en undersökning reducerar missförstånd och irrelevanta svar hos informanten. Det är också viktigt att begränsa frågorna för att komma in på djupet i ämnet vi undersöker, på detta sätt undviker vi irrelevant information såsom detaljerade beskrivningar på en viss artefakt som saknar relevans för undersökningen. Det är också viktigt att kunna förstå när det blir för lite eller för mycket information, båda fallen anses vara en stor nackdel.

Jacobsen (2002) hävdar att det är extra viktigt att skicka en kopia på det slutliga arbetet, detta för att ge informanterna möjligheten till att få se hur informationen behandlats i en empirisk studie, att informationsinnehållet har tolkats på ett rimligt sätt och att de kanske kan dra nytta av det empiriska resultatet. Samtliga intervjupersoner blev erbjudna en kopia på den slutliga upplagan av denna undersökning och alla har visat ett stort intresse av att få en kopia.

4. EMPIRI & ANALYS

I tabell 4.1 presenterar vi informanternas bakgrund, kompetens och antal år de arbetat som projektledare. Denna information styrker deras kompetens inom det område som vi studerar. Vi har förkortat ordet *intervjuperson* till IP och tilldelat de A, B och C. Detta har vi gjort för att underlätta för läsaren.

Tabell 4.1 Informanternas bakgrund och kompetens

Informant	Utbildning	Jobbtitel	Antal år	Antal projekt	Övrigt
<i>IPA</i>	Magisterexamen i informatik	Projektledare	15	Ca.20 små och stora	Arbetar både med traditionella och agila projekt
<i>IPB</i>	Master of science in business technology	Scrum Master & sekundär Product Owner	5	1 stort	Arbetar i Scrum projekt
<i>IPC</i>	Civilingenjörsutbildning i elektroteknik	Senior konsult	16	5 stora	Certifierar Scrum Masters

4.1. Arbetsprocess

4.1.1. Roller

Samtliga informanter beskrev att när de arbetar enligt Scrum metoden i stora projekt delar de upp projektet i delprojekt. Två av informanterna arbetar i varsitt stort projekt som tillämpar Scrum metoden. Den tredje informanten har tidigare erfarenheter av Scrum i stora projekt. Enligt Schwaber (2004) är det på detta sätt som man ska arbeta med Scrum i stora projekt. I projekt som kräver större arbetsinsatser än ett Scrum Team klarar av att hantera rekommenderar Schwaber (2004) att projektet delas upp på flera Scrum Team som därigenom arbetar parallellt på samma projekt.

Alla tre informanter beskrev att deras *Scrum Team* följer Scrums beskrivning för hur team ska utformas. Det finns inga fördefinierade roller, dock finns det inom gruppen personer som är specialister på vissa områden inom systemutveckling men de har ingen officiell roll. Enligt Schwaber (2004) ska det inom ett Scrum Team inte finnas fasta roller som till exempel, testare utvecklare eller analytiker. Alla inom teamet ska kunna arbeta med samtliga uppgifter som de har. Samtidigt menar IPA att de försöker behålla alla personer i teamet kvar under projektets gång men inte alltid lyckas vilket leder till att sammansättningen av gruppen förändras under projektets gång. IPB säger samma sak och menar att detta är svårt i stora

projekt då dessa oftast pågår över en längre period och detta gör det svårt att behålla personer under hela projektet. IPC beskriver att i de stora projekt han varit på tidigare har han alltid strävat efter att behålla sammansättningen av gruppen. De tre informanterna påpekar alla vikten av att inte byta ut gruppmedlemmar under projektets gång. Schwaber (2004) menar att för att teamet ska kunna bli presterande kräver det att de får arbeta ihop sig under en längre period. Sammansättningen på teamet bör inte ändras.

IPA säger också att det i stora projekt, förutom de ordinarie *Scrum Teamen* som arbetar med valda delar av projektet, tillkommer ett team som arbetar tvärfunktionellt nära projektorganisationen. I de fall det finns flera Scrum Team anser Stober & Hansmann (2009) att tvärfunktionella team bör införas som ansvarar för det totala projektet. Ett team som inte arbetar med detaljerna utan snarare verkar övergripande med de olika delarna som delprojekten arbetar med.

Samtliga informanter nämner att inga förändringar görs i Scrum Master rollen när det kommer till stora projekt, denne arbetar enligt Scrum. Detta innebär att en Scrum Master ska fungera som en coach för teamet samt stödja de i deras arbete under sprinten samt säkerställa att Scrum följs (Schwaber, 2004). Dock tillägger IPC att i stora projekt får en Scrum Master även ta rollen som projektledare vilket innebär att denne får hantera fler än ett team.

IPA säger att rollen *Product Owner* inte förändras i stora projekt. De håller sig till en Product Owner som arbetar med de uppgifter som Scrum föreskriver. Dessa är att säkerställa att produkten hela tiden utvecklas utifrån en aspekt på affärsnyttan samt sammanställa alla behov och krav som utgör grunden för förändringar av produkten (Schwaber, 2004).

Tabell 4.2 Förändringen inom roller ur intervjupersonernas perspektiv

Förändring i roller	Intervjupersoner		
	IPA	IPB	IPC
Scrum Team	Ja	Ja	Nej
Scrum Master	Nej	Nej	Ja
Product Owner	Nej	Ja	Ja

IPB berättar att rollen som Product Owner har delats upp i två roller. Dels finns det en lokal Product Owner för Scrum Teamet vilket IPB arbetar som. Sedan finns där en Global Product Manager som har en roll som Chief Product Owner och denne representerar kundsidan. Enligt Schwaber (2004) kan en Product Owner antingen komma ifrån kundsidan eller från det egna företaget. IPB säger vidare att Chief Product Owner ansvarar för det stora projektet och att produkten som utvecklas ligger i fas med marknaden vilket enligt Schwaber (2004) är en huvuduppgift för Product Owner. Anledningen till uppdelningen av rollen är enligt IPB att Scrum kräver en tillgänglig Product Owner och detta är inte möjligt eftersom Chief Product Owner reser mycket till de olika delprojekten runt om i världen. Således tilldelades IPB rollen som en lokal Product Owner. En Product Owner bör finnas på plats under projektet och det

ska vara lätt att nå eftersom han är den som bestämmer vilka funktioner produkten ska innehålla (Schwaber, 2004; Stober & Hansmann, 2009).

Kommunikationen angående projektet, produkten och sprintarna sker mellan IPB och Chief Product Ownern. Dock har IPB inte lika stor kontroll över projektet som en Product Owner ska ha enligt Scrum. IPC berättar att de i stora projekt delar in projektet i delprojekt där varje delprojekt får en Product Owner. Sedan utses någon av dem till Chief Product Owner och blir därmed den som är högst ansvarig för projektet och produkten. Eventuellt kan Chief Product Owner behöva kompletteras med roller som arkitekt som formulerar kraven. Dock menar IPC att han försöker att inte ha för många roller utan strävar istället efter det tvärfunktionella i Scrum. Stober & Hansmann (2009) anser att i stora projekt kan Product Owner rollen delas upp hierarkiskt med en Chief Product Owner som arbetar nära företagsledningen och styrgruppen samt ett antal ordinarie Product Owner som arbetar mot Scrum Mastern och Scrum Teamet. Tabell 4.2 visar informanternas svar på om rollerna förändras när de arbetar efter Scrum i stora projekt.

4.1.2. Artefakter

Eftersom stora projekt har delprojekt väljer IPA att dela upp *Product Backlog*en i delar. De olika teamen arbetar sen med den egna backloggen som representerar teamets kompetensområde och är anpassad efter dem. IPB säger att de har en primär Product Backlog som gäller för hela projektet globalt och sen har de delbacklog som består av utvalda delar av huvudbackloggen och som de olika delprojekten arbetar med. IPC menar att de ibland kan behöva skala Product Backloggen för att varje team ska ha sin del att arbeta med. Detta är dock inte något han föredrar då det komplicerar projektet. IPC rekommenderar endast en Product Backlog oavsett hur stor produkten och projektet är, men att istället ha olika vyer av Product Backloggen. Varje område inom projektet har en egen vy av Product Backloggen som de koncentrerar sig på. I Scrum föreskrivs enbart en Product Backlog per projekt för att inte öka komplexiteten (Schwaber, 2004). Men Stober och Hansmann (2009) menar att det går bra att dela upp Product Backloggen i mindre listor när projektet är stort. Det ska finnas en huvudsaklig Product Backlog, men teamen som arbetar i delprojekten bör erhålla en mindre backloglista med formuleringar på vad de ska arbeta med och vad som prioriteras (Stober & Hansmann, 2009).

Tabell 4.3 Förändringen inom artefakter ur intervjupersonernas perspektiv

Förändring i artefakter	Intervjupersoner		
	IPA	IPB	IPC
Product Backlog	Ja	Ja	Ja
Sprint Backlog	Nej	Nej	Nej

När det gäller *Sprint Backlog*en säger alla tre informanter att inga förändringar görs när projektet är stort. Sprint Backloggen ska enligt Schwaber (2004) vara en detaljerad lista med

uppgifter som teamet ska arbeta med under den kommande sprinten. Tabell 4.3 visar informanternas svar på om artefakterna förändras när de arbetar efter Scrum i stora projekt.

4.1.3. Ceremonier

När det gäller *Sprint Planning* möten förekommer det enligt informanterna inga förändringar när projektet är stort. Dessa möten går till på samma sätt som när man arbetar i små projekt. Det enda som skiljer mellan informanternas svar är hur ofta mötena genomförs. IPA säger att de har planeringsmöte var fjärde vecka. IPB säger att de arbetar i två veckor mellan mötena. IPC säger att de antingen kör fyra veckor eller varannan vecka beroende på hur den kommande sprinten ser ut. Under *Sprint Planning Meeting* träffas Scrum Mastern, Scrum Teamet och Product Ownern för att planera den kommande sprinten och strukturera krav och funktioner med utgångspunkt från Product Backloggen (Schwaber, 2004).

Enligt informanterna görs det inga förändringar i *Daily Scrum* när projektet är stort. Alla informanterna nämner att Daily Scrum är ett möte mellan Scrum Team och Scrum Master. IPA och IPC tillägger att andra intressenter är välkomna att närvara på mötet men endast för att lyssna och observera. Medlemmarna i teamet är de enda som får prata. Enligt Schwaber (2004) är Daily Scrum ett kort möte mellan Scrum Team och Scrum Master som hålls varje dag för att dela information med varandra och uppdatera varandra om läget i sprinten.

IPA beskrev att på deras *Sprint Review Meeting* (sprintdemo) är det enbart Scrum Team, Scrum Master och Product Owner som närvarar. Istället för att hålla dessa möten mellan sprintarna säger IPA att de håller sprintdemo kontinuerligt, i stort sätt varje vecka. Sedan har de en sprintdemo för andra intressenter. I stora projekt är det mer pengar involverad och således fler intressenter. IPA menar att det då är bättre att hålla ett eget möte med företagsledningen för att visa upp hur budgeten för projektet har använts och hur långt de har kommit. Projektet som IPB arbetar i utförs själva mötet enligt Scrum, dock tillåts inga intressenter eller andra involverade personer, bara teamet, Scrum Mastern och Product Owner. IPB rapporterar sen till Chief Product Owner om det som de har gått igenom på mötet.

IPC säger att de följer Scrum när det kommer till sprintdemo. IPC anser att demonstrationen är det bästa mötet, då de får tillfälle att visa upp vad som har gjorts. Han ser gärna att, förutom Scrum Team och Product Owner, andra intressenter och berörda personer bjuds in till dessa möten. IPC anser att en gemensam sprintdemo bör hållas med samtliga Scrum Team, eventuellt mellan varannan sprintdemo, för att öka förståelsen för projektets helhet. IPC menar att detta ökar förståelse för produkten och förbättrar samarbetet inom projektet. Schwaber (2004) menar att förutom Scrum Team, Scrum Master och Product Owner bör även andra intressenter som till exempel slutanvändarna medverka under sprintdemo. Detta dels för att teamet ska kunna visa upp vad de utfört och hur långt de är i utvecklingen, men också för att få återkoppling och synpunkter på produkten från intressenterna som ser produkten ur ett annat perspektiv (Schwaber, 2004). Tabell 4.4 visar informanternas svar om ceremonierna förändras när de använder Scrum i stora projekt.

Tabell 4.4 Förändringen inom ceremonier ur intervjupersonernas perspektiv

Förändring i ceremonier	Intervjupersoner		
	IPA	IPB	IPC
Sprint Planning Meeting	Nej	Nej	Nej
Daily Scrum	Nej	Nej	Nej
Sprint Review Meeting	Ja	Ja	Nej

4.2. Kommunikation

4.2.1. Kommunikation grupp

När det gäller val av kommunikationskanaler svarade två av de tre projektledarna att kommunikationen förändras i stora projekt. Därmed är det inte möjligt att sköta kommunikation synkront som Scrum föreskriver. Samtliga intervjuade poängterar att det är väldigt viktigt med synkron kommunikation för att kunna läsa kroppsspråket på personen de diskuterar med. Tabell 4.5 representerar informanternas svar om de anser att det blir en förändring eller ej.

Tabell 4.5 Förändringen inom kommunikation ur intervjupersonernas perspektiv

Förändring i kommunikation	Intervjupersoner		
	IPA	IPB	IPC
Grupp	Ja	Nej	Ja
Kund	Ja	Ja	Nej
Ledning	Nej	-	Ja
Mellan grupper	Ja	Ja	Ja

IPA tycker att den största förändringen med kommunikation med grupperna i stora projekt är när de befinner sig på distans. I projektet som IPA administrerar idag finns det två grupper i Sverige och ett i Polen. Med de svenska grupperna fungerar kommunikationen ”face to face” med det polska teamet sker större del av kommunikationen med videokonferenser. IPA anser att videokonferenser har nackdelar. Dessa är att det blir dyrt och uppstår fördröjning i ljud och bild som kan leda till missuppfattningar. Turk et al., (2002) tar upp att nackdelen med videokonferenser nämligen att de är dyra och inte lika effektiva som önskas. För att stärka banden med distansgruppen, bjuder IPA in dem för att arbeta och diskutera eller åker IPA över till dem. Schwaber (2004) rekommenderar att det ska finnas en plan för hur hanteringen av kommunikation ska ske med distansgrupper.

IPC förklarar att kommunikationsvägarna förändras i ett stort projekt. Det behövs någon form av mjukvara när kommunikation och information ska spridas samt fysiska informationsplatser som ska finnas nära grupperna. IPB har inte behövt förändra sättet att kommunicera med gruppen.

Kock (2007) förespråkar synkron kommunikation när kunskap ska förmedlas. Detta ligger i linje med vad IPA använder sig av när han kommunicerar med sina grupper. Att försöka förmedla kunskap asynkront kommer att ta längre tid med större risk för missuppfattning och felolkning av meddelandet (Kock, 2007).

4.2.2. Kommunikation kund (Product Owner)

I frågan om kommunikation med kund förklarar IPA att det uppstår en kommunikationsproblematik. Det som händer är att kravet på dokumentation ökar i samband med projektets storlek och kostnad, detta kan i sin tur hämma produktutvecklingen. Schwaber (2004) menar att en av anledningarna till att agila arbetsmetoder kom till var för att fokus skulle vara på utveckling och inte på dokumentation vilket innebär att det inte ligger i linje med vad som rekommenderas i Scrum. IPA provade att minska på dokumentationen för att fokusera på produkten, men stötte på problem när kunden inte kunde följa arbetet genom dokumentationen.

IPB menar att kommunikation förändras när det kommer till beslutsfattande i de fall Product Owner inte finns tillgänglig. Som alternativt kommunikationsmedel använder IPB asynkront medel som E-post. Enligt IPC blir det inte några förändringar utan det räcker med asynkron kommunikation vilket styrks av Kock (2007) som rekommenderar asynkron kommunikation när det gäller spridning av allmän information till större grupper eftersom det blir effektivare och billigare. Vidare diskuterar Kock (2007) att känslig information såsom etisk, laglig eller personlig information bör hanteras synkront för att minska risken att obehöriga tar del av informationen, vilket IPB upplevde som påfrestande när han behövde använda sig av asynkron kommunikationen för att diskutera ett beslut med kunden.

4.2.3. Kommunikation med företagsledning

Gällande kommunikation med ledning var det endast en projektledare som konstaterade att valet av kommunikationsmedel förändrades. IPC poängterar att även om han arbetar med Scrum fullt ut behövs det någon form av kommunikationskanal mot ledningen. ”*Ta seden dit man kommer och använd de företags kommunikationskanaler, men skära av de kanaler som inte medför ett värde för projektet*”. IPA använder synkron kommunikation med ledningen. Schwaber (2004) argumenterar för att kommunikationen ska ske synkront vilket inte stämmer överens med hur IPC hanterar kommunikationen. IPA tillämpar synkron kommunikation med ledningen och detta ligger i linje med vad Schwaber (2004) rekommenderar hur kommunikationen ska äga rum.

