



LUNDS UNIVERSITET

Ekonomihögskolan

Agila metoder – Hur teori omvandlas till faktiska arbetssätt

Kandidatuppsats, 15 högskolepoäng.
Institutionen för Informatik, Lunds Universitet.

Framlagd: Juni, 2012

Författare: Johan Kvarnryd
Henrik Åkerberg

Handledare: Nicklas Holmberg

Examinatorer: Markus Lahtinen
Claus Persson

Abstrakt

Uppsatsen berör hur organisationer går till väga för att tolka och anpassa agila arbetsmetoder för att dessa ska passa verksamheten på ett fördelaktigt sätt. Det finns en mängd metoder att utgå ifrån för att uppnå ett agilt arbetssätt, de företag vi varit i kontakt med har alla nämnt att det är metoderna Scrum och Kanban som oftast används. Vi undersöker hur processen går till när ett beslut tas för att följa en viss metod, vi har därför intervjuat personer med erfarenhet av en sådan beslutssituation för att få ett uttalande om tillvägagångssättet. I studien undersöks också hur den agila metoden och således också arbetssättet sprids till övriga intressenter inom organisationen. Vi behandlar relevanta delar i teoriavsnittet så som metoder för utveckling, projektledning och kommunikationsteorier. Vi har upptäckt att organisationernas arbetssätt skiljer sig från metodens teoretiska arbetssätt. Genom att analysera skillnaden har vi utifrån metodval, metodanvändning, kombination av metoder, produktägare och kundperspektiv, formellt och informellt arbetssätt samt kommunikationsstruktur resonerat kring organisationers arbetssätt.

Innehållsförteckning

1 Inledning	1
1.1 Bakgrund	1
1.2 Problemformulering	3
1.3 Forskningsfråga	3
1.4 Syfte	3
1.5 Avgränsningar	4
2 Teori.....	5
2.1 Projektadministration	5
2.2 Vattenfallsmodellen	5
2.3 Agil systemutveckling.....	7
2.3.1 Lean Software Development	8
2.3.2 Kanban.....	10
2.3.3 Scrum	10
2.3.4 Extreme Programming	12
2.3.5 Feature-Driven Development	15
2.3.6 Crystal	15
2.3.7 Sammanställning av agila metoder	16
2.4 Kommunikation och kunskapsspridning.....	17
2.4.1 Problematik vid kunskapsspridning	18
2.4.2 Kommunikationsprocessen.....	19
2.4.3 Kommunikationskanaler	20
2.5 Förändringsarbeten	21
2.6 Liknande studier av agila metoder.....	22
3 Metod.....	24
3.1 Tillvägagångssätt	24
3.2 Typ av intervjuer.....	25
3.3 Urval av intervjupersoner.....	26
3.4 Intervjuguide.....	27
3.5 Etiska aspekter i en undersökning	28
3.6 Reliabilitet och validitet	29
4 Empiriskt resultat.....	30
4.1 Metodval	30

4.2 Metodanvändning	31
4.3 Kombination av metoder.....	32
4.4 Produktägare och kundperspektiv.....	34
4.5 Formellt och informellt arbetssätt.....	34
4.6 Kommunikationsstruktur.....	35
5 Diskussion.....	37
5.1 Metodval	37
5.2 Metodanvändning	38
5.3 Kombination av metoder.....	39
5.4 Produktägare och kundperspektiv.....	40
5.5 Formellt och informellt arbetssätt.....	41
5.6 Kommunikationsstruktur.....	42
6 Slutsats	43
7 Bilagor	45
Bilaga 1, Intervjufrågor.....	45
Bilaga 2, Intervjuprotokoll, P1	46
Bilaga 3, Intervjuprotokoll, P2	51
Bilaga 4, Intervjuprotokoll, P3	64
Bilaga 5, Intervjuprotokoll, P4	71
8 Referenser	81

Figurförteckning

Figur 2. 1 Scrum.....	11
Figur 2. 2 XP-flödesschema.....	14
Figur 2. 3 Process för kunskapsspridning.....	18
Figur 2. 4 Lärandeprocessen	19
Figur 2. 5 Kommunikationsprocessen.....	20

1 Inledning

1.1 Bakgrund

Den miljö som organisationer agerar i idag blir allt mer komplicerad. Denna komplexitet skapar både problem och möjligheter (Turban, Aronson, Liang & Sharda, 2010) I denna kontext ställs det även allt större krav på mjukvaruutvecklare att förbättra kvalitén på sina produkter. Antagandet att mjukvaruutvecklingsprocessen direkt påverkar produktens kvalitet har motiverat många organisationer att utveckla sina metodval (Magdaleno, Lima Werner & Mendes de Araujo, 2011).

Plandrivna utvecklingsmetoder och kvalitetsstandarder har implementerats för att göra utveckling mindre kaotiskt och mer förutsägbart. Agil utveckling har uppstått som ett alternativ till plandrivna utveckling. (Magdaleno et al, 2011) Enligt Magdaleno et al, (2011) har agil utveckling korta utvecklingscykler och hög kundinvolvering för att underlätta anpassning till en föränderlig situation. Traditionell systemutveckling har visat sig ineffektiv för att uppnå utsatta mål, därför har agil utveckling framstått ur frustrationen som funnits kring gamla metoder med strikta fasindelningar och hög dokumentationsgrad. (Bhalerao & Puntambekar, 2009)

År 2001 etablerades ett manifest, utifrån ett antal gemensamma principer, som ligger till grund för vad agil utveckling är idag (The Agile Manifesto). De principer som ligger bakom detta manifest är enligt The Agile Manifesto följande:

”Vi följer dessa principer:

- *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.”*

Dessa principer ligger till grund för framförallt Extreme Programming (XP), Scrum, Feature Driven Development (FDD), Crystal Methodology, Dynamic System Driven Development (DSDM), Adaptive Software Development (ASD), Open Source (OS), Agile Modeling (AM), och Pragmatic Programming (PP). Det har observerats att alla dessa metoder gör sig bäst i mindre utvecklingsprojekt. Kritiker har uttryckt att agila metoder inte har några klara bakomliggande riktlinjer och istället bygger på en ad-hoc inställning till utveckling. Det har också uttryckts att det är svårt för den genomsnittlige utvecklaren att förstå och helt ut använda agila metoder vid systemutveckling. (Bhalerao & Puntambekar, 2009)

Oavsett inställning så har många forskare konstaterat att agil utveckling är väldigt populärt, men också att det innebär svårigheter att anpassa metoderna till specifika situationer. Särskilt om utvecklingsprojektet i fråga är av större karaktär (Elshamy & Elssamadisy, 2007) eller om man vill kombinera flera utvecklingsmetoder (Magdaleno et al, 2011). Vid större projekt är det vanligt att dela upp utvecklingen i delprojekt. Då uppstår dock ett behov för ett ramverk för hur dessa ska koordineras och bilda en helhet. (Elshamy & Elssamadisy, 2007)

De antaganden som ligger till grund för agila metoder stämmer inte nödvändigtvis vid t.ex. storskalig systemutveckling. Det krävs en bredare förståelse för uppbyggnad av organisationer utifrån ett agilt tankesätt. Att bara förändra utvecklingsprocessen tar inte i beräkning att denna process ska passa in i ett större sammanhang. För att få en agil organisation måste flera dimensioner kombineras. (Kettunen & Laanti, 2007) Om en korrekt anpassning sker, har det visat sig att agil utveckling kan ge avsevärda fördelar, även i projekt med hög komplexitet eller begränsade resurser. Detta stämmer dock främst på projektnivå. Agila organisationer fortsätter att vara problematiska. (Mishra & Mishra, 2011)

Agila metoder bygger ofta på att de appliceras i en specifik kontext eller att de är tillräckligt flexibla för att kunna anpassas till situationen av utvecklarna själva (Stepanek, 2005). Detta får oss att ifrågasätta hur dessa metoder tolkas, anpassas och ibland även kombineras för att passa in i specifika situationer och bilda ett konkret arbetsätt för organisationer. Vi anser att en extremt detaljerad metod endast kan passa i en kontext. Alla andra behöver specificeras för att passa den specifika kontext de används i. Detta leder oss till antagandet att organisationer alltid gör någon form av översättning från teori till praktik. Denna översättning kan ske formellt eller informellt, men måste enligt oss göras i någon form förr eller senare.

1.2 Problemformulering

Det senaste decenniet har ett flertal agila utvecklingsmetoder utvecklats, baserade på The Agile Manifesto från 2001 (Bhalerao & Puntambekar, 2009). Enligt Lalsing, Kishnah & Pudaruth (2012) har populariteten för de agila metoderna ökat, det medför att många systemutvecklingsföretag rör sig bort från äldre metoder, som exempelvis vattenfallsmodellen, för att istället införa agila utvecklingsmetoder. De traditionella metoderna har relativt förutsägbara steg att följa, medan agila metoder är mer anpassningsbara (Lalsing et al, 2012).

Jämfört med äldre metoder där det har funnits klara direktiv om vilken uppgift en projektdeltagare ska arbeta med, är det inom den agila utvecklingen viktigt att projektdeltagarna har stora kunskaper om de olika arbetsinriktningar som existerar i projektet, eftersom de kan behöva utföra en mängd olika arbetsuppgifter (Stober & Hansmann, 2010).

De traditionella arbetsmetoderna är uppbyggda kring en hierarkisk struktur och karaktäriseras av att vara dokumentationsdrivna. I kontrast till det är de agila metoderna kommunikationsdrivna, vilket sätter höga krav på en väl fungerande kommunikation inom organisationen och främst bland projektdeltagarna. (Stober & Hansmann, 2010)

Singh (2012) skriver att trots de agila metodernas popularitet så finns det fortfarande problem som måste behandlas innan metoderna kan användas till sin fulla kapacitet och för att de ska passa alla typer av miljöer. Han har exempelvis identifierat kommunikationsproblem och oklara krav för användning av metoderna som problem som måste beaktas.

1.3 Forskningsfråga

Problemformuleringen leder fram till vår forskningsfråga.

Hur anpassas agila metoder till att fungera i en organisations verksamhet och hur kommuniceras arbetssättet till medarbetare?

1.4 Syfte

Genom att undersöka hur organisationer arbetar med anpassning av utvecklingsmetoder och kommunikation inom organisationen, vill vi beskriva hur agila utvecklingsmetoder används i agila systemutvecklingsprojekt. Vi avser med denna studie öka förståelsen för hur agila metoder tolkas, anpassas och kombineras för att i praktiken fungera med en organisations verksamhet.

1.5 Avgränsningar

Vi kommer endast undersöka hur översättningen av agila metoder till praktik utförs. Vi kommer inte undersöka om dessa metoder är effektiva eller hur arbetssättet är uppbyggt i detalj. Vår frågeställning utgår ifrån denna översättning och möjligtvis dess bidrag till ett effektivt arbetssätt, men arbetssättets effektivitet i sig är inte av intresse för vår undersökning.

2 Teori

2.1 Projektadministration

Organisationer som arbetar med projekt delar ofta in dessa i projektfaser för att förbättra den verksamhet som arbetar med projektet. Faserna kallas för projektets livscykel. (Deenen, 2007) Enligt Macheridis (2009) kallas dessa faser för definitionsfas, planeringsfas, genomförandefas samt reflektionsfas.

Definitionsfasen innebär att en förstudie genomförs för att bestämma om uppdraget ska accepteras eller inte. Arbetet leder till ett projektbeslut och fasen avslutas med att ett kontrakt mellan beställare och uppdragstagare inrättas. Planeringsfasen går ut på att skapa en struktur för projektet och bestämma en deadline för när det ska vara färdigt samt att milstolpar och beslutspunkter under projektets livscykel fastställs. Vidare är genomförandefasen den fas i livscykeln där projektet aktivt utförs. I den sista fasen, reflektionsfasen, sker överlämnandet av projektresultatet till beställaren och projektgruppen upplöses. Det är här viktigt att reflektera över hur arbetet har utförts, eftersom reflektion leder till kompetensutveckling. (Macheridis, 2009)

Enligt Christensen & Kreiner (1997) bör utvärderingen i reflektionsfasen ge en rättvisande bedömning av det genomförda arbetet med avseende på vad som var ämnat att göras och vad som faktiskt utfördes i projektet. Författarna talar om substantiell utvärdering, vilket är menat att svara på ifall projektets resultat är användbart och optimalt i ett vidare sammanhang för organisationen eller beställaren. Utvärderingen ska ses ur ett strategiskt perspektiv för att kunna besvara vilken kultur och motivation som ska eftersträvas vid framtida projekt. Det är enligt Macheridis (2009) viktigt att i reflektionsfasen planera för vilka framtida projekt medarbetarna ska engageras i samt att projektresultatet överförs till beställaren på ett sätt som inte inkluderar fortsatt ansvarstagande från de som arbetat med lösningen och kommit fram till resultatet.

Arbetet med utvärderingen kan användas som ett beslutstillfälle för att fatta beslut om projektets problemrelevans och betydelse, detta för att ge organisationen en investering i en lärprocess som vid framtida projekt besparar organisationen en nyttolös implementering av ett vanligt förekommande problem. I den här fasen kan organisationen avgöra om fler resurser ska satsas på problemets lösning eller inte. (Christensen & Kreiner, 1997)

2.2 Vattenfallsmodellen

Projektledning är inget nytt koncept. Det kan antas att projekt så som byggandet av pyramiderna i Egypten och Sydamerika krävde någon form av projektledning. Den moderna projektledningen kan spåras till 1950-talet, och här finns det en dokumenterad och noga motiverad form av projektledning. Framförallt militära projekt ledde fram till en rad

schemaläggningstekniker för projektledning. Dessa byggde huvudsakligen på sekventiella uppgifter, händelser och milstolpar. Dessa tekniker kan liknas vid exempelvis dagens processmodeller. (Stober & Hansmann, 2010)

Den mer moderna varianten av dessa föråldrade metoder kallas för vattenfallsmodellen. Denna metod passar för både stora och små projekt. Till grund ligger fem faser: krav, design, implementering, testning och support. Varje fas måste avslutas helt innan nästa fas påbörjas. Varje projekt börjar med kravfasen. Först identifierar utvecklarna intressenterna och tar kontakt med de som bedöms vara viktiga för att projektet ska lyckas. Intressenterna får sedan komma med önskemål på det system som projektet ska mynna ut i. Utifrån detta underlag i kombination med resurser och kunskap gör utvecklarna en analys som leder till en kravspecifikation som definierar det färdiga systemet. Tillsammans med intressenter lägger utvecklarna sedan upp en tidsplan för arbetet och en plan för hur intressenterna ska fortsätta att involveras. Det är väldigt viktigt att kravspecifikation och planering är noga dokumenterade och att det efterföljs ordentligt. (Stober & Hansmann, 2010)

Designfasen går som namnet antyder ut på att ta fram en noggrann design av systemet. Denna design ska sedan direkt kunna översättas till kod i nästa fas. Denna fas involverar ofta t.ex. användarfall, UML-diagram och flödesscheman. Det är viktigt att identifiera exakt hur systemet ska underlätta eller automatisera processer för den slutgiltiga kunden. (Stober & Hansmann, 2010)

När designen är klar påbörjas implementeringen. Här översätts designen till kod, komponent för komponent, eller objekt för objekt. Komponenterna testas var för sig. I slutet av kodningsfasen eller början på testningsfasen integreras komponenterna till en helhet. Här tenderar överraskningar att framstiga, eftersom komponenterna inte alltid interagerar som man förutsett. Det är oftast i denna fas som det uppstår problem gentemot planeringsschemat. Att utveckla ett perfekt system är oftast en önskan men samtidigt en omöjlighet. Dessutom kan lösningar av oförutsedda problem dra ut på tiden. (Stober & Hansmann, 2010)

Testfasen kan ibland vara svår att direkt skilja från implementeringen. Testningen går ut på att identifiera problem som uppstått under kodningen. Idealet är att hitta alla oegentligheter och fixa dessa så att systemet fungerar som det ursprungligen var avsett att göra. Ofta tvingas avvägningar göras mellan att hitta alla buggar som finns och färdigställa systemet inom avsatt tid. Det tenderar att vara svårt att få en överblick över testningen. En vanlig observation är dock att högre funktionalitet och en bredare plattform leder till en längre och mer invecklad testning. (Stober & Hansmann, 2010)

När systemet väl har levererats till kunden ska det också underhållas. Detta är supportfasen. Support delas ofta upp i tre nivåer. Nivå ett består i många fall av en telefoncentral. De tar emot samtal, löser enklare problem och vidarebefordrar svårare problem till lämplig instans. Nivå två är specialister som kan lösa svårare problem. De är experter på ett eller ett antal system, i vissa fall kan de till och med vara specialiserade på en del av ett visst system. Nivå tre hanterar problem indirekt. De löser buggar och vidareutvecklar systemet i s.k. patchar.

(Stober & Hansmann, 2010)

Fördelen med vattenfallsmodellen är dess klara och effektiva struktur. Vad som ska presteras bestäms tidigt och sedan fokuseras allt arbete på att producera detta resultat. (Stober & Hansmann, 2010) Nackdelen är samma struktur. Det går inte att anpassa systemet till ändrade omständigheter. Detta gör metoden högst problematisk, speciellt i längre projekt där situationen utvecklas avsevärt från projektstart till leverans. Därför har vattenfallsmodellen i många fall visat sig ineffektiv för att uppnå utsatta mål. (Bhalerao & Puntambekar, 2009)

Vattenfallsmodellen bygger också på det felaktiga antagandet att varje fas är fullt sekventiellt beroende av den föregående. För att fullt avskilja dessa faser så måste varje fas producera någon form av artefakt som kan överföras till nästa. Detta kan vara ett dokument, en plan, etc. Implementeringsfasen är den enda fas som producerar något som faktiskt är värdefullt för kunden, d.v.s. ett system. Därför är vattenfallsmodellen inte effektiv, den fokuserar väldigt lite på det arbete som faktiskt genererar värde för kunden. (Stepanek, 2005)

För att möta de brister som observerats kring vattenfallsmodellen så har flera modifierade modeller utvecklats. En är att dela in implementeringen i flera delfaser med individuella milstolpar. Detta ger en klarare bild över var i utvecklingen man befinner sig och tillåter också testning redan innan all kodning är avslutad. En vidareutveckling av denna tanke är att arbeta med en inkrementell vattenfallsmodell. Projektet delas då in i delprojekt som alla går igenom de första fyra faserna innan nästa påbörjas. Detta gör att krav och designfasen upprepas och utvecklingarna kan anpassa systemet till nya utvecklingar och lärdomar efter hand som det utvecklas. (Stober & Hansmann, 2010)

2.3 Agil systemutveckling

Traditionell systemutveckling med hög fokus på planering och dokumentation har visat sig ineffektivt, och bristfälligt vad gäller reaktivitet och kundfokus. Som en reaktion har agila metoder växt fram ur den frustration som bildats. (Bhalerao & Puntambekar, 2009)

Agil projektledning kan spåras till en artikel av Takeuchi & Nonaka i Harvard Business Review i januari 1986 (Cervone, 2011). Det agila tankesättet visade upp konkreta resultat med Kaizen inom Toyota som sedan ledde vidare till Lean production, Kanban och just-in-time (JIT) (Stober & Hansmann, 2010).

Idén fick dock inte fäste inom systemutveckling förrän med Sutherland & Schwaber på en konferens 1995. De diskuterade då bristerna med traditionell systemutveckling och den första agila systemutvecklingsmetoden. Idag existerar flera olika former av agila utvecklingsmetoder, men alla bygger på samma grundidéer som tvärt avviker från traditionell utveckling. (Cervone, 2011)

2001 enades en rad ledande tänkare inom agil utveckling om ett gemensamt manifest:
"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan"
(The Agile Manifesto)

Agil projektledning har sina rötter i detta manifest men har anpassats något för att passa projektledning istället för systemutveckling. Agil projektledning framhäver två viktiga aspekter. Den första är att korta iterationer minimerar risker. Den andra är att direkt kommunikation bör prioriteras över utförlig projektdokumentation. Anledningen bakom dessa aspekter är att de hjälper projektgrupper att anpassa sig till de oförutsägbara och föränderliga krav som utvecklingsprojekt ställs inför. (Cervone, 2011)

2.3.1 Lean Software Development

Enligt Petersen & Wohlin (2011) är Lean inte en agil metod, den är dock nära besläktad och fungerar väl i kombination med agil utveckling. Det som skiljer Lean från agila metoder är dess fokus på "början till slut" och det värdeskapande flödet däremellan. Lean systemutveckling enligt Stober & Hansmann (2010) bygger på s.k. "best practices" d.v.s. riktlinjer som har visat sig fungera i verkligheten. Lean består av sju sådana riktlinjer:

1. **Eliminera slöseri** – Slöseri kan uppstå på många vis inom systemutveckling. Ingenjörer tenderar att fokusera för mycket på briljant teknologi, till den punkt där de glömmet det faktiska värdeskapandet för verksamheten. All utveckling som inte relaterar till kundvärde bör undvikas. Slöseri gäller även för projektadministration. Utförliga kravspecifikationer för funktioner som aldrig implementeras är exempel på slösad tid. Även planer inför en osäker framtid kan ses som slösad tid om den inte är klart kopplad till arbete som kommer skapa värde i någon form. Bygger planerna på osäkra antaganden är också deras värde osäkert.
2. **Fokusera på lärande** – Planering är användbart till en viss gräns, men lärande är oundgängligt. Prototyper är en källa för feedback och förbättring. Lärande inkluderar också att analysera misslyckanden, undersöka orsaken och försäkra sig att de inte kommer uppstå igen. En vanlig teknik är "5W". Den innebär att man måste fråga sig "Varför hände detta?" fem gånger för att nå den verkliga orsaken. Det är viktigt att ha attityden att varje misslyckande är en möjlighet till att förbättras och lära sig av.

3. **Bygg in kvalitet** – Om testningen visar alltför många defekter så fungerar den övergripande processen inte som den ska. Testdriven utveckling leder till fungerande kod. Genom att integrera kod successivt undviker man den oundvikliga mardrömmen som uppstår när all kod integreras i efterhand. Kvalitet är dock inte bara felfri kod. Kvalitet antyder att kundens förväntningar har mötts. Därför är det viktigt att förstå kundens definition av värde.
4. **Skjut upp förbindelse** – Eftersom ingen kan skapa en perfekt projektplan måste utvecklingen läggas upp i små delar, istället för en full specifikation. Detta kräver att man delar upp åtaganden mellan grupper så att de kan göra framsteg på egen hand. Systemets övergripande struktur bör ha en komponentbaserad struktur som tillåter att funktioner läggs till utan förbehåll. Utmaningen är att hålla systemet öppet för utvecklingar och förändringar. Definitiva beslut bör skjutas upp så länge som möjligt och tas utifrån välgrundad kunskap.
5. **Leverera snabbt** – Att överbelasta utvecklarna saktar ner framstegen. Utvecklingsteam är som datorer, långa köer saktar ner arbetet. Inkrementella leveranser minimerar köbildningen av uppgifter och låter utvecklarna fokusera helt på den uppgift de har för tillfället. En jämn pålitlig arbetstakt är fullt kompatibel med snabba leveranser, hög kvalitet och låga kostnader.
6. **Respektera människor** – En motiverad grupp med ett gemensamt mål har ett hållbart övertag. Detta antyder ett effektivt ledarskap och ett positivt arbetsklimat. Ett sådant klimat släpper fram kreativa talanger. Alla intressenter och deltagare i ett projekt bör agera mot ett gemensamt mål.
7. **Optimera helheten** – Individuella uppgifter eller funktioner har inget annat värde än i vad de bidrar med till helheten. I slutändan är det ett system som ska levereras till kunden och detta system kommer skapa kundvärde som en helhet. En design kan vara bra, men om den inte kombineras med en funktionell produkt mister den sitt värde helt.