4.2.4. Kommunikation mellan grupper

När det gäller kommunikationen mellan grupper i projektet hävdar samtliga projektledare att det förekommer förändringar beroende på följande faktorer tid och plats. En förlängning av

mötestiden är nödvändig för att alla grupprepresentanter ska ha möjlighet att yttra sig i det här avseendet följer projektledarna i vad som förordas i Scrum. Kommunikationen förändras när deltagarna befinner sig i olika delar av världen och använder videokonferens som kommunikationsmedel mellan team. IPA förklarar att ljud och bild inte alltid är synkroniserat i en videokonferens och risken för oklarheter och missförstånd ökar, vilket kan påverka arbetseffektiviteten i projektet. Projektledarna använde sig av flera andra kommunikationsmedel för att säkerställa informationen. Paasivaara et al., (2008) argumenterar för att grupper ska använda sig av flera kommunikationsmedel när avstånden mellan grupperna blir för stora. Detta stämmer överens med informanterna om hur de använde sig av olika kommunikationsmedel för att upprätthålla nivån på kommunikationen.

4.3. Projektadministration

4.3.1. Möten

IPA anser att det är viktigt med stormöten i stora projekt eftersom det är i dessa möten teammedlemmar skapar förståelse för projektets helhet. Detta sker genom informationsspridning som behandlas i stormöten. IPB berättar att det är *programchefen* som samlar alla projektmedlemmar där programchefen berättar övergripande om deadlines, status på projektet, när produkten ska släppas på marknaden och var projektet befinner sig i förhållande till tidsschemat. IPC har varit rådgivare åt många organisationer och har upplevt varierande kvalitet på deras stormöten. IPC menar att stormöten som tillämpar envägs kommunikation är meningslösa. Istället rekommenderar IPC att stormöten är korta och där deltagarna kan ställa frågor och presentera statusen på sitt delprojekt.

IPA tycker att det kan införas ett extra möte som han kallar för *Krismöte*. Deltagarna i ett krismöte består endast av de berörda personerna oavsett roll eller uppdrag. IPB och IPC rekommenderar att man ska hålla sig till de traditionella Scrum mötena. Tabell 4.6 består av informanternas synpunkter som berör mötesstrukturen inom stora Scrum projekt.

Tabell 4.6 Förändringen inom möten ur intervjupersonernas perspektiv

Förändring i Möten	Intervjupersoner		
	IPA	IPB	IPC
Samtliga involverade personer bör få möjlighet att delta i ett möte	Ja	Ja	Ja
Ändra struktur eller upplägg på Scrum möten	Ja	Nej	Nej

Paasivaara et al.,(2008) framhåller möte som ett viktigt verktyg för informationsspridning och menar att möten bidrar till att öka kompetensen hos projektdeltagarna. Alla informanter anser att det är viktigt med stormöten. IPC menar att stormöten är avgörande i de fall informationen är grunden till schemauppläggning och arbetsplanering. Schwaber (2004) menar att strukturen på ceremonierna i Scrum ska förbli intakt även i stora Scrum projekt. Detta ligger i linje med vad informanterna anser och de motiverar sina svar med att arbetseffektivitet ökar och återkoppling förbättras med oförändrade Scrum ceremonier.

4.3.2. Styrning

IPA: ”Beslutprocessen är oftast Top-Down, även i agila projekt är det Top-Down...” med detta menar IPA att det är en hierarkisk beslutprocess. IPB poängterar att beslutprocessen i projektet är hierarkisk. IPB förklarar att programchefen är den högst ansvariga. Programchefens tidplan bygger på 20 andra tidsplaner. Detta innebär en stor risk för eftersom det räcker att ett delprojekt inte håller sin deadline eller att programchefen får felaktig information av projektledarna för att misslyckas. IPC hävdar att när det blir fler involverade i projektet behövs det en samordning och menar att en hierarkisk beslutprocess är att föredra.

IPA anser att medarbetarna ska utvecklas genom speciella utbildningar som lyfter kunskapsnivån hos dem, men inom en rimlig nivå. Detta innebär att progressen av ett Scrum Team kan försämrats på grund av en medarbetares kunskapsbrist. Det får inte vara en stor brist eftersom det kan bli mer fokus på utbildning än själva uppdraget. IPB har upplevt utbildningar på ett annorlunda sätt. Det är seniorer i ett team som utbildar juniorer i de fall ett Scrum Team består av både seniorer och juniorer. Det är sällan ett Scrum Team enbart består av juniorer. I sådana fall införs en kompetensökande Sprint.

IPC hävdar att det finns olika sätt att lösa kompetensbrist i ett Scrum Team. IPC rekommenderar samma resursutvecklingsprocedur som IPB. Kompetensutveckling kan även ske inom teamet. Teamet ansvarar för att identifiera interna brister och komma med lösningsförslag som oftast leder till att teamet kompletteras.

Både IPA och IPC hävdar med bestämdhet att delegering är nödvändigt när projektledare vill utföra en uppgift genom andra personer. IPA: ”att delegera innebär att man har kontroll.”

IPC menar att coachning och rådgivning tilldelas rollerna inom Scrum. I stora projekt är det organisationerna som väljer att ha en projektledare. En del av projektledarens roll innebär att stödja projektmedlemmar med coachning och rådgivning. IPB tycker att dessa uppgifter ska utföras av de som har mest kunskap inom ett område. IPA förklarar hur coachning och rådgivning utförs av olika roller i stora Scrum projekt. Dessa utförs på grupp- och individnivå.

Tabell 4.7 demonstrerar informanternas perspektiv på vilka värden som ändras. Dessa värden avgör faktorns/projektstyrningens roll, nackdelar och fördelar, viktighetsgrad och potential inom Projektadministration.

Tabell 4.7 Förändringen inom styrning ur intervjupersonernas perspektiv

Förändring i Styrning	Intervjupersoner		
	IPA	IPB	IPC
Hierarkisk beslutprocess	Ja	Ja	Ja
Utveckla resurser via utbildningar	Ja	Nej	Nej
Delegering	Ja	-	Ja
Annan än projektledaren som coach och rådgivare	Ja	Ja	Ja

Kerzner (2009) hävdar att en projektledare ska kunna fatta beslut och att beslutfattandeprocessen snarare är en vardaglig händelse i ett projekt. För de engagerade i projektet sitter projektledaren högst upp i hierarkin. Scrum Master eller Product Owner innehar beslutmakten till en viss grad. Gruppmedlemmarna har beslutmakten inom teamet, de bestämmer på vilket sätt de ska lösa uppgifterna. Under intervjuerna har informanterna lagt fram sina rekommendationer och perspektiv gällande beslutprocessen och samtliga tyckte att beslutprocessen ska vara av en hierarkisk uppbyggnad.

Schwaber (2004) menar att det är positivt med utbildningstillfälle där medverkande i projekt får möjlighet att utveckla sin kompetens för att kunna fullgöra arbetsuppgifter effektivt och ta ansvar för utfört arbete. I praktiken har de flesta av informanterna haft ett annat perspektiv på utbildning. De anser att det är teamet som ska identifiera brister och lägga fram lösningsförslag som oftast innebär en komplettering med någon person som har den nödvändiga kompetensen, alternativt bistår seniorer fåkunniga juniorer.

Delegeringen är projektledarens ansvar, genom att utnyttja arbetarnas färdigheter och förmågor kan högre produktivitet uppnås (Kerzner, 2009). Delegering som uppgift fördelas mellan projektledaren och Scrum Mastern i stora Scrum projekt (Schwaber, 2004). Två av tre informanter menar att delegering är nödvändigt om uppgifter ska utföras genom andra personer, i synnerhet när det gäller ett stort projekt. Svensson & von Otter (2001) anser att en projektledare ska kunna delegera på ett korrekt sätt. Rätt uppgift till rätt person.

Coachning och rådgivning är i första hand projektledarens uppgift (Kerzner, 2009). Att styra ett projekt är att hantera genomförandefasen i ett projekt. Det är i denna fas projektledaren ser till att de aktiviteterna som genererats ur de uppsatta planerna fullbordar målen (Christensen & Kreiner, 1997). Informanternas håller inte med om att projektledaren ensam utför dessa uppgifter, istället fördelas uppgifterna mellan de olika rollerna i Scrum. Scrum Master och Product Owner får var och en sin del av uppgifterna rådgivning och coachning. Projektledaren coachar i sin tur också, dock är målgruppen annorlunda och typen av rådgivningen är bland annat mindre teniskt orienterad.

4.3.3. Befogenhet

IPB och IPC anser att rollernas befogenheter i Scrum inte ska ändras. Varje roll har sin egen befogenhet och rollerna tilldelas i början av projektet. De tycker även att rollernas befogenheter ska vara intakta genom hela projektet.

IPB och IPC menar att konflikter vanligen uppstår mellan likartade roller som befinner sig på antingen olika eller samma nivå i hierarkin som till exempel Product Owner och Chief Product Owner. Det kan uppstå konflikter när projektledare inte är tydlig under stormöte. Detta på grund av att en del medarbetare baserar arbetsstruktur och schemauppläggning på informationen som projektledaren förmedlar i stormöte. Tabell 4.8 innehåller informanternas synpunkter om befogenhetsfrågor inom stora Scrum projekt.

Tabell 4.8 Förändringen inom befogenhet ur intervjupersonernas perspektiv

Förändring i befogenhet	Intervjupersoner		
	IPA	IPB	IPC
Involverade roller tilldelas befogenheter	Nej	Nej	Ja
Justeringar av befogenheter	Nej	Nej	Nej
Konflikter som uppkommer mellan de olika rollerna i projektet	Ja	Ja	Ja

Schwaber (2004) tar upp konflikter som kan uppstå mellan Scrum Masters i stora Scrum projekt. Dessa konflikter kan uppstå när Scrum Masters tycker att de har rätt i en fråga som kan beröra flera Scrum Teams.

Christensen & Kreiner (1997) hävdar att tilldelningen av befogenheter ska ske i de initiala faserna av ett projekt och bör tilldelas en projektledare. IPA anser att befogenheter ska tilldelas av styrgruppen. IPB konstaterar att befogenheter ska tilldelas i början, innan projektet påbörjas. IPC har en annan synpunkt och hävdar att befogenhet ingår i roller. Projektledaren får tilldela befogenheter om det innebär effektivisering av arbetet inom ett uppdrag.

5. SLUTSATSER

5.1. Undersökningsresultat

Huvudsyftet med den här uppsatsen har varit att identifiera vilka konsekvenser det blir vid införandet av agila arbetsmetoden Scrum i stora projekt. Vi har genomfört intervjuer och analyserat empirisk insamlad data och har kommit fram till att det sker förändringar i faktorer inom följande tema:

- *Arbetsprocess*
- *Kommunikation*
- *Projektadministration*

Vi har valt att presentera resultaten i ett ramverk som är uppdelad i tre tabeller, en för varje tema (tabell 5.1 - 5.3). Detta ramverk kan ligga till grund för organisationer som vill införa arbetsmetoden Scrum i stora systemutvecklingsprojekt. Varje rekommendation innebär ett tillämpningsförslag. Dessa tillämpningsförslag beskrivs i korthet under rubriken *Beskrivning*.

5.1.1. Rekommendationer inom temat Arbetsprocess

När Scrum tillämpas i stora systemutvecklingsprojekt krävs det inga stora förändringar på själva arbetsprocessen. Varken litteraturen eller informanterna beskriver några större förändringar i roller, artefakter eller ceremonier. Viktigt att påpeka är att det inte finns en allmän *Best Practice* för hur Scrum ska tillämpas i stora projekt. Arbetsmetoden anpassas till den enskilda organisationen och rådande situationen, vilket bekräftas av litteraturen och informanterna.

Följande förändringar är viktiga att genomföra enligt informanterna: dela upp stora projekt i små delprojekt där varje Scrum Team arbetar med ett delprojekt, arbeta med tvärfunktionella team, komplettera med flera Product Owners eller projektledare som dels arbetar nära teamet men också nära företagsledningen. Utöka antalet möten för att Scrum Team såväl som intressenter ska få en bättre förståelse för produkten och projektet. I tabell 5.1 presenterar vi vilka förändringar som bör utföras inom arbetsprocessen, därefter presenterar vi en beskrivning av de olika rekommendationerna.

Tabell 5.1 Förslag på tillvägagångssätt inom arbetsprocess i stora Scrum projekt

TEMA	FAKTOR	REKOMMENDATION	BESKRIVNING
ARBETSPROCESS	Roller	Behåll sammansättningen i Scrum Teamet	Sträva efter att bevara Scrum Teamets sammansättning oavsett hur stort projektet är eller hur länge det varar. Teamet behöver arbeta ihop sig en längre period för att kunna prestera optimalt.
		Ha inga fördefinierade roller inom teamet	Inom teamet kan det finnas specialister inom ett område, men undvik officiella roller som till exempel testare och databasadministratör.
		Ha mer än en Product Owner	I stora projekt räcker det inte med en Product Owner som arbetar med alla Scrum Team parallellt. Det behövs fler som arbetar med varsitt Scrum Team och en som arbetar nära företagsledningen
	Artefakter	Dela upp Product Backlogen	För att underlätta arbetet för Scrum Teamet bör en Backlog som är anpassad till det enskilda Teamet skapas. Men den huvudsakliga Product Backlogen som gäller för hela projektet bör göras tillgänglig så att samtliga team kan ta del av den och ha en förståelse för vad andra team arbetar med samt skapa en förståelse för projektets helhet
	Ceremonier	Förändra inte Daily Scrum	Låt Daily Scrum förbli ett möte mellan Scrum Master och Scrum Teamet. Förutom Product Ownern ska helst inga andra vara närvarande. Syftet är att teamet ska dela information med varandra och uppdatera varandra på sprintens läge
		Komplettera Sprint Review Meeting	Arbete i delprojekt kräver att, förutom de ordinarie Sprint som hålls inom delprojektet, även utöka med liknande Review Meeting för andra intressenter. Stora projekt involverar många intresserade personer som vill ha information om produkten samt ha tillfälle att ge återkoppling. Ett möte för alla delprojekt bör införas. Detta för att skapa en förståelse för produkten

5.1.2. Rekommendationer inom temat Kommunikation

En av slutsatserna som är gemensam för informanterna är å ena sidan att *synkron* kommunikation är att föredra men å andra sidan var det inte möjligt att bara använda sig av detta på grund av olika faktorer som tid, plats och storlek. Informanterna menade att det uppstod missuppfattningar som kunde försena projektet. En annan slutsats var att kravet på dokumentationen ökade med projektets storlek på bekostnad av produktutvecklingen. Tabell 5.2 ger exempel på de rekommendationer med tillhörande beskrivning som kan används för att förbättra kommunikationen i ett stort projekt.

Tabell 5.2 Rekommendationer för kommunikation

TEMA	FAKTOR	REKOMMENDATION	BESKRIVNING
KOMMUNIKATION	Team	Använd synkron	Synkron kommunikation fungerar bäst när kunskap ska förmedlas. Videokonferenser tillhör synkron kommunikation men upplevs som bristfällig och kan leda till missuppfattningar.
		Använd asynkron för informationsspridning	När information ska spridas till alla projektdeltagare lämpar sig asynkron kommunikation som är billigare och effektivare.
	Kund	Synkron ska användas	Hantering av viktig och känslig information
	Ledning	Använd befintliga kommunikationskanaler	Använd företagets kommunikationskanaler för att minska missförstånd.
	Mellan grupper	Synkron lokalt och distans	Lokalt: Längre mötestid. Distans: Videokonferenser tillhör synkron kommunikation men kan upplevas som bristfälliga och kan leda till missuppfattningar.

5.1.3. Rekommendationer inom temat Projektadministration

Under temat Projektadministration har informanterna samma uppfattning om att beslutprocessen bör vara hierarkiskt uppbyggd. Informanterna använder sig även av *stormöten* för att informera alla medarbetarna om projektets status. Tabell 5.3 illustrerar vilka ändringar och justeringar bör utföras inom temat Projektadministration och på de tillhörande faktorerna, det presenteras även en beskrivning på varje rekommendation i tabellen nedan.

Tabell 5.3 Förslag på tillvägagångssätt inom projektadministration i stora Scrum projekt

TEMA	FAKTOR	REKOMMENDATION	BESKRIVNING
PROJEKTADMINISTRATION	Möten	Införa stormöten	Varannan vecka eller varje månad bör medarbetarna samlas till stormöten. Syftet för detta möte är projektöversikt.
		Möten ska vara intakta	Alla Scrum möten ska hållas intakta i struktur och procedur för att uppnå högre arbetseffektivitet.
	Styrning	Införa en hierarkisk beslutprocess	Beslutprocessen ska helst vara hierarkiskt organiserad, vilket kan reducera konflikter inom ett projekt.
		Undvika att lägga upp speciella utbildningar	Speciella utbildningar kräver tid och pengar. Det rekommenderas att senior personal ska lära upp junior personal.
		Delegera	Delegering är ett måste om projektledare vill utföra uppgifter genom andra personer. Att delegera är att ha kontroll över projekt.
		Dela upp coaching och rådgivning på andra roller	Projektledarens uppgifter delas upp bland Scrum rollerna, vilket kan harmonisera styrningsflödet i stora Scrum projekt.
	Befogenhet	Undvika att tilldela extra befogenheter	Extra befogenheter som tilldelas efter att projektet påbörjats komplicerar rollerens beteende i ett projekt.
		Undvik justeringar på befogenheter	Förändringar i befogenheter kan ställa till med konflikter och komplikationer med personer som har likartade roller.
		Var beredd på att lösa konflikter	Konflikter uppstår vanligen mellan likartade roller som befinner sig på antingen samma eller olika nivåer i hierarkin.