Många av dessa riktlinjer är svåra att genomföra i realiteten. De är mer av en vision att jobba efter än ett mål som faktiskt går att uppnå. Exempelvis krävs det mycket erfarenhet mjukvaruutveckling för att kunna dela upp arbetet i små uppgifter och eliminera de onödiga. Speciellt eftersom samma produkt endast utvecklas en gång till skillnad från det ursprungliga användningsområdet bilproduktion. (Stober & Hansmann, 2010)

Fallstudier har visat att Lean är effektivt för att upptäcka problem och korrigerera dessa. Utvecklingsteam som använder en hybrid av agil och Lean utveckling producerade dessutom mer kod än de som använder bara agila metoder. I längden upptäckte de också allt färre problem. (Petersen & Wohlin, 2011)

2.3.2 Kanban

Kanban är ett senare tillskott till Lean. Det går att använda Kanban utan Lean men de är i grunden ett gemensamt tankesätt som började med Kaizen inom Toyota. Grunden i Kanban är flödeshantering inom bilindustrin. (Suárez-Barraza & Ramis-Pujol, 2010) Ett pull-flöde förespråkas framför ett mer traditionellt push-flöde. Detta innebär att bilarna inte följer ett klassiskt löpande band. Istället läggs objekt åt sidan när en avdelning är klar med dem, och plockas ut av en annan avdelning när de har användning för det. (Jyothi & Rao, 2012) Kanban har sedan transplanterats för att användas inom systemutveckling. Det grundläggande målet har förblivit detsamma, att förebygga överbelastning och flaskhalsar. (Turner, Ingold, Lane, Madachy & Anderson, 2012)

Kanban är ett sätt att grovt schemalägga sitt arbete på ett hållbart vis. Denna schemaläggning läggs upp på exempelvis en anslagstavla, gärna på en gemensam plats där alla berörda kan se den. På anslagstavlan sätts ett antal kolumner ut baserat på faser, så som mål, köläggning, avslutad, etc. Sedan skrivs uppgifter upp på kort, exempelvis post-it-lappar, och delas in i kolumnerna. Tanken är att ingen kolumn får ha mer än ett visst antal uppgifter åt gången. Uppgifter ska avslutas innan man tar sig an nya, och skulle alltför många uppgifter ansamlas i en kolumn ska alla hjälpas åt att avsluta dessa och få ner antalet till önskvärd nivå. På detta vis undviks dels att en person blir överbelastad, men också att andra inte tvingas vänta på att en person ska avsluta uppgifter som deras uppgifter är sekventiellt beroende av. (Turner et al, 2012)

2.3.3 Scrum

Scrum introducerades av Takeuchi, DeGrace, Schwaber, m.fl. i mitten på 90-talet. Det bakomliggande konceptet är en förenkling av projektadministration. Scrum bygger på en flexibel process med iterativ utveckling. Den består av tre roller, tre dokument och tre möten. (Stober & Hansmann, 2010) Rollerna är Scrum-teamet, Produktägaren samt Scrum-mästaren (Hneif & Hock Ow, 2009).

Ett Scrum-team är uppbyggt av 5-10 stycken projektmedlemmar och det är dessa som arbetar med projektet på heltid. Teamet är självorganiserat och det finns inte inom gruppen någon bestämd ledarroll. Medlemmarna i teamen kan byta plats med varandra, men det sker endast mellan sprintarna, vilket vi återkommer till senare. (Cervone, 2011)

Produktägaren är beställare, oftast en kund, men produktägaren kan även vara en person som arbetar i den egna organisationen. Han eller hon har till uppgift att säkerställa att teamet arbetar med väsentliga saker utifrån ett affärsperspektiv. Produktägaren har också till uppgift att förvalta och prioritera uppgifter inom projektets Product Backlogg, vilken arbetas fram i samarbete med övriga Scrum-teamet. Vi återkommer till innebörden av Product Backlogg senare. (Softhouse)

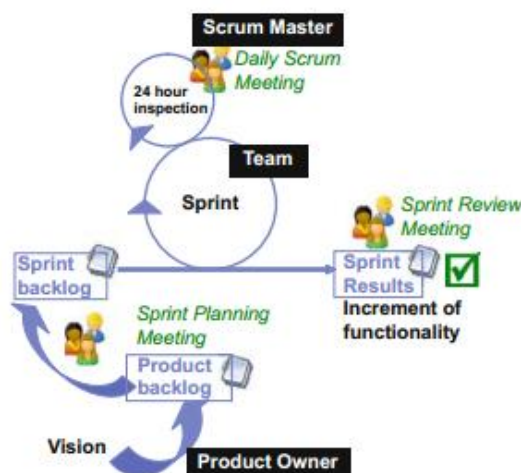
Scrum-mästaren har som uppgift att administrera Scrum-teamet. Han eller hon ansvarar för att dagligen kalla teamet till ett möte som kallas Daily Scrum Meeting. Koncentrationen i Scrum-mästarens arbete ligger i att ge teamen de bästa förutsättningarna för att projektet ska drivas framåt och att de uppsatta målen ska kunna nås. (Softhouse)

Ett centralt begrepp inom Scrum-metoden är Product Backloggen. Den innehåller en prioriterad lista över alla objekt som är relevanta för en specifik produkt. Listan kan bestå av buggfixar, förbättrade krav från kunden, funktionalitet för en mer konkurrenskraftig produkt och uppgraderingar av teknik. De högst prioriterade kraven överförs till en Sprint Backlogg, vilket är grunden till vad som kommer ske i det faktiska arbetet. Kraven som kommer från Product Backloggen styckas upp i mindre uppgifter som sprids ut bland projektmedlemmarna, det är dessa uppgifter som nu utförs i den så kallade Sprinten. (Vlaanderen, Jansen, Brinkkemper & Jaspers, 2011)

När ett planeringsmöte för sprinten har hållits, kan arbetet i sprinten starta. En sprint skiljer sig från de faser som finns i ett traditionellt projekt, en sprint är begränsade till en månads lång iterationscykel, i varje sprint utvecklas funktionaliteten hos produkten ytterligare. Under arbetet i en sprint bör ingen yttre påverkan tillåtas som kan riskera att störa Scrum-teamets arbete. (Cervone, 2011)

I många projekt, men inte nödvändigtvis i alla, börjar varje sprint med ett Daily Scrum Meeting. Mötet hålls varje dag och pågår runt 15 minuter. Närvarande vid mötet är Scrum Mastern, som har ordförandeskapet, och Scrum-teamet. Syftet med de dagliga mötena är att säkerställa att arbetet fortskrider på ett ändamålsenligt och obehindrat sätt samt att projektdeltagarna kan framföra sina åsikter. För att analysera eventuella problem ska gruppmedlemmarna svara på följande tre frågor:

- Vad har du gjort sedan förra mötet?
- Vad tänker du göra inför kommande möte?
- Är det något som hindrar dig från att utträtta ditt planerade arbete? (Cervone, 2011)



Figur 2. 1 Scrum (Stober och Hansmann, 2010, s. 43)

Stober & Hansmann (2010) skriver att effektiv användning av Scrum kräver erfarenhet och planering. Flera av komponenterna inom Scrum behöver specificeras utifrån den givna organisationen eller den givna situationen. Exempelvis behöver en sprints längd anpassas till projektets behov. Detta blir extra viktigt eftersom sprintarnas längd inte får ändras när de väl påbörjats, och inga uppgifter ska vara ”nästan” avslutade. Uppgifter ska vara avslutade eller skjutas upp helt till nästa sprint.

Enligt Cervone (2011) är fördelarna med agil projektledning och i synnerhet med Scrum-baserade tillvägagångssätt dess enkelhet. Nya funktioner kan utvecklas och testas i korta iterationscykler och eftersom varje medarbetare har ett stort ansvar för sin del i projektet, får projektledaren en bredare, mer översiktlig bild över projektet. Genom att driva ett projekt efter ett agilt tankesätt uppnås det en omfattande kommunikation inom projektet, det bidrar till att teamet blir organiserat och effektivare. I slutändan kan detta leda till en ökad produktivitet för alla inblandade parter. Vidare säger Hneif & Hock Ow (2009) att studier visar att Scrum-metoden inte är, trots sina fördelar, lämplig att använda för produkter när tyngdpunkten ligger på användbarhet. Metoden misslyckas med att identifiera användarbehovet hos användaren. Det sker eftersom produktägaren främst koncentrerar sig på företagsfrågor och därmed glömmar bort användarbehovet.

2.3.4 Extreme Programming

Enligt Stober & Hansmann (2010) och Stepanek (2005) bygger extrem programmering (XP) på flera koncept. De är inte nya men har tagits till en ”extrem” nivå. De är som följer:

- **Testdriven utveckling** bygger på att ett test tas fram direkt efter en funktionalitet har definierats för utveckling. När själva utvecklingen av funktionaliteten sedan sker kan den ständigt testas mot det testfall som togs fram som en direkt koppling till kravspecifikationen. Fördelen med testdriven utveckling är att den ger omedelbar och konkret feedback.
- **Programmering i par.** Att programmera i par ger fördelar i form av klarare perspektiv. Fyra ögon ser mer än två och det finns alltid någon som kan ifrågasätta irrationellt tänkande. Det krävs dock vana för att arbeta effektivt i par.
- **Refactoring** är en term som innebär att man ändrar kod utan att ändra funktionalitet. Det kan innebära att man effektiviserar koden eller helt enkelt rensar upp den och gör den mer överskådlig. Refactoring är en aktivitet som bör utföras kontinuerligt, det är dock viktigt att alltid utföra testning efteråt för att försäkra att koden fortfarande fungerar som den ska.
- **Enkelhet.** Utvecklingen ska hela tiden fokusera på funktionalitet. Ingen kod ska skrivas som inte är direkt nödvändig i nuläget. Om kod tros behövas imorgon ska den inte skrivas idag, situationen kan förändras innan dess.

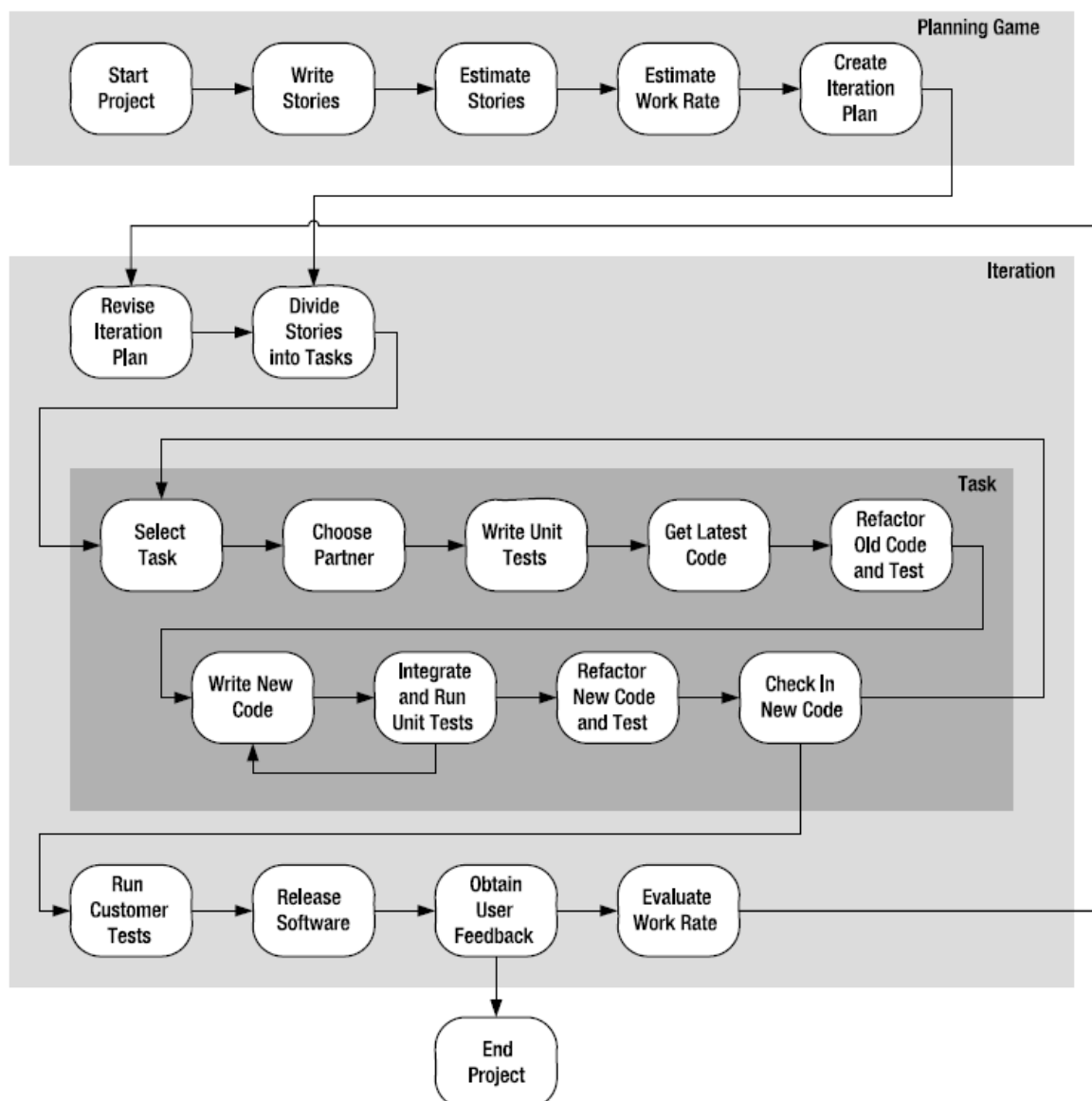
- **Planering.** I början av ett projekt läggs en grov plan upp för vad som ska levereras och när. Hela teamet och kunden närvarar vid möten där denna plan läggs upp. Denna plan förändras efter hand men fungerar som ett ungefärligt mål att arbeta mot. XP bygger på tvåveckorsiterationer som resulterar i leveransfärdig kod, komponent för komponent.
- **Små leveranser.** Varje iteration ska leverera ett fungerande system som kan ges till kunden. Kunden ska kunna stoppa utvecklingen efter vilken iteration som helst och ändå ha ett fungerande system. Det ska också vara möjligt att byta riktning och prioritera om funktioner efter varje iteration.
- **Kontinuerlig integration** är en naturlig följd av små frekventa leveranser. Systemet ska efter varje iteration fungera helt och integration måste därför vara ett ständigt arbete.
- **Kontinuerlig testning.** Frekvent och automatiserad testning försäkrar att koden fungerar som den ska och gör att det inte uppkommer några oförutsedda problem senare i utvecklingen.
- **Kollektivt ägarskap av kod.** Alla utvecklare ska kunna bidra till all kod. Om en person anser sig kunna bidra så ska detta inte hindras. Detta bygger på små team om ca 10-12 utvecklare som mest.
- **Hållbar takt.** Alltför hög arbetstakt fungerar bara kortsiktigt och gör att arbetet lider på längre sikt. Hållbar takt är mest effektiv i längden.
- **Kodningsstandarder.** För att upprätthålla kvalitet och försäkra sig om att man senare kan underhålla systemet är det viktigt att hålla sig till gemensamma konventioner för kodning. Riktlinjer för dessa ska vara enkelt åtkomliga för alla projektdeltagare.
- **Kontinuerlig kundkontakt.** Det är vitalt att involvera kunderna i projektet. Endast kunden kan definitivt avgöra om utvecklingen lever upp till dennes krav och önskemål.

De flesta av dessa koncept är fullt skalbara, vissa kan dock ställa till problem. Små leveranser kan bli svårt att uppehålla i större projekt. Ju större projektet är desto längre tid tar det att sammanställa. Om iterationerna fortfarande ska vara två veckor innebär det att större projekt får mindre andel tid på faktisk utveckling. Kollektivt ägande av kod kan också bli problematiskt. Vid en viss storlek blir det opraktiskt eller t.o.m. omöjligt för individer att få total överblick av ett system. Därför blir det nödvändigt att dela upp projektet i delprojekt. (Stober & Hansmann, 2010)

Enkelhet är också ett värdefullt mål att sikta på, men i vissa fall är det inte praktiskt. När olika

team utvecklar olika delar av en helhet kan det bli svårt att ha samma helhetsfokus, och vissa bitar kod tvingas utvecklas utan en klar bild av vilken funktionalitet den bidrar till. Slutligen kan planering inte ske med hur många personer som helst. Det har visat sig opraktiskt att ha mer än 8-10 personer närvarande vid sådana möten. Detta kan avhjälpas något genom att ha gruppmöten för att etablera ståndpunkter som en representant sedan tar med sig till det verkliga mötet. På så sätt kan ett möte med 8-10 personer på ett relativt bra sätt representera åsikterna av upp till 100 personer. (Stober & Hansmann, 2010)

Kritiker har kallat XP för glorifierad hackning, men kräver i själva verket stark disciplin. Till skillnad från exempelvis Crystal så har den ett specifikt flödesschema som ska följas. (Stepanek, 2005)



Figur 2. 2 XP-flödesschema (Stepanek, 2005, s. 76)

2.3.5 Feature-Driven Development

Funktionsdriven utveckling (FDD) är ett försök att få ett mått av kontroll över agil utveckling. Utvecklingen ska kunna anpassas till förändrade omständigheter, men det ska hela tiden finnas ett klart fokus. Detta fokus är funktionalitet och kundvärde. (Hunt, 2006) FDD inkluderar flera traditionella element som tonats ner i andra agila metoder. Detta inkluderar planering, design, dokumentation, specifikt ansvar för individuella utvecklare, etc. (Rychlý & Tichá, 2008)

FDD är enligt Rychlý & Tichá (2008) en iterativ, inkrementell systemutvecklingsprocess som bygger på fem processer:

- **Utveckla en övergripande modell** – En modell för problemområdets domän och systemets roll där tas fram av domänexperter och utvecklare.
- **Bygg en lista över funktioner** – En funktion i det här fallet är en liten levererbar del av systemets funktionalitet. Funktionen ska ha ett klart kundvärde. Dessa funktioner delas in i grupper och kopplas till den övergripande modellen.
- **Planera utifrån funktioner** – Grupper av funktioner prioriteras utifrån deras kundvärde och bedömd tidsåtgång för att utveckla. Dessa delas sedan upp mellan funktionsteam som sedan inom teamet delar upp individuella ansvar.
- **Design utifrån funktion och utveckling utifrån funktion** – De två sista processerna itereras för varje enskild funktionsgrupp. Teamet tar fram en specifik design för funktionerna och implementerar sedan dessa.

På en skala från vattenfallsmodellen till XP kan FDD ses som en mer moderat form av agil utveckling. Ett försök att utifrån best practises få det bästa av två världar. (Rychlý & Tichá, 2008)

2.3.6 Crystal

Crystal metoderna har utvecklats av Alistair Cockburn. Han intervjuade utvecklingsteam och fann att de ofta ignorerade formella metoder och arbetade på ett sätt som de fann lämpligt. Crystal är hans försök att beskriva hur de faktiskt arbetade. Det var ett arbetsätt som faktiskt fungerade väl. Crystalmetoderna är därför mer beskrivande än föreskrivande, och ger på så vis en flexibilitet som gör den lättare att använda i praktiken. (Stepanek, 2005)

Cockburn observerade att team av olika storlek även arbetade på olika sätt och hade olika behov. Därför utvecklade han Crystal Clear för 2-8 utvecklare, Crystal Yellow för 9-20, Crystal Orange för 21-50, o.s.v. Enligt Stepanek (2005) bygger alla metoderna dock på sju gemensamma egenskaper:

- **Frekventa leveranser** – Tanken bakom dessa korta iterationer är ett återkommande fenomen inom agila metoder. Även Cockburn såg behovet av att frekvent utvärdera och omvärdera resultat och planer.
- **Reflektiv förbättring** – En stor styrka med korta iterationer är att det går att reflektera över det som presterats och lära sig inför framtiden. Ju oftare detta gör desto mer lär utvecklarna sig, och desto bättre blir slutprodukten.
- **Nära kommunikation** – Kommunikation är en viktig funktion inom agil utveckling. Det är den som ska hålla ihop utvecklingsteamet när struktur och dokumentation minskas. I Crystal Clear-metoden ska hela teamet samlas i ett gemensamt rum och ha möten där alla kan komma till tals samtidigt.
- **Personlig säkerhet** – Denna punkt handlar inte om fysisk säkerhet. Tanken är att alla deltagare ska känna sig säkra i sin rätt att föra fram idéer, synpunkter, problem och direkta misstag utan att bli attackerade. Detta i sin tur leder till förbättringsmöjligheter.
- **Fokus** – Inom Crystal har fokus två betydelser. För det första ska deltagare ha möjligheter att fokusera på en uppgift åt gången. En fragmenterad arbetsdag är en ineffektiv arbetsdag. För det andra ska teamet ha tydliga mål som de alla arbetar mot. Varje uppgift ska utföras med dessa mål i åtanke.
- **Enkel åtkomst till expertanvändare** – Utvecklarna ska ha så enkel åtkomst som är praktiskt möjligt till de expertanvändare som valts ut. Dessa användare håller kunskaper om hur det faktiskt är att arbeta med den process som systemet ska stödja. Därför har de en unik förmåga att utvärdera utvecklarnas framsteg.
- **Teknisk miljö med automatiserade tester, konfigurationshantering och frekvent integration** – Crystal har ingen specifik integrationsfas i slutet av utvecklingen. Därför måste testningen ske kontinuerligt, och automatisering är ett effektivt sätt att uppnå dessa. Även konfiguration och integration bör ske fortlöpande, detta försäkras att komponenterna fungerar som en helhet.

I Crystal ser vi många gemensamma element med de andra agila metoderna vi tagit upp. Det som utmärker dem är dels att de bygger på praktiska observationer, och dels att de är väldigt flexibla. Crystal innehåller moment och målsättningar men ingen specificerad process att följa. Detta lägger både mer press och mer möjligheter hos utvecklaren som använder metoden. (Stepanek, 2005)

2.3.7 Sammanställning av agila metoder

Inom Agil utveckling ser vi flera återkommande aspekter, iterationer, kundfokus, kontinuerlig testning, utförlig kommunikation, etc. Vi ser också en avvägning mellan att ge konkreta

riktlinjer och att inte begränsa utvecklaren. Vi anser att detta stämmer väl med vår syn på glappet mellan agil teori och agil praktik.

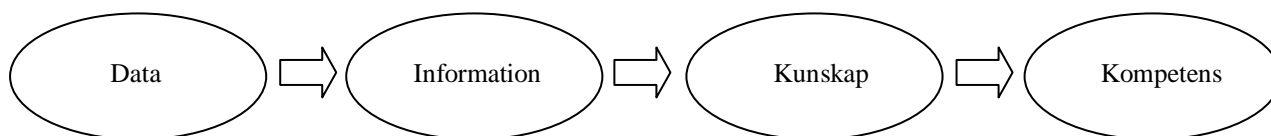
Lean och Kanban är som sagt inte utvecklade för just systemutveckling. Därför kan de inte appliceras direkt på systemutveckling utan viss anpassning (Turner et al, 2012). Resultatet påminner till stora delar om det vi ser inom agila metoder. Reflektivt arbete, hög kommunikationsgrad, snabba leveranser, helhetstänkande, och att göra kvalitetsarbetet centralt och integrerat i produkten är element som går väl igen inom de agila metoder vi tagit upp. Vi ser att alla de agila metoderna på olika sätt implementerar samma tankesätt som tas upp i det agila manifestet. Sedan kan varje enskild metod bidra med sina egna tillskott. XP är exempelvis ensamt bland de metoder vi utforskat om att förespråka kodning i par.

Vi ser en klar gemensam koppling till det agila manifestet. Flera metoder delar element och idealen som ska eftersträvas är väldigt lika. Vi anser därför att vi är berättigade att föra en gemensam undersökning av agila metoder som ett övergripande begrepp. Vi ser klara likheter som gör att metodernas användning kan undersökas tillsammans.

2.4 Kommunikation och kunskapsspridning

För att en organisation ska ha möjlighet att sprida information som framkommer från en utvärdering av ett projekt, eller för att sprida information om hur en anpassad agil arbetsmetod är tänkt att fungera, bör de arbeta med kunskapsspridning och kommunikation. Det är enligt Macheridis (2009) viktigt att arbeta med det för att uppnå god inläring och kompetensutveckling. När en person tillägnar sig ny kunskap eller utvecklar nya färdigheter, brukar detta vanligen uppfattas som lärande. Skillnaden mellan dessa två begrepp, kunskap och färdighet, förklaras genom att kunskap innebär att en person har insikt i hur något fungerar eller sker, medan färdigheter innebär att personen har en förmåga att använda kunskapen för att få något att fungera eller ske (Jacobsen & Thorsvik, 2008). Det är därför av stor vikt att organisationer arbetar aktivt med lärande för att möjliggöra utvecklingen av färdigheter hos projektdeltagare som arbetar med agila utvecklingsmetoder. Genom att sträva efter att uppnå en hög grad av kompetensutveckling och sprida kunskapen om arbetssätt till övriga organisationen kan framtida projekt utföras med större säkerhet.

För att kunna sprida kunskap krävs det enligt Macheridis (2009) data som via en process (se figuren på nästa sida) omvandlas till information, som i sin tur omvandlas till kunskap och som i sin tur omvandlas till kompetens. Data som tolkas av mottagaren och görs förståelig blir till information. Informationen analyseras och görs begriplig och blir därmed kunskap. Kompetens uppstår när mottagaren kan använda kunskapen i olika sammanhang och situationer.



Figur 2. 3 Process för kunskapsspridning (Macheridis, 2009, s. 160)

2.4.1 Problematik vid kunskapsspridning

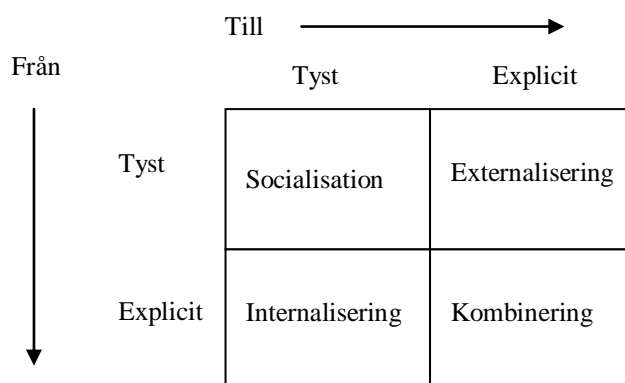
Problemet som kan uppstå när projektmedlemmar tillägnar sig ny kunskap eller utvecklar nya färdigheter är att mycket av det lärande som sker stannar hos den enskilde individen (Jacobsen & Thorsvik, 2008). Det medför att kunskapen om hur ett visst arbetsätt utförs, från en anpassad agil metod, kan vara dold för övriga deltagare. I det här läget innehar individen så kallad tyst kunskap. För att uppnå ett kollektivt lärande och upprätthålla en god kompetensutveckling måste kunskapen spridas vidare, vilket dock kan vara svårt ifall kunskapen är tyst.

Enligt Gerami (2010) är den tysta kunskapen personlig. Det handlar om erfarenheter som individen har angående olika situationer. Dessa är specifika, svåra att formalisera och kommunicera samt svåra att dela med andra inom organisationen. Motsatsen till tyst kunskap är explicit kunskap (Gerami, 2010). Vilket enligt Jacobsen & Thorsvik (2008) förklaras som erfarenheter och förhållanden som är lätta att identifiera. Den här typen av kunskap finns nedskrivet och sätts ofta i system som strukturer, rutiner och procedurer för att organisationens medlemmar lätt ska kunna ta del av den.

En grundläggande åtgärd från projektgruppens sida bör vara att tydligt klargöra för samtliga projektdeltagare hur de ska förhålla sig till ett visst arbetsätt efter en anpassad metod. Det är därför angeläget att denna typ av information snabbt utvecklas till explicit kunskap som enkelt kan tillgodoses av samtliga projektmedlemmar. Det är också viktigt att den tysta kunskapen omvandlas till explicit för att stödja utvecklingen av lärandet i organisationer. Det finns fyra stycken former för att omvandla kunskapen mellan tyst och explicit. Jacobsen & Thorsvik (2008) förklarar begreppen som följande:

1. **Socialisation** innebär att kunskapen förflyttas från tyst till tyst kunskap. Det sker genom att kunskap sprids mellan människor utan att de talar med varandra eller aktivt försöker förmedla kunskapen. Det kan till exempel vara en anställd som observerar hur en kollega arbetar, och på så sätt lär sig hur han/hon själv ska utföra arbetsuppgiften. Socialisation är dock en begränsad form av kunskapsöverföring eftersom det krävs att de anställda arbetar nära varandra och ofta kan se varandra.
2. **Externalisering** innebär att kunskapen går från att vara tyst till att bli explicit kunskap. För att den ska bli det krävs det att individens tysta kunskap presenteras via skrift eller muntligt tal, och görs tillgänglig för andra inom organisationen.

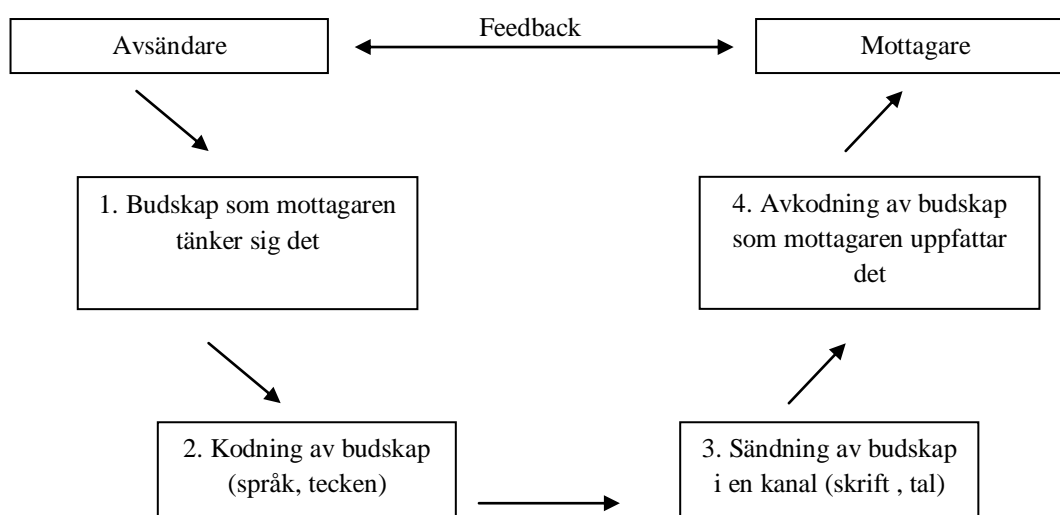
3. **Kombinering**, innebär att kunskapen omvandlas från explicit till explicit. Denna form används för att sprida viktig information, exempelvis analyser av organisationens potentiella konkurrenter, vilket kan ge upphov till att en ny syn eller arbetsätt anammas i en viss situation. Detta handlar om att kategorisera information som redan finns i organisationen, således medför det att den här formen av lärande inte är särskilt kraftfull.
4. **Internalisering**, innebär att kunskapen förändras från att vara explicit till att bli tyst. Processen medför att individer tar del av den explicita kunskapen och anpassar den till sina egna tysta kunskaper.



Figur 2. 4 Lärandeprocessen (Gerami, 2010, s. 235)

2.4.2 Kommunikationsprocessen

Det krävs en process där personer eller grupper utväxlar information med varandra för att möjliggöra spridning av kunskap inom projektgruppen, alternativt inom hela organisationen. Kommunikation definieras som en kontinuerlig process där medlemmarna kommunicerar mellan individer och grupper, både internt och extern, för att upprätthålla och förändra organisationen. (Jacobsen & Thorsvik, 2008) Kommunikationsprocessen kan ses som en följd av handlingar där informationen överförs från en sändare till en mottagare. Sändaren vill förmedla något och mottagaren är informationens adressat. Processen illustreras av figuren på nästkommande sida.



Figur 2. 5 Kommunikationsprocessen (Jacobsen & Thorsvik, 2008, s. 296)

Kommunikationsprocessen startar med att en sändare kodar informationen, genom att formulera och uttrycka det som ska förmedlas. Val av symboler, både verbala och/eller icke-verbala signaler, ingår i kodningen för att kunna förmedla informationen till mottagaren. Därefter måste sändaren välja vilken kanal som ska användas för att förmedla budskapet. Det kan exempelvis ske via skrift eller tal. (Jacobsen & Thorsvik, 2008)

Efter de två ovannämnda stegen måste mottagaren avkoda informationen. Det innebär att mottagaren måste tolka budskapet och bilda sig en egen uppfattning av vad sändaren vill förmedla. Till sist återstår bara återkopplingen där mottagaren svarar sändaren på det mottagna budskapet, till följd av det byter de inblandade parterna roller med varandra och kommunikationsprocessen börjar om på nytt med samma handlingar som precis har beskrivits. (Jacobsen & Thorsvik, 2008)

2.4.3 Kommunikationskanaler

Informationen som förmedlas i kommunikationsprocessen kan hanteras på olika sätt, genom exempelvis muntlig eller skriftlig kommunikation (Macheridis, 2009). Det kan enligt Jacobsen & Thorsvik (2008) lättare uppstå problem vid muntlig kommunikation i jämförelse med skriftlig kommunikation, men det finns även en begränsning i den skriftliga kommunikationen på hur mycket information som kan överföras i ett budskap. Därutöver kan det även ta lång tid innan sändaren får någon respons från mottagaren vid skriftlig kommunikation.

Fördelen med den muntliga kommunikationen är att det är lättare att överföra stora mängder information och att sändaren kan förvänta sig en snabb respons på budskapet. Generellt sett är den bästa typen av kommunikationskanal en kanal som möjliggör för överföring av många

signaler samtidigt, ger snabb återkoppling och tillåter sändare och mottagare att vara personliga och anpassa budskapet till varandra (Jacobsen & Thorsvik, 2008).

Eftersom agila utvecklingsmetoder är kommunikationsdrivna (Stober & Hansmann, 2010) är det viktigt att det existerar god kommunikation inom projektgruppen. Det är därför väsentligt att det tidigt i projektets livscykel redogörs för vilka kommunikationskanaler som finns till förfogande. Om det ses nödvändigt bör deltagarna och andra intressenter utbildas för att garantera att kommunikationsmedierna används på rätt sätt vilket bidrar till att bevara projektets effektivitet. (Macheridis, 2009)

Vidare kan kommunikationen inom en projektgrupp delas in mellan formell respektive informell kommunikation. Den formella kommunikationen följer alltid vissa regler och sker inom vissa ramar på ett formellt sätt. (Macheridis, 2009) Det innebär att information förmedlas inom organisationen genom planerade arrangemang och upplägg. Den följer organisationsstrukturen och är direkt relaterad till den verksamhet och arbete som utförs. Det förekommer både vertikal och horisontell kommunikation i en organisation. Information som förmedlas via den vertikala kommunikationen har sin sändare och mottagare på olika hierarkiska nivåer inom organisationen, här sker ett informationsutbyte mellan över- och underordnad. Gällande den horisontella kommunikationen sker informationsförmedlingen på samma hierarkiska nivå, mellan medarbetare som har liknande status. (Jacobsen & Thorsvik, 2008)

Informell kommunikation förekommer ständigt mellan anställda då de stöter på varandra på arbetsplatsen. Ett informationsutbyte kan ske informellt innan eller efter ett viktigt möte, där det ofta innehåller diskussioner om arbetsplatsen, arbetssätt och arbetsuppgifter. Den informella kommunikationen har en stor potential för lärande, det har visat sig att denna form av kommunikation sprider information snabbare än den formella kommunikationen. Även förtroendet för informationen från de underordnade har enligt författarna visat sig vara större vid informell kommunikation. (Jacobsen & Thorsvik, 2008)

2.5 Förändringsarbeten

Vid införande av nya agila arbetsmetoder drabbas organisationens verksamhet av en förändring. Det kan innebära att organisationen hittar nya sätt att utföra en existerande uppgift eller förändringar i hur arbetsuppgifter delas upp och samordnas. (Jacobsen & Thorsvik, 2008) Förändringsarbeten kan skiljas mellan två grupper, episodisk förändring och kontinuerlig förändring. Episodisk förändring har en början och ett slut, den är oftast också avsiktlig med ett konkret mål. Kontinuerlig förändring är en evolution som kan vara avsiktlig eller improviserad. Införandet av en ny arbetsmetod så som agila metoder är en episodisk förändring. Att sedan utvärdera och följa upp användningen av denna metod är en kontinuerlig förändring. (Landy & Conte, 2007)

Det finns flera barriärer mot förändring, både hos individer och hos organisationer. Individer kan känna att förändring är ett hot mot deras ekonomiska eller sociala säkerhet.

Organisationers barriärer grundas i det faktum att en organisatorisk eller processförändring gör ändringar i organisationens själva fundament. Ett mer allmänt problem är rädslan för det okända. För att överkomma dessa barriärer har det tagits fram förändringsmodeller. En av de äldsta modellerna som ligger till grund för många av dagens mer invecklade modeller är unfreeze-change-refreeze-modellen (Landy & Conte, 2007), den ser ut som följer:

1. **Unfreeze** – Individerna blir medvetna om de åsikter och värderingar de håller i nuläget.
2. **Change** – Individerna inför nya åsikter och värderingar.
3. **Refreeze** – De nya åsikterna och värderingarna stabiliseras.

Denna grund har lett fram till en lång rad modeller men grundtanken är att förändringar först bör grundas i organisationen, sedan genomföras, och slutligen följas upp för att stabilisera förändringen. (Landy & Conte, 2007) En bra metod för förändring ger inte bara en ren förändring, den försäkrar även stödet för förändringen från intressenter. En ny process att arbeta utifrån fungerar inte automatiskt bara för att den är effektiv på pappret, den måste accepteras av de som ska följa den. Något som är ännu värre än en processförändring som fastnar i analysstadiet, är en perfekt processdesign som inte implementeras. (Harmon, 2007)

2.6 Liknande studier av agila metoder

Det finns en rad tidigare studier inom vårt problemområde. Hayes & Richardson (2008) har undersökt implementeringen av agila metoder. De fann att det är av vikt för företag att noga utvärdera sin situation och sina behov innan de implementerar agila metoder. De pekade exempelvis på att agila metoder gör sig bäst i små grupper. Därför rekommenderade de att företag gör efterforskning och läser fallstudier från likande bolag innan de implementerar agila metoder som arbetssätt.

Pikkarainen, Haikara, Salo, Abrahamsson & Still (2008) gjorde sin undersökning av kommunikationen inom agila projekt. De undersökte Scrum och XP som visade en positiv effekt på kommunikation både inom och utom arbetsgruppen. Den ökade informella kommunikationen minskade också behovet av dokumentation. Kommunikationen med externa intressenter var dock fortfarande problematisk och varken Scrum eller XP erbjöd tillräckliga mekanismer för kommunikation med dessa.

Moe, Dingsøyr & Dybå (2010) fann brister i Scrum vad gällde att ha självstyrande team. Scrum föreslår självstyrande team, men i praktiken fann de ett behov av klara riktlinjer för uppdelning av ledarskap och ansvar som Scrum inte tillhandahöll. De skriver att individuell autonomi ställer till problem för självstyrande team. Den överblick som främjas av Scrum gav

en bra grund för feedback, koordination och backup, men inget direkt ramverk för hur det skulle genomföras.

Jyothi & Rao (2012) tar upp kombinationen Lean-Kanban och hur denna metod som är framtagen för bilindustrin kan användas tillsammans med agila metoder för att skapa ett fördelaktigt arbetsätt inom systemutveckling. De nämner dock att det krävs ett klart ramverk för hur detta ska genomföras och de har inte själva undersökt om det kan göras på ett fördelaktigt vis.

Sammantaget ser vi i dessa fall tecken på att agila metoder inte är en perfekt lösning. De kan inte appliceras fördelaktigt i alla situationer och även i fördelaktiga situationer verkar de behöva kompletteras av andra metoder eller specificeringar för att bilda en fullständig helhet. Detta stämmer väl med den bild vi utgått ifrån. Agila metoder verkar behöva tolkas, anpassas eller kombineras i många situationer.

3 Metod

För att besvara forskningsfrågan har vi valt att utgå från en intensiv uppläggning vilket innebär att vi går på djupet gällande användningen av agila metoder. Enligt Jacobsen (2002) är metoden att föredra för att få fram så många nyanser och detaljer som möjligt, samt att försöka få en förståelse av förhållandet mellan undersökningsenheten och det sammanhang som undersökningen ingår i. Jacobsen (2002) hävdar att metoden ska användas när författarna vill förklara vad som sker i en särskild situation eller vill förklara en specifik händelse. Det ligger i linje med vad vi vill åstadkomma med studien, därför väljer vi den intensiva uppläggningsen.

Gällande vår insamling av empiri har vi beslutat att utgå från en kvalitativ ansats. Den kvalitativa metoden innebär att vi som undersökare för en öppen intervju med respondenter angående vårt undersökningsområde. Metoden lägger vikt vid detaljer, nyanser och det unika hos varje uppgiftslämnare samt att den sätter få begränsningar för de svar som uppgiftslämnaren anger. (Jacobsen, 2002) Det passar oss att välja denna ansats eftersom det är lämpligt att använda metoden då intresse finns för att skapa klarhet i ett fenomen. Den kvalitativa uppläggningsen syftar i regel på att ta reda på hur människor tolkar och förstår en viss situation (Jacobsen, 2002).

Vidare skriver Jacobsen (2002) att kvalitativa ansatser ofta blir intensiva, vilket innebär att undersökarna väljer ut ett eller några få fall, alternativt ett fåtal enheter för att utföra intervjuer med personer som varit med om en viss händelse. Anledningen till att bara enstaka tillfällen studeras är för att få en så fullständig bild som möjligt av en situation eller händelse. Det passar oss bra att utgå från dessa metoder eftersom vi strävar efter att få en tydlig förklaring till hur agila metoder tolkas från teorin och sedan används i praktiken.

3.1 Tillvägagångssätt

Vår process för att hitta lämpliga intervjupersoner började med att vi valde att besöka ett förutbestämt antal företag där vi personligen kunde presentera oss och förmedla vårt syfte med arbetet och intervjuerna. Vi fick ett positivt gensvar av samtliga företag och fick så småningom inbokade tider för att utföra intervjuer.

Innan vi påbörjade arbetet med att samla in empiri studerade vi teori angående systemutveckling, projektadministration, kommunikation, kunskapsspridning och lärande inom organisationer. Vi ville innan intervjuerna utfördes skaffa oss en bred kunskapsbas för att kunna föra en god diskussion med våra uppgiftslämnare.

Intervjuerna har vi utfört på företag som har sitt säte i Lund. Det har underlättat för oss eftersom vi har kunnat utföra intervjuer öga mot öga, vilket enligt Jacobsen (2002) är att föredra. För att inte uppehålla intervjupersonerna allt för länge från sitt arbete har vi siktat på

att utföra intervjuerna på max 30 minuter. Dock har det i de flesta fall gått snabbare än så men vi har ändå fått den information vi varit ute efter.

Vi har spelat in samtliga intervjuer för att kunna hålla en flytande konversation samt för att kunna återge uppgiftslämnarnas uttalande på ett korrekt sätt. Det inspelade materialet har vi transkriberat för att underlätta vårt arbete med att analysera empirin. Analysen har utförts genom att informationen har kategoriserats efter ett antal ämnen som vi anser vara vitala att diskutera kring. Kategoriseringen utför vi för att kunna se vilken data som liknar varandra och behandlar samma fenomen eller ämne, alternativt se ifall data skiljer sig åt och belyser olika fenomen eller ämnen. En kategorisering är en förutsättning för att en jämförelse ska kunna ske mellan olika intervjuer, det ger oss möjligheten till att diskutera samma ämne från intervjuerna men belysa dem ur olika synvinklar. (Jacobsen, 2002) Ämnesindelningen baseras på informationen som har lämnats av intervjupersonerna men också från centrala begrepp i vår frågeställning. Kategoriseringen bidrar till ett ramverk som vi följer i vår sammanställning av empiri samt i diskussionsdelen, vilket försäkrar att vi diskuterar relevant information i förhållande till frågeställningen.

Vi har valt att kategorisera empirin efter metodval, metodanvändning, kombination av metoder, produktägare och kundperspektiv, formellt och informellt arbetsätt samt kommunikationsstruktur. Vi väljer dessa ämnen för att vi vill analysera hur organisationerna går tillväga vid ett metodval, hur de använder de valda metoderna samt hur de kombineras med varandra. Vidare vill vi analysera produktägarens och kundens perspektiv vid användning av en agil metod. Vi vill även diskutera formellt och informellt arbetsätt för att förstå skillnaden mellan vad som tydligt demonstreras av en metod och vad som är underförstått. För att möjliggöra diskussion kring organisationens sätt att förmedla information till sina medarbetare analyserar vi organisationernas kommunikationsstruktur. Ämnesindelningen ger oss möjlighet att resonera kring en organisations arbetsätt med agila metoder. Analysen av empirin, tillsammans med teoriavsnittet, ligger till grund för vår diskussionsdel.