5.2. Kritisk diskussion

Det har visat sig vara svårt att hitta ett passande ramverk till vår studie. Vi sökte först efter artiklar som behandlade ramverk för stora agila projekt men det fanns inget sådant ramverk. Därefter fortsatte vi söka igenom flera vetenskapliga artiklar som föreslår ett ramverk för projektledning ur ett BPR (Business Process Redesign) perspektiv. Sökningen gav oss inget positivt resultat. Således, valde vi att skapa ett eget ramverk som är anpassat för vår uppsats.

Vårt ramverk har varit till stöd genom hela undersökningen, ramverkets riktlinjer har följts i intervjuerna trots att vi har haft öppna frågor vilket är en fördel. Ett möjligt problem i vårt ramverk är att det inte har granskats i större eller flera sammanhang av forskare.

Vi valde att använda tre informanter vilket kan anses vara en brist, men de informanter vi valde besitter djupa kunskaper och täckte gott och väl vår frågeställning och detta gav oss möjligheten att gå in på djupet av ämnet.

För att beskriva Scrum har vi till största delen använt upphovsmännens litteratur. Detta medför å ena sidan en exakt beskrivning av arbetsmetoden Scrum men å andra sidan föreligger det en avsaknad av kritik mot densamma. Införandet av Scrum i stora projekt är ett relativt nytt fenomen och utbudet av vetenskapliga artiklar i ämnet är begränsat. Vi har fått använda oss av empiriska studier om hur organisationer har gått tillväga för att införa Scrum i stora projekt.

5.3. Slutord

Scrum har tagit systemutveckling till en ny nivå genom att fokus ligger exklusivt på kund och produkt. Arbetsmetoden Scrum har visat sig vara framgångsrik i små systemutvecklingsprojekt. Detta ligger till grund för organisationer som väljer att införa arbetsmetoden Scrum i stora projekt. Scrum tar det bästa ur föregångarnas systemutvecklingsmetoder vilket får Scrum att framstå som en briljant utvecklingsmetod som utmärker sig inom systemutveckling. Rollerna i Scrum är produkt- och funktionsorienterade vilket innebär att eftersträvas i Scrum att involvera utvecklare med hög kompetens och flera års erfarenheter. Scrum ställer högre krav på projektdeltagarna. Dessa blir i sin tur tilldelade roller som är exklusivt anpassade till Scrum.

I Scrum har utvecklingen inom ämnet projektledning försummats vilket kan försämra Scrums rykte inom systemutvecklingsbranschen. Detta kan anses vara en *stor* brist i det fallet då rollen *projektledare* förblir enda lösningen i sammanhanget *Scrum i stora systemutvecklingsprojekt*. Därför skulle en evolution inom området projektledning inte vara helt fel i nuläget.

Bilaga 1 – Agile Manifesto

Mellan den 11:e och den 13:e februari träffades 17 personer i Utah, USA. Dessa personer var representanter för en rad olika utvecklingsmetoder som till exempel Scrum, Extreme Programming (XP), Adaptive Software Development och Chrystal. Alla dessa metoder uppkom som ett alternativ till de traditionella, dokumentstyrda, utvecklingsmetoder som fanns, bland annat livscykelmodellen eller vattenfallsmodellen. Det som framkom var The Agile Development Manifesto, en deklARATION som undertecknades av alla deltagare (Agile manifesto, 2001). Under mötet kom man fram till 12 principer som det agila manifestet står för (Agile manifesto, 2001):

- Vår högsta prioritet är att tillfredsställa kunden genom tidig och kontinuerlig leverans av värdefull programvara.
- Välkomna förändrade krav, även sent under utvecklingen. Agila metoder utnyttjar förändring till kundens konkurrensfördel.
- Leverera fungerande programvara ofta, med ett par veckors till ett par månaders mellanrum, ju oftare desto bättre.
- Verksamhetskunniga och utvecklare måste arbeta tillsammans dagligen under hela projektet.
- Bygg projekt kring motiverade individer. Ge dem den miljö och det stöd de behöver, och lita på att de får jobbet gjort.
- Kommunikation ansikte mot ansikte är det bästa och effektivaste sättet att förmedla information, både till och inom utvecklingsteamet.
- Fungerande programvara är främsta måttet på framsteg.
- Agila metoder verkar för uthållighet. Sponsorer, utvecklare och användare skall kunna hålla jämn utvecklingstakt under obegränsad tid.
- Kontinuerlig uppmärksamhet på förstklassig teknik och bra design stärker anpassningsförmågan.
- Enkelhet – konsten att maximera mängden arbete som inte görs – är grundläggande.
- Bäst arkitektur, krav och design växer fram med självorganiserande team.
- Med jämna mellanrum reflekterar teamet över hur det kan bli mer effektivt och justerar sitt beteende därefter.

Deltagarna på mötet kom att kalla sig ”The Agile Alliance”. Sedan mötet har agila systemutvecklingsmetoder blivit ett väldigt populärt ämne inom organisationer som arbetar med utveckling av mjukvara (Kosh, 2005). The Agile Alliance har växt och blivit de agila metodernas ”röst”. De håller bland annat i föreläsningar och arrangerar evenemang.

Bilaga 2 – Intervjufrågorna

Med dessa frågor har vi täckt in det vi vill undersöka och vi kommer att använda frågorna efter behov i intervjuerna eftersom de första frågorna i varje del är väldigt öppna förväntar vi oss att få ut den information som är relevant de andra frågorna har vi som ett stöd att ställa beroende vilket håll intervjun.

TEMA	INTERVJUFRÅGOR
Inledande frågor (N)	<p>N1. Vad har du för utbildning?</p> <p>N2. Hur många år har du arbetat som projektledare?</p> <p>N3. Har du varit projektledare i traditionella projekt innan du började arbeta agilt?</p> <p>N4. Ungefär antal projekt du har varit projektledare på?</p> <p>N5. Vad är ett stort projekt enligt dig? Antal personer och antal miljoner?</p> <p>N6. Anser du att det går att införa scrum helt i stora projekt?</p> <p>N7. Använder ni scrum fullt ut i stora projekt?</p> <p>N8. Hur använder ni er av scrum i stora projekt?</p> <p>N9. Använder ni er av scrum of Scrums i stora projekt?</p> <p>N10. Hur ser infrastrukturen ut i ett stort projekt?</p>
Kommunikation (K)	<p>K1. Vilka kommunikationsmedier använder ni för att informera?</p> <p>K2. Grupp : Fördelar med de valda medierna</p> <p>K3. Grupp : Nackdelar med de valda medierna</p> <p>K4. Kund : Fördelar med de valda medierna</p> <p>K5. Kund : Nackdelar med de valda medierna</p> <p>K6. Ledning : Fördelar med de valda medierna</p> <p>K7. Ledning : Nackdelar med de valda medierna</p> <p>K8. Hur sker kommunikationen mellan Scrum Teamen, nackdelar och fördelar?</p> <p>K9. Kan samma information tolkas olika beroende på vem som förmedlar?</p> <p>K10. Hur gör man för att minska missförståndet av information?</p> <p>K11. Är det något du vill tillägga?</p>
Möten (M)	<p>M1. Har ni stormöten med alla som är med i projektet?</p> <p>M2. Hur fungerar dessa stormöten?</p> <p>M3. Håller ni möten enligt Scrum metoden, om inte, vilka förändringar införs?</p> <p>M4. Är det något du vill tillägga?</p>

Styrning (S)	<p>S1. Hur går beslutsprocessen till? S1.a. Blir beslutsprocessen mer hierarkisk i ett stort projekt? S1.b. För-/nackdelar med processen</p> <p>S2. Hur hanterar man deltagare som är ointresserade av att dela med sig av kunskap?</p> <p>S3. Scrum ställer höga krav på projektmedlemmarnas kompetens hur löser ni det i stora projekt?</p> <p>S4. Krävs det någon speciell utbildning? om ja, vilka roller behöver utbildning ?</p> <p>S5. Hur mycket kan man delegera utan att förlora kontrollen över projektet ?</p> <p>S6. Hur sker coachningen och rådgivning av projektmedlemmarna?</p> <p>S7. Hur håller man människor utan befogenheter men med egenintresse borta från projektet?</p> <p>S8. Hur gör man för att undvika konkurrens mellan teamen?</p> <p>S9. Vilka samordningar sker mellan uppgift och medarbetare för att nå bästa resultat?</p> <p>S10. Är det något du vill tillägga?</p>
Befogenhet (B)	<p>B1. Hur tilldelas befogenheter till roller?</p> <p>B2. Hur sker justeringarna på befogenheter?</p> <p>B3. Vilka konflikter kontra roller kan uppkomma?</p> <p>B4. Är det något du vill tillägga?</p>
Arbetsprocess (R)	<p>R1. Har ni samma team under hela processen eller ändras sammansättningen?</p> <p>R2. Har projektmedlemmarna fasta roller (enligt scrum ska sådana inte finnas)?</p> <p>R3. Konkreta exempel på skillnader i systemutvecklingen när ni arbetar i stora jämfört med små Scrum-projekt?</p> <p>R4. Planerar ni en sprint i taget eller flera (en stor sprint eller flera små)?</p> <p>R5. Hur fungerar det att få alla att fylla i Sprint Backlogen?</p> <p>R6. Lyckas ni hålla tidsramen för sprinten (30 dagar enligt Scrum)?</p> <p>R7. Genomförs sprint retrospektivt (går igenom de erfarenheter som har gjorts och de lärdomar som har dragits) med alla projektdeltagare?</p> <p>R8. Är det något du vill tillägga?</p>

Bilaga 3 – Intervju A – IPA

Inledande Frågor

N1

IPA: Jag är magister i informatik det är en fyraårig utbildning och min första tre år gick jag på systemvetenskaplig i Växjö och det sista året gick jag på Göteborgs Universitet

N2

IPA: Jag har arbetat som projektledare i 12 år

N3

Ja det agila kom ju runt 2000 agil programvaruutveckling ska man nog säga inte agil projektledning vilket är ganska nytt och jag har arbetat knappa 10 år med agila metoder då och de sista 5 åren har varit med scrum och glidit in i agil projektledning och idag kallar jag mig agil projektledare

N4

IPA: Ungefär ett om året är väl realistiskt att gissa på sen har det varit många små i början som teknisk projektledare där det är kortare cykler, som administrativ projektledare brukar det vara 1 till 2 år är projekten så säg att jag varit projektledare i ett 20 tal projekt det betyder väl att jag varit projektledare för många små projekt i början och nu på senare tid lite större projekt de stora projekten är på 1 år och de små där under.

N5

IPA: När det gäller tid så är ett stort projekt större än ett år och ett litet är då mindre än ett år det har ju då med komplexiteten att göra är det större än ett år så är det så pass svårt så att det tar lite längre tid. När det gäller pengar så är det ungefär 15 miljoner tycker jag att projekten börjar bli stora. Och alla projekt jag kör nu är över 15 miljoner. Jag tycker ett år är en bra skiljelinje för stora projekt i USA brukar man säga 1 miljon dollar men då blir det 6-7 miljoner kr.

SR: Antal personer?

IPA: När det blir fler än tio tycker jag att det börja bli stort för då behöver man skala upp med nivåer då, mer än 14 börjar det bli nivåer och hierarkier i projekt men 10 personer på ett år då är det stort eftersom varje person kostar 1,5 miljoner under ett år räknar man på

N6

IPA: Nej det går inte för när det börjar handla om mycket pengar så blir kontrollen och intresset mycket större och mer fokusering på en finansiell uppföljning som är något svår att övertyga intressenterna som betalar. Att köra rent agilt det blir så mycket pengar. Om någon investerar 20 miljoner så vågar de inte lita på att det blir vad det blir men det blir iallafall något. Om det är mindre pengar så tror jag att man kan ha den acceptansen. Försök låna 20 miljoner på en bank och säg att du ska köra agilt och de kommer inte att ge dig pengarna och lite så är det med stora projekt det är väldigt svårt att köra helt agilt utan då får man wrappa runt med traditionell metodik runt detta

N10

IPA: Du har dels det som är scrum, Scrum Teamet och du har ett krossfunktionellt team som sköter både utveckling och test så det blir en enhet i projektorganisationen då är det utvecklingsprocessen där test och utveckling ingår det kan vara ett team på 7-10 personer, men 5 kanske är det bästa. Sen har du andra roller som man använder för att klä scrum, klä utvecklingsprocessen inte scrum. och då har man valt scrum som utvecklingsprocess och dessa roller kan vara driftsättningspersonal, det kan vara projektkontroller som externa inte som intressenter.

N9

IPA: Ja absolut när det blir för många utvecklare får man göra grupperna så pass intressanta och små så att det blir effektivt, säg är det bara en jättescrum så tjänar man inte så mycket på det, hälften av teamet är inte intresserade av vad den andra hälften gör och det blir inte heller relevanta diskussioner på exempel daily standups. I det nuvarande projektet så kör jag 2 Scrum Team i Sverige och 1 i Polen faktiskt i samma projekt. Och för att få Scrum Teamen att kommunicera med varandra har vi scrum of scrums. I scrum berör man och hanterar saker som Scrum Teamet kan hantera men om leveransen är något större om man har delar som interagerar eller kräver integration med varandra så har man ett Scrum Team som t ex hanterar kartfunktionen och ett annat team som hanterar kommunikationen med telekom nätet och ett annat team hanterar databas och rapporter och dom bör ibland, oftast så klarar dom sig i sin grupp ett bra tag men ibland när de ska interagera och integrera med andra komponenter inom samma projekt då måste man, då krävs det en dialog. Som scrum of scrums kan hantera väldigt bra.

Kommunikation

K1,K2,K3

IPA: Jag använder det bästa

SR: och det är

IPA: Det bästa är ju alltid den mänskliga dialogen, med dagliga möten det man kallar face to face. Det är inte det du och jag gör nu då utan det är när du kan läsa även om mina ben gungar och ger uttryck för någon form av nervositet, när man kan läsa hela människan och sättet man säger saker inte bara vad man säger utan hur man säger det. Det sättet använder jag mest då. Det kan jag inte göra så mycket med mina distans scrums som befinner sig i polen i detta projekt utan då får man försöka använda det näst bästa eller alternativ där under men det absolut sämsta är något som har väldigt lite bandbredd vad det nu skulle vara, om man nu förlöjligar det så skulle det kunna vara sms men det skulle man aldrig tänka sig att kommunicera med i projektet, men som minst så försöker vi ha sådana här möten som vi har nu jag och du (videokonferens) där man använder ljud och bild och delar på gemensamma resurser som dokument

SR: Finns det någon nackdel att använda sig av videokonferens?

IPA: Ja det tror jag folk är väldigt ovana, i alla fall om man tittar på dom blir ju mer och mer vana men jag som anser mig vara IT-proffs och väldigt modern tycker att det är lite obekvämt med videokonferens det har väl med ålder att göra kanske, det kanske är lättare för de yngre som växer upp med det. Det finns en fördröjning i videokonferens som kan vara svår man får inte direkt feedback exempelvis en fiffig kommentar den kommer så sent att den kanske bara blir ett störmoment istället i alla fall så som tekniken är idag men det blir bättre och bättre, de små nickningarna eller bekräftelserna som man behöver för att känna sig levande i dialogen missas där så det är en nackdel. Känner man varandra väl kan man hantera det men känner man inte varandra måste man investera ganska mycket i väldigt rika kommunikationsmedia innan man kan gå ner till videokonferens. Jag har ju mina team som får komma och hälsa på mig då och då mest för att jobba bara för att vara på plats och får acceptans för sitt sätt att vara och sedan kan de åka till sina siter och jobba om det nu är Polen eller USA eller så åker jag över och

K1,K4,K5

IPA: Normalt sett är det en styrgrupp som bestämmer ungefär som styrelsen bestämmer i ett företag så bestämmer styrgruppen i ett projekt och styrgruppen tillsätts av ledningen. Ett väldigt kraftfullt sätt är att faktiskt säga till styrgruppen ”så här långt har vi kommit” och så visar man en fungerande demo, nu är inte alla intresserade av det men oftast är styrgruppen under tidspress i stora företag kanske styrgruppen hanterar flera hundra projekt och dom vill egentligen att saker ska gå snabbt på presentationen, du har 15 minuter att presentera projektets problem och har du inga problem så blir de besvikna för att du tagit upp deras tid. Styrgruppen är oftast intresserad av pengar och progress i form av leverabler och i progressen, pengarna kan man säga att så här mycket har vi spenderat. Vi har planerat att spendera 10 miljoner och vi har spenderat 10 miljoner och för detta har vi gjort följande så kan man visa en kort demonstration, det är jättekraftfullt och det är de inte vana vid styrgruppen oftast är de vana vid att få se papperspresentationer och sedan när allt ska produktionssättas så hör de om problem och bland den en positiv möjlig produktionssättning. Jag tycker det är jättekraftfullt Någon gång har jag använt en demo för att visa progressen och det är egentligen inte riktigt legalt, men den gången jag gjorde det var det lite häftigt, styrgruppen uppskattade, men det var inte riktigt ok egentligen då. Som projektledare har du egentligen ingen riktig makt det har styrgruppen.

K1,K6,K7

IPA: Kommunikationen med Product Owner?

SR: mm

IPA: Ja du

SR: i scrum så ska Product Ownern vara med dagligen på möten mm

IPA: Ja det där är extremt intressant.

SR: Hur kommunicerar man med Product Ownern om han inte kan vara med hela tiden och för att informationen inte ska misstolkas.