3.2 Typ av intervjuer

I vårt val angående vilken typ av intervjuer vi genomför har vi bestämt att utgå från en öppen individuell intervju. Den kännetecknas av att undersökare och personen som intervjuas samtalar med varandra som i en vanlig dialog. Vi väljer den här typen av intervju eftersom det är mest lämpligt att använda denna metod när relativt få enheter ska undersökas, eller när undersökarna är intresserade av den enskildes inställning och uppfattning om en situation. (Jacobsen, 2002)

Den öppna intervjun är till en viss grad strukturerad genom att det finns en lista över vad som ska tas upp under intervjun. I övrigt är den ostrukturerad och fortlöper som ett vanligt samtal utan någon form av begränsning eller styrning från intervjuarens sida. (Jacobsen, 2002) Vi har utarbetat en intervjuguide där vi dels har strukturerade frågor som berör intervjupersonens erfarenheter av agila projekt. Vidare har vi ett antal mer öppna frågor som leder till diskussion

om det aktuella ämnet. Intervjuguiden ger oss riktlinjer till vad som bör behandlas under intervjun för att säkerställa att vi berör väsentliga områden.

Eftersom vi utför en besöksintervju öga mot öga får vi en personlig kontakt med uppgiftslämnaren vilket leder till att känsliga ämnen lättare kan diskuteras. Vi har också möjligheten att observera intervjupersonens kroppsspråk vid en besöksintervju, det underlättar för oss att uppfatta hur djupt vi bör gå in på en fråga. Om vi upptäcker att personen ser besvärad ut kan vi undvika att gå över gränsen och istället byta inriktning eller fråga för att minimera risken att intervjupersonen sluter sig och blir en sämre informationskälla. (Jacobsen, 2002)

3.3 Urval av intervjupersoner

Vi koncentrerar oss på ett fåtal företag och personer att intervju. Vi gör det för att uppnå en djupare och mer detaljerad insikt i företagets hantering av olika utvecklingsmetoder som finns att tillgå. Vårt urval har gjorts i eftersträvan att få olika perspektiv snarare än att få representativa svar. Alla våra intervjupersoner har arbetat med agila metoder i skarpa situationer vilket gör deras erfarenheter verkliga och relevanta. Däremot har de olika perspektiv och deras upplevelser är baserade i deras subjektiva verklighet. Denna verklighet påverkas av framförallt kontext/roll i situationen och deras individuella egenskaper, i form av personlighet, kunskap etc.

Vi har huvudsakligen vänt oss till små och medelstora utvecklingsföretag, med det menar vi företag som har tillräckligt få anställda för att vara koncentrerade till ett kontor. Vi har gjort detta eftersom vi fått uppfattningen att större företag tenderar att ha en mer strikt policy angående examensarbeten, vilket inte främjar studenternas autonomi.

Vi har blivit väl bemötta av de företag vi kontaktat och har därför kunnat föra en bra dialog angående urval. Vi har låtit företagen själva komma med förslag på vilka de tror kan vara lämpliga för oss att intervju. Av ett företag hänvisades vi t.o.m. till ett av de andra företagen som menade att ett produktföretag kunde ge oss ett perspektiv på agila metoder som ett konsultföretag inte kunde.

Våra intervjuer har gett oss en bredd av perspektiv. Vi har besökt tre företag med olika sätt att arbeta. Ett mer strukturerat konsultföretag, ett mindre strukturerat konsultföretag och ett produktföretag. Detta har varit intressant eftersom de har tydliga skillnader i sina relationer till kunder och hur de agila metoderna relaterar till dessa.

Ett konsultföretag utvecklar system specifikt för en kunds behov vilket ger ett specifikt kundperspektiv som dock är externt (Gordon & Starr, 1985) Produktdrivna företag utvecklar en produkt mot en intern beställare för att senare säljas mot en bredare grupp externa kunder (Turner et al, 2012). Detta ger ett mindre specifikt kundperspektiv men håller det internt. Vi ansåg det högst intressant att undersöka båda perspektiven och hur skillnaderna påverkade deras användning av agila metoder.

Vi har i vår undersökning inte tagit hänsyn till aspekter så som kön eller etnicitet. Vårt urval är inte menat att vara representativt. Vi har strävat efter olika perspektiv men utifrån skillnader i roller och företag. Vi har gjort antagandet att andra aspekter inte spelar in lika mycket, men det hade varit möjligt att göra en annan studie där man undersökt just detta antagande. För oss och vår frågeställning kändes det dock inte relevant.

Sammantaget anser vi att vårt urval ger oss en bred bas för undersökning av ämnet. Vi har flera synvinklar på ett fenomen som hjälper oss analysera och förstå det. Vi har därför en bra grund för att bygga vår empiri som i sin tur leder till vår diskussion. De personer vi intervjuat är:

P1 – Har studerat Systemvetenskap vid Lunds Universitet. Arbetar nu på ett konsultföretag som systemutvecklare. Personen har sedan starten av sin yrkeskarriär arbetat med utveckling av mjukvara med en agil utgångspunkt i arbetssättet. Projektteamen personen har deltagit i har varit väldigt Scrum-baserade. Personen brukar ha en drivande roll i teamet och är en av dem som alltid är med och försöker modifiera processen. (Konsultbolag 1)

P2 – Har studerat Systemvetenskap tre år i Växjö. Sista året och även efter utbildningen har personen arbetat som systemutvecklare. Personen gillar att koda men har även varit Scrum-master under en tid. Personen har varit på flera företag som arbetat med agila metoder, och är nu konsult på ett företag som håller på att införa Kanban. (Konsultbolag 1)

P3 – Har studerat till Civilingenjör i informationsteknologi vid Uppsala Universitet. Personen arbetar som platschef på ett konsultföretag, och fungerar samtidigt som projektledare där. Företaget personen jobbar på nu arbetar med agila projekt men personen har också fem års erfarenhet från andra företag av att arbeta med projektledning där ett agilt arbetsätt har använts. (Konsultbolag 2)

P4 – Har studerat till Civilingenjör i elektroteknik på Lunds Tekniska Högskola. Personen har arbetat med agila metoder de senaste tre åren. Innan dess arbetade personen utifrån vattenfallsmodellen. Nu arbetar personen som projektledare på ett företag som använde Scrum tills för 1,5 år sedan. De överväger just nu sin metodanvändning och använder olika versioner av Scrum och Kanban under tiden. (Produktföretag 1)

P5 – Har studerat till Civilingenjör i teknisk matematik på Lunds Tekniska Högskola. Arbetar nu på ett mjukvaruutvecklingsföretag som Director Of Products. Personen har varit produktägare i ett renodlat Scrum-arbete men har också erfarenhet som utvecklare i ett produktteam som arbetade agilt. (Produktföretag 1)

3.4 Intervjuguide

Vi har utformat 12 stycken frågor, se Bilaga 1, som tillsammans med teoriavsnittet ligger till grund för att en diskussion och slutsats ska växa fram. Intervjuerna inleds med allmänna frågor som berör uppgiftslämnarens utbildning och erfarenhet av att arbeta i agila projekt. Vi ställer till en början sådana frågor för att få en bild av personens relevans i undersökningen.

Det är även enligt Jacobsen (2002) bra att börja med enklare allmänna frågor som inte kräver så mycket av intervjupersonen för att undvika att samtalet låser sig. Vilket det kan göra om intervjun inleds med komplexa och invecklade frågor.

Intervjufrågorna går sedan vidare till att bli mer öppna, där uppgiftslämnaren får möjlighet att fritt prata kring ämnet. Vitsen med en öppen intervju är att få beskrivningar med uppgiftslämnarens egna ord (Jacobsen, 2002), därför försöker vi styra intervjun så lite som möjligt för att få fram det som intervjupersonen anser vara viktigt angående det aktuella ämnet. För att säkerställa att vi får den information vi önskar ställer vi följdfrågor och gör förtydliganden under intervjuens gång. Det gör vi även för att säkerställa att vi har förstått innebörden av uppgiftslämnarens uttalande så vi kan återge informationen på ett korrekt sätt.

De öppna frågorna behandlar vilka metoder som har använts inom organisationen och hur ett projekt påverkar vilken metod som används. Frågorna behandlar också hur arbetssättet sprids inom organisationen och hur det lärs ut till andra medarbetare samt hur metoderna kombineras eller hur komponenter väljs ut. Anledningen till att vi har utarbetat ett antal frågor är för att vi vill ha dessa som riktlinjer till att säkerställa att uppgiftslämnaren talar om saker som är relevanta för vår undersökning. Frågorna stämmer även överens med ramverket som används för sammanställning av empiri samt för diskussion.

3.5 Etiska aspekter i en undersökning

Vid en intervju är det viktigt att reflektera över förhållandet mellan undersökare och uppgiftslämnare med avseende på etiska aspekter. Jacobsen (2002) talar om tre grundkrav som en undersökning bör ha som mål att uppfylla. Dessa är informerat samtycke, krav på privatliv och krav på riktig presentation av data. Nedan följer en redovisning av vad författaren skriver om dessa begrepp samt hur vi arbetar för att uppfylla dessa krav.

- **Informerat samtycke** - Intervjupersonen ska på frivillig basis delta i undersökningen. För att underlätta för uppgiftslämnaren att enklare fatta ett beslut om att vara med i studien eller inte, ska uppgiftslämnaren erhålla information om undersökningens syfte. (Jacobsen, 2002) Vi meddelade vid vår första kontakt med företagen vårt syfte och vilka typer av områden som skulle behandlas vid intervjun. På så sätt kunde en intresseförfrågan ställas till lämpliga kandidater hos företagen vilket mynnade ut i att samtliga tillfrågade samtyckte och ställde upp på en intervju.
- **Rätt till privatliv** - Syftet är att minska möjligheten att kunna identifiera individer utifrån de data som sammanställs från intervjun. Det ska inte gå att urskilja vad en viss uppgiftslämnare har svarat på en fråga eller vad den har gjort i en viss situation. (Jacobsen, 2002) För att uppnå detta har vi beslutat att använda en hög grad av anonymitet hos våra intervjupersoner och vi har även informerat uppgiftslämnaren om detta innan intervjuens start.

- **Krav på riktig presentation av data** - Detta handlar om att uppgiftslämnaren ska bli korrekt återgiven, informationen ska återges fullständigt och i rätt sammanhang. Undersökarna bör undvika att använda information som är taget ur sitt sammanhang till att argumentera för något som uppgiftslämnaren inte ursprungligen har menat. (Jacobsen, 2002) För att säkerställa att vi verkligen har förstått intervjupersonens mening i sammanhanget har vi stannat upp där oklarheter funnits och diskuterat frågan närmare för att få en bättre förståelse. Vi har även spelat in konversationen med uppgiftslämnarens tillstånd för att kunna återge intervjupersonens uttalanden korrekt.

3.6 Reliabilitet och validitet

I arbetet med en undersökning är det viktigt att försöka minimera problem som kan uppstå gällande giltighet (validitet) och tillförlitlighet (reliabilitet). Vid användning av kvalitativa metoder måste en kritisk granskning göras för att bedöma om slutsatserna är giltiga och går att lita på. (Jacobsen, 2002) Det innebär att vi håller oss kritiska till kvaliteten på vårt insamlade empiriska material, vi måste även kritiskt granska källornas förmåga att ge oss korrekt information om det vi undersöker.

För att uppnå en så hög grad av validitet och reliabilitet som möjligt har vi intervjuat personer som har god kunskap om ämnet. Informationen kommer från en förstahandskälla, dvs. uppgiftslämnaren har personligen varit med om händelsen som diskuteras. Även vårt tillvägagångssätt med öppna intervjuer där intervjupersonen fritt får tala om en händelse eller ämne bidrar till att öka validiteten, eftersom det enligt Jacobsen (2002) ofta tillskrivs större validitet då uppgiftslämnaren inte blir styrd av undersökarna. Det leder till att information som ligger närmare uppgiftslämnarens egentliga uppfattning delges.

Intervjuns resultat kan påverkas av det sammanhang där informationen samlas in. Det kallas för kontexteffekt och innebär att kontexten antingen är artificiell eller naturlig. Vi har därför valt att utföra intervjuerna i en för uppgiftslämnaren naturlig miljö för att minimera risken för en negativ inverkan på resultatet. Genom att utföra intervjuerna i en för uppgiftslämnaren känd miljö stärks reliabiliteten och validiteten ytterligare. (Jacobsen, 2002)

Vi ser utifrån detta att vi får en relativt hög validitet eftersom intervjupersonerna får tala utförligt om ämnet, och förtydliga exakt vad de menar. Vi får dock en tämligen låg reliabilitet eftersom vårt urval är begränsat. Vi får en bra bild av vad våra intervju personer tycker och varför de tycker så, men vi kan inte generalisera vad utvecklare allmänt anser. Vi får istället en mer ingående motivering till varför en anpassning eller kombinerad metod görs. IT-branschen som helhet motsvarar inte nödvändigtvis de fynd vi gjort, men fynden kan ändå ge relevanta insikter i problematik som kan existera inom fler organisationer i IT-branschen.

4 Empiriskt resultat

4.1 Metodval

I våra intervjuer är det huvudsakligen två utvecklingsmetodiker som nämnts som relevanta, Scrum och Kanban. Scrum beskrivs ofta som den mer strikta metoden och Kanban som den mer flexibla. P4 och P5 uttrycker att Scrums sprintar inte är tillräckligt agila för alla situationer. De kände ibland ett behov av att kunna anpassa backloggen från dag till dag, om t.ex. nya buggar kom in. P2 framförde sin åsikt att Kanban, på gott och ont, inte tillförde särskilt mycket till arbetssättet.

P2 - ” Kanban säger inte så mycket mer än att du får inte göra mer än så här många saker åt gången” (s. 53)

Flertalet av dem vi intervjuat förespråkar egentligen att använda metoderna som inspiration för att hitta ett eget arbetsätt. P5 berättade om att de på hans företag tidigare kört Scrum till vad han beskriver som en nästan religiös nivå. Det fungerade bra, men situationen förändrades och därmed metodbehovet och metodvalet. P1 har också arbetat relativt fullständigt utifrån Scrum tidigare. Detta val kom från utvecklarna själva i en kontext där de mer eller mindre var helt fria att lösa sin uppgift på det sätt som de fann lämpligt. De hade dock ingen formell metodspekifikation. De var överrens om Scrum och vad detta innebar kom de fram till tämligen ad hoc.

Metodvalen som gjorts har kommit från en rad olika källor. P3 och P4 talar framförallt om projektledarna, deras erfarenheter och hur de tillsammans arbetar fram ett arbetsätt utifrån sina observationer. P3 berättade att när erfarna projektledare diskuterade sina observationer om vad som lyckades, så överensstämde cirka 90 %. De resterande 10 % kunde de komma fram till genom kompromiss. P4 berättade en liknande historia, men istället för att kompromissa hade de inom de olika teamen arbetat aningen olika. De delade dock erfarenheter och observationer mellan sig.

P1 och P2 underströk vikten av att metoden kom från utvecklarna själva. De förespråkade självstyrda team som fick information, resurser, och krav utifrån men själva bestämde vad de gjorde utifrån dessa. P2 uttryckte bl. a. att han ansåg en metod som kom uppifrån i organisationen som dödfödd. P5 berättade dock om ett metodval som kommit uppifrån i organisationen och blivit lyckat. Detta anammades snabbt av utvecklarna. Huvudsakligen p.g.a. den frustration som uppstått ifrån att det tidigare inte funnits någon formell arbetsmetod alls. De införde Scrum för hela organisationen utom Maintenance, och resultatet blev väldigt lyckat.

P5 – ”Det var otroligt effektivt, jag tror aldrig nånsin att vi kommer vara så effektiva...” (s. 76)

P2 ifrågasatte hela konceptet med att sprida metoder. Bara för att en metod passade i en kontext innebär inte att den passar i en annan. Han menade att alla bör utveckla sin egen

metodik. Andras framgångar och hur de nått dessa kan agera som inspiration men bör inte tas som sanning i den egna organisationen. Han menar att ingen annan kan lära dig hur det är bäst för din organisation att arbeta.

P2 – *”...om man tar certifiering i Scrum. Jättekonstigt. Vad har du gjort? Du har certifierat dig i en metodik som någon annan har utvecklat för att vara bra på hans företag. Det är skitbra om du vill jobba på hans företag”* (s. 61)

P2 kommenterar också vattenfallsmetoden. Den är impopulär idag men kan vara användbar i rätt kontext, och har varit populär förr. På samma vis föreställer han sig ett framtida avståndstagande från agila metoder, och en ny metodik som tar dess plats. Han berättar att han sett flera metoder komma och gå i popularitet och att det kommer fortsätta så länge folk försöker använda metodiker som inte är anpassade till dem.

4.2 Metodanvändning

Trots att intervjupersonerna huvudsakligen hade arbetat med samma två metoder, Scrum och Kanban, så var deras användning av dessa metoder ganska skilda. P1 berättade om en arbetsplats där de använt Scrum, men de hade aldrig riktigt diskuterat vad det innebar. Alla ”visste” vad Scrum var. De var sammansvetsade som grupp och hade tämligen lika synsätt, så om något problem dök upp i arbetssättet löste de detta snabbt och informellt.

P2 hade huvudsakligen upplevt en inkrementell implementering av Scrum. Han beskrev det i form av en slags behovstrappa. Den började med morgonmöten. Efter det kom jämna releasecykler. Efter det kom retrospektiv- och planeringsmöten. Slutligen kom produktägaren och prioriterad backlogg. Detta var en trappa han upplevt på flera arbetsplatser men han hade aldrig upplevt det som fullständigt lyckat. Han påtalade att produktägaren ofta glömdes bort, och den rollen är enligt P2 *”halva grejen”*. Detta går vi in närmre på senare.

P2 talade även om en allmän oförståelse kring vad agila metoder är och vad de kan göra. *”Man tror att man ska få en silver bullet, man köper den i en låda”* (s. 61). Han ansåg att en kritisk lärdom för att lyckas var att identifiera två aspekter: *”Man får identifiera det problem man ska lösa”* (s. 60) och vad en agil metod kan åstadkomma *”Man ska inse att du kan inte bara komma med en metodik bara för att nån ball kille har kommit på att göra såhär i Kalifornien, och han är nu jätterik. Då funkar kanske precis lika bra i Staffanstorp. Det gör det inte nödvändigtvis.”* (s. 60). Han uttryckte därför att det krävdes en djupare förståelse för vad man försökte göra och varför.

P3 berättade att hennes företag skilde tydligare på utvecklingsmetodik och projektledningsmetodik. De använde Scrum som utvecklingsmetodik, men PROPS, Prince2 eller PPTS för projektledning, vilket är exempel på vattenfallsmetodiker. Scrum och hur de använde den metoden kom framförallt ifrån erfarna projektledare som diskuterat fram en gemensam tolkning/anpassning. Hon underströk vikten av projektledning för att kommunicera med övriga organisationen angående exempelvis budget, milstolpar, etc., men också inom

teamet. Scrum som metod nämner inte mycket personligt ansvar, men hon ansåg att det upprätthålles ändå, genom personliga interaktioner och informella krav och ansvar.

P3 – ” Nä, individen har ju fortfarande sitt eget ansvar. Scrum är ju lätt att misstolka om man vill det, t.ex. det här med att man inte behöver dokumentera – det är skitbra, man behöver inte hålla några tider – det är också skitbra, och man gör det man hinner och det man inte hinner gör man i nästa sprint, alltså så kan man ju också tolka det om man vill men det är helt fel, det är ju inte det som är syftet med metoden, så visst ansvar på individen är ju fortfarande detsamma. Jag som chef vill fortfarande att Kalle och Pelle måste ju ta sitt eget ansvar och får de en uppgift så ska de ju leverera det som de har sagt, och den ska ha god kvalité, det försvinner ju inte.” (s. 67)

P4 och P5 arbetar på ett företag som tidigare kört Scrum tämligen genomgående. De hade då via konsulter fått en fast definition av Scrum som de följde till punkt och pricka. P5 - ”...vi körde by the book...”. De fick hjälp från en konsultbyrå som kom och utbildade dem i användningen av Scrum. De hade en inlärningsperiod på cirka en månad, men sen fungerade det väl tills situationen förändrades och de tvingades släppa den inarbetade metoden. När vi besökte dem höll de på att utvärdera och utarbeta en ny arbetsmetod. De arbetade olika i olika projekt men huvudsakligen utifrån Scrum och Kanban.

Exakt hur dessa användes byggdes dock upp av projektledarna själva efter gemensamma diskussioner utifrån erfarenheter. De försöker i nuläget hitta en form av ”minsta gemensamma nämnare” (- P4 och P5) som de kan utgå ifrån när de tar fram en gemensam metod. Om där finns något som är så bra att alla anammat det så anser de att det är en perfekt utgångspunkt.

Gemensamt för alla intervjupersonerna kan man säga ett antal saker. Ingen ansåg att en metod kunde användas ”rakt av” i vilken situation som helst. I de flesta fall kände de ett behov av att anpassa metoden till deras specifika krav, och att formellt eller informellt täcka upp de tomrum som metoderna oundvikligen innehåller. Produktägarrollen var enligt alla problematisk inom Scrum. Alla lade också större vikt vid sociala interaktioner än vid formella metoder. Möten var bättre än handböcker. De flesta hade rört sig bort från formella metoder och mot ett mer allmänt agilt tankesätt.

4.3 Kombination av metoder

Som vi nämnt tidigare har vi märkt att de vanligaste metoderna att arbeta med är Scrum och Kanban. P1 säger att det är projektet som ska styra vilken metod som används, företaget bör börja med att se vilka krav de har på projekten och därefter välja vilken metod som ska användas. Sedan kan delar läggas till eller tas bort beroende på vad som krävs och passar bäst för situationen. P1 säger att det är bra att utgå så minimalt som möjligt med metoden för att sedan utöka den om det behövs.

Vidare säger P1 att de har kombinerat Scrum och Kanban i sitt arbete. De startar med att se efter vilka krav som finns och vilka projekt som ska utföras, det ligger till grund för att välja vilka eller vilken metod som ska användas och vilka delar som ska väljas ut, ”...eftersom vi

inte ska ha färdiga releaser så kör vi en kanbanboard som bara flyter men vi tar in tex. morgonmöten från Scrum, för det vill vi ha, och vi la även in ett retrospektiv i slutet av varje vecka, så där har vi plockat lite på det hållet.” (s. 49)

På företaget där P3 arbetar har de en metod för projektledning och en metod för mjukvaruutvecklingen. För att leda projekt används vattenfallsmodellen som tydligt beskriver i vilken fas projektet befinner sig i, medan utvecklingsmetoden är mer Scrum-influerad. Då det ofta uppkommer frågor om projektets status underlättar det ifall ett vattenfalls-tänk tillämpas. Trots att företaget arbetar efter ett agilt tankesätt så uppkommer det alltid frågor om hur långt projektet har fortskridit, därför används projektledarmetodiken för att ha möjlighet till att sätta upp tydliga milstolpar. Hon menar att det är svårare att göra det i Scrum då hon anser att Scrum egentligen är en ganska tunn metodik, mer ett tankesätt enligt henne.

P3 säger att företaget har kombinerat erfarenheter från erfarna projektledare som vet att en viss metod fungerar att arbeta efter. Utifrån dessa erfarenheter har best practice valts ut från metoderna och sammanställts till en helhet. Ifall de upptäcker på vägen att det inte fungerar så utför de nödvändiga ändringar. Företaget kombinerar metoder på det här viset eftersom de enligt P3 är ett informellt företag. Problematiken hon upplever uppstå är att hitta balansen mellan att jobba agilt och att jobba efter ett vattenfalls-tänk.

Det enda företag som implementerat Scrum till fullo är det företag där P4 och P5 arbetar, de valde att använda metoden med alla dess komponenter utan att kombinera den med någon annan metod. P5 upplevde implementeringen framgångsrik och det medförde en klar förbättring från tidigare mer ostrukturerade arbetsätt. P4 – ”...det blev en jättestor skillnad från att köra lite så här källarkodning till Scrum....” (s. 79)

P2 säger sig inte förstå poängen i varför man ska kombinera de två metoderna Scrum och Kanban. Han talar om att problematiken ligger i användningen av Kanban som medför att mängden folk som får vara med i varje kolumn på Kanban-boarden minskas, detta för att tvinga folk att koda tillsammans med varandra. Han säger att det är svårt att kombinera en linjeutvecklingsmetodik, som Kanban är, med Scrum som är tänkt att dela upp uppgifter i kortare projekt, ”*Sen tror man att dom här två ska lira tillsammans.*” (s. 52) Han säger att det får folk att utforma egna metoder i stället för att följa en metod enligt dess teori. P2 säger även att om en kombinerad av dessa två metoder sker skapas en tungrodd modell utifrån saker som inte ska vara tungrodda.

P4 hävdar att det mer eller mindre går att göra Scrum av Kanban. Senaste tiden har företaget haft lite olika projekt där vissa har arbetat med Scrum och i vissa andra projekt har det varit mer oklart gällande vilken metod som använts. För att få en gemensam metod i alla projekt funderar de på att börja använda Kanban vilket enligt P4 kan anpassas till stor del. De använder Kanban som en inspirationskälla men har för avsikt att inte använda metoden rakt av, ”...Jag tror inte det finns nått företag, det hade varit konstigt med ett företag som skulle köpa en process rakt av utan att fundera igenom och gallra, eller lägga till.” (s. 71)

4.4 Produktägare och kundperspektiv

Alla intervjupersoner uttryckte en problematik kring det som inom Scrum kallas produktägaren. P2 berättade om problemet att få in kundens åsikter oavsett vem som blev produktägare. Det optimala var enligt honom att kunden själv skulle föra fram sina tankar. När det faktiskt försökts uttryckte han det som följer: *”Men dom kommer inte regelbundet. I helvete heller! Det är alltid då mycket viktigare att göra andra grejor”* (s. 54). Han har också varit med om att ha en projektledare som produktägare, men det har blivit fel eftersom denne inte är opartisk och kan ha problem att föra fram kundens perspektiv. Säljare har en bättre kontakt med kunden men har ingen förståelse för vad som går att utveckla. Han har inte heller upplevt något engagemang från säljare vad gäller agil utveckling.

P3 hade helst velat ha en kund som produktägare men har i praktiken kört en slags hybridmodell. Kunden och projektledaren delar i princip upp ansvaret. Kunden uttrycker vad denne önskar och projektledaren försöker översätta detta till en backlogg. P4 och P5 har också erfarenhet av att arbeta såhär och menar då att man för en dialog mellan projektledare och kunder för att få fram något som är både önskat, behövt och realistiskt genomförbart. Detta kräver att kunden visar intresse men inte att denne sätter sig in i agila metoder. Vissa kunder har insikt i agila metoder och hade kunnat skriva en egen backlogg, men P3, P4 och P5 säger att det inte utnyttjats.

P4 och P5 har som sagt följt en hybridmodell i sina kundprojekt, d.v.s. när de integrerar sin produkt hos en specifik kund. När de arbetar med sin grundläggande produkt inhouse har de dock ingen specifik kund att föra en dialog med. Då har de interna produktägare som utför marknadsanalyser och lägger utifrån dessa upp konceptuella features. Produktägarna får sedan i uppdrag att översätta dessa till mer konkreta features och en tidsplan för utveckling.

P4 – *” Till exempel: Nu vill vi ha ett nytt cartoon-filter där alla ser ut som seriegubbar om man har tagit en bild. Det är kanske på den nivån eller lite mer detaljerat som produktägaren tar det, sen blir det upp till respektive projektteam att bryta ner det här till nånting som är mer definierat, och planera in när det skulle kunna genomföras i tiden”* (s. 74)

P1 berättade att han och hans team arbetat helt mot en intern produktägare. De fick en klar uppgift för vad de skulle prestera och det fungerade väl för dem. Hur väl det motsvarade förväntningarna för den slutgiltiga kunden är dock svårare att besvara.

Gemensamt kan man se att de upplevde det svårt att hitta en person som på egen hand kan uppfylla produktägarrollen. Det krävdes både en förståelse för vad som önskas och vad som kan presteras och därför har det i flera fall blivit en uppdelning på två personer.

4.5 Formellt och informellt arbetsätt

Vi fick intrycket från flera intervjupersoner att stora delar av metoderna delas informellt. P5 talar bland annat om att en viss planering av sprintar sker i förväg, där buggar, önskemål kommer in som inte kan lösas direkt. Då lägger de in att den ska utföras nästa sprint, eller den

därefter. Budgetering utförs också, men är inte en formell del av metoden. P4 berättade även om hur en gemensam kärna i arbetssättet utarbetades informellt mellan projektledare. Erfarenheter delades och tankar diskuterades. I flera fall har man haft kontinuerliga möten med ett väldigt vagt mål om att diskutera projekt. Både P3 och P4 berättade om sådana möten. Arbetssätt diskuterades ofta men fritt och utan dokumentering.

Enligt P1 behövdes ingen genomgång av deras tolkning av Scrum. Alla ”visste” vad det innebar. Inget av företagen vi besökte kunde i nuläget säga exakt vad som var deras arbetssätt. De var inspirerade av Scrum och Kanban, men de följde dem inte helt, och metoderna täckte inte in mer än en bråkdel av vad de faktiskt gjorde. När vi frågade om hur man visste hur arbetet skulle utföras hade alla olika versioner av samma svar. De pratar med varandra. På så vis gjorde de en användbar praktik av ett agilt tankesätt.

4.6 Kommunikationsstruktur

Övergripande har vi märkt att det är främst den informella kommunikationsstrukturen som används i projektsammanhang. P4 säger att kommunikationen mellan projektledare sker via en informell kommunikation, men inom projektet menar han att det även är viktigt att använda en mer formell struktur för att få projektdeltagarna att förstå vad som ska utföras. Samtliga intervjuade personer talar om att projektteamen som finns i verksamheten är små, det leder till att det blir naturligt med en informell kommunikation mellan projektdeltagarna och projektledare. P5 säger att deltagarna i projektteamen känner varandra väl mellan teamen vilket gör att den informella kommunikationen lättare faller på.

P3 och P4 angav att de i det dagliga arbetet går runt bland projektdeltagarna och säkerställer att de arbetar med sina uppgifter utan problem. P2 menar att det underlättar att vara få deltagare i ett projektteam då det främjar ett initiativ till att hjälpa sina medarbetare och att samtidigt upprätthålla en konversation som bidrar till att projektets deltagare vet vad andra gör och vilka problem som eventuellt har uppstått.

Trots att det under det dagliga arbetet sker en informell kommunikation säger P4 att dagliga möten hålls för att säkerställa att alla inblandade intressenter får en bild över hur situationen för arbetet ser ut. P1 säger att det på en tidigare arbetsplats inte fanns någon metodik för hur kommunikationen skulle ske inom företaget, och hur projektdeltagarna skulle få en gemensam bild av arbetsmetoden. Det var upp till var och en att lära sig arbetet med metoden och de flesta hade vetskap om hur de skulle gå tillväga i arbetet, P1 - *”Ja, nä... det fanns nog inget sånt, jag vet inte om vi hade nått informationsblad och lite så som vi hängde upp, men jag tror de flesta har rätt så bra koll på det”* (s. 48)

P3 påpekar också att det känns onödigt att lägga tid på att skriva en handbok för hur arbetet ska utföras då hon menar att ändå ingen kommer att läsa den, hon lägger hellre tid på att personligen gå runt bland medarbetare och se efter hur arbetet fungerar samt för att ställa kontrollfrågor.

Det vanligaste sättet att sprida information är via muntlig kommunikation och det sker enligt de intervjuade personerna både formellt och informellt via vardagligt arbete och på planerade möten.

5 Diskussion

5.1 Metodval

Som vi nämnt i empirin är det i princip bara Scrum och Kanban som våra intervjupersoner överhuvudtaget övervägt. De ser dem som de enda reella alternativen. När de vill ha en mer vattenfallslik metod så gör de en egen kombination av vattenfallsmodellen och Scrum. Istället hade de kunnat överväga exempelvis FDD som är utvecklat just med samma tanke. Fördelen med FDD hade varit att den dels redan utgör en färdig helhet och dels är baserad på best practices. Att istället utveckla en egen metod med ungefär samma utgångspunkt kan helt klart ge fördelar i form av anpassning till det specifika företaget. Vi ser dock klar anledning att ifrågasätta varför FDD inte övervägts i just den situation som metoden är framtagen för.

Vill de ha något friare än Scrum så hade de kunnat överväga XP, eller Crystal som inte har ett klart flöde överhuvudtaget. Istället för att använda en ofullständig metod hade de då kunnat använda en fullständig sådan, utan att den begränsade deras anpassningsförmåga till en föränderlig vardag. Även i detta fall hade de fått metoder som är grundade i best practices. Även om Scrum också är det så kan det samma inte sägas för Scrum med modifikationer.

Istället väljer de Kanban med insprängda element från Scrum. Exempelvis kan de ta allt från Scrum förutom Sprinten och kalla det Kanban. De kan då motivera det impopulära valet att inte köra Scrum helt ut med att de i själva verket kör Kanban. De verkar dock inte ha undersökt Lean direkt. Trots att Kanban är så nära sammankopplat med just denna metod. Vi fann inte heller några tecken på klara ramverk för hur Kanban och Scrum ska kombineras. Något som efterfrågats tydligt i tidigare studier som vi nämnt. Detta kan dock bero på att de inte anser sig använda Scrum, utan endast inspireras av den.

De överväger inte i närheten av alla metoder som finns, och det är också sällan som de hittar en metod som de faktiskt kan använda som den är. En av intervjupersonerna var öppen för att man skulle utveckla sina egna metoder, och alla uttryckte att man borde använda dem främst som inspiration. Vi tror detta är ett nödvändigt koncept i många situationer men som verkar aningen tabu i praktiken, alla vill inte erkänna att de anpassat sin metod. Vi ser dock en tendens att anpassa metoder informellt utifrån organisatoriska behov för att skapa ett praktiskt arbetsätt. Vi ser ingen anledning att inte formalisera denna anpassning, annat än sociala konventioner.

Implementeringen av agila metoder har enligt vår empiri kommit från tre huvudsakliga källor. Gemensamt intresse bland utvecklare, gemensamt intresse bland projektledarna eller en ensam inspirerad chef. Oavsett källan för idén så har alla påpekat vikten av att hela organisationen är överrens och motiverad. Detta stämmer väl överrens med unfreeze-change-refreeze-modellen som vi tog upp tidigare. Även de misslyckade implementeringsförsök som beskrevs förklarades med just brister i vad som kan tolkas som unfreeze-fasen.

En lyckad implementering bygger enligt både teori och empiri på att den motiveras, planeras och grundas i den egna organisationen. Just motiveringen läggs det extra vikt på av Hayes &

Richardson (2008) som identifierat just brister i motiveringen som ett stort problem inom implementering av agila metoder. Landy och Conte (2007) skriver att det är viktigt att låta metoden grundas djup nere i organisationen och låta den växa fram långsamt så att alla berörda får en förståelse över arbetssättet.

I allmänhet upplevde vi ett slags socialt grupptryck där man skulle följa Scrum eller Kanban för att alla andra gjorde det. Med tiden verkade dock alla intervjupersoner, även om de inte ville släppa titlarna, röra sig bort från dessa metoder och mot en slags egen metod utgående ifrån det agila manifestets principer. Man gick allt mer bort från en agil metod och mot ett agilt tankesätt. Detta anser vi rimmar väl med det agila manifestet som ligger till grund för just Scrum och Kanban. Det tyder på enkelhet och anpassning till förändringar.

5.2 Metodanvändning

Vi kan se tydliga tecken på att en agil utvecklingsmetod ofta anpassas för att på bästa sätt möta verksamhetens krav och mål. Det finns en gemensam bild från de undersökta företagen om att specifika delar väljs ut från metoderna för att sedan sättas in i en kontext där de passar bäst. Även de tidigare studier vi tagit upp har funnit att metoder inte ger klara riktlinjer för exempelvis ledarskap, ansvar och extern kommunikation utan behöver kompletteras eller anpassas för att möta organisationers behov. Det har framkommit under intervjuerna att det är taktiskt att införa en metod steg för steg, genom att starta så minimalt som möjligt för att sedan lägga till delar som visar sig vara betydelsefulla för verksamheten.

Det är ovanligt att metoden implementeras rakt av och att den används med alla dess delar enligt en instruktionsbok. Övergripande finns det en inställning hos företagen att se till vilka behov företaget har och därefter bestämma hur metoden ska anpassas. Våra intervjuer har gett oss bilden av att det är svårare för en större organisation att implementera en agil metod utan anpassning än det är för en mindre organisation. Även användningen av mindre projektgrupper underlättar för att använda metoden som det är tänkt att den ska fungera, vilket stämmer överrens med vad som nämnts i tidigare studier. Agila metoder förespråkar mindre arbetsgrupper, men den maximala lämpliga storleken skiljer sig mellan metodernas teori och uppgiftslämnarnas faktiska erfarenheter. Från intervjupersoner har vi fått en lämplig storlek om maximalt 5-6 personer, teorin föreslår upp till 10 personer.

Istället för att börja använda en metod rakt av och tro att det ska lösa alla problem som eventuellt finns inom organisationen, anser flera av de personer vi intervjuat att organisationer bör utvärdera sina mål och hur metodiker kan stödja dessa. Detta stämmer även med den tidigare studie som Hayes & Richardson (2008) har utfört. Det går inte att implementera en metod rakt av och tro att det får bukt med problem så som t.ex. dåliga programmerare.

Ett återkommande val är att många av företagen vi pratat med väljer att använda en Product Backlogg, att utföra dagliga Scrum-möten och att använda rollen produktägare i sin verksamhet. Gemensamt bland företagen är också att den del som oftast plockas bort från metoden är sprint-delen eftersom den begränsar arbetet till viss del. Framförallt om de

behöver anpassa sig till förändringar på en kortare basis som exempelvis supportavdelningen i vissa fall behöver göra.

Vi har insett att många företag inte reflekterar över att det använder sig av en Product Backlogg. De tar det för givet att ha en dokumentation över vad som ska göras och vilka krav som finns på projektteamet att de inte tänker på att de använder sig av en del från Scrum. En av våra intervjupersoner uttryckte att en fördel med Kanban var att de kunde bygga sin backlogg på ett sätt som passade dem, och han såg inte motsägelsen i att Kanban inte har en backlogg. Vidare tycker vi det är bra att företagen uppmanar till formell kommunikation i form av dagliga möten med projektteamens medlemmar, för att upprätthålla deras förståelse över det pågående arbetet och användningen av metoden. Som tidigare nämnts i teoridelen så är informell kommunikation bättre för att sprida information, men genom dagliga möten säkerställs att information sprids på regelbunden basis och på ett vis som är direkt relaterat till verksamheten.

5.3 Kombination av metoder

De vanligaste metoderna att kombinera inom vårt urval är Scrum och Kanban. Kanban används för att hantera ett företags flöden och enligt en av de intervjuade kombineras metoden med Scrum för att komplettera arbetsättet med delar som anses behövas för att utföra arbetsuppgifterna samt för att hålla koll på vad som sker inom projektet. En annan intervjuperson säger att de använder Scrum kombinerat med deras egna erfarenheter av hur ett projekt ska hanteras. I den situationen tillämpas delar från Scrum som anses vara nödvändiga för att driva verksamheten framåt. Under projektets gång kan företaget välja att plocka in eller utesluta delar för att förhöja projektets förmåga att leverera ett bra resultat.

Det förekommer även på ett företag en kombination av vattenfallsmodellen och arbete med Scrum. De arbetar på så sätt för att enklare kunna rapportera exempelvis budgetuppföljning till berörd chef. De anser att det underlättar att ha tydliga milstolpar, vilket inte på samma sätt finns i ett agilt arbetsätt, för att kunna besvara kunden på frågor som berör tidsplanering, kostnader osv. Det kan dock som vi nämnt tidigare ifrågasättas om inte FDD ensamt hade kunnat fylla samma roll och möjligen göra det bättre.

Det är dock lite olika åsikter i den här frågan från de intervjuades sida. Det finns de som förespråkar en kombination medan det främst är en intervjuperson som inte förstår varför man ska kombinera just Kanban med Scrum. Eftersom de två metoderna härstammar från olika branscher och organisationsstrukturer så uppstår det enligt uppgiftslämnaren ett komplicerat sätt att arbeta på, när metoderna i själva verket ska bidra till ett enkelt sätt att arbeta. Kanban, tillsammans med Lean härstammar från Toyotas arbete utifrån en linjeorganisation. Denna grund är avsevärt skild från grunden i det agila tankesättet och de är därför enligt vår intervjuperson svåra att sammankoppla till en effektiv helhet.

Både intervjupersoner och tidigare studier som vi tagit upp är dock eniga om att en kombinerad inte kan ske friktionsfritt. Jyothi & Rao (2012) efterfrågar ett klart ramverk för

hur kombineringsen ska se ut, och vi har från intervjupersoner hört att metoder oundvikligen går emot varandra på någon punkt. Ett undantag skulle möjligtvis vara Lean och Kanban, men Kanban ses också som en del av Lean vilket kan ge ett felaktigt antagande att de är två skilda metoder. Vårt urval har dock inte sett Lean som ett relevant alternativ att arbeta efter, så en kombination Lean och Kanban har automatiskt blivit utesluten.

Fenomenet med att ta de mest vitala delarna från olika metoder och sammanfoga till en helhet som passar verksamheten anser vi vara ett bra koncept för att uppnå ett väl fungerande arbetsätt. Som våra intervjuade säger så är det inte givet att en viss metod fungerar på ett företag bara för att det fungerade tillfredsställande på företaget där metoden växte fram. Vi ser dock utifrån vår och andra studiers fynd anledning att understryka vikten av att detta görs utifrån en punkt av förståelse och motivering av de val som görs. Om organisationer utifrån det agila manifestet plockar väl fungerande komponenter från olika metoder, och utvecklar en skraddarsydd metod för den egna organisationen, anser vi det logiskt att denna metod skulle vara väl anpassad till den specifika organisationens behov.

Sammantaget så tror vi det är positivt att dra inspiration från flera metoder och hitta en mellanväg som passar den egna organisationen. Däremot är det motsägelsefullt och ofta omöjligt att tro att man kan köra två metoder fullt ut jämsides utan ett klart ramverk däremellan. Även med ett klart ramverk är det inte säkert att en kombination av två effektiva metoder bildar en effektiv helhet.

5.4 Produktägare och kundperspektiv

Produktägarrollen i Scrum har av flera intervjupersoner beskrivits som problematiskt. Tanken är att man ska få in kundperspektivet i backloggen. Då uppstår en problematik i dynamiken. Backloggen ska matcha både kundens önskemål och utvecklarnas möjligheter. Det är svårt att hitta en person som kan uppfylla båda dessa perspektiv. Från både teori och intervjupersoner har vi uppfattat ett önskemål om att produktägaren bör vara en kund. Vi har dock också hört under intervjuerna att de flesta kunder saknar både intresse och kunskap för att vara praktiska produktägare.

Kunderna beskrivs som att de inte har tillräckligt med tid att lägga på en ständigt förändrande backlogg. Den enda väl fungerande varianten vi hört är att dela rollen i två. Den delas då mellan kunden och en projektledare. Kunden uttrycker en konceptuell version av sina önskemål som projektägaren sedan översätter till en konkret backlogg. Oftast måste det dock föras en dialog för att försäkra att projektledaren tolkat rätt och att kunden med facit i hand faktiskt är nöjd med utvecklingens framskridande.

Vid inhouse-utveckling av en egen produkt verkar det enklare att hitta en specifik produktägare. Denne ska huvudsakligen utföra marknadsanalyser, och utifrån dessa lägga upp en plan för utvecklingen. Även i detta fall verkar det dock som projektledarna måste hjälpa till att översätta detta till en konkret backlogg. Sammantaget verkar det näst intill omöjligt att ha en prioriterad backlogg utan en dialog. Kunden kan inte prioritera uppgifter optimalt utan att

veta hur lång tid den tar, och projektägare kan inte prioritera dem utan att veta hur viktiga de är för kunden.

Dock anser vi att vissa problem uppstår ur en missuppfattning av vad produktägaren ska göra. Flera uppgiftslämnare verkar utifrån våra intervjuer anta att produktägaren ensam ska ta fram och prioritera backloggen. Vår uppfattning utifrån teori är istället att hela Scrum-teamet ska skapa backloggen tillsammans, men att produktägaren är ytterst ansvarig för den och alltid har sista ordet.

Om man gör den tolkningen så blir det betydligt lämpligare att ha en kund som produktägare om denne visar intresse. Detta eftersom produktägaren då inte behöver veta vad som är genomförbart eller inte. Denne behöver endast utifrån diskussion med utvecklingsteamet ha möjlighet att uttala sig om vad som är viktigt för kunden, och utifrån detta prioritera backloggen. Om produktägaren istället ska vara ensamt ansvarig för utarbetningen av backloggen uppstår en situation där denne måste vara väl införstådd med både utvecklarnas och kundens perspektiv. Som det framkommit i intervjuerna finns det få som ensamma kan fylla en sådan roll.

Vi spekulerar att Scrum kanske hade behövt en tydligare struktur för hur backloggen ska utarbetas, med klara riktlinjer och ansvar. Tanken som vi förstått det är att alla ska kunna lägga till och ändra i backloggen ohindrat, och att dessa ändringar sedan ska godkännas av produktägaren. Detta verkar dock inte fungera optimalt i praktiken, ingen vi intervjuat har arbetat efter denna princip. Detta tror vi kan bero på att produktägaren är den enda som har ett formellt ansvar för backloggen. Vi anser att det delade ansvaret för backloggen borde formaliseras till viss grad, för att markera för projektdeltagarna att de bör bidra till den. En bra backlogg bygger utifrån teorin på att alla gemensamt bidrar till den.

5.5 Formellt och informellt arbetsätt

Vi har utifrån intervjuerna uppfattat att stora delar av företagens arbetsätt var framtagna, uppehållna och utlärdas via socialisering. Detta är inte nödvändigtvis negativt. I minst ett fall verkar det ha fungerat väldigt bra att förlita sig på ett sådant socialiseringsbaserat, informellt arbetsätt. Dock verkar det inte fungera väl när organisationer växer sig större. Det blir då ett ökat tryck på att förstå varandra även mellan grupper. I det fallet menar vi att det finns en brist på externalisering, uppgiftslämnarna har svårt att kommentera eller motivera de delar av arbetsättet som är underförstådda. Det blir då också svårt att dela med sig av eller utvärdera det befintliga arbetsättet. Ett av företagen arbetar just nu på att externalisera den ”*minsta gemensamma nämnaren*”. Vi tror det är positivt ifall arbetsättet görs medvetet, eftersom det ökar möjligheten att bearbeta och utveckla det.