IPA: Det här är väldigt intressant om man tittar på det projektet vi arbetar med just nu, det utspelar sig i en så ganska traditionell projektmetodik

SR: mm

IPA: Där beställaren oftast har blivit övertygad genom att man dokumenterar det så man jobbar väldigt mycket med krav och olika typer av dokument olika artefakter och i ett av dom projekten så har jag valt att inte göra så och stötta på en hel del problem då i det sättet. Problemet är att det är så mycket pengar som ska släppas till projektet för att göra viss leverabler så är det en del som känner sig maktlösa man vill gärna säga ok när man ser det man ska få och det är man van att få via specar

SR: mm

IPA: Så att det där är svårt, du går inte runt och säger till någon lita på mig det blir bra för det finns ingen som litar på dig när det handlar om mycket pengar och så är det i den här miljön folk litar säkert men det räcker inte för att övertyga Product Ownern vill ha en viss nivå på dokumentationen, så vi har gått halvvägs vi tycker det är ok vi gör en del av dokumentationen tillräckligt mycket för att övertyga men väldigt mycket med bilder inte så mycket med formell kravhantering där man skriver ex systemet ska ha en hand när man hovrar över en länk som är klickbar med andra ord detaljerat. Utan vi säger att det ska bete sig på ett visst sätt man säger beteenden ganska övergripande sätt. Och så ger man bilder att så här kan det se ut och sedan säger man att vi kan detaljera den mer men då är vi faktiskt inne i programvaruutveckling där vi antingen kan spendera pengarna i pappersformat eller göra det i kodformat där i den dialogen så, jag finner den lite svår ett av projekten för att jag inte är riktigt klar över ifall Product Ownern förstår även om jag försöker förklara så är jag inte så säker på om han förstår. Men det är en pedagogisk uppgift jag tror ju på agila projektmetodik där man ska kommunicera i en dialog om vad som kommer att ske, vad dom kommer att få och sedan visa det med fungerande programvara då har vi gjort så här, det här med sprintdemo övertygar en hel del när dom får se vad det är dom får, så vi har försökt göra en liten shortcut där vi gjort lite dokumentation och sedan så får ni det här ungefär (programvaran) ett ganska snabbt sätt att hoppa in i demonstrationerna men inte hålla sprintdemo efter varje sprint utan hålla dem kontinuerligt i stort sett i varje veckobygge vi bygger oftare än en vecka men vissa veckobyggen lägger vi

ut i en viss miljö där Product Ownern har möjlighet att säga ”det här behöver du inte specas” säger Product Ownern vilket innebär att Product Ownern har fått det som inte behövde specas och då ser Product Ownern vinsten i att inte behöva överspeca, tricket var att försöka visa inte genom trust me utan göra det innan de hann reagera på vissa enkla saker, här har ni lösningen men den är inte färdig men ni kan ge synpunkter på hur ni vill ha det och vi kan dokumentera det i papper och göra pappersprototyper eller så kan vi få dom i en form av backlogg och göra kodning så tricket i det här traditionella projektet var att försöka hoppa in i en sprint-demo lik process och det tror jag fungerar då. Kommunikationen har då varit att visa att så här kommer det att se ut titta det är nästan klart.

K8

IPA: Teamen är egentligen inte intresserade av vad andra Team medlemmar i andra team gör men dom är oftast intresserade vad produkten av ett team blir, det vill säga om jag ska integrera med navigationsutrustning så är inte jag som Team medlem intresserad av en viss detalj i navigationsscrummen utan jag är intresserad av det som är interfacet mot den leverablen så vissa gånger har jag gjort misstag som projektledare att låta Scrum Teamen då ibland mötas och berätta vad de gör i detaljer round the table eller stand up för att man tror att det ska öka förståelsen för helhetslösningen, men det har varit väldigt negativ feedback på det så att de är egentligen inte intresserade vad den enskilde projektmedlemmen gör men dom är intresserade av leverablerna

SR: och leverabler är?

IPA: Vad leverabler är för något?

SR: mm

IPA: Leverabler är ju det som levereras resultatet av en uppgift på olika nivåer på högsta nivån så är det scoopet(vad ämnar projektet göra) och sedan bryter man ner det i underleverabler så en leverabel kan vara ett interface på lägre nivå då. Något man levererar hela eller delar av en produkt.

K9

IPA: Oja, helt klart det är ju det behöver inte sägas mer än ja definitivt.

K10

IPA: Det finns ju en potentiell risk där men för mig är det klinkande klart var den avgränsningen går så jag har inga problem med det då.

SR: Ok hur sköter du det?

IPA: En scrummaster han sköter sitt scrumteam. Det börjar med sprintplanering och sedan så de en sprint och där är det scrummastern kung i sitt scrumteam. Men projektledaren sköter hela projektet.

SR: Så under den här sprinten får inte du komma in överhuvudtaget menar du?

IPA: Nej så fungerar det inte, man är fortfarande lika välkommen som vilken annan människa som helst i scrum prata och dricka kaffe, men jag har ingen rätt att modifiera sprintmålen, men jag har fortfarande rätt att intressera mig för hur det går och om det finns några risker för det hanterar jag, om individer vill vara lediga för det hanterar jag också. Scrummasterns ansvar är att utvecklingsprocessen går vidare sen har projektledaren ett större ansvar för hela projektet och därav delaktig så gränserna är tydliga.

K11

IPA: Jag tycker man ska säga att det här med agila metoder är väldigt duktiga på kommunikation där man kommunicerar leverablerna= resultatet på ett bättre sätt och att det finns en kultur där man försöker leverera färdigt en tjänst innan man börjar på nästa istället för att börja med alla tjänster parallellt där det finns en risk för att det inte blir en leverans alls. Jag tycker om det här sättet man kommunicerar framdriften väldigt bra och så finns det vissa principer som när det gäller att skala organisationen som jag tycker agila processer är duktiga på.

Sen finns det en hel del vetenskap om olika medier och generellt så säger man att det med minst bandbredd är sämst och det som har mest bandbredd för människan är det visuella sen hörsel, så kan man både se och höra så är det bra och kan man även känna så är det bättre och känna kan vara att visa en färdig funktion.

Möten

M1

IPA: Ja vi ganska traditionella projektmöten. Teammedlemmarna är väl medvetna om vad som gäller för scrumteamet vem som gör vad på daglig basis, men oftast så saknar de information om vad som sker runt omkring och de uppskattar att få veta det. Så vad som händer på projektnivå eller andra nivåer beror på hur man har skalat initiativet det kan vara programnivå eller portföljnivå som ligger högst upp då som alla andra är de väldigt intresserade av att försöka förstå sin tillvaro och då ger projektmöten eller stormöten som du kallar det. Projektmötet tillgodoser en del av det behovet av information.

M3

IPA: Ja, Vi kör då ganska traditionellt med scrumprocessen, med sprintplanering, själva sprinten och sedan demo och retrospective och sen tillbaka in då.

SR: har du som projektledare daily scrum med dina Scrum Masters

IPA: Nej det har vi inte utan vi har scrum of scrums som är samma karaktär eller typ som daily scrums men dom är inte daily utan dom är 2 ggr i veckan ex tisdag torsdag och det betyder att då så går man runt så får scrummastern säga vad dom gjorde sist, vad dom ska göra framöver och om dom har några hinder som dom behöver hjälp av mig för att lösa, så det är ett sätt att skala det här. Dom kan ju självklart ta kontakter med varandra vissa uppgifter kan vara att de ska integrera med varandra och då är det inget dom behöver ta upp det på scrum of scrum utan det hanteras bara i utvecklingen mellan utvecklarna och mellan teamen. Men om det är något annat då som ledighet, reser, byggservrar eller testmiljöer sånt som dom behöver hjälp att hantera. Så det är lämpligt med 2 möten i veckan med scrummasters fast det styrs mycket beroende på hur stort projektet är och man kan ha fler möten som vem som helst får kalla till med dessa möten brukar vara krismöten och det händer ibland att det behöver kallas till extramöten beroende på var alla befinner sig så kan man använda livemeeting ett nätmöte det brukar vara allvarliga saker och oftast är det saker som inte kan vänta en dag

M4

IPA: Ja det är ju att det finns externa beroenden på miljö då och hur hela projektet sköts man kan ju köra ett traditionellt projekt med en scrum utvecklingsprocess så finns det olika typer av mötesbehov då. Och det är ju inte helt enkelt att hålla sig till ren scrum och det beror på hur organisationen ser ut förövrigt

SR: Har storleken på projektet betydelse?

IPA: Nej det är nog främst till det habitat den miljö som projektet ska utföras så finns det möjligheter då och det betyder då egentligen att möjlighet och begränsningar du kan inte köra väldigt agilt om miljön inte är det ex vissa system på kärnkraftverket kan inte annat än att köras traditionellt så det har inte med storlek att göra utan mer med organisationens signatur eller hur organisationen är i sig. Om den är platt, hierarkisk, matris eller projektorganisation så får man anpassa sin utvecklingsprocess till det först och främst det vill säga att man får ta seden dit man kommer det går inte att köra scrum i alla miljöer på samma sätt utan det kräver en viss förståelse för miljön, det ställer höga krav på att man gör sitt yttersta för att förstå organisationskulturen och hur det fungerar för att ha en möjlighet att göra ett bra projekt. Det tycker jag ibland att man missar om man är alldeles för ny med scrum man blir för religiös så tror man att oberoende på kultur så kan man bete sig hur som helst. Bara man säger att man får ta seden dit man kommer så förstår alla. Det betyder samma sak då (when in rome do as roman do)

Styrning

S1,S1.A,S1.B

IPA: Beslutsprocessen är oftast top-down även i agila projekt är det top-down även om leverablerna sker längst ner så sker styrningen uppifrån sedan är det självklart starka återkopplingar från grupperna nere och upp så försöker man ta hänsyn till det för att få en snabb styrning. Låt oss säga när saker blir väldigt stora som t ex vissa program (steget över projekt) portfölj (steget över program) Astra kan ha en portfölj eller två och i portföljen har de ett 10-tal program som genererar 100-talet projekt och i ett av de här projekten kan det finnas scrumteam det är så man skalar hierarkiskt och det är inte säkert att alla projekt är agila det kan vara väldigt traditionella i samma program men syftet med att göra något överhuvudtaget är att förändra ett nuläge till ett bättre läge helt enkelt ämnar man med ett projekt att göra något till det bättre. Så beslutsvägen uppåt och neråt, men så säger man så här där jobbet sker låt säga i utvecklingsprocessen där ska man inte lägga några hinder i form att jag ska styra den som är mest kompetent att utföra sitt jobb om jag ska styra den personen kommer jag att styra fel, man försöker inte mikromanagement för det blir fel utan man låter den personen som vet hur jobbet ska utföras också styr den delen. Där är då scrum den lägsta nivån där man låter de här bubblorna existera under 2-3 veckor och styrningen måste bli hierarkisk när projektet blir stort även om organisationen är platt så blir det hierarki i den platta organisationen så stora. Jag försöker minnas om man har skalat projekt på ett annorlunda sätt men jag kan inte minnas att jag varit med ett projekt som har skalats på ett annat sätt än hierarkiskt.

S2

IPA: Man skickar ut dom

SR: rakt av?

IPA: Japp rakt av

S3,S4

IPA: Mm det är nästan så att scrum i vissa fall inte kräver det utan det finns en stor acceptans för folk som inte är up to speed ja då säger man att man ska kompetensutveckla dem med parprogrammering osv . alla måste ju förstå vad scrum är innan de kan köra scrum och har de aldrig gjort det så är det väldigt dåliga resurser att ha för då måste man lära upp dem i det här agila och det är jobbigt för det handlar om att förändra ett tankesätt, sätt att arbeta och det kan ta många projekt innan det fungerar så den kompetens man vill ha är att de arbetat agilt eller förstår det agila och sedan i form av teknik kompetens det kan man alltid bygga upp. Fast man vill inte ha sådana individer utan man försöker att få individer som kräver så lite som möjligt fast det är problem i stora projekt för det är fört många inblandade och oftast får du ta det som finns sen får du hantera det och utveckla dina resurser så oftast brukar man börja med en liten scrumkurs i början av projektet där man bjuder in någon celebritet som får berätta om scrum eller utbilda alla deltagare, det går ju ganska snabbt att bli certifierad scrummaster. Det är lättare att bli certifierad scrummaster än att gör en 5-kamp på Liseberg eller nästan lika svårt då så det är väldigt lätt att utbilda folk i scrum men om de tar till sig scrum det är en annan fråga. Det är lätt att förstå vilka moment som ingår men att leva scrum är inte lätt och det är problemet med scrum

S5

IPA: Man ska delegera det innebär att man har kontroll. Man delegerar en väldigt tydlig uppgift som ska utföras på en tydlig tid med väldigt tydliga resurser sen begär du helt enkelt återkoppling . delegering är A och O om du ska utföra något som du personligen inte kommer att utföra utan det kommer utföras av någon annan så du gör beställningen så det handlar egentligen att ha kontroll men då får man vara tydlig i sin delegering vad det är som ska uppnås under vilken tid, med vilka resurser och när vill du ha respons tillbaka i Scrum delegerar också Product Ownern saker han vill ha utförda med vilken prioritering och hur lång tid det ska ta men han frågar teamet ”hur lång tid tror ni det tar” det är ett sätt att hantera komplex delegering – eftersom ni (scrumteamet) ska utföra det så säg hur lång tid det tar att utföra det. Så du kan delegera allt och ändå ha kontroll bara du är tydlig. Är man dålig på att delegera så tappar man snarare kontroll då kommer folk att göra det de själva finner bäst med det de tror att de ska göra.

S6

IPA: Jag ger alltid feedback till mina projektmedlemmar och det betyder att jag pratar med de om beteenden som är bra eller dåliga. Om det är bra så fortsätter jag att be dem fortsätta och förstärka det och om det är dåliga beteende så ber jag dem vara medvetna om de, det är inte alltid enkelt att förändra ett beteende men jag ber dem vara medvetna om det. Jag tycker det fungerar ganska bra men det handlar inte om agilt utan alla är i behov av att få lite synpunkter på hur man uppfattas för det är väldigt svårt att veta hur man uppfattas man ser ju sig själv inte på det sättet utan det är oftast andra som gör det. Jag arbetar med feedback med mina medarbetare även scrum har ju den feedbacken men den sker ju i grupp på retrospectiven men det finns även feedbackmodeller på individer som man ska göra

S8

IPA: Konkurrens är bra, konkurrensen är god, mellan projekt innebär det där gruppen vill bevisa att de är duktiga och kanske bättre än andra det tycker jag är ganska sunt det är lite tävlingsmoment i det, den kan ju bli negativ om dom skulle börja förstöra för varandra men det är väldigt sällan det händer. Jag tycker konkurrens är bra men det finns en viss konkurrens som handlar om jobb security som kan vara svår är du inte riktigt effektiv så kommer någon annan se till att du inte finns eller kan agera så det är väldigt nyttigt med en viss mängd konkurrens och ibland ha den flåsandes men inte för ofta då kan den bara bli nedslitande

S10

IPA: Nej.

Befogenhet

B1

IPA: Projektledarens befogenheter delas ut av styrgruppen. Projektledaren får befogenheter att leverera projektet. Projektledaren i sin tur delegerar uppgifter till bl.a. scrum master.

B2.

IPA: Det sker inte mycket av justeringar rörande befogenheter. De sätts upp som en del av det ramverk projektet skall verka i.

B3.

IPA: Konflikter kan uppkomma på många sätt. Oftast sker det missförstånd rörande vilken roll Linjen kontra Projektet har. Inom ett projektteam finns det inte väldigt mycket utrymme för konflikter. De konflikter som uppkommer får lösas.

B4.

IPA: nej.

Arbetsprocess

R1

IPA: Man försöker ha samma team faktiskt, det är målet men oftast brukar någon bytas ut men man försöker ha samma team

R2

IPA: Vissa har det fast inte majoriteten, det finns ju de som är väldigt duktiga då en specialistroll där det inte finns utrymme eller behov heller av kompetensspridning hos alla för att det är en så liten specialiserad del skulle kunna innebära någon expertkompetens för en viss databas som man vet att man använder 1 eller 2 ggr om året då har man den rollen. Man försöker ha det brett så att alla kan ta varandras uppgift, men en del är mer intresserade än andra och då skapas en expertroll

R3

IPA: Ja stora projekt är lite tyngre dokumentmässiga men det har att göra med att det oftast är mer pengar involverade. I små projekt kanske man mer är intresserade av att lära sig något av projektet. Men om det blir säg 100 miljoners investering i ett projekt så kan det vara jobbigt att inte få ut något av det. Friheten blir mindre desto större projektet blir

R4

IPA: Det beror på hur bråttom man har då när man har som mest bråttom planerar man en sprint i taget men alla sprintar grovt. Men har man inte bråttom så kan man planera alla sprintar

R5

IPA: Det fungerar lite olika det är väldigt projektspecifikt och individspecifikt. Det finns egentligen bara ett sätt och det är att följa upp loggen och de som inte fyller i får förklara varför de inte fyller i den och sedan får man säga till dem att fylla i loggen nästa gång och fortsätter det så är det ett felaktigt beteende som man får försöka korrigera och fortsätter det så får de ta konsekvenserna av sitt felaktiga beteende

R6

IPA: Oftast inte då 9 fall av 10 så går man över det är sällan man går under för det finns alltid saker man behöver göra som ex fler test som gör att man inte kommer under på så sätt kan man säga att scrum tenderar att utnyttja den tid som finns för det finns en stor frihetsgrad för utvecklaren att bestämma vad som skall göras och den uppgiften utförs jag tror iofs att det är bra att göra så de dom gör extra brukar vara fördelaktigt

R7

IPA: Nej bara med sprintdeltagarna

R8

IPA: Nej

Bilaga 4 – Intervju B – IPB

Inledande frågor

N1

IPB: Jag har en bachelor of science inom datateknik och master of science inom business technology

N2

IPB: 2,5 år inom ledarrollen men då hade jag inte den roll jag har idag

N3

IPB: Jag var inte formellt sett projektledare på mitt gamla jobb hade ett mer tekniskt ansvar mer åt systemarkitektur inte leda personer på det sättet utan det var ett mer tekniskt ansvar.