Socialisering av kunskap verkar fungera bäst inom grupper på upp till cirka sex personer. Större grupper kräver någon form av externalisering av kunskap och formalisering av arbetsätt. Vi tvivlar på att kunskapen måste dokumenteras men den bör i vår mening åtminstone verbaliseras. Jacobsen & Thorsvik (2008) skriver att det är viktigt att implicit

kunskap om arbetssätt görs explicit. De lägger dock inget krav på att det görs genom dokumentation. En tanke med det agila tankesättet och något som också observerats av intervjupersonerna är att dokumentering inte skapar kundvärde. Det innebär dock inte att kunskap inte ska spridas, ifrågasättas eller till viss grad formaliseras. Agilt arbete kräver tydlig kommunikation.

5.6 Kommunikationsstruktur

Den struktur som förekommer mest bland företagen vi varit i kontakt med är den informella kommunikationsstrukturen. Eftersom alla företag håller sina projektteam inom den storleksordningen som Scrum-metodiken rekommenderar är det fördelaktigt att använda sig av en daglig informell kommunikation utöver det formella morgonmötet, Daily Scrum Meeting. Det medför att projektdeltagarna uppdaterar varandra om vad som sker i projektet vilket leder till att samtliga medlemmar har vetskap om varandras status i arbetsuppgifterna. Enligt en uppgiftslämnare främjar informell kommunikation ett tätare samarbete och ger större möjligheter till att kunna hjälpa varandra vid problematiska situationer. Jacobsen & Thorsvik (2008) skriver att den informella kommunikationen har stor potential för lärande samt att förtroendet för informationen stärks. Därför anser vi informell kommunikation positiv att anamma inom organisationer. Dock inte uteslutande.

Om en metod utarbetas och implementeras utan dokumentering eller kontinuerlig uppföljning, ser vi en fara i att det kan leda till att kunskapen som finns om arbetssättet internaliseras helt. För att undvika det är det fördelaktigt att skriva ned informationen vilket medför att kunskapen blir explicit. Samma process kan genomföras via verbalisering men det krävs i så fall mer arbete för att överföra kunskapen till nya personer. Vi menar därför att det skulle vara fördelaktigt ifall en manual fanns tillgänglig som hänvisar till befintligt arbetssätt, även om en sådan manual troligen inte anses användbar i det dagliga arbetet. Denna manual hade enligt oss varit lämplig för nya projektdeltagare och externa intressenter. På sikt finns det dock fördelar med att projektdeltagare socialiseras av medarbetare och internaliserar kunskapen.

Som tidigare nämnts arbetar företagen med små team, och företagen i sig själva är också små. Det anser våra uppgiftslämnare bidra till att det naturligt uppstår en informell kommunikationsstrategi eftersom medarbetarna arbetar nära varandra. De ger exempel på att det finns en god personlig kontakt mellan medarbetarna till följd av att de arbetar nära varandra. Det är därför inte nödvändigt att det krävs en formell kommunikationsstrategi för att förmedla information inom organisationer med en eller ett fåtal arbetsgrupper.

6 Slutsats

Den frågeställning vi arbetat utifrån är: *Hur anpassas agila metoder till att fungera i en organisations verksamhet och hur kommuniceras arbetssättet till medarbetare?* Frågan har lett oss till ett antal observationer, argument och slutsatser. Vi har observerat att de företag vi besökt egentligen endast ser Scrum och Kanban som reella alternativ för agilt projektarbete. Vi har konstaterat att de sällan har haft möjlighet att använda dessa metodiker utan anpassning eller klart subjektiv tolkning. Användningen av framförallt Kanban skiljer sig påtagligt från beskrivningen i teori.

Vi har uppfattat att agila metoder ofta ses ur ett komponentperspektiv istället för som en helhet. Delar kan implementeras utan att till fullo använda sig av en specifik metod. Product Backlogg är ett sådant exempel. Produktägare och sprintar å andra sidan orsakar i flera fall problematik som gör att organisationer vi undersökt väljer bort dessa komponenter och istället använder sig av Kanban med tydliga inslag av Scrum.

Vi har även sett en klar tendens bland våra intervjupersoner att de allt mer rör sig bort från agila metodiker och närmar sig ett mer allmänt agilt tankesätt där de kan utarbeta ett eget arbetssätt på relativt informell basis. Ju mer de förstår metoderna, desto mer verkar de överge dem. Vi kan gå så långt som att jämföra agila metoder med en agil krycka. Vår undersökning antyder att agila metoder bör ses som en inspiration till eget arbetssätt snarare än ett arbetssätt som är färdigt för implementering.

Vi har konstaterat att arbetssätt till stor del existerade i form av tyst kunskap inom vårt urval. Om man accepterar antagandet att utveckling av arbetssätt leder till en bättre produkt, anser vi det också logiskt att man bör vilja göra denna kunskap explicit så att den kan behandlas mer analytiskt. Framförallt när organisationer växer verkar de få minskade möjligheter att dela kunskap via socialisering, speciellt mellan grupper som är beroende av varandra, och då krävs det att kunskapen externaliseras så att den kan delas på ett mer öppet och formellt sätt.

Vi har genom vår undersökning fått en väldigt intressant inblick i utvecklingsföretags vardag. Vi har sett en stor diskrepans mellan teori och praktik. Vi har observerat ett antal ämnen som vi anser förtjänar fortsatt efterforskning. Vi tror först och främst att det hade varit givande att upprepa vår undersökning med ett bredare urval. Vi har upptäckt specifika frågor inom vårt problemområde som vi ser som intressanta för framtida forskning. Produktägarrollen i arbetslivet är något vi tycker förtjänar mer undersökning. Relationen mellan Scrum och Kanban kan undersökas djupare, och även varför så många metoder inte ses som reella alternativ.

Vi anser att vi producerat ett givande exempel av hur agila metoder faktiskt används, hur arbetssättet kommuniceras, och vilken problematik som existerar. Vi har visat upp reella problem och potentiella lösningar inom det agila arbetssättet som kan ge viktiga lärdomar även om de inte kan generaliseras. Agila metoder fungerar i dagsläget inte optimalt inom vårt urval. Det finns en klar skillnad mellan det formella arbetssättet och det faktiska arbetssättet. Vi ser dock ingen anledning att angripa det agila tankesättet, och tror att den allt mer ökade

förståelsen för det agila tankesättet i dessa fall leder till en positiv evolution av det agila arbetsättet.

7 Bilagor

Bilaga 1, Intervjufrågor

1. Vad har du för utbildning?
2. Vad har du för tjänst?
3. Vad har du för erfarenhet av agila projekt?
4. Vad har varit din huvudsakliga projektroll?
5. På vilket sätt påverkar ett projekts utformning er metodanvändning/metodval?
6. Vilka agila utvecklingsmetoder har använts i era projekt?
7. *Om bara en metod använts:* Hur har ni anpassat metoden till era projekt?
8. *Om flera metoder använts:* Vilka metoder har ni valt att kombinera, hur, varför?
9. Har speciella delar ur metoderna valts ut? Vilka? Varför?
10. Hur bestämmer ni vad som ska väljas ut ur metoder och hur ni tolkar dessa element?
11. Har ni en kontinuerlig utvärdering/utveckling av era metodval?
12. Hur kommuniceras det utvalda arbetssättet till övriga organisationen? (Hur lärs det ut?
Skriftlig/muntlig kommunikation?)

Bilaga 2, Intervjuprotokoll, P1

H – Tänkte börja med lite frågor om dig, lite kort bakgrund.

P1 – Mmm

H – Vad har du för utbildning?

P1 – Systemvetenskap

H – Kandidat eller?

P1 – Ja, det kan man säga, jag har inte avslutat alla kurserna ännu, men jag har skrivit C-uppsats och så vidare.

H – Ok. Vad har du för tjänst här?

P1 – Ja, jag vet inte vad jag kallas, systemutvecklare/konsult.

H – Ok

P1 – Nånting, rätt så luddigt hittills tror jag.

H – Ok, vad har du för erfarenhet av agila projekt?

P1 – Ehm, agila projekt... vad ska man säga, allting vi gör nu förtiden känns det som är agilt, så från det att jag började jobba så har det varit med utgångspunkt av att det ska vara väldigt agilt. Så framförallt förra jobbet så 3,5 nästan 4 år i ett team, väldigt scrumbaserat.

H – Ok

P1 – Sen här håller vi väl på i dess linda nu då och det är lite mer kanbaninspirerat. Vi kör än så länge så lättviktigt som möjligt, så om man börjar med i stort sett ingen metodik alls och så lägger man på element för element som det känns att vi behöver.

H – Ok, vad har du haft för huvudsaklig projektroll?

P1 – Alltså, i scrum så är man ju alltid... där har jag alltid varit teammedlem, alltså där har vi ju så få roller.

H – Ja, det är ju bara dem tre

P1 – Precis, ja. Och i kanban har man egentligen inga roller alls, utan vi kör på.

H – Ok, och informellt då? Har du...

P1 – Hmm, jag har väl alltid tagit en drivande roll och varit en av dem som alltid försökt modifiera processen och så.

H – Mmm, få se här... på vilket sätt påverkar projektets utformning er metodanvändning och metodval?

P1 – Det påverkar nog väldigt mycket. Det ska ju alltid vara det man gör som driver metodiken, inte tvärtom. Eller det bör alltid vara, sen är det inte alltid så. Så det är framförallt det vi försöker göra, att vi ser på det konkreta behovet och så lyfter vi in den metoden vi behöver. På förra jobbet så var det lite annorlunda, det var vi väldigt beroende av tex kontinuerliga releasecyklar, så därför valde vi scrum kan man väl säga framför allt. Och sen då gjorde vi ledet av att vi körde in en hel full plan av implementation av scrum med alla

certifieringar och kurser osv och sen började vi därifrån anpassa. Så det kom egentligen från två olika håll.

H – Ok

P1 – Men det bör alltid vara projektet som styr metodiken.

H – Ni nämner att ni har tagit den i omgångar, så att säga kanban, hur har ni liksom... vilka segment har ni valt först och hur har ni gjort det för att anpassa till ert arbetsätt?

P1 – Det man kan säga framför allt det är väl att när vi började med att sätta igång med inhouseutvecklingen här, så började vi då med att se vad vi hade egentligen för krav och dem projekten som finns just nu, så hade vi inget behov av kontinuerlig release tex, utan vi ville mer ha ett flöde för utveckling, och då blir kanban mer naturligt. I och med att det inte har ett slut och ett mål utan det bara flyter på även i backloggen, så det var nog mest där vi började, så egentligen började vi med att vi behövde ett behov, eller vi hade ett behov av att visualisera vårt arbete. Så då blev det liksom en kanbanboard, det var det första, för vi ville se vad var och en höll på med och vad som skulle göras närmast. Så det var väl det första liksom, och sen började vi då rita upp vilka kolumner behöver vi på vår board? Och det är samma sak, hålla det så minimalt som möjligt till en början och sen om vi behöver så lägger vi till en kolumn eller drar ifrån osv. men, så man börjar väldigt enkelt, och kanban är så väldigt lättviktigt, så där behövs liksom... vi har inte behövt så mycket mer.

H – Hur, alltså... mer organisatoriskt sett, hur har ni bestämt vad som bestäms liksom, är det nån som... eller är det mer en allmän diskussion, och tänkte också på hur det tolkas?

P1 – Ofta är det, på förra jobbet så var det i princip så att teamet var väldigt självbestämmande, så vi var ett utvecklingsteam, vi vill förändra det sättet vi jobbade på, vi vill jobba med scrum – så går man fram till ledningen och sen sätter man igång. Och sen är det liksom ingen som lägger sig i hur vi organiserar det så länge det fungerar.

J – Är det främst i scrum-arbete eller kan det förekomma i andra?

P1 – Det förekommer nog överallt, alltså jag tror nästan all metodik så här kommer nästan alltid från teamet, det beror säkert lite på hur stort företagen är, nu har jag bara jobbat på lite mindre, men då är det liksom, jobbar du på ett utvecklingsföretag så är det alltid utvecklingen som är i centrum, det är aldrig som i andra organisationer att du låter affärerna styra för mycket, utan det är produkten och kreativiteten hos utvecklarna som alltid måste styra.

J – Ok

P1 – Och det är liksom ingen i management som bryr sig om om man jobbar med scrum eller kanban. De vill ju bara åt mjukvaran, så det kommer liksom alltid innefrån. Och största problemet då blir snarare hur du får, säg 10 utvecklare att komma överrens om hur man jobbar eftersom att alla ändå är ganska olika.

J – Men det är upp till den gruppen att bestämma?

P1 – Ja, min erfarenhet är att det alltid är upp till gruppen, men sen är det ju annorlunda, du kan ju komma in i ett befintligt team osv och då skiljer det sig säkert.

H – Tänkte på det här med tolkning av metoder tex specifika argument som backlog och så, har ni gått utbildningar i det eller?

P1 – Nej, alltså när det gäller scrum så hade vi scrummaster, produktägare på certifieringar, men teamet är nog självlärt kan man säga. Och där har vi, där började vi som sagt med att köra allting enligt, där finns ju liksom böcker, så nu kör vi så här och sen så började vi modifiera om vi märkte att det inte funkade.

J – Hur gjorde ni för att lösa kommunikationen och kunskapsspridningen inom gruppen, hade ni nån metodik för det?

P1 – För just det agila?

J – Ja, just i er grupp där, att nu bestämmer vi att vi kör det här, då måste ju alla vara med på det.

P1 – Ja, precis...

H – Få en gemensam bild av vad det är liksom

P1 – Ja, nä... det fanns nog inget sånt, jag vet inte om vi hade nåt informationsblad och lite så som vi hängde upp, men jag tror de flesta har rätt så bra koll på det.

J – Så det var upp till var och en där att sätta sig in i det?

P1 – Ja, lite så. Det som är styrt är det att nu ska vi ha ett morgonmöte, då ska du dyka upp där klockan nio liksom, det är mest det som finns utifrån, och sen sätter man upp, vad heter det... vi hade liksom våra egna workingrules som vi satte upp på ett papper och då är det det som ska gälla liksom.

H – Ok, Har ni haft någon kontinuerlig utvärdering eller utveckling av ert medtodal?

P1 – Mmm, alltså i scrum så ingår ju det, om man nu kör fullt ut scrum så har man ju ett retrospektiv i slutet av varje cykel, då utvärderar man ju hela arbetet och då kan det ju dels vara vad som har gått bra eller dåligt med utvecklingen med också utvärdera hur metoden har fungerat, vi hade cyklar på tre eller fyra veckor så var tredje var fjärde vecka så satt vi liksom allihop i en grupp och tittade på vad vi hade gjort under den här perioden och då även eventuellt gjorde vi justeringar.

J – Det var då nånting ni kunde ta med er till nästkommande projekt?

P1 – Ja, alltså på förra jobbet jobbade vi med samma produkt hela tiden, så det var därför vi lyckades tror jag med att slimma vår metod så bra för att det var hela tiden samma sorts utveckling, annars varierar det ju jättemycket med projekt

J – Då är det svårt också att ta med sig nåt gammalt från tidigare projekt eller?

P1 – Precis, eftersom du alltid ska utgå ifrån projektet

J – Ja, det är det som ska styra...

P1 – Ja

J – För att återgå till vår fråga lite grand på om ni anpassar eller kombinerar olika metoder, kan man köra scrum tillsammans med nån annan agil utvecklingsmetod?

P1 – Mmm, alltså du kan ju köra element av det, vi hade ju... alltså... på både scrum och

kanban jobbar man med en tavla i centrum, och det vi gjorde var att vi hade lite av kanbans kolumntänk i vår scrumboard så vi förde in lite av det, jag vet inte om ni har koll på exakt...

H – Just kanban har vi inte...

P1 – I princip så är det en tavla där du har x antal kolumner som säger: det här är vad vi ska göra, det här är vad som pågår, det här är vad som ska testas och det här är färdigt

J – Ok

P1 – Och hela syftet är att i varje kolumn förväntas finnas x antal läranden, så du hela tiden begränsar work in progress, för att om det då uppstår en flask... en propp nånstans, så ska alla hjälpas åt att lösa den så man kan gå vidare, så det är lite, kommer väl lite av det här leantänket att du jobbar här mer på en produkt från start till mål istället för löpande band

J – Mmm

P1 – Så att du hjälper liksom till att, medan scrum har riktigt inte det utan där är det lite mer: jag kan sitta på min del i flera veckor liksom utan att det påverkar dem andra.

H – Så det blir lite mer sekventiellt i kanban som i lean tex?

P1 – Ja, precis, alltså du har inte den här cykeln som du har i scrum utan där jobbar du ju från start till mål och sen börjar man om, medan i kanban flyter det bara på hela tiden, bara en strid ström och det enda som är viktigt är att det inte stoppar upp nånstans pga nåt hinder. Så kan man väl säga. Och det vi har gjort nu som sagt, eftersom vi inte ska ha färdiga releaser så kör vi en kanbanboard som bara flyter men vi tar in tex. morgonmöten från scrum, för det vill vi ha, och vi la även in ett retrospektiv i slutet av varje vecka, så där har vi plockat lite på det hållet

J – Så det går att kombinera?

P1 – Absolut. Och det är väl som allt annat, allting är verktyg och så tar man det man gillar.

J – Ja, precis, det som passar för stunden så att säga.

P1 – Och särskilt då man kör kortare projekt, då tror jag man måste göra så, för det blir väldigt konstigt att ha en väldigt strikt metodik som där vi ska trycka in allting i

J – Mmm

P1 – Speciellt då på ett konsultföretag där du aldrig vet riktigt vem som kommer jobba med vad för att det är sån omsättning hela tiden med folk som kommer in efter ett uppdrag och så ska det passa alla och så ska alla vara nöjda, så man måste vara väldigt flexibel tror jag.

H – Nått speciellt som är extra viktigt skulle du säga just när man snackar om anpassning av metodval?

P1 – Vad är smart egentligen alltså? Man måste börja nånstans och så måste man hela tiden utvärdera

H – Ok, så utgå från enskilda komponenter?

P1 – Ja, kanske, eller framförallt vara flexibel liksom och... investerar man för mycket och säger att nu ska vi köra det här så är det lätt att du håller fast vid det fast det inte funkar, och

det är väl ok att köra typ scrum men man måste vara beredd att ta bort de element som inte fungerar för en och inte låsa in sig.

J – Och det är som sagt oftast alltså, ja det kommer inte från ledningen att nån säger att nu ska ni köra det här utan det är upp till gruppen här nu att...

P1 – Ja, det brukar vara så

H – Ja, att ni får en uppgift, så ska den lösas

P1 – Precis

J – Ganska självstyrande?

P1 – Än så länge har det varit det för mig i alla fall, jag tror inte jag skulle trivas i en organisation som inte var det.

J – Då har vi fått lite kött på benen...

H – Är det nåt du vill tillägga?

P1 – Nej, inte så där spontant.

H – Då tackar vi för oss!

Bilaga 3, Intervjuprotokoll, P2

P2 – Så, jag har ju egentligen sett två ställen som har försökt använda agila metoder, varav ett bra och ett mindre bra, och några andra ställen som inte har försökt använda några metoder alls, mer eller mindre. Det är lite situationsberoende, vad som är bra och vad som inte är bra. Så, om man säger så att göra saker med agila metoder gör inte saker absolut bra, bara för att man kommer med det här. Det som det står i den berömda texten, det är ingen silver bullet som gör allting bra automatiskt.

H – Nä...

P2 – Det viktigaste är att förstå, när det gäller agila metoder, är att det är en symbios av tre entiteter, när man tittar på Scrum egentligen. Det är ett utvecklingsteam en Scrum-master och en produktägare. Dom flesta kommer till eventuellt ett utvecklingsteam, kanske en Scrum-master som oftast inte fungerar som en Scrum-master, som oftast fungerar som en utvecklare med lite extra tid för det. Sen så har du produktägaren i den tredje rollen, som nästan alla misslyckas kapitalt med. Möjligtvis inte [Företag 1] där jag var ganska länge. Dom gjorde det rätt så bra. Men det är den svåraste rollen och det är däri som hela... det är däri som man lyckas eller om man inte lyckas, och det är däri som man faktiskt förstår riktigt det här egentligen ska gå ut på. Du förstår vad jag menar? Det är också däri utmaningen ligger. Jag vet inte. Ni får leda in mig på det ställe där ni vill ha mig för annars bara pratar jag löst så här.

H – Lite det som är tanken med agila metoder som jag förstått det är kundfokus. Och det är produktägaren som ska representera det, att allting ska riktas mot ett kundvärde på något sätt .

P2 – Mmm, så står det.

H – Så står det?

P2 – Så står det! Och det är också jävligt svårt för att det som ni borde ha hört är att det är jättesvårt att få kunder att jobba så här för att dom vill oftast inte vara med. Det är ett jävla motstånd för att få dom. På [Företag 1] försökt vi att tvinga in dom. Vi tvingade ganska bra in dom, till att få in dom. Dom var tvungna att lägga in ärenden i ärendehanteringssystemet. Dom var tvungna att kommunicera och förstå hur det skulle utvecklas iterativt.

Jag hade ett uppdrag på tre månader här borta på [Företag 2]. Dom jobbade mot interna kravställare, där beställarrollen fanns inte utan det fanns en produktägare som samtidigt också tänkte ut vilka features som skulle in. Och det funkar inte riktigt, att vara både produktdesigner... en produktägare är en planerare, en planerarroll och helt avhäng ju av att det är han som gör att det hela fungerar. Annars så blir det en form av storm i vattenglas.

Jag skulle vilja ta det som en form av behovstrappa. Dom flesta som försöker göra någonting med Scrum når upp till den första nivån. Den första nivån kan vi ungefär säga att det här är: vi klarar av att hålla daily möten med varandra. Då har man fan kommit rätt bra långt. Dom flesta klarar inte att hålla daily möte med varandra. Man skiter i det och så blir det varannan dag, nån gång i vecka och så håller man inte samma tid. Då är det det först liksom. Så lyckas man komma lite högre än så. Då lyckas man hålla release i jämna cykler eller intervaller.

Kommer man lite högre än så så klarar man av att hålla retrospektiv- och planeringsmöten. Kind of, sort of, liksom. Och kommer man lite högre än så, då... när du väl är så pass långt kommen i processen, då ligger egentligen ganska mycket utanför att teamet är självbestämmande och snarare på bordet utav produktägaren. Då börjar vi prata om att vi ska ha en prioriterad backlog. Vi ska ha koll mer än så här långt framför näsan vad som kommer att komma in och vad som kommer hända under en närmre tid. Vi ska kunna samla in kundkrav och sånt.