N4

IPB: Det är ett projekt och det är det projektet jag är på nu och det har varit igång sedan mars 2008

N5

IPB: Är det mer än 50 personer anser jag det vara ett stort projekt

N6

IPB: Definitivt men som allt annat så krävs hårt arbete och det finns många potentiella problem beroende på organisationens kultur är och hur människorna är men de är en del av organisationen, men det finns många faktorer om Scrum ska fungera i stora projekt.

N7

IPB: Hundra procent scrum om det nu finns något som är hundra procent scrum. vi har det som mål att använda scrum så gott det går

KJ . Vad är din roll i det nuvarande projektet?

IPB: Jag är man kan säga att jag är Product Owner för scrumteamet som är ett av de här 20 delprojekten då och anledningen att jag är Product Owner där beror på att den tidigare Product Ownern fick en annan roll som global product manager det vill säga han representerar marknaden, kunderna på det företaget vi är på och den produkten dom vill ha så det är han som ansvarar för hur det stora projektet ska fungera vad som behövs och inte behövs. Den här personen kan inte vara tillgänglig för alla delprojekt då måste han klona sig själv, eftersom vi kör scrum då scrum kräver en Product Owner som närvarar så blev jag tilldelad rollen som produktägare för vårt projekt, men det betyder inte att jag har hundra procent kontroll över hur systemet ska se ut så givetvis måste jag prata med han som är huvudProduct Ownern och få input från honom men under resans gång har jag förstått hur systemet ska se ut så jag sköter min roll där jag beslutar vad som ska göras och inte göras men det krävdes mycket tid och jag behövde läsa på väldigt mycket om systemet
Jag sa att vi kör fullt ut och inom scrum finns inte projektledaren finns ju bara tre roller Product Owner, scrummaster och team så egentligen kan jag inte säga att jag är projektledare i den bemärkelsen. Jag har en roll där jag representerar Product Ownern i vårt projekt på daily basis. Att säga att vi kör scrum fullt ut och sedan säga att jag är projektledare det går inte ihop, men det är en ledarroll jag har det är det

N9

IPB: Vårt projekt är ett av 20 delprojekt som pågår just nu för att leverera ett större system då inom bygg (automationsbranschen) vi är det enda delprojektet som kör scrum medans de andra 19 delprojekten kör en egen definierad agil arbetsramverk om man nu får kalla det så. Som dom kallar för Hamlet den är väldigt lik scrum men den är anpassad efter den verksamhet jag representerar så att jag kan inte säga att vi kör scrum of scrums där jag befinner mig på uppdrag 17 av projekten befinner sig i Skandinavien, 2 i USA och ett i Australien

N10

IPB: Projektet består av 20 delprojekt där varje projekt har en testansvarig, en motsvarighet till projektledare beroende på vilken metod projektet använder sig av vi har testare, vi har utvecklare och en teknisk dokumentansvarig. Projektledarna i delprojektet har ansvar för att det finns progress att rapportera till programchefen att allting är enligt schema eller inte är enligt schema så det handlar mer om att följa upp så att det är hela organisationen stora projektets se på en projektledare sedan är det upp till varje projekt i sig att ha sina egna visioner och hur man löser den interna strukturen för att uppnå målet. Så varje projekt har en projektledare som rapporterar till programchefen och han i sin tur har ansvar för att deadlines hålls för det här systemet som ska komma ut på marknaden det är på en väldigt hög nivå programchefen arbetar på. Medan projektledarna har resursansvaret då dom ser till att folk är på arbetsplatsen och arbetar och hålla människor på gott humör. Att man håller de deadlines man har så att hela det stora projektet kan rulla fram då. Så är visionen.

kommunikation

K1,K2,K3

IPB: Scrum bygger på hög kommunikation mellan deltagarna det är var det handlar om sedan finns det givetvis saker som är ristade i sten. Vi har daily scrum varje dag kl 09:00 där de berättar vad de gjorde, vad de ska göra och om det finns något som har hindrat dom från att göra det som de ska göra. Givetvis ska man även prata med varandra under dagen, men det är teamets puls att vi dagligen har en rutin där vi informerar varandra om vad vi ska göra det är inte ett sett att rapportera till någon eller några utan teamet ska veta vad folk gör och hjälpa varandra om man nu har ett problem så att inte 2 st arbetar på samma sak så det är väldigt viktigt att man informerar varandra om vad man ska göra men det här är bara en del av en mer komplex beskrivning hur ett scrumteam egentligen ska fungera. Jag vet inte hur djupt in jag ska gå i detta. En best practice är face to face det bästa man kan använda andra medel som mail, msn, telefon det fungerar men är inte så lätt om man samarbetar med någon från USA och köra videokonferens och så kommer inte hela kommunikationen fram genom linan då, det är en helt annan grej när man sitter bredvid varandra och pratar man får en helt annan känsla av vad den andra personen försöker säga då. Ibland kan ansiktsuttryck förstärka den information man vill få ut. men face to face kan man säga är A och O i scrum prata med varandra prata med intressenter

K1,K6,K7

IPB: Han är, det är, när vi väl träffas så är kommunikationen bra men samtidigt som jag sa innan då skälet till att jag blev Product Owner för vårt team är ju då att han inte kunde vara tillgänglig hela tiden han reser runt väldigt mycket Australien, USA. det företaget vi är på projekt har 114 000 anställda och det finns 1 global product manager kan man säga och man kan räkna ut själv att han är svår att få tid med så att det handlar om att ställa rätt frågor och planera tiden och verkligen utnyttja den tid man får med honom så att jag kan göra så bra arbete som möjligt så att vi får ut rätt produkt i slutändan, men givetvis att kunna ha bra kommunikation är att man har en bra domänkunskap om det man kommunicerar om med domänkunskap menar jag att om jag utvecklar ett system som är en automatisering inom byggfastighet kräver en viss tröskel man behöver komma över och förstå deras fackspråk när jag väl förstod det så har det fungerat klockrent jag vet vad han vill förmedla till mig jag tolkar han sinformation på rätt sett och förmedlar vidare utan att det blir några missförstånd vi pratar samma språk just nu kan man säga. 70% av vår kommunikation sker face to face 20% via msn 10% telefon på ett ungefär när det gäller msn så är det mail msn kan man säga men mest face to face. Vi har roliga diskussioner han försöker hitta tid till mig. Nackdelarna är när man behöver ett akut svar på någonting som inte jag kan fatta ett beslut utan att ha ett bollplank, men det sker mindre sedan ett halvår eftersom jag har kommit in så pass mycket i min roll så det blir mindre frågor och jag kan oftast ta beslutet själv

K8

IPB: Det finns ju formella och informella möten. De formella mötena är då alla projektledarna träffas en gång i veckan där dom tar upp beroenden mellan delprojekten problem som man kan uppmärksamma de andra delprojekten med det är spikade möten. som jag nämnde innan så finns de olika projekten i Skandinavien, USA och Australien och många av deltagarna är med via videolänk då. Projektledarna som befinner sig inom samma område och byggnad fungerar kommunikationen bättre med för att man har lärt känna dem och det är högre till tak man kan diskutera tuffare med varandra. Medan det är svårare att kommunicera med jänkarna för att dom har ett annat sätt att vara på dom är mer rätt på medan vi är mer defensiva i våra uttalanden medan jänkarna är väldigt offensiva och med offensiv menar jag att de ser inte problemen utan snarare möjligheterna projekten här i Sverige vill säkra lite mer. Sedan så finns det givetvis möten där teammedlemmar från ett projekt bokar möte med ett annat team vissa är bättre på det än andra

K9,K10

IPB: Säg att vi har ett problem om man t ex en deadline på ett scrum en vecka ofta bygger det inte på att projektledaren har planerat dåligt utan test tog längre tid än det borde göra eller att utveckling tog för lång tid, frågar du en utvecklare så kan det betyda att projektledaren hade en för optimistisk planering det blir en form av blaming i vissa fall. Men man pratar olika språk oftast en projektledare där jag befinner mig pratar mer om milstolpar, datum och personer, medan en utvecklare snarare ser hur hm hur ska man förklara det på ett bra sätt. Man kan säga att projektledaren inte har den tekniska förståelsen för vad som görs därav kan det bli fel ibland när det gäller planeringen av projekten utvecklaren har den tekniska kompetensen men samtidigt inte lika bra på att synka sig med de andra projekten vi pratar om 20 delprojekt som ska levereras samtidigt ungefär

IPB: Givetvis så kan information misstolkas och scrum har ju som mål att rätt man eller kvinna ska ta diskussionen och inte en speciellt utvald person som ska prata om allting med andra projekt till exempel så det finns inga uttalade ansvariga vissa delar i projektet den som kan något om ett visst problem ska representera teamet när man pratar med andra projekt det vill säga att Product Ownern eller scrummastern behöver inte vara de mest lämpliga att diskutera en viss sak med andra projekt utan det kan vara en av teammedlemmarna de som utvecklar kanske besitter den största kunskapen inom ett område man eller kvinna för att ta den diskussionen på så sätt om alla har om du har ett stort projekt som består av 20 delprojekt scrum of scrums då träffas alltid rätt personer istället för att ha en formell person som har hand om kommunikationen då är det lätt att man missar något att man förmedlar information till ett annat projekt som är helt felaktig pga att man inte har den förståelse som krävs för att föra den diskussionen det trycker scrum väldigt mycket på. Ett sätt att minska den risken är helt enkelt att köra scrum hehe.

K11

IPB: Nä det sammanfattar väl att säga mer face to face diskussioner och mindre dokumentation skicka rätt person på rätt möte istället för någon som är utvald på förhand.

Möten

M1

IPB: Det projektet jag är med i (han menar sitt scrumteam) där sitter alla tillsammans vi sitter i ett öppet landskap

KJ . men ni är fortfarande en del av huvudprojektet

IPB: Ja precis och där finns det olika forum att träffas det finns ju dom här, dels har vi när projektledarna träffas 1 gång i veckan sedan finns det andra formella träffar där programchefen kommer och samlar alla projektmedlemmar kan bli att man har ett stort möte på något hotell med 200 pers där man pratar om övergripande deadlines och var projektet befinner sig nu totalt sett, när något ska släppas ut på marknaden och var vi befinner oss om vi är försenade eller inte. Så det finns några sådana här möten då, det gör det. Det viktigaste mötet är de som sker mellan teammedlemmar i ett projekt det är dom viktiga. De formella är bara att, det är något som lever kvar känns det som ibland att man måste ha dom här stora formella mötena och träffas sen

om det är något som har med organisationskulturen att man ska se och träffa varandra. man ska ju helst träffas mellan mötena inte bara på de formella mötena

M3

IPB: Vilka möten vi håller

KJ . ja om ni kör enligt scrum när det gäller möten

IPB: Ja vi försöker efterlikna följa scrums rekommendationer att vi kör iterationer då 2 veckors iterationer så att vi har ju dels så har vi sprintplanering där vi planerar vad vi ska göra de närmsta 2 veckorna sen har vi exekveringen av det vi har planerat, sedan har vi planering inför nästkommande sprint igen så det går i loopar hela tiden då att varje sprint startas med en sprintplanering och avslutas med en demo med det vi planerade då. Och målet och innehållet som ska göras 6 veckor av Product Ownern i det här fallet jag då i mitt projekt. Men det viktiga jag kan tillägga är att teamet måste kommitta dom måste komitta till innehållet som jag lägger fram kommitta dom inte så kan inte sprinten starta det är ju det scrum bygger på att det är teamet som kommittar inte att jag delegerar om att ”det här måste ni göra” utan jag lägger fram en rekommendation, får feedback från teamet sen tillsammans då vissa fall halvägs, vissa fall närmst mig och i vissa fall närmst dom då.

KJ . Jag undrade om ni kör möten efter scrummetoden och det gör ni?

IPB: Ja det gör vi och det handlar om 4 timmars sprintplanering varannan vecka då och daily scrums som max får hålla 15 minuter där alla i teamet säger vad dom gjorde igår. vad de ska göra idag och om de haft några problem med det dom har arbetat på och fokus ligger på vad man ska göra och inte så mycket på vad man gjorde annars är det lätt att man hamnar i en form av rapporteringsläge där man rapporterar och det vill man undvika eftersom det är teamet är ju dom som kommittar till arbetet vid sprintplaneringen då ska man inte behöva få en rapport av varje individ som visar om man jobbat eller inte jobbat i slutändan (paus telefonsamtal)

M4

IPB: Nej.

Styrning

S1,S1.A. S1.B

IPB: Om man ska se det hierarkiskt vi kan börja med rollnivå vi har vår programchef som ansvarar att projektet levereras i tid då. Han i sin tur har 20 projektledare under sig som rapporterar till honom om status och progress i varje projekt sedan så har vi en annan aspekt av det hela arbetet som ska utföras då har vi en global product manager (personen i vårt projekt som var Product Owner från början då) men som lämnade över ansvaret till mig och han kommer med informationen på vad som ska göras och så det kommer från den änden då. Så det finns olika hierarkier beroende vilken fråga man vill diskutera närmre då men styrning om man säger så här så är det programchefen som har det yttersta ansvaret, det är han som ansvarar att systemet kommer ut i tid, styr ner mot projektledare som har ansvar att se till att delprojektet går framåt för det räcker att ett projekt inte går rätt framåt så kommer allting efter så styrningen är så där.

KJ . Finns det någon nackdel med det här

IPB: Ja definitivt när man har den här uppdelningen att du har en programchef som har en tidsplan som baseras på 20 andras tidsplaner är chansen extremt stor att det blir fel och jag kan nästan med gott samvete alltid blir fel det är kanske därför det heter plan men jag vill säga att planera ett stort projekt är en konst speciellt inom systemutveckling eller mjukvaruutveckling så att det kräver ju extremt stor del av informationen är korrekt som förmedlas till programchefen det räcker att informationen är fel från 1 eller 2 projektledare har fått något helt om bakfoten eller progressen så förstör det allt så det är väldigt känsligt sättet att styra det på

S2

IPB: Det är ju också en av de här fördelarna i scrum då scrum bygger på att det är teamet som kommittar och inte någon projektledare som kommittar till arbetet och vad jag menar med det egentligen är så länge teamet så länge tillsammans beslutar att ok det här arbetet ska vi slutföra på 2 veckor så är det dom som beslutar att det ska göras så d et är dom som kan ställas ansvariga om något inte levereras om jag som person delegerar att ni ska göra detta och inte något annat den här mängden så kan teamet överkommita som dom inte har en chans att leverera då är det fel att dom ska ställas ansvariga för misslyckandet.

KJ . jag ställer om min fråga hur hanterar du deltagare i ditt projekt som inte vill för det bygger mycket på tyst kunskap om det då är en person som inte vill dela med sig av sin kunskap och som inte trivs att jobba efter scrum och tycker det här är jobbigt och helt enkelt inte vill kommita hur gör du med en sådan deltagare

IPB: Då är det viktiga att först och främst försöka få teamet scrum bygger ju på att man kör retrospective efter varje sprint så sätter man sig ner 1 timma och pratar igenom vad som gick bra, vad som gick dåligt och vad vi kan göra bättre till nästa gång och det är helt enkelt att få teamet att komma in i sådana diskussioner som berör insats och att teamet identifierar problemet att det kanske finns personer som inte gör allt och försöka få den personen att bli mer scrummig av sig fungerar det inte att teamet kan lyfta upp dom problemen på retrospective så får man i värsta fall ta tag i det själv helt enkelt sätta sig ner med den personen och förklara för honom att scrum bygger på tillit och att man tillsammans kommittar och att man hjälps åt accepterar inte personen det så får man se över möjligheten att lyfta ut den här personen från teamet till något annat projekt, men givetvis ska man försöka få personen att anpassa sig.