Då ligger det helt plötsligt på den här lite bortglömda entiteten, som att det kommer. Det är därför folk pissar på sig med Scrum, för dom glömmar bort den här killen som är halva grejen. Det är inte så svårt att komma till första nivån, det är inte så svårt att hålla möten en gång om dagen, titta varandra i ögonen och säga: jag gjorde lite grann igår. Men det är dom andra grejerna som kräver organisatoriska förändringar, och det är dit det är svårt att nå för att det är där du har en... där måste du in och ta nästa. Plockar du då produktägaren från en... en utvecklare som blir produktägare. Brukar bli ganska kasst för att dom ser väldigt sällan längre än implementationsdetaljer. Det blir oftast så.

Plockar du dom däremot från säljsidan. Då får du istället folk som inte förstår att saker och ting tar tid att utveckla och något i den stilen. Det finns en jävla massa Scrum-certifieringar. Något som jag tycker är ganska mycket trams. Fan vet hur många produktägar-certifieringar där faktiskt är. Med tanke på att han är halva symbiosen. Sen den tredje är kund-jäkeln. Men då ska man ha kommit så långt att man har en fungerande produktägare innan kunderna ens börjar påverka. Få bort kunderna ut utvecklingsteamet det är ju fenomenet Scrum-master som gör vad han ska, och få det andra på plats. Det fick dom aldrig på det här bolaget. Dom hade i och för sig andra problem.

J – Men, nu har du pratat mycket om Scrum. Har du erfarenhet av att jobba nånting med andra typer av agila utvecklingsmetoder?

P2 – Det finns väll bara en till som folk håller på med och det är Kanban.

J – Det var den som [P1] nämnde.

P2 – Ja, precis.

J – Har du varit med om att kombinera dom på nått sätt?

P2. Ja det är... Ja. Jo. Jag har varit med om diverse försök i det. Jag har inte riktigt förstått poängen i det om jag ska vara ärlig. Kanban ger ju... Problemet är det man drar ner mängden folk som får vara i varje kolumn för att tvinga folk att koda tillsammans med varandra. För att man tycker att man ska programmera för att det är så jävla bra, det har andra sagt. Det är inte riktigt min åsikt, att det är så jävla bra alltid. Det blir lite grann så här att om man nu försöker kombinera dom här två. Kanban jobbar ju inte med en iterativ utveckling, utan det är ett sätt... den är ju väldigt avspeglad utav det som den kommer ifrån, d.v.s. Toyota.

Att bilproduktionen skulle rulla på. Då har man hela tiden linjeorganisationstänk. Sen är det nån jävel som har läst om det här fantastiska och tänkt att: det här kan jag ju köra på mjukvaruutveckling också. Så tar man någonting som är en linjeutvecklingsmetodik varpå man kombinerar detta med Scrum som är en teknik för att göra... för att dela upp nånting i

kortare projekt. Sen tror man att dom här två ska lira tillsammans. Det får folk att konstruera alldeles egna metodiker istället för att köra ”by the book” eller vad man ska säga.

Om du kombinerar dom här två. Då lyckas man bygga en tungrodd modell utifrån saker som inte ska vara tungrodda. Scrum säger inte så jävla mycket mer än att du ska releasa var nånting-vecka, månad eller vad du nu bestämmer dig för, och att du ska ha planeringsmöte och att du ska ha en prioriterad backlogg och du ska ha det här. Kanban säger inte så mycket mer än att du får inte göra mer än så här många saker åt gången. Det är ungefär det den säger. Du ska inte ha mer än fem saker pågående på en gång. Savar det upp för mycket i den kolumnen så får vi hjälpas åt med grejorna. Sen brukar folk använda Kanban till att tvinga folk att programmera, storheten på det hela.

Men om du då kombinerar dom här två tillsammans med varandra, så tycker jag i alla fall i min åsikt att man skapar sig en processtyrd organisation, som går ifrån det som man egentligen pysslar med, d.v.s. utveckla grejor till att leta efter att optimera. Det hände ganska mycket på [Företag 1] att man ganska mycket optimerade utvecklingsmetodik, man kanske istället skulle kollat lite på vem kodar dåligt här, eller nånting i den stilen för att i slutänden är det ju det.

Det är lite en av min pet peeves med agil utveckling i allmänt är att det är väldigt väldigt lätt att ha ett långt möte om agil utveckling. Att sitta på ett retrospektiv och snacka en massa skit om... om du ska ha tre eller fyra kolumner, ene extra kolumn för analyser och tester och sånt här. Alla kan ha en åsikt och alla kan få lov att ha rätt. Så det går att ha hur många såna här retrospektiv som helst. Och det finns någon gräns där det där blir kontraproduktivt. Där du inte kan optimera din organisation längre utan... bara gör bot på dom sakerna som är utanför.

Det är också lite min erfarenhet av vad som hänt när du har... vad som hänt när du inte har en adaption av agil utvecklingsmetodik... utan ditt eget utvecklingsteam. När det gäller det så... Kanban säger inte så jävla mycket om resten av organisationen. Den ställer väldigt lite krav. Scrum ställer betydligt mycket mera krav. Den ställer krav på en produktägare. Den ställer krav på att folk ska förstå att vi releasar så här. Kanban säger inte ett dugg om vilken cykel du ska ha, eller var issue:arna kommer ifrån. Den säger bara: du ska inte göra mer än fem saker åt gången.

J – Den specificerar sig in på utvecklingsteamet då?

P2 – Ja, den är väldigt lokaliserad, och är det så att du bara vill ha en sån metodik och du inte kan påverka nånting i din organisation så kan du lika väl köra den som att köra... försöka ha en större metodik, som Scrum till exempel. Du förstår vad jag menar tror jag. Det är en smalare metodik. Sen vet jag inte om det finns så många andra metodiker som folk försöker köra. Det finns det säkert, som inte jag känner till. Extreme programming kanske kan nämnas liksom...

H – Den verkar ju populär men den är ju... extrem

P2 – Asså, ja. Ja? Vad säger dom egentligen? Du ska koda i par?

H – Ja, där är några saker. Det är så olika beroende på vem man snackar med. Vissa säger att där är ingenting, där är bara dom här sakerna. Koda i par och så är där andra

saker. Andra säger att här har vi den exakta sekventiella kedjan. Det är strikt och så. Vissa säger där är ingenting alls det är bara ad hoc liksom. Ingen tanke alls.

P2 – Jag tror att det har inte fått en formalisering på det sättet som dom andra metodikerna har. Men som sagt jag uppfattar tydligt att där finns en gräns. Framförallt finns där en gräns för när det jobbet du gör för att förbättra din egen arbetsmetodik, det kan aldrig nå utanför din organisation för dom vägrar lira tillsammans med alla andra. Dom vägrar att förändra sina processer så att dom ska kunna lira bättre med dig. Då börjar dom istället mikrooptimera sina egna små resurser som du har inom din egna lilla organisation. Det är liksom fem grabbar som sitter och kodar, så ska du försöka optimera din egna utvecklingsprocess därifrån.

J – Så du har aldrig upplevt en bra kommunikation. Alltså att man sprider den här metodiken till organisationen.

P2 – Jag har aldrig sett att dom varit speciellt intresserade. Det närmsta man kan få dom till det är daily stand up. Och jag har aldrig sett dom göra det regelbundet. Du kan eventuellt få dom till att gå med på dina retrospek... eller inte dina retrospektiv men dina sprint reviews. Typ. Men dom kommer inte regelbundet. I helvete heller! Det är alltid då mycket viktigare att göra andra grejor. Jag vet inte riktigt men, det blir... Det har alltid varit dödfött där. Dom kommer och så tittar dom på din review liksom. Sen så går dom ut igen, och så får du ingen direkt feedback.

Dom gånger dom också säger att vi kanske också vill börja lite med agila utvecklingsmetodiker så går det aldrig. Det verkar inte finns något intresse för det. Trots att jag kan tycka att om du har ett gäng säljare så borde dom definitivt vara intresserade av vad andra gör, men kulturen finns inte där och då kör man liksom solo här. Sen har jag ju sett organisationer där det inte finns någon utvecklingsmetodik alls. Det är väl inte alltid bättre, men inte alltid sämre heller. Att köra agila utvecklingsmetoder... Jag tror inte på att det är en lösning för allting. Jag har för övrigt varit Scrum-master också, i ett par månader ska jag säga. Så jag var det där bort på [Företag 2]. Jag försökte styra upp lite där.

J – Vi har lite grundläggande frågor. Lite strukturerade.

H – Du har täckt upp lite, men vi kan köra åtminstone bakgrund. Vad har du för utbildning?

P2 – Jag är ju systemvetare. Mycket atypisk sådan, men det är jag.

H – Vad är en typisk systemvetare?

P2 – Jag tror inte det är nån som sitter hemma och kodar i assembler.

H – Nä, i och för sig...

P2 – Jag borde nog varit datalog men det är jag inte. Jag har läst tre åt systemvetenskap i Växjö. Så snabbt som möjligt ut, in. Det sista året arbetade jag samtidigt också, jag tyckte det var för lätt.

J – Det är schyst att få in foten direkt så.

P2 – Ja, det var inte så jävla svårt. Jag gick in på [Företag 3]s reception och tyckte jag kunde bli timmanställd. Det tyckte dom också. Den första timmanställda på [Företag 3]. När det begav sig.

H – Vad har du för tjänst?

P2 – Vad sa du?

H – Vad har du för tjänst här?

P2 – Vad jag har för tjänst? Konsult står det på visitkortet så det är väll det.

J – Du hade varit Scrum-master förut också sa du?

P2 – Ja, men det är ju i ledet av ett uppdrag.

J – Ja, i och för sig, men du har haft den erfarenheten i alla fall?

P2 – Ja ja, visst.

H – Erfarenhet vad gäller projekt har du täckt ganska ingående.

P2 – Ja, det tror jag i alla fall. Jag har inte sagt så mycket om projekt som inte varit agila men det behöver ni... ja. Jag tror ni har hygglig bra koll på det.

H – Och du har varit Scrum-master samtidigt som utvecklare?

P2 – Ja, jag... Det är sällan som Scrum-mastern får vara dedikerad Scrum-master. Vi hade en dedikerad Scrum-master på [Företag 1]. Det är väldigt trevligt att ha en dedikerad Scrum-master. Det är ett städjobb, Scrum-master, vilket folk inte förstår. Det är definitivt ingen prestige-grej. Det är ett jävla piss-jobb. Det handlar om att du ska kalla in till dom här mötena och att du ska se till att saker och ting som kommer att pinka på teamets vardag, typ att kundskrälet inte levererar tillräckliga spec:ar och sånt där. Det ska du söka upp idet dagliga jobbet. Du ska agera sköld, så att det inte kommer en massa säljare och petar i koden och sånt där. Så, det är en sköld och en säkerhets...

Det finns en jävligt trevlig bloggartikel, "Five Whys" finns den på Roy Osherove om hierarkier och om kaos i Scrum-teamet. Den är värd att läsa. Men, nej, som sagt, jag har aldrig fått vara agil ledare på heltid. Jag tror inte jag skulle vilja vara det heller för jag är inte sånn. Jag gillar ju att koda. Det tycker jag, det ska man göra. Det är inte riktigt min grej. Då skulle jag i så fall gå mot nått annat som jag inte vill göra.

H – Vi undrar också på vilket sätt som projektets utformning har påverkat metodanvändning och metodval.

P2 – Ja, asså grejen är ju det att metodvalen är ju oftast. Oftast så kommer det med någon som tycker att det är jävligt fräckt, och Scrum har varit det fräckaste nu, bra länge, och innan dess så var det andra saker som var fräckt. Nu är det väl lite grann så här att Scrum är "go-to"-valet. Om du över huvud taget ska ha någon metodik, så är det det man väljer rakt upp och ner. Vill man vara lite "on-the-edge" så kör man Kanban, men Scrum är nästan på liksom.

Antingen så har vi ingen metodik alls, och då styrs vi av några okända faktorer så som när kunden känner att det är tid för en release eller nått i den stilen. Mycket vaga, eller projektorganisationer som har mycket dokumentation och folk som inte gör något annat liksom. Men annars om man har nån organisation så brukar folk ha att man ska gå emot Scrum. Är man lite tuffare så försöker man göra Kanban. Ska man försöka vara extra supertuff så försöker man göra en kombination, vilket jag tycker är dumt.

H – Tolkar jag rätt att tolkning och översättning till praktik har varit mycket mötesbaserat, långa, och att alla ska komma fram och så här, utdragen process?

P2 – Ja, asså, ja. Vad ska man säga? Du menar att översätta den här teorin till praktik?

H – Ja, precis.

J – Är det nån som tolkar och säger så här ska vi jobba, eller får alla vara med?

P2 – Alla i gruppen brukar få vara med, ibland till absurdum, att det tar över, att man tror man ska kunna lösa allting med metodiker och det gör man inte. Grejen är det att agila saker gör saker bättre i team som redan är ganska bra. Har du nått annat som regnar på paradisen, om du har en chef som är dum i huvudet eller något, eller du har en produkt som är superkass. Du gör väl inte så mycket bättre med att klistra Scrum ovan på, och tro att allting ska bli bra.

Du kan heller inte förvandla ett gäng kassa kodare till bra kodare genom att tvinga dom att hålla retrospektiv en gång var tredje vecka. Det funkar inte så. Har du dåliga förutsättningar, framförallt är det dåligt folk egentligen, så blir det inte bra. Ska man utveckla mjukvara så får man ha folk som kan utveckla mjukvara, annars blir det så att du försöker lösa saker vi metodikerna, vilket alla tycker känns jättebra för då behöver vi aldrig prata om att vissa inte förstår vad rekursion är för nånting.

Då kan alla sitta på retrospektiven och prata med varandra jättelänge, om att det kanske är bättre om vi alla siktar tydligare på att hålla... på att vi måste vara tydliga med att börja varje morgon med att säga att jag behöver hjälp med nånting. I själva verket står det en elefant i rummet och skriker: två av oss här är skitkassa på att koda. Då blir det inte bra i slutändan. Man kan inte dölja en kvalitetsbrist i folk som jobbar där, eller att man får kassa spec:ar från folk. Dom beställer saker och ting som vi som team borde ha auktoritet att kasta ut och säga det här kan du inte göra för du vet inte ens vad du vill ha.

Om du fortsätter att ta in dom och har ett undermåligt kvalificerat team så spelar det ingen roll vilka agila metoder du väljer. Däremot kan du få ett team av väl inkörda kodare till att koda bättre med varandra om du har en metodik du kan följa. Det är det som är jävligt viktigt. Dom här metodikerna får ett genomslag genom att... det är ett sätt att försöka göra saker bättre som adresserar någonting som kanske är lite högre upp på skalan över vad folk egentligen skulle behöva adressera. Förstår du vad jag menar med det här, att man försöker gärna lösa problem som kanske inte ska lösas med metodikmässiga skäl?

Försöker göra dom bättre på det sättet, och det är en fara med dom här grejorna tycker jag. Det är just det att andra kan känna sig rätt så hemma, och det finns inga rätt eller fel om det vilket gör att man kan prata hur mycket som helst om det. Man löser inte ett grundproblem.

H – Jag tänkte på det här med väldigt specifika delar, hur har det sett ut?

P2 – Jag tror det följer den här jävla behovshierarkin. Jag har aldrig sett dom gå i nån annan ordning än så. Jag tänkte jag skulle blogga om det där någon gång, det vore så fint med Maslows behovshierarki och Scrum. Det går alltid i den ordningen och det tar nästan alltid stop nån stans därvid. Kommer man över dom här då är man... Det är lättast att hålla ett möte varje dag 09:15 eller vad man nu ska ha. Men, även det där är rätt svårt ibland att få folk att komma i tid på morgonen och respektera att vi ska stå upp på våra daily standup. Det kan vara väldigt motigt, även det som man kan tycka är en enkel banal sak.

J – Väljer man bort dom här andra delarna eller blir det bara att man inte hinner med dom?

P2 – Det brukar alltid se ut så att man gör en ansatts. Vad folk oftast gör är att då tar dom dit någon form av konsult som ska föra fram det fantastiska med agila utvecklingsmetodiker. Så kommer han, och så sätter man upp dom här grundpelarna. Vi ska ha dom här mötena. Nu ska vi ha en tavla på den här väggen och så ska vi ha lappar på den så vi kan se vem som jobbar med vad. Sen ska vi ha ett retrospektiv, ett planeringsmöte och du är produktägare. Så, nu kör vi grabbar. Sen börjar man med det, och så tycker alla det känns rätt så bra. Fan, jag vet vad alla gör, och jag vet vad som finns i sprinten. Jag vet om allt går åt helvete. Så kan man göra lite diagram och så är alla nöjda.

Sen brukar nyhetens behag försvinna. Dom här metodikerna börjar slira eftersom dom inte har växt fram i organisationen av sig själv på något sätt. Precis på det sättet var det på det uppdraget där jag var Scrum-master. Jag fick veta att jag skulle på något sätt lyfta upp det här teamet som hade slirat. Det var inte så jävla nog. Vi kunde ha daily standup varannan dag. Vi behövde inte alltid ha retrospektiv, och vi brukar ha vår estimering av ärenden samtidigt som vi har vår planering. Det är väl inte så farligt liksom? Ha aldrig det! Det är förfärligt. Det är hemskt. Det pissar på precis allting. Sen kommer du dit och försöker dra upp dom här gränserna. Jag vet inte om dom har lyckats hålla sina gränser nu när jag inte är där längre, men... nu tappade jag tråden här...

J – Det är svårt för dom att uppnå allting...

P2 – Ja, och hålla uppe allt det här. Det slirar gärna när man känner att det inte funkar. I synnerhet så slirar det när du inte känner att du får gehör. När man själv känner att vi når upp till en nivå. Vi har våra retrospektiv, våra planeringsmöten även när vi har jättekasst underlag från produktägaren så har vi vår daily standup. Om du inte får gehör uppifrån då, att produktägaren kommer aldrig med planerade backlogs, utan han hittar på vad han ska ha på planeringsmötet. Jättevänligt, supervanligt, för folk förstår inte vad produktägaren ska göra. Då reflekteras det ganska snabbt tillbaka på utvecklingsteamet att vad fan ska vi hålla på här. Det enda vi kan göra på våra daily standups är att säga vad vi gör och tror vi ska göra imorgon men ingen riktigt vet eftersom vi aldrig får en planerad backlog av en annan Pelle. Då kommer man inte längre. Det känns liksom hopplöst och så vandrar man ner igen i icke-metodik-land liksom.

H – Kör man då någon utvärdering eller uppföljning av metoden?

P2 – Ja, gör man det? Nä, det tror jag inte man gör. Det finns ett dogm som säger att vattenfallsmetoder det är dåligt, det ska vi inte göra. Det måste vi göra ibland för att vår kund eller organisation såhär. Vi vill köra Scrum. Varför det? För att Scrum är bra. Sen så har man inte egentligen utrett det där. Det kommer: vi vill köra detta. Det är bara att titta på arbetsannonser och sånt där. Det står vi ska jobba med agila metoder och sånt där. Och varför? För det gör alla andra. Det hade kunnat var vad fan som helst. Jag tror inte det är någon verklighet, dom... inte vad det är för nånting egentligen, utan dom har...

J – Dom har läst på tapeten...

P2 – Ja men visst

H – Det gäller ju för alla förändringsarbeten. Om man inte har motivation innan eller uppföljning sen så...

P2 – Men... när det här började på [Företag 1], som lyckades väldigt väl med sin Scrumimplementation. Där började vi underifrån. Du började med att nån hade hört om Scrum i ett ganska tidigt skede och vi ska ha morgonmöten. Sen försökte man implementera det här. Vi hade morgonmöten... Jag var med från början i princip vilket inte [P1] var. Vi hade morgonmöten varannan kanske var tredje dag. Det var bra nog. Sen så växte behovet succesivt istället för att komma in med... Annars blir det så att säg du har fem kodare som sitter så och kodar lite grann, och sen ibland kommer det en chef och skriker på dom lite grann vad dom ska göra.

Sen försöker du införa Scrum på detta och du tar den här cookie-cutter-approachen: såhär gör vi för det står i boken. Det är så alla gör. Då har du försökt införa nånting som inte fötts underifrån. Det har inte fötts ur en acceptans ifrån dom som faktiskt införde det. Det är ingen som förstår och har sett den gradvisa förbättringen. Det är det som gör att hela businessen med att ta in en agil konsult är ganska konstig. Du kommer inte ha den förståelsen som du kommer få utav att börja med ett daily standup varje morgon, eller varannan morgon om det känns för tungt. Det gör ingenting att vi gör fel enligt boken för vi bygger egen intern förståelse, och då funkar det.

Men i och med att Scrum när vi började på [Företag 1] inte var dogm på det sättet. Vi var ju jättefärska på det. Det var ingen annan som jobbade såhär, och det var ingen som visste riktigt hur man skulle göra. Då kunde man ta lite grann i taget. Nu är de så att om du säger att vi har morgonmöte varannan dag men inte mer då kommer alla titta på dig som att du gör fel. Nä, du gör kanske inte fel. Du gör kanske det som är moget i din organisation just nu. Dit vi kan nå. Det är lite sådär svårt. Så som det är just nu i alla fall

J – Jag tänkte lite på det här. Den interna förståelsen. Är det gruppvis eller finns det nån kommunikation om det är två utvecklingsteam till exempel eller fler? Pratar man något tillsammans...

P2 – Jag har egentligen inte varit med om mer än ett utvecklingsteam ska jag väl säga. Vi hade två teams på [Företag 1] ett tag, som jobbade parallellt med varandra, men det var fortfarande samma produkt. Det var bara en uppdelning för att det var lättare att planera. Sen ditchade vi det för att det var jobbigare att planera. Det blev mer konstigt än vad det var bra. Jag har inte varit med i en organisation där det varit ett större utvecklingsteam som försökte köra Scrum. Det finns en falloff där. Jag tror personligen att den ligger på sådär fem-sex personer. Mer än så kan du inte ha i ett Scrum-team. Det pissar på sig.

J – Ja, eftersom det är interaktionen hela tiden...

P2 – Ja, men det är ingen som bryr sig. Om du sitter fem personer runt ett bord så kan du precis på vad alla dom gör. Du kanske vet att jag vill hjälpa han. Är du femton personer så skiter du fullständigt i vad tio av dom gör. I slutändan skiter du i vad alla gör utom dig. Du blir aldrig personligt accountable för vad du gör för det finns ingen som håller koll på vad du gör. Det här personliga ansvaret för vad du gör, den försvinner ”out the window” vid mer än sex personer i Scrum-team.

H – Det bygger på att folk har översikt?

P2 – Det är ingen som bryr sig längre. Då är det bara att säga: Jag mhmhmhmm och idag tänker jag göra mhmhm det här. Sen så sitter man och surfar reddit. Det går skitbra. Det händer för man blir inte accountable. Är du upp till fem-sex så går det att hålla koll på varandra. Det gör man. Det är en personlig relation. Det är inte bara en i gänget.

J – Är det också så att man hjälper varandra då? Om man stöter på problem så kan man ta hjälp från gruppen av den kunskap som finns?

P2 – Ja, men har du tio eller så där så får du inte den. Det är observer effect egentligen. Om ni har hört nån gång om det i socialpsykologi. Läser ni fortfarande det? Det fick jag läsa.