S3

IPB: Menar du i scrumprojektet eller i stora hela? Vilken nivå

KJ . vi får ta det på din nivå på teamet

IPB: Som sagt givetvis så. Scrum bygger på nu kommer vi in på det här igen genom att ha ett team som pratar mycket med varandra så hjälps åt och tillsammans löser problemen så lär man sig väldigt mycket under resans gång. Personer som har en medelkunskap kan efter några månader ha väldigt stor kunskap pga kunskaps spridningen som finns inom teamet. Men givetvis startar man ett nytt projekt på 8 personer och kunskapen rent tekniskt sett är väldigt låg då kommer det bli otroligt tufft i början. Det är som sagt Scrum berättar egentligen meer om hur projektet ska rulla framåt och mer hur projektcyklen ska vara fyller man inte den med rätt kompetens då kommer man ingen vart det gör man inte. Men scrum skapar större förutsättningar för att man ska kunna lära sig av varandra än vad andra traditionella metoder gör där man oftast där det finns specialister som inte delar med sig av sin kunskap lika bra som scrum som scrum rekommenderar då scrum bygger på tvärfunktionella teammedlemmar det vill säga personer som kan göra många olika saker, göra rätt saker i rätt ögonblick. Klart har man låg teknisk kompetens kvittar det om man kör scrum eller andra traditionella metoder man kommer att misslyckas men scrum skapar större förutsättningar för att det ska gå bra

S4

IPB: Det beror på hur teamets sammansättning är har du 4 seniora och 4 juniorer i projektet så fungerar utbildningen som så att seniorerna utbildar juniorerna, men har du 8 juniorutvecklare så får aktiviteterna som tas in i först sprinten vara av den karaktären så att det blir en kompetens ökande sprint. Det vill säga att istället för att utföra det riktiga arbetet att man lägger upp kunskapsinsamlade aktiviteter då. Så man har som sprint mål att vi ska kunna göra ett litet program nått enkelt demo bara med vissa aktiviteter kopplat till demot för att bygga upp en viss kompetens då så det beror helt på individerna i projektet, men det är väldigt ovanligt att man har 8 juniorutvecklare i ett projekt oftast försöker man ha en mix

S6

IPB: Det är samma grej som jag sa innan om kommunikation den ska alltid ges av den person som besitter störst kunskap om ämnet samma gäller coachningen där man ska inte försöka låsa sig i formella roller utan den personen som besitter den största coachkunskapen ska vara den som coachar givetvis är det bra att har du

exempelvis ett team med 4 seniora utvecklare och 4 juniorutvecklare är det en väldigt bra ide att försöka ha parprogrammering i början där den seniora lär upp junioren så att kommer upp på banan det är väl ett sätt. Parprogrammering är inte något scrum rekommenderar det är en best practice det är något man fyller scrumtavlan med. Så det är väl ett bra sätt parprogrammering

S7

IPB: Det är egentligen samma svar som jag gav om personen som inte kunde anpassa sig till scrum det handlar om att om man som scrummaster eller teammedlem upptäcker ett visst mönster hos en viss person då som inte är uppskattat så gäller det att identifiera det på retrospectet och ta upp de problemen, det ska inte vara scrummastern eller Product Ownern som lägger fram det ska vara teamet själva som på något sätt det gäller att få scrummastern in i diskussionen eller att själva lyfta fram det på retrospectet att tillsammans lyfta fram problemen och åter igen där lyckas inte det så får man ta tag i personen i fråga och diskutera om man gör det av egenintresse eller för projektets bästa. Men det bästa är om teamet själva kommer in i den diskussionen det skapar ansvarskänsla över att man är med i något att man själva kan identifiera problemen och lyfta fram dem retrospectet är ju nyckeln där eftersom retrospectet tar upp vad man gjort dåligt, bra och vad man kan göra bättre. Så förhoppningsvis är ju att eliminera egenintresse hos individer är något man kan göra bättre då ur projektets synvinkel

KJ . jag tänkte mer på människor som är utanför projektet som har egenintresse och försöker påverka teamet

IPB: Jasså du menar så ja ja det är scrummasterns uppgift att skydda teamet mot utomstående. När teamet kommit till ett innehåll ett sprintinnehåll under 2 veckors tid, så bygger den kortsiktiga 2 veckors plan att man jobbar ostört man kan inte ha personer som är utanför teamet komma in och ta eller låna personer under 2-3 dagar för då faller det teamet har kommit till då så det är väldigt viktigt att teamet får jobba ostört. Givetvis finns det situationer som man behöver prata med någon i ett annat projekt men det ska försöka hållas på minimal nivå. Man ska jobba ostört men om något är akut så får man plocka in det som en aktivitet i nästa sprint så får man planera att hjälpa med just det problemet som ytterligare en intressent har då. Arbeta ostört är A och O. det är svårt att komma och uppnå mål om man inte får arbeta enligt planerat

S10

IPB: Nej

Befogenheter

B1

IPB: Befogenheter tilldelas i början innan projektet startar man har en scrummaster, Product Owner och teammedlemmar i en ideal värld ska dem vara intakta dom ska ju inte pendla rollerna så befogenheter ges i början och ska hållas sen kan man ta diskussionen på en annan nivå så länge man efterföljer scrums rekommendationer så gör man rätt en teammedlem ska inte kunna ta saken i egna händer och gå utanför scrumprocessen då ska ju scrummastern vara där och coacha och berätta vilka regler som gäller. Varje roll har sina befogenheter ex Product Ownern ska inte gå in och vara scrummaster och ta hans roll då blir det konflikt i det hela

B3

IPB: Absolut det kan vara en enkel grej att, jag nämnde innan projektledarna har veckomöten där dom tillsammans sätter sig ner och pratar om problem man har i sitt projekt är det en projektledare som inte berättar allt kan det bli väldigt fel, eftersom dom kan planera något som är baserat på felaktig information då kan det bli helt gale. Sen är det på delprojektnivå där kan inte Product Ownern komma in och bestämma vad teamet ska kommit då kan det bli en konflikt

B4

IPB: Nej.

Arbetsmetod

R1

IPB: I vårt fall har teamet förändrats projektet har varit igång 2,5 år slutar folk och börjar folk och då tvingas man ta in nya personer i projektet, men projektet i sig lever vidare men sammansättningen förändras men Product Owner och scrummaster är stabila det är sällan det sker förändringar där. Men teammedlemmar givetvis. Att få ett team att arbeta jättebra kräver att de arbetar tillsammans en längre tid så att man kommer in i de olika team faserna, att man känner tillit och ansvar dom bitarna då så har du ett team som hela tiden förändras fastnar gruppen i rollsökarfasen och man kommer aldrig upp på den nivån när ett team blir högpresterande. Det är inte bra att byta personer mellan projekten det skapar oro och förvirring det blir hack i cykeln, scrum bygger hela tiden på att bli bättre och med bättre menar jag på teamnivå för att kunna mäta förbättringar över en tid kräver att projektet innehåller samma personer. Så vi undviker att flytta runt dem, de fall det har hänt så har folk slutat eller misskött sig men det är väldigt sällsynt

R2

IPB: Om man går enligt scrummetoden så är de tre rollerna fasta det rullas inte runt bland rollerna

R3

IPB: Små projekt är givetvis enklare att kontrollera och få en exakt planering på. När du har ett stort projekt så är det många delprojekt som ska fungera tillsammans och det kan hämma kommunikationen man kan efterlikna det vid en familj är det många så är det lätt att glömma bort någon och det blir subgrupperingar

R4

IPB: sprintplaneringen är att planera den sprint som man kommer att börja arbeta på när sprintgenomgången är klar givetvis så måste man arbeta med sprintar som kommer längre fram men det är Product Owners ansvar att göra de prioriteringar som krävs och att kravställningarna som är i tur i backloggen nedbrutna och gjorda på ett så pass bra sätt att teamet under en sprintplanering ta sig an uppgiften förstå vad den innebär och kunna 100 procentigt göra en exakt plan hur man ska implementera det där så det ställer stora krav på Product Ownern att han kan förmedla det som är viktigt. Man försöker planera en max två sprintar i taget för det är ingen idé att planera fler för världen kan förändras, man kan kanske inte ens implementera något som ligger fyra sprintar framåt i tiden när den väl blir aktuell man försöker ligga på en kontrollbar tidsram

R5

IPB: det bygger på att teamet är moget att dom har förståelse för varandra och kan ärligt säga till varandra vad som är möjligt och inte möjligt så alla i teamet måste vara överens om innehållet i Sprint Backloggen för att man ska kunna starta sprinten man ska inte starta en sprint ifall 1-2 i teamet anser det vara omöjligt att lösa uppgiften utan då får man komma till en sådan nivå där alla är nöjda det bygger på att alla kommittar om det är någon som stretar emot så är det ett ärende för retrospecten på vad som kan göras bättre inför nästa sprint då

R6

IPB: det bygger på att kommitta teamet på vad dom tror dom hinner med och inte blir påtvingade något så hålls tidsramarna bra i de fallen Product Ownern kör en egen agenda och försöker få teamet att överkommita ja då blir det oftast att man inte hinner med arbetet och inte tidsramen, tidsramen avser sprintiterationen som är 2 veckor i vårt fall så det beror på vem som kommittar till arbetet

R7

IPB: Ja för att en retrospect ska vara bra så ska scrummaster och Product Owner och teammedlemmar vara närvarande felar någon så får man inte full effekt av det klart någon kan vara sjuk ibland men man försöker få alla att vara med på den och det är väldigt viktigt att det bara är teamet som är med. Det får inte vara andra intressenter utifrån som är med och lyssnar för det kan vara väldigt känsliga saker som man diskuterar och det ska stanna inom teamet så det är en stängd session

R8

IPB: nej

Bilaga 5 – Intervju C – IPC

Inledande Frågor

N1

IPC: Jag har civilingenjörsutbildning i elektroteknik från LTH.

N2

IPC: Jag har varit projektledare och första gången jag var det, det var drygt för 12 år sedan. Sedan har jag varit i olika roller, oftast i rollen som konsult hos kunder. Jag har varit projektledare mest i förbättringsprojekt med en del skarpa-leverans projekt.

N3

IPC: Jag har jobbat som projektledare båda i traditionella projekt och i rollen som Scrum Master i agila projekt.

N4

IPC: Fem

N5

IPC: Stort projekt, 50 personer eller fler. Budgeten är 50 miljoner och uppåt

N6

IPC: Scrum är utvecklad för att vara ramverk för produktutveckling och DEPLOYMENT (att sprida den, att sätta den på marknad eller releasa det) men man pratar också om det är för komplexa produkter, Scrum pratar väldigt lite om projekt och projekt formen, där också kan man ju utveckla produkter där det krävs många människor och jag anser att det går att göra, det är inte enkelt och det tar tid och det kräver mycket eftertanke ifrån de involverade och det kräver rätt mycket förståelse ifrån de involverade också. Men jag anser att det går att göra.

N7

IPC: Jag har inte använt Scrum fullt ut i stora projekt, däremot i mindre, ja!

N8

IPC: För min del handlar det om produktstrukturen, för att om man utgår från produktstrukturen och Product Owner rollen, funderar man över det så det räcker det bara med en Produkt Owner också behöver den personen kompletteras med roller som har mer teknisk karaktär än vad man lägger på Product Ownern i utgångsfallet. Och vi behöver någon form av hierarki när det gäller Product Owner. Man kan tänka sig ha en Chief Product Owner (CPO) som högst ansvarig, och sedan Product Owner som kan befinna sig i de olika områdena av produkten. Ut ifrån oftast blir det någon form av hierarki och de här enskilda Product Owner kan ibland behöva kompletteras med roller som arkitekt och liknande som stöter i formuleringen av kraven. Jag försöker att undvika att ha för många roller eftersom det som jag anser man ska sträva efter är de tvärfunktionella i Scrum, att man skapat tvärfunktionella Team och där tycker jag att arkitekt rollen passar bättre, det kan vara rent lite känslomässigt där organisationer har lite svårt att åstadkomma. Jag är helt övertygad om att det är det bästa sättet.

Sedan beroende på hur många områden och hur många team jobbar i respektive område, eventuellt så behöver man skala artefakten Product Backlog också. Jag gillar att inte göra det för komplicerad, att sträva efter för

komplexa sätt att lösa saker på därför vill jag gärna ha en Product Backlog oavsett hur stor produkten är, men att man anstränger sig för att ha olika vyer av det. Om man delar upp produkten i olika områden så har varje område sin egen vy av Product backlog, därför behöver man vara på en mer övergripande nivå på de övergripande kraven kan man behöva portionera och tagga på något vis, t.ex. område A och område B och ibland båda, beroendet emellan och liknandet. Så man kan få ut ett antal vyer från Product backlog så att den enskilda Teamet kan se en del av det så att de inte ska se hela alltihopa förutom den delen som gäller just dem, dvs. en enda Product Backlog. Nästa steg handlar det om hur vi jobbar med samordningen och synkroniseringen mellan de olika Team. Pratar vi stora projekt så pratar vi om rätt många team.

Vi använder oss av centraliserade Scrum of Scrums, där alla team ska samordnas en nivå ovanför, och en nivå till men det är relativt sällan man stöter på det. Den decentraliserade varianten är att man litar på teamen och att de har tillräckligt drivkraft för att förstå vilka andra team som de behöver eller kan synkronisera sig med och att man då ger var och team i uppgift att identifiera de andra teamen och vilka man behöver synkronisera sig med och se till att skicka team medlemmarna till dagliga möten, dvs. Scrum möten. Det är en styrka i det, det är snabbare kommunikation, man får oftast en större förståelse på helheten i produkten. Det är inte alltid så att organisationer vågar lita på teamen att klara av att göra det här, det kan vara en nackdel, men jag tycker inte att det är en nackdel.

Kommunikation

K1

IPC: Det jag ska informeras om är hur planen ser ut i det grova långsiktiga planen, i Scrum pratar man om Release Plan, och det är Product Owner som ansvarar för det och lämpligt, jag är förespråkare av fysiska informationsplatser, man sätter saker på väggar, meeting rooms osv... någon skärm som finns på en central plats i företaget kan vara bra att använda, intranät kan vara bra att använda för att sprida information.

K2 & K3, K6 & K7

IPC: Den informationsspridning som jag anser behövs det är från Product Owner och vad som anses ska göras och status i förhållande till hur vi ligger till ifrån det som planerats. Säg att vi har 8 sprintar, hur ligger vi till efter 4 sprintar, hur ligger vi till efter 5 osv. och vad som behöver justeras och/eller ändras, det är vad som gäller information från Product Owner.

Sedan har vi information från Teamen, hur det går för oss i innevarande sprint. I större projekt kan det vara nyttigt att använda oss av vissa mjukvara som kommunikations och informations spridnings hjälpmedel, är det ett eller två team så rekommenderar jag att undvika det.

K4 & K5

IPC: Jag anser att det ska vara Product Owner som sköter det. Det är inte alla organisationer som gör det fullt ut på det viset, vissa känner sig att ha behovet av den mer traditionella projektledarrollen. Jag ser det såhär ibland att även om man jobbar med Scrum fullt ut så kan man behöva ha gränssnitt mot ledningen eller mot övriga organisationen och det är någon form av begräsning, man ska behålla sig till de standarderna och kommunikationerna som finns inom ett företag även om, jag tycker man ska kära av en del av kommunikationen om de inte har ett värde, ibland finns de för ett speciellt syfte såsom rapportering som jag anser att Product Owner som ska sköta det, sedan har man rapportering om hur man ligger till i sprinten och det är då teamet som ansvarar för det, det tycker jag det räcker bra.

K8

IPC: Är man väldigt många team så kan det behövas samordning, man kan behöva införa möten som kanske inte föreslagits i Scrum. Man har Scrum of Scrums möten, man kör decentraliserade kommunikationsmodellen och samordningsmodellen, så deltagandet i de dagliga mötena kan behöva köras lite längre.

K9

IPC: Ja, det är därför man är så noga i Scrum att ha en person, vi säger att product owner är en person och ibland räcker det inte och då har man fler, det kan hända att samma information ges av olika människor på olika sätt även om de inte haft den interaktionen, den risken finns. Jag gillar att göra så att det som är huvudansvarig, kallas för Chief Product Owner, har mer övergripande ansvar sedan går det ju inte om det är många människor, för en enda som ska informerar alla om allting, utan på något visst ha kompletterande information och möjlighet att ställa frågor och det ligger lite utanför Scrum kan man säga, utan det är mer vanligt sätt att hantera många människor och att se till och ge folk möjligheten och göra det tydligt, om jag har en fråga ”vad ska det ändras?”, på engelska säger man det på ett ganska bra sätt då Scrum Mastern utför sina uppgifter ”Use all your senses, including common sense”, det låter inte lika roligt på svenska. [leende]

K10 & K11

IPC: Det iterativa, att du jobbar i iterationer och att du följer återkopplingen på det som visas upp i slutet av varje sprint, det är ju ett sätt att minska risken för missförstånd som kan uppstå.

Jobbar du i 18 månader sedan visar du upp de sedan säger de ”nej, det var inte just det jag ville ha”, så är det ju rätt jobbigt, jobbar du bara en månad eller två veckor då är det ju inte lika stort problem. Sedan strävar man efter att undvika sådan naturligtvis, men just det här iterativa inslaget genom, jag brukar prata om korta återkopplingsmötena, det dagliga mötet, att ha Product Owner tillgängligt i daglig basis vilket är en av de stora utmaningarna när man tillämpar Scrum i stora projekt.

Jag gillar att hålla mig till de mötena som finns i Scrum, och där finns ju tre möten, det är sprintplaneringen i början och sprint preview i slutet och sedan det dagliga mötet. Det dagliga mötet för ett team ska hållas kort, har man olika nivåer då man tänker sig ha Scrum of Scrums så blir Scrum of Scrums mötet lite annorlunda jämfört med ett dagligt möte för ett team, det oftast behöver vara lite längre, har en annan karaktär. Det dagliga Scrum mötet för ett team handlar om synkronisering, vad du har gjort sedan igår, vad du planerar att göra till imorgon och vilka hinder som gjort att du inte blev färdig och då kommer vi överens om att jag ska hjälpa teamet med detta och det andra och bli av med problemet innan vi sätter igång med ett nytt, det är synkroniseringen vi är ute efter i teamets dagliga möte.