J – Nej.

P2 – Tio poäng socialpsykologi. Tråkig bok. The observer effect är att om du inte observerar nånting, och inte kan bli personligt accountable för det så är du mindre benägen att ingripa. Om du ser nån som ligger på gatan avsvimnad så hjälper man till, men chansen sjunker desto fler man är i crowden. Om han ligger utanför ditt hus

J – Hur många som har sett honom.

P2 – Precis. Andra har sett honom och... Det funkar precis likadant i ett Scrum-team. Be om hjälp av femton personer, det är ingen som bryr sig. Men om du går fram till någon och säger kan du hjälpa mig med detta så får du det. Så hela metodiken blir ganska kass och jag tror att tröskeln ligger vid sådär fem-sex.

J – Det låter... Det är förståeligt.

P2 – Det är många som inte förstår det där.

J – Nä, det finns alltid olika åsikter liksom.**H – Jag läste någon artikel där dom försökte iföra det i projekt med mer än 50 användare.**

P2 – Coolt. Det gick inte, va?

H – Inte helt bra!

P2 – Det kan man också fråga, varför man försöker? För det är inte tänkt till att ha det. Du kan inte applicera Scrum på allting bara för att du hittat den här... Bara för att du hittat en ny hammare är inte allting en spik. Folk verkar tro att det är så. Man har fått det här nya fina verktyget och det har ju funkade bra för nån annan. Bara för att det funkar bra för nån annan, betyder inte att det funkar jävligt bra för dig. Det är inte alls säkert att du ska ha en agil utvecklingsmetodik överhuvudtaget. Det är inte säkert att du ska ha nån utvecklingsmetodik alls.

Det finns organisationer som fungerar alldeles utmärkt utan utvecklingsmetodiker. Har du tillräckligt bra utvecklare så behöver du ingen jävla utvecklingsmetodik. Om du bara har ninjor som sitter och kodar som är ap-bra på det dom gör, och som brinner för det och känner att dom har understöd och förstår vilka spec:ar dom har för att göra saker och ting. Du behöver ingen jävla Scrum för att hålla koll på dom. Dom kommer göra bra jobb ändå. Man försöker lösa ett problem, och det är sällan jag faktiskt ser att problemet är inom utvecklingsteamet. När problemet är inom utvecklingsteamet då är det ändå nånting som Scrum inte löser.

Om det är utvecklingsteamet så brukar det bero på att någon av dom är dålig. Det är inte så jävla behagligt ämne att snacka om det. Det löser du inte genom att försöka tvinga andra att hjälpa honom. Då har du troligen redan gjort dig trött på den jäveln som inte lär sig. Det är lite så. Försöker man lösa det med Scrum så... det funkar liksom inte sådär jävla bra. Däremot om man lyckas få Scrum att funka med Produktägare och sånt där, så att du har en specad backlog och grejor som kommer in. Då börjar man få benefits av hela teamet, men du kan inte lösa defekter inom teamet, att nån är kass, det håller inte.

H – Så det är lite det som är kontentan? Rätt situationer, och inse vad det är?

P2 – Ja, de tycker jag. Inse vad du försöker lösa först. Vad är problemet egentligen? Frågar du ett utvecklingsteam då vad som är problemet. Det är oftast vi vet inte riktigt vad som ska göras. Jaha. Oftast är det nån form av kommunikationsproblem med ledningen så då kan det lösas i form av att vi får uppgifterna prioriterade rätt till oss i lagom doser och med bra spec:ar. Allting av det här ligger ju utanför utvecklingsteamet. Utvecklingsteamet kommer heller inte att säga att utvecklingsteamet är kasst för att dom som jobbar där inte är så jävla bra. Även om det tyvärr ofta är anledningen. Ni ska se så mycket kod det finns om är dåligt. Bli inte såna. Bli inte såna som skriver dålig kod.

Det är mycket sånt det handlar om. Man får identifiera det problem man ska lösa. Man ska inse att du kan inte bara komma med en metodik bara för att nån ball kille har kommit på att göra såhär i Kalifornien, och han är nu jätterik. Då funkar kanske precis lika bra i Staffanstorp. Det gör det inte nödvändigtvis. Det som kommer därifrån var nånting som dom hade utvecklat framåt. Dom hade evolutionärt kommit fram till var schysst som fan. Sen har dom i efterhand skrivit ner en pärm om saker som blev skitbra, och sen försöker dom dela med sig av det här som fallit från himlen. Vi gjorde såhär och så blev det skitbra.

Jag är övertygad om att hade du haft samma team som fick sätt sig ner och komma fram... sen låter du dom bestämma eller självbestämma, så kommer dom komma fram till en metodik som kanske helt motsvarar den hatade vattenfallsmodellen, och som dom kommer tycka är skitbra. Sen kan dom klistra på det här med nått nytt ballt namn och sen kan dom sälja det till en massa andra människor. Det kommer funka bra för vissa men inte alla, men det kommer funka skitbra för dom som tog fram modellen för det är deras modell. Eftersom dom tog sig tiden att lufta fram det från ingenting så är det garanterat perfekt för dom.

Det är det som gör det konstigt att komma med metodiker till folk, och sen säga att om du gör så här så blir det jätte jättebra. Varför det? För att det blev det för dom. Är dom ditt utvecklingsteam? Är dom din organisation? Dom lär ju inte vara det. Då blir det konstigt. Det är därför vissa lyckas. Det kommer komma en backlash på det här med agila metoder, därför att alla har provat det och det blir inte bättre. Det är klart som fan det inte blir bättre. Du har inte tänkt ut det här. Daily standups kanske inte var för din organisation. Din organisation kanske ska ha strikta release-cykler där det ramlar ner som i ett vattenfall. Det kanske är skitbra för dig.

Det är en tidsfråga innan det kommer komma nånting tillbaka på det där. Sen är det nån annan som kommer sitta och tycka vi gjorde på nått annat sätt och sen kommer du ha en ny våg där alla tror att det här... för det här är inte det första. Vi har haft vattenfallsmodellen, sen hade vi

software engineering-tänket, där du skulle låtsas att alla som skrev kod var ingenjörer. Gör på ett ingenjörsmässigt sätt så blir det bättre. Det gick sådär. Sen så hade vi Rational Unified Process. Jättefint, man köpte dyra verktyg och sånt där. Så skulle man rita grafer och så såg allting bra ut ett tag. Jättheadaption och sen blev det kass med det också. Scrum kanske redan har nått sin peak. Ramlar den ner så kommer det nånting nytt där.

För dom som adopterade metodiken från början och gjorde alla på riktigt så är det här fortfarande ”the shit”. Därför att den är född utav dom själva.

H – Så det är kanske den tolkningen man ska göra istället att dom utvärderade själva sina behov utvecklade en egen metod och den funkade bra för dom.

P2 – Ja, absolut. Annars har du tagit nån annans metod. Nån annan som har tänkt ut nånting.

J – Man kan ju kanske ta delar från den.

P2 – Ja, absolut, och det ska du. Man ska vara inspirerade kanske.

H – Men inte ta den rakt av.

P2 – Nä, men det gör ju folk. Det är ju det som händer hela tiden när man tar in en agile-konsult. Jättekonstigt. Eller om man tar certifiering i Scrum. Jättekonstigt. Vad har du gjort? Du har certifierat dig i en metodik som någon annan har utvecklat för att vara bra på hans företag. Det är skitbra om du vill jobba på hans företag. Det är jätteskumt. Jag förstår att man vill göra det. Man tror att man ska få en silver bullet, man köper den i en låda. Ta den på så kommer alla springa hand i hand över ängarna, och så når man en bit på vägen. Det är därför som det kanske inte alltid är ett misslyckande att säga att vi har våra daily standups ibland.

Du har kanske utvecklat en organisation som når till ett läge att vi inser att det är ingen som blir glad av daily standup. Vi har weekly standup därför att hälften av våra utvecklingsmedlemmar dom jobbar halvtid hemifrån, eller är på affärsresor i Korea. Vad fan vet jag?

J – Man anpassar metoden.

P2 – Ja, precis. Man låter den växa fram underifrån. Man inser att det kommer inte färdigt i en låda och kommer göra allting bra. Man inser: vad kan jag fixa med agil utvecklingsmetodik? Ska jag bara försöka kör Scrum som det är, så ligger minst hälften på en man och det är produktägaren, som är superduper viktig och övergår allting annat. Om jag får ha ett team som skiter i sina retrospektiv och sina daily standup men har kick-ass produktägare som förstår teamet och ger dom en bra prioriterad backlog, eller tvärt om ett team som har alla retrospektiv och gör allt som dom ska, hjälper varandra till höger och vänster men som har en produktägare som petar i naveln.

Vad väljer jag? Jag tar den med bra produktägare. Det är klockrent. Ska man börja med förändringsprocess så ska man börja där, men det är mycket svårare ha tag på nån som troligen är lite såhär kostym-Pelle. Dom brukar vara det. Och så ska du få han att jobba efter en metodik. Du har säljar-typen. Dom är mycket mer intresserade av att vara ute på kundmöten och dricka vin. Det är mycket svårare. Det är inte dom som agile-rörelsen riktar in sig på att fånga. Man hör evangelisterna är ute på utvecklarkonferenser och talar till dom

redan troende om att det är bra att ha morgonmöten. Det kanske inte är där man borde vara och predika.

Det kanske är där den lågt hängande frukten finns, men det är inte den som är mognast. Lite så är det.

H – Jag tror vi fått med våra frågor.

P2 – Ja jag brukar prata mycket.

J – Ja.

H – Det är intressant

P2 – Jag har sett rätt mycket faktiskt. Det blir ju så. Jag har varit igång rätt länge. Jag har jobbat sedan 2001, gick jag ut. Så det är ett tag. Det har varit väldigt annorlunda ställen. Vi hade ju allt från ingen utvecklingsmetodik till... Just nu där jag är nu har vi ingen. På [Företag4]s IT-avdelning. Dom har ingen metodik alls. Nästan inga möten heller. Å andra sidan har dom principen den som skriker högst får vad han vill ha. Det varuhuset som brinner upp mest just nu får hjälp. Så funkar det.

J – Har dom det där strukturerat ändå?

P2 – Nej. Nu funkar liksom inte importen till [Företag 5] längre. Per fixar du det? Jag är inte säker på att dom ska ha en utvecklingsmetodik. Jag frågade dom: har ni nån metodik? Nej. Lägg inte ner för mycket tid på saker för det kommer nått nytt. Dom har teammedlemmar som inte är ett riktigt team heller för alla har distinkta områden. Nån sitter i sharepoint, nån sitter i .net, nån sitter i biztalk... Har du något direkt behov av samhörighet och samkommunikation? Ja, kanske ibland, men oftast inte.

H – Dom tycker dom kan fixa det informellt?

P2 – Ja, och jag är inte säker på att dom ha fel. Jag tror snarare att dom har rätt. Å andra sidan jobbade jag... 2002-2006 som datorspelsutvecklare. Jag gjorde spel till Xbox och PS2, och gamecuben ett tag. Där var tio utvecklare. Då skulle man kunna tänka sig att man hade Scrum-team. Nu är det här före Scrum i och för sig, för att det här är så länge sedan. Vad hade vi där istället? Vi hade i princip ingenting. Vi hade ett projekt som bara löpte på, mot ett releasedatum som ingen riktigt visste när det var. Det var roligt och det var okej. Jag var inte så gammal så jag kunde arbeta till kl 23:00 sex dagar i veckan

H – När släpps spel någonsin på utsatt tid?

P2 – Nja, det släpptes ju när det släpptes. Ja, jo. Det finns spel. Men där var ingenting. Vi hade en projektledare ett tag, men han gillade vi inte, så han sparkade vi sen. Vi tyckte han var så in på micken. Han förstod inte. Där var en annan kultur. När det gäller datorspel ska du vara en jävla superprogrammerare och är du inte det så är du inte värd nånting. Vi har en sån kultur i den här branschen, som ibland kan saknas här. Här är snarare så att här accepterar man vilka jävla idioter som helst. Det finns så mycket horribel kod som är skriven av folk.

Det är lite olika kultur. Där hade nog behövts nånting. Samtidigt är det frågan om det blivit bättre för där fanns folk som inte tyckte om varandra. Sånt blir ju aldrig bättre. Har du två folk som ska jobba tillsammans som inte gillar varandra så blir det skit. Ska du försöka ha en

Scrum-team som ber om hjälp från varandra eller orkar lyssna på varandra på morgonmötet, när Kalle och Lasse tycker den andre är dum i huvet.

H – Att tvinga fram mer kommunikation där gör det ju inte bättre.

P2 – Nej, dom vägrade prata med varandra. Dom jobbade på olika plattformar. Kunde jag sitta emellan och skriva kod som skulle användas av båda. Så funkar det ibland bara. Ah, det är bra. Ska vi...

J – Ja, jag tror vi fått med det mesta vi efterfrågade, så det är lugnt.

H – Tack så mycket

P2 – Varsågoda

[Vi tackade för oss och sa hejdå]

Bilaga 4, Intervjuprotokoll, P3

P3 – Är det första intervjun ni gör?

H – Det är tredje, vi gjorde två igår

P3 – Det är bra, då är ni rutinerade.

H – Ja, vad har du för utbildning?

P3 – Jag är civilingenjör i informationsteknologi från Uppsala.

H – Ok, och din tjänst här?

P3 – Platschef men jag är också projektledare så det är relevant.

H – Ok, vad har du för erfarenhet av agila projekt?

P3 – Vi jobbar med agila projekt här på [Företaget 1] och vi har även de senaste fem åren, när jag har jobbat med projektledning har det varit mer eller mindre agilt skulle jag säga.

H – Så det är fem år?

P3 – Ja, men inte här, men alltså på olika ställen, men de senaste fem åren skulle jag säga att man har blivit mer och mer agil i sitt projekt tänkande, sen kanske det inte har varit några renodlade agila projekt någonsin, det kanske inte kommer hända heller men det tänket har i alla fall funnits där under den tiden.

H – Vad har varit din huvudsakliga projektroll, har det varit projektledare eller?

P3 – Mmm

H – Sen undrar vi lite, hur projektets utformning påverkar metodanvändning och metodval?

P3 – Hur tänker du då?

H – Alltså, från projekt till projekt beroende på kunder? Vad man ska prestera mer, hur det påverkar ert metodval eller metodanvändning.

P3 – Vi väljer alltid samma metod för våra projekt är mer eller mindre lika, och så har det nog varit även på tidigare företag där jag jobbat, man har alltså en inriktning, man gör inte så särskilt olika typer av projekt utan det är mjukvaruutvecklingsprojekt och det är en kund som beställer nånting, sen vad det nånting är, det kanske varierar men metodvalet faller sig ganska naturligt eftersom det är mjukvara vi pratar om.

J – Vad har ni kört med för metoder då? Eller vilken metod?

P3 – Om man tittar på utvecklingsmetod egentligen, jag vill ändå skilja på det, utvecklingsmetod och projektledningsmetod, många gånger så är ju utvecklingsmetoden kanske scruminfluerad medan då scrum är i mina ögon en blandning av en projektmetod och en utvecklingsmetod lite grand i samma låda liksom. Men om man tittar på rena projektmetoder då, så är det ju vanligt att använda PROPS eller Prince2 eller PPTS och kombinerat då med Scrum eller...

J – För att passa då till mjukvaruutvecklingen främst?

P3 – Mmm, ja och det här andra laget då kanske jag ser att man behöver för den rena projektuppföljningen alltså om man glömmer bort att det är mjukvara utan bara det här med det administrativa att man, ja men man behöver rapportera sin budgetuppföljning tex. till någon chef eller styrelse eller styrgrupper eller vad det nu är, det behöver man ju alltid göra oavsett om man gör ett agilt projekt eller inte så finns det alltid nån där som vill veta vad vi har för ekonomi i detta

J – Mmm

P3 – och det är alltid någon som vill veta hur långt har vi kommit, så de frågorna finns alltid där även om man säger vi jobbar agilt, vi tar det som det kommer, men det funkar inte riktigt så i praktiken, så därför vill man gärna ha liksom några tydliga milstolpar då och det kommer ofta från projektmetodiken då, scrum är ganska tunt egentligen, det är nästan ett tankesätt då.

J – Ok, den (Scrum) har ju också vissa milstolpar att följa den också, som typ Daily Scrum...

P3 – Precis, men det är inte så strikt så här liksom go live eller... som vi lever ju också i en verklighet att vi måste offerera projekt... om man har en egen produktutveckling internt, då är man ju sin egen kund, då slipper man det där med avtal och kontrakt och olika leverantörer ska offereras samma lösningar och så, eftersom vi måste göra det så måste vi alltid göra en förstudie. Förstudie är ett vattenfalls-tänk, det existerar ju liksom inte i den agila världen men vi måste göra det för vi måste på något sätt ha en specifikation i vårt kontrakt, vad är det vi ska göra för nånting och vad kommer det att kosta?

Det låter jättefint på pappret med att kunna säga att man ska bygga på förtroende och bla bla, det funkar inte riktigt så, ännu i alla fall i den riktiga världen utan de vill ha ett pris och så vill de kunna gå till en annan leverantör och se: ok, vi vill bygga detta, vad kostar det när ni bygger det? Så den typen av milstolpar liksom ok, nu kör vi igång implementationsfasen till exempel. Och nu har vi accepterat produkten, nu går vi in i någon form av support och maintenance-fas, alltså dem saknar liksom...

H – Ok, tänkte på det här alltså det att ni tex använder Scrum, har det kommit från utvecklarna själva eller har det kommit uppifrån?

P3 – Det är nog en blandning, just på [Företaget 1] så är det... vi har... det är positivt från alla håll tror jag.

J – Har du varit med om någon kombinerad av olika agila metoder? Om ni kör Scrum, så väljer ni in någonting annat, en annan metod också, försöker kombinera dem?

P3 – Skulle säga att vi inte... alltså Scrum är väl... vi använder nog Scrum kombinerat med egna erfarenheter och plockat russin ur kakan och lite det vi själva känner för. Så skulle jag uttrycka det.

J – Ok, så ni kombinerar efter egen... det ni behövde så att säga.

P3 – Ja.

J – Ok

P3 – Så vi försöker ha morgonmöten och vi försöker jobba i sprintar och vi har sprintdemo retroperspektiv, så alltså många av dem här nyckelbegreppen inom scrum har man ju plockat ur.

H – Hur har det gått till när ni har plockat ut russin ur kakan så att säga och kommit fram till ert arbetssätt, hur har det gått till?

P3 – Jag skulle säga att det är en eller två eller några erfarna personer som har ett liksom, under lång tid, säg en tio års period där erfarna projektledare har ett arbetssätt som dem tycker att det här funkar av erfarenhet, och då har vi på nått sätt tagit best practice från dem, snackat du tycker så, du tycker så men 90% är lika, ok de andra 10% vad gör vi med det? Vi plockar Det från dig och Det från dig, ja det är vår metod liksom. Försöka ha det så praktiskt som möjligt och sen så kör man och kommer man på på vägen att det här funkar inte, vi borde göra så här istället liksom, så ändrar vi det, vi är ett ganska informellt företag så, vi är inte strikta och har inte liksom hur mycket dokumenterade processer som helst utan vi tycker det är viktigare med att folk verkligen jobbar likadant i det dagliga jobbet snarare än att vi har en hög med fina beskrivningar liksom som ändå ingen följer.

J– Så ni har en bredare kommunikation genom hela företaget istället för sett liksom i en viss projektgrupp?

P3 – Ja, vi har ju inte så många projektledare, vi har fyra stycken, så att jag menar, det är ganska enkelt att samordna dem, är man ett bolag med flera hundra anställda och 40 projektledare då kanske man behöver ha en handbok som säger att så här gör vi liksom.

H – Jag tänker på det här med uppföljning och utvärdering av metod, är det mer informellt då eller?

P3 – Ja, vi har en gång i månaden ett sånt här projektuppföljningsmöte och det är ett generellt forum för projektledare där man pratar om projekten i sig men där pratar vi också om liksom... man hjälper varandra med saker som inte fungerar och så där. Vi försöker hålla det ganska praktiskt och informellt.

H – Mmm... ok...

P3 – Var det svar på din fråga?

H – Ja, absolut. Så det blir lite så här halvformellt liksom... alltså vad är liksom, man har möte om ett projekt allmänt och tar upp det som man känner behövs helt enkelt?

P3 – Ja

J – Nä, men det är lite det, det vi har pratat om nu som vi är ute efter så att det...**H – Alltså hur ska vi säga det här... vad har varit kritiskt så att säga för att få metoden att fungera liksom?**

P3 – Det som är svårt kan man ju säga är det som är kruckset, det är det att hitta balansen mellan att jobba agilt och att jobba vattenfall, alltså att tillgodose dem här... alltså den traditionella världen, alltså att vi är leverantör då det som jag var in lite på innan, liksom hur stelbent måste man vara för att göra offerter och hur, vilka traditionella moment måste finnas

under själva implemtationsprojektet för att våra kunder förväntar sig det liksom, det är det svåra.

H – Mmm

P3 – Alltså, vi skulle gärna jobba fullt ut agilt, det skulle vi jättegärna göra med våra kunder, att man hade en backlog och att man liksom prioriterade efter den och att man hade kanske en budget att röra sig vid, vi vill göra av med så många miljoner under det här halvåret, det här är vår önskelista, vi börjar uppifrån och ner liksom, det skulle vara fantastiskt för oss att jobba så. Jag är helt övertygad om att kunderna skulle bli nöjdare... men det är lite lurigt att få till det i praktiken eftersom man inte, det är lättare att kanske om man gör release två, tre eller fyra, men just när det är ny kund för oss och vi är ny leverantör för dem, vi har inte jobbat tillsammans innan, alltså det förtroendet finns ju inte där av naturliga skäl utan de vill ju ha liksom några, vad heter det? Alltså, de vill ju vara säkra på att de får det de betalar för och vill veta vad de betalar för. Så längre fram sen om man ska göra en release två tex, händer ju att vi gör det, då tror jag att det hade varit lättare att, alltså man känner varandra och man har jobbat ihop, att kunna släppa lite på de tyglarna från båda håll.

H – Tänkte fråga, bla en annan vi pratat med hade upplevt vissa problem och tänkte se om det går igen, han hade upplevt det här med scrum då att, han tyckte det tog bort ansvaret från den individuella utvecklaren, liksom att det kunde bli problem där, är det nånting ni har upplevt?

P3 – Nä, individen har ju fortfarande sitt eget ansvar. Scrum är ju lätt att misstolka om man vill det, tex det här med att man inte behöver dokumentera – det är skitbra, man behöver inte hålla några tider – det är också skitbra, och man gör det man hinner och det man inte hinner gör man i nästa sprint, alltså så kan man ju också tolka det om man vill men det är helt fel, det är ju inte det som är syftet med metoden, så visst ansvar på individen är ju fortfarande detsamma. Jag som chef vill fortfarande att Kalle och Pelle måste ju ta sitt eget ansvar och får de en uppgift så ska de ju leverera det som de har sagt, och den ska ha god kvalité, det försvinner ju inte. Men det är lätt att använda som argument att kunna gömma sig