Scrum of Scrums mötet har en annorlunda karaktär och det kan handla om problemlösning också och därför det behöver vara lite längre och kanske behöver kompletteras med någon roll, i stora sammanhang man kan behöva lägga till någonting utöver de vanliga Scrum och kanske någon synkronisering eller arkitekt roll, jag brukar rekommendera att man inte gör det utan snarare ha virtuella team som har ansvar för det övriga, eller att man tänker i termer av att ha integrationsteam där de har ett antal team som utvecklar och deras resultat integreras av något team så man har den hierarkin i detta sammanhang.

SR: om man har de här stora projekten och anta att man har bara en Product Owner, vad blir konsekvenserna?

IPC: Är det en person som varit en traditionell produktchef så måste han eller hon sannolikt jobba externt mot kunder, marknad och liknande, så det går nästan inte att räkna med bara en produktchef, det har jag svårt för mig att få. Utan man behöver hitta några vägar där det går att lösa och när man jobbar tillsammans, och det är inte bara Scrum som har det som en framgångsfaktor, men samarbete och kommunikation är framgångsfaktorer i Scrum, och det är ju i de flesta sammanhang. Men det är extra viktigt i Scrum.

Möten

M1 & M2

Jag har sett några stormöten, och rådgivit organisationer som har haft större projekt som har haft olika mötesformer. Stormöten för alla som är med i projektet hålls, och det är väldigt väldigt vanligt och de varierar i kvalitén, många har väldigt dåligt kvalitét i och med det handlar om envägs kommunikation i stor möten och det är, jag personligen har lite svårt för det, för jag ser inte riktigt nyttan med det.

Sedan har jag sett i organisationer, det finns, till exempel SKANIA-IT i Södertälje, de har möte där de kallar det för ”puls möten” och det är inte bara ett stort projekt med stormöte en gång i veckan utan alla, alla som jobbar i utvecklingsorganisationen samlas i ett stort rum där varje projekt har en stor tavla som man skjuter upp och drar ner och det kan vara många tavlor i det här rummet och de har ett sådant möte en gång i veckan och tar en timme, ungefär. Där ser man till att kort dra svar som berör projektet och folk kan ställa frågor, man har handskakning när man går från en fas till en annan, de försöker jobba ”Agile” också och är inte linje - inspirerad

alls. Det är liksom hjärtat i deras informationsspridning, det här mötet och rummet och det är den typen av möten och rummen tittar man på.

M3

Jag tror att man behöver, förr eller senare samla alla som jobbar med produkt med jämna mellanrum där man ska kunna ge möjlighet att lyfta upp frågor eller sprida information. Man håller mötena kort och försöker arbeta med de mötena som finns till bud enligt Scrum och ett av de bästa mötena är det där är ju demonstrationen (Sprint Preview), där vi visar upp vad vi har gjort, demonstrationen kan ju användas till informationsspridning också. Säg att man bjuder in inte bara Product Owner och teamet som har utvecklat utan alla som är berörda av det vi jobbar med just nu. Man kan också tänka sig göra en gemensam demonstration på allt som har gjorts efter sprintarna av alla team vilket kan kräva lite extra jobb naturligtvis, men just möjligheterna med att informera och sprida information genom existerande möten kan man lätt glömma bort.

DI: De dagliga mötena (Daily Scrums) ska väl kunden vara med, eller?

De dagliga mötena är till för team medlemmarna där de ska synkronisera sig, det är de som ska jobba, de som ska prata, sedan är det tillåtet för Product Ownern att vara närvarande och lyssna, det är tillåtet till vem som är intresserad att vara närvarande och lyssna, men det är bara team medlemmarna som får prata, så att säga. Sedan är det sund förnuft som gäller där också, det är ett kort möte för att synkronisera och sedan är det oftast så att man identifierar behovet av vad som ska diskuteras, de saker som har kommit upp under mötet, detta kan man ta efter bestämda mötestiden, hitta lösningar och/eller ha diskussioner. Det är den typen av omvägar som man får göra. För att undvika, om vi är 7 i teamet, det kan finnas en fråga som berör bara 3 stycken, då är det bättre att de övriga 4 kan gå till arbetet och vara produktiva istället för att bara sitta ner på mötet och bara rulla tummarna.

KJ: man jobbar inte som robotar, det står att alla ska vara med så ska alla vara med...

Ja, och det är just därför vi håller mötena kort, det är viktigt för mig att veta vad du lyckas med och inte lyckas med för det är ju vi som team som ska lyckas tillsammans eller misslyckas tillsammans alltså det är inte jag som individ eller du som individ som ska göra det.

DI: kommer Product Owner vara med dessa möten, är det ofta Product Ownern är med?

Jag tycker att Product Ownern har nytta av att vara med på de här mötena och det är lite beroende på hur långa sprintar man har och hur bra man distribuerar det. 1, 2 eller 3 gånger i veckan kan man säga att det är vettigt och vara med. Själv tycker jag att det är bäst och vara med varje dag, men det är väldigt få som gör det.

M4

IPC: Det är bara det att man lätt glömmet bort att man faktiskt kan använda Scrum mötena till informationsspridning också.

Styrning

S1

S1.A. & S1.B.

IPC: Ja, det blir mer hierarkiskt. Sedan är det så att de beslut som, sedan är det ju vad du gemför med, gemför du med ett stort projekt som jobbar traditionellt eller gemför du med ett litet projekt som jobbar traditionellt eller projekt som jobbar enligt Scrum?

DI: det vi är mest intresserade av är ju hur det är när det är Scrum och i stora projekt.

IPC: Ju, det blir ju mer nödvändigt med att ha det mer hierarkiskt. Så fort man blir fler så behövs det en samordning. Där man ska behöva särskilja, dela upp i teams som utvecklar sina olika delar och man strävar efter att de olika delarna ska vara så oberoende av varandra som möjligt. Jag brukar säga att man försöker hitta det ideala läget, men det ideala läget uppstår väldigt sällan. Vi försöker minimera beroenden och hitta sätt att uppleva så lite beroende under sprintarna som möjligt. Just för att de här beslut- och liknande procedurerna ska kunna ligga nere på teamet, det är de som producerar besluten om hur vi gör, exakt vilken lösning vi väljer. Ju fler ledare desto mer övergripande beslut behövs, så fort har vi tagit beslut så har vi egentligen begränsat oss. Så,

nackdelen i en beslutsprocess i ett stort projekt om man får må filosofiskt än att behöva vara på ett visst sätt skulle jag säga. Är du få tal människor, är det bara vi 4 då klarar vi att sköta detta utan några direkta processer och vi kommer fram till lösningar och hitta sätt att jobba på. Men är det 10 andra team då är det nackdel när det är en stor del att göra. Många gånger så handlar det om att man kan utveckla saker på ett team, men ledtiden blir ju noll om det är stora produkter som ska utvecklas.

S2

IPC: Det kan vi prata om i flera timmar och flera dagar, hur man hanterar sådana människor. I grunden och botten så handlar det om att, att jobba i team kräver detta ett antal saker, bl.a. så är det väldigt många som är överens med mig när jag säger att det första som man måste tänka på är att team medlemmarna litar på varandra, om jag som team medlem ska dela med mig och min kunskap så litar jag på att de övriga, om jag ska ställa dumma frågor så litar jag på övriga team medlemmarna att de inte få det i nacken efteråt.

För att om vi litar på varandra, då klarar vi av att ta en diskussion som är en kreativ och fruktbar diskussion om hur vi ska lösa problem A, hur vi ska lösa problem B, vi kan komma till ganska högljuda och animerade diskussioner, om vi litar på varandra då klarar vi av att säga [så här tycker jag och så här tycker jag] och sedan säger du vad du tycker och så gemensamt i teamet kommer vi fram till en lösning som vi kan komma överens om att det gör vi tillsammans. Förutom att lita på varandra, ingen i teamet ska behöva känna någon rädsla för konflikter och klarar man av det så klarar man av att ha ett gemensamt åtagande och klarar vi det så klarar vi av att ställa varandra till svars och klarar vi det så klarar vi av att vara resultat orienterade. Och hela det här som jag beskriver nu, det är inte jag som kommit på just den utan det är en person som heter "Patrick Lencioni" som har skrivit en bok som heter "The Five Dysfunctions of a Team". Och det har jag och mina kollegier haft väldigt mycket nytta av att lära oss den modellen och jobba med Team, att öka sin förmåga att lita på varandra, att hantera meningsskiljaktigheter och att komma fram till gemensamma åtagande osv. så man blir resultat orienterade, och många gånger så handlar det om tillit och troende för varandra.

S3

IPC: När man väl har satt igång och formerat teamen, vem ska ingå i det teamet och vem ska ingå det andra teamet osv. så ligger ju väldigt mycket ansvar på team medlemmarna att identifiera, har vi i vårt team tillräckligt kompetens för att lösa den uppgiften då har vi åtagit oss, och ser vi att vi behöver öka vår kompetens osv. det gäller att ha förmågan att identifiera de kompetenserna som vi inte har.

S4

IPC: Alltså hur man löser det, det kan vara på olika sätt, utbildning är ett sätt att lösa det på. Att man ökar kompetensen hos de som finns i teamet genom att addera eller komplettera med en team medlem som är eller har efterfrågad kompetens kan vara ett sätt att lösa detta på. Det finns inte alltid ett sätt att lösa saker och ting på. Det som jag gillar att göra är att ge teamet ansvaret för att identifiera behovet och komma med lösningsförslag och det alternativet tycker jag är det bästa. Det kan finnas begränsningar naturligtvis.

SR: är det inte lite svårt när det 40-50 man som ska stå och var och en ska fram och berätta vad han eller hon har gjort osv.?

IPC: Man ska göra det för det dagliga Scrummet för Teamet, och man säger att Scrum Team ska vara från 5 till 9 stycken, om man har Scrum of Scrums så har man representanter från de teamen så det är också relativt lågt antal så kör du Scrum "by the book" så är du nästan aldrig 40-50 man.

S5

IPC: Jag anser att man kan delegera och att man ska delegera rätt mycket, skälet till att jag anser det är att det handlar om att delegera hur vi ska lösa en uppgift, det handlar inte om att delegera vad vi ska göra, det ansvaret ligger på Product Owner, den ena eller det är flera som behövs. Det jag menar med "hur vi ska göra?", det finns alltid flera olika sätt att lösa en uppgift på, och det är just därför jag tycker att man ska delegera, för att varje Team ska hitta sitt sätt att göra bra resultat. Jag är en sådan person som tycker att det är oerhört tråkigt när en person ska komma fram till mig och berätta för mig om hur jag ska lösa en uppgift, jag gillar att komma på det själv och göra efter mitt huvud. Jag tycker att det är bättre om jag får lägga mina aspekter på saker.

S6

IPC: Ja, alltså i ett stort projekt kan det behövas en projektledare, ett stort projekt enligt Scrum har per definition ingen projektledare, det är organisationerna som väljer ha det, och en traditionell projektledare har ju ett antal uppgifter, allt från de dagliga rapporterna som kan hamna på en traditionell projektledare, att stödja medarbetare i projektet och att sköta övergripande rapportering. Det här är ett exempel på hur uppgifter som hamnar på en traditionell projektledare fördelas på rollerna i Scrum, den dagliga rapporteringen, uppföljningen under Sprinten är ju Teamets ansvar. Att stödja team medlemmar, att ge de rätt förutsättning för att lyckas det är Scrum Masterns ansvar och det övergripande långsiktiga planeringen och rapporteringen, det är Product Owners ansvar.

Så, den coachningen som sker, sker oftast från Scrum Mastern till team medlemmarna, är det till hela teamet eller en del av team medlemmarna. Jag går tillbaka till föregående frågan, ”hur det sker?”, det är väldigt mycket upp till respektive Scrum Master, men det som är en vanlig rekommendation är att man ska göra det ur ett faciliterande perspektiv, att ställa frågor, att berätta om observationer och synpunkter, på alla möjliga sätt, fråga teamet än att tala om för teamet, för det är också sätt att hjälpa teamet att bli självgående. Scrum mastern som har coachar rollen så behöver man oftast agera annorlunda om man är Scrum Master för ett Team som är precis sammansatt, det är kanske inte alls läge för att jobba på detta sätt och då kan man behöva göra mer.

Har man scrum team som jobbat länge och är sammansatta under längre perioder så behövs det ingen Scrum Master för då vet vars i teamet vad de ska göra och hur. Jag brukar berätta, när jag håller kurser, det är inte bara jag som har goda erfarenheter om teamarbetet, hur team samarbetet fungerar bra, när jag var ung och var ute och segla så var vi 4-5 personer, vi visste vad vi ska göra, var och en visste hur och vad han ska göra, alla hade egna ansvarigheter, vi behövde inte ens prata med varandra efter ett tag, skulle det dyka upp något utan för det normala så löste vi det tillsammans.

Det är nummer ett fascinerande, och det är den känslan jag vill, jag vill hjälpa teamet när jag jobbat som Scrum Master och jag tycker att alla Scrum Masters borde sträva efter. så, hur det sker, det kan finnas behov av att lära sig frågeteknik till exempel det finns material som man kan hitta, en bok som heter ”Powerful Questions”, där pratar mycket om termer som vilka frågor är vettiga att ställa och hur man ställer frågor. T.ex. så undviker man att ställa frågor som kan besvaras med ett Ja eller Nej, i det läget man vill gärna ha en diskussion. ”Powerful Questions” är för att få de att tänka på frågan, uppmanas att tänka och bli mer kreativa i den situationen. Det är många Scrum Masters som söker utbildningar inom coaching och coachingstekniker.

S7

IPC: Det finns alltid intressenter, jag brukar kalla de för intressenter, det finns alltid många olika typer av intressenter. Jag gillar att bygga från grunden och en grundläggande del, det handlar om att berätta om arbetssättet för intressenter, att det är såhär tänkt fungera, här har vi den rollen och den har den rollen och ni ska betraktas som intressenter och vi behöver era medverkande och vi behöver era åsikter. Detta ska kommuniceras till Product Owner för det är Product Owner som ska bestämma om vad som ska göras, i vilken ordning det ska göras grundad på vilket som prioriteras i Product Backloggen. Vi behöver alla era synpunkter på vad som ska göras, vi behöver inga synpunkter på hur det ska göras utan det lämnar vi till teamet. Och ni är välkomna, en gång i veckan eller en gång varannan vecka eller en gång om månaden eller hur vi har de här Sprint Reviewerna, ge en återkoppling på det vi visar upp, det som är färdigt, att vi kan mäta vår progress i färdiga grejer och det som brukar höras av teamen är att de påstår att de är 95% färdiga, det ska inte vara så.

S8 & S9

IPC: Om frågan är såhär: hur gör vi för att undvika konkurrensen mellan team medlemmarna, så handlar det mer om att lyfta det från en enskild individ till team - sammanhanget. Och det är likadant här, att lyfta det från det enskilda teamet till det stora helheten, just det där med samarbete, vi vill bygga en produkt och vi behövs, allihopa. Det handlar inte om att vi ska lyckas med vår del och vi ska inte lyckas med vår del utan det handlar om att få det till att bli mer att lära av varandra också, har vi någonting som vi har fattat, gör vi på det här viset så blir det så himla bra, vi ska visst sprida kunskapen till flera och har vi problem så informerar vi resten om det. Så man lyfter det från individen till nästa nivå. Inom ”Lead Software Engineering”, Scrum har fått vissa influenser från Lead, man pratar om ”Optimize the Whole” som ”Optimera Helheten”, och det säger ju Scrum i sådana här sammanhang.

S10

IPC: Product Owner huvudansvarar för styrning och i form av att ha en bra Product Backlog som är uppdaterad, och sedan vad det än som ska kunna ta beslut när det bli olika syn på saker och ting, om man har en hierarki av Product Owners så är det någon som behöver kunna lösa ut saker och ting, vi är på samma nivå som Product Owners och vi behöver komma överens, ibland kan vi behöva hjälp med att lösa ut saker.

Befogenhet

B1 & B3

IPC: I Scrum ändrar man inte befogenhet på rollerna, Product Owner har befogenhet att bestämma vad som ska göras, vad som är viktigare än det andra och har befogenheten att prioritera mellan saker och ting. Har jag ansvar att göra det så man maximerar värdet av det som insatsen ger så, är man Product Owner och har fem team eller tjugofem team, det spelar ingen roll utan det gäller att prioritera så att vi får ut så mycket värde av insatsen för vår iteration som enligt Scrum finns. I övrigt så är det befogenheter för team då att ta beslut om hur vi ska, dels, det viktiga, viktiga att det är teamet som har befogenheten att bestämma hur stort åtagande gör vi för varje Sprint, det ska man ju aldrig tumla på. Och sedan också befogenhet att bestämma hur vi implementerar, men givetvis i samarbete med Product Owner då de har dialoger eller ställa frågor.

B3

IPC: Konflikter kan uppstå då Product Ownern vill vara med och bestämma om hur teamet eller flera team ska lösa en uppgift, det är en typisk konflikt. En annan konflikt kan vara då teamet förlitar sig om att Scrum Mastern ska hjälpa de på något vis med de tekniska besluten, men det är fel det där, Scrum Mastern är inte någon tekniker som det ibland påstås i vissa projekt, utan en Scrum Master är en coach, en facilitator och ett stöd, någon som har ansvar om att ta hand om allt det som egentligen inte har med implementeringen att göra. Det är team medlemmarna som ska fokusera på det de har åtagit så Scrum Mastern ska inte hålla på och leta efter saker som de saknar och laga servrar osv..Det kan hända en del konflikt och bråk om någon hierarki av Product Owner finns, mellan två Product Owners som ligger på samma nivå naturligtvis.

KJ: Scrum Mastern och Projektledaren, vilka konflikter kan det uppstå då det är ett större projekt?

IPC: Ska du köra ett stort projekt enligt Scrum så har du ingen projektledare, du har en Scrum Master för varje team, du kan tänka att du gör olika indelningar, du kan antingen, antingen kan du göra som jag sa tidigare, att du har ett antal team som levererar någonting och sedan har du ett integrationsteam och de har en Scrum Master och de jobbar för att det ska bli lätt för underliggande team att lösa sina uppgifter och sedan underlätta integrationen, så man har en annorlunda uppgift som integrationsteam än ett rent utvecklings- eller implementeringsteam. Så jobbar jag enligt Scrum så kan jag ha den typen av uppsättning eller så har man, som vanligt, olika team som tillhör olika område av produkten och enligt Scrum så har vi ingen projektledare för man har en Product Owner här och sedan kan man ha en CPO (Chief Product Owner). Sedan är det vissa organisationer som tycker att vi behöver en projektledare och då väljer de göra det, det är inte helt enligt Scrum och göra men, det kan komplicera till det. Det är nog någon som ska hålla ihop trådarna. Och ja, det kan ske vissa konflikter då det finns olika flera olika identiteter ja.

Är det ett stort projekt så mappar jag gärna upp Produkt Ownern till projektledare. Är det medium och små projekt så mappar jag upp Scrum Mastern till projektledare.

Arbetsprocessen

R1

IPC: Nu är ju nackdelen att jag inte är inne på ett projekt just nu. Det där är väldigt olika, och det är många som håller med om att team som jobbar tillsammans och går genom olika faser och först gör de så att de lär känna varandra, klämma lite på varandra, förstå vad det är för uppgift, sedan kommer man till en konflikt fas och så försöker man reda ut det och försöka hitta normerna för det här teamet och sedan så får man gå vidare och bli presterande. Det är många av de här teorierna som finns om gruppsamverkan som säger att ändrar man sammansättningen då kan man bara förvänta sig att produktiviteten går ner för ett team över en tid och det är därför man ska sträva efter att ha samma team och de som glömmer bort det blir inte så framgångsrika som de som klarar av att komma ihåg det.

R2

IPC: Jag har aldrig varit med och jobbat i ett projekt där jag har något att säga till om alla har sina fasta roller inom teamet, utan man har sin kompetens och utifrån den så bidrar man på olika sätt, jag är inte testare, utvecklare eller CR-ansvarig utan jag har kompetens inom något område och då bidrar jag med den kompetensen jag har annars skulle det bli tokigt.

SR: Men för att återgå lite till den förra frågan dom måste ju en då det kanske blir av kommunikation kanske fastän dom jobbar i samma sammansättningar i teamen så måste de väl vara insatta i vad de andra teamen gör? Jo men det, kanske inte vad alla gör men dom som man har behov av där det finns beroenden det är jätteviktigt

R3

IPC: Det konkreta är informationen är det stora projekt med många team så räcker det inte med tavlorna på väggarna utan då behöver man oftast något verktyg för att sprida informationen och ibland är det distribuerat när man inte sitter på samma ort och det är klart att behövs mer så det blir. Så en annan konkret skillnad är att det blir lite mer tungrovt ju fler man är så är det ju det märker man i andra sammanhang också

SR: man förlorar lite av det här agila

IPC: Det finns dom som säger att ju större projekt desto mindre lättörligt. Jag brukar uttrycka mig som så att desto mindre projekt desto större frihet kan du ge ju fler det är ju större det är desto mer begränsningar måste du ge till det enskilda teamet man har inte lika stort manöver utrymme.

SR men testning, kodning m m sköts på samma sätt här?

IPC: Ja det är så jag gjorde faktiskt en retreat på tveat Bob Martin eller onkel Bob Martin som är en software engeneering guru och han skrev något i stil med att ”oberoende av stort, litet kortsiktigt långsiktigt what ever så bra resultat kräver bra kodning från början till slut” och det ligger mycket i det har du en liten uppgift som du slarvar med blir det inte bra och har du en stor uppgift och slarvar så blir det inte bättre. Så disciplin är lätt att glömma bort och att jobba med scrum alltså lättörligt kräver mycket disciplin man förväntar sig hög disciplin på enskilda teammedlemmar och hela team också naturligtvis

R4

IPC: Flera definitivt jag brukar kalla det för release planering och det handlar om 2 saker som jag ser det och det en är att det är några som är intresserade av att veta hur det ligger till i det stora hela någonstans ska vi vara färdiga antingen så har vi den här mängden krav ska vi implementera och då är de intresserade att få veta när de kan få det eller så har vi lovat att leverera till ett visst datum och då kan de vara intresserade av att få veta hur mycket som vi verkligen vill ha kommer med till det datumet så att ha fast omfattning och fast datum fungerar väldigt sällan för man kan behöva variera någonting så därför tycker jag definitivt att en releaseplan som är på längre sikt än en kort sikt det är det ena. Det andra fallet är team medlemmar de behöver förstå lite vad som väntar runt hörnet också för dom kan behöva förbereda sig mentalt dom kan behöva göra någon, jag brukar använda XP-termen spyke känner ni till det? Det är att man gör en tidsbegränsad analys av ett tekniskt problem och inom XP så gör man det ofta genom att koda ,hacka ihop någonting för att förstå någonting och det är ett bra sätt att jobba i scrum också titta lite framåt att kunna hjälpa product owner att bryta ned det som skall komma om 2 eller 3 sprintar så man vet lite vad som väntar bakom nästa dörr också

SR men i stora projekt har man en stor sprint eller delar man in det i flera sprintar

IPC: Det där är en mer definitionsfråga vad du menar med det en eller det andra men det man vill åstadkomma det är att varje team gör ett åtagande så här mycket klarar vi av att göra under den här sprinten sen är tanken att det ska vara oberoende av varandra så att du ska kunna jobba intensivt med bara det så det är ändå då skulle jag säga att det lutar åt en sprint för varje team. Men sen kan det vara så att resultatet från team till ett annat team behöver integreras och det är att rekommendera att köra en kontinuerlig integration under sprintens gång så vi ser effekten av det alltså behövs det en samordning och ett samarbete också och det är inte alltid du klarar att ha oberoende utan ditt team och mitt team måste arbeta på något sätt.

R5

IPC: Det är teamets uppgift det är teammedlemmarna gemensamt som ska göra det . jag tycker enligt min erfarenhet att det fungerar bra det är klart att det är lite stötning och blötning i början när man inte är van vid det och det är likadant med det dagliga mötet att det har jag varit med om flera ggr att människor har varit vana att

jobba 1-2 veckor innan de behöver visa vad de har gjort och plötsligt är det någon konstig som heter scrummaster och säger att nu måste du berätta varje dag vad du har gjort. Och det krävs ju övning innan man klarar av det folk kan känna sig stressade och pressade om man inte är van vid det men det fungerar och det är enkla saker det går oftast att förklara vad som är meningen det styrkan och fördelen och nackdelen med att inte göra det.

R6

IPC: Ja det är konsten att inte släppa på det utan kravet är att har vi sagt att vi ska ha 4 veckors sprintar ja då är det 4 veckor som gäller och det är för att få in en fast takt i arbetet och det som händer, det kan hända en av två ytterligheter dels så har vi förmågan att jobba undan så fort att vi blir färdiga innan sprinten är slut och vad man säger då är att teamet ska ta något nytt från backloggen och överraska positivt eller så stöter vi på sådana svårigheter så vi klarar inte av att slutföra allt vi har åtagit oss inom ramen för sprinten och det är därför man har begreppet sprintmål . när man har ett tydligt övergripande mål för resultatet vad man ska uppnå för resultat under en sprint och oftast är det så att ett åtagande från ett team under en sprint det är större än sprintmålet som jag illustrerar en product backlog så här och så säger teamet att,(ritar på tavlan) product owner säger att målet med den här sprinten är detta då säger teamet bra då behöver vi göra följande det är inte alltid det blir strikt prioriteringsordning och sedan konstaterar man då att ja om vi gör dem så har vi ändå utrymme att göra någonting mer så då väljer vi utifrån prioriteringsordningen. Och har man problem under sprinten så använder man burndownchart och följer upp hur man ligger till och har man problem så syns det på burndownchart. Och då tittar man på hur mycket det återstår av sprinten för att uppnå sprintmålet och resterande uppgifter kan man ta bort det. Sen går man i mål med sprinten har längden lyckades inte göra allt vi hade åtagit oss men vi uppfyllde sprint målet och sedan får man justera sitt val utifrån det naturligtvis. Och i det stora hela om man tittar så är det oftast inte av avgörande betydelse om ett visst krav görs i sprinten eller i en annan sprint däremot så fort det händer vet vi någonting och det är därför takten är så viktig i det hela, vi ska få in en takt så att vi kan göra avstämningar för det ger oss kortare återkopplingscyklar. Sprintlängden ska man eftersträva att hålla fast och då kan det hända att man valde 4 veckors sprintar kanske vi hade fått bättre resultat om vi hade kört 2 veckors sprintar då är det ok att byta till det och köra med det.

R7

IPC: Man ska definitivt göra det för varje team med alla teammedlemmarna se så kan det behövs göras retrospectivt som går utanför teamgränserna för om vi är många team som ska samarbeta så räcker det kanske inte bara att titta på hur det fungerar inom mitt team eller någon annans team utan vi måste titta på samarbetet också och där får man hitta sätt att göra det på och jag tycker det är vettigt att kanske inte varje sprint nödvändigt vis men åtminstone varannan eller med relativt smidigt med utvalda representanter från teamet för att göra något övergripande retrospective också så i någon mening blir svaret på den frågan Ja. Nackdelen med att behöva göra det varannan gång är att komma ihåg så lång tid tillbaka är inte lätt 8 veckor vad hände för 8 veckor sedan. Det talar för att man ska göra det varje sprint. Jag tror man ska göra det varje sprint.

SR enligt statistik så lyckas scrumprojekt bättre än vanliga projekt i stora projekt blir det bättre med scrum då också eller har man förlorat det man tjänar annars?

IPC: Jag ser möjligheten att tjäna väldigt mycket på att använda scrum och agile även i stora projekt det är inte alla som lyckas skörda frukterna av det långt ifrån för det kräver rätt mycket, det kräver mycket av många människor det är det som är kruset. Vi kanske skulle kunna framföra riverdance vi fyra men om vi var fyra tusen kanske vi inte skulle lyckas

DI är det något du vill tillägga?

IPC: Jo som man säger om agile scrum också det är att ”the art of the possible” så därför är en av de absolut viktigaste sakerna i scrum och agile är att man hela tiden använder sig av inspect och adapt tittar på hur jobbar vi just nu kan vi göra något för att förändra det för ett bättre resultat, gör vi den ändringen granskar den och eventuellt en ny anpassning igen inspect och adapt och för att kunna göra en sådan granskning , inspect så krävs det en öppenhet och det är därför som ett av värdeorden i scrum är öppenhet vi måste visa upp saker precis som vi uppfattar dem och precis som de ser ut annars så hur ska vi kunna ändra dem till det bättre om vi inte visar upp hur det ser ut så det är det jag skulle vilja tillägga

Bilaga 6 – Tabeller och figurer

Tabell Nr.	Tabell Namn	Sida
2.1	Agile and Plan-Driven Method Home Grounds (Boehm & Turner, 2003)	10
2.2	The Five Critical Agility/Plan-Driven Factors (Boehm & Turner, 2003)	11
2.3	Ramverket för Scrum i Stora Projekt	23
4.1	Informanternas bakgrund och kompetens	27
4.2	Förändringen inom roller ur intervjupersonernas perspektiv	28
4.3	Förändringen inom artefakter ur intervjupersonernas perspektiv	29
4.4	Förändringen inom ceremonier ur intervjupersonernas perspektiv	31
4.5	Förändringen inom kommunikation ur intervjupersonernas perspektiv	31
4.6	Förändringen inom möten ur intervjupersonernas perspektiv	33
4.7	Förändringen inom styrning ur intervjupersonernas perspektiv	34
4.8	Förändringen inom befogenhet ur intervjupersonernas perspektiv	36
5.1	Förslag på tillvägagångssätt inom Arbetsprocess i stora Scrum projekt	38
5.2	Förslag på tillvägagångssätt inom Kommunikation i stora Scrum projekt	39
5.3	Förslag på tillvägagångssätt inom projektadministration i stora Scrum projekt	40
Figur Nr.	Figur Namn	
2.1	Scrum Uppbyggnad (Softhouse AB)	12
2.2	För- och nackdelar i kommunikation	16

REFERENSER

- Avison, D. & Fitzgerald, G. (2006) *Information system development: methodologies, techniques & tools* Berkshire: McGraw-Hill Education
- Boehm, B. & Turner, R. (2003) *Balancing Agility and Discipline – A Guide for the perplexed* Boston: Pearson Education
- Christensen, S. & Kreiner, K. (1997) *PROJEKTLEDNING – Att leda och lära I en ofullständig värld*. Lund: Academia Adacta
- Cleland, D. I. & King, R. W. (1968) *Systems analysis and project management*. New York: McGraw-Hill
- Coram, M. & Bohner, S. (2005) “The Impact of Agile Methods on Software Project Management” IEEE International Conference and Workshops on the engineering of Computer-Based Systems
- Cronholm, S. (2008) Using Agile Methods? – expected effects. Accepted to the 17th International Conference on Informations Systems Development (ISD2008)
- Earl, M. (1978) *Prototype systems for accounting, information and control*. *Accounting, Organizations and Society* Vol. 3 No. 2
- Engquist, A. (2002) *Kommunikation på arbetsplatsen Chefen, Medarbetaren, Gruppen* Stockholm: Prisma
- Gould, J.D., Boies, S.J. & Ukelson, J. (1997) *How to Design Usable Systems*, i Helander, Landauer & Prabhu (Red.), *Handbook of Human-Computer Interaction*, New York: Elsevier Science
- Gulliksen, J. & Göransson, B. (2002) *Användarcentrerad systemdesign*. Lund: Studentlitteratur
- Gustavsson, T. (2007) *Agile – konsten att slutföra projekt*. Karlstad: TUK Förlag
- Hanakawa, N. och Okura, K. (2004) *A project management support tool using communication for agile software development*. Proceeding of the 11th Asia-pacific Software Engineering Conference '04
- Highsmith, J. & Cockburn, A. (2001) Agile software development: the business of innovation, *IEEE Computer*, Vol. 34, No. 9, 120-122
- Jacobsen, D. I. (2002): *Vad, hur och varför? Om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen*. Lund: Studentlitteratur
- Kerzner, H. (2009) *PROJEKT MANAGEMENT – A System Approach to Planning, Scheduling, and Controlling*. 10th edition. New York: Wiley
- Kock, N. (2007) *System analysis & Design Fundamental*. California: Sage Publications

Kosh, K. (2005) *Agile Software Development Evaluating the Methods for Your Organization* Boston: Artechhouse

Kvale S. (1996): *Interviews: An Introduction to Qualitative Research Interviewing*. California: Sage Publications

Larman, C., Cockburn, A. & Highsmith, J. (2003): *Agile and Iterative Development: A Manager's Guide*. Boston: Addison Wesley

Lewis, J. (1998) *Team-Based management*. New York: Amacom

Paasivaara, M., Durasiewicz, S. och Lassenius, C. (2008) *Distributed Agile Development: Using Scrum in a Large Project*. IEEE International Conference on Global Software Engineering

Royce, W.W (1970) *Managing the development of large software systems*, Proceedings of IEEE Wescon.

Schwaber, K., (2004). *Agile project management with Scrum*. Washington: Microsoft Press.

Stober, T. & Hansmann, U. (2009) *Agile Software Development - Best Practices for Large Software Development Projects*, Berlin: Springer

Svensson L. & Casten von Otter (2001) *Projektarbete – teori och praktik*. Stockholm: Santérus Förlag

Turk, D., France, R. och Rumpe, B. (2002) Limitations of agile software processes, in 'In Proceedings of the Third International Conference on Extreme Programming and Flexible Processes in Software Engineering (XP2002), Springer-Verlag, p. 43–46.

VersionOne (2009) State of Agile Development Survey 2009
(<http://pm.versionone.com/StateOfAgileSurvey.html>)

Internet:

Agile adoption survey 2008: <http://www.ambysoft.com/surveys/agileFebruary2008.html>
[besöktid 28-4-2010, kl. 11:00]

Agile manifesto: <http://agilemanifesto.org/principles.html> [besöktid 29-4-2010, kl. 10:00]

Computer Sweden: <http://computersweden.idg.se/2.2683/1.306713/jattarna-blir-agila>
[besöktid 11-4-2010, kl. 09:00]

Computer Sweden: <http://computersweden.idg.se/2.2683/1.306718> [besöktid 11-4-2010, kl. 09:00]

Computer Sweden: <http://computersweden.idg.se/2.2683/1.306715> [besöktid 11-4-2010, kl. 09:00]

Scrum uppbyggnad: http://www.softhouse.se/c/document_library/get_file?uuid=4e2b533f-5068-4db0-aaa5-3f340a0fe980&groupId=10222 [besöktid 3-5-2010, kl. 13:00]

VersionOne (2009) State of Agile Development Survey 2009

<http://pm.versionone.com/StateOfAgileSurvey.html> [besöktid 25-03-2010, kl. 09:00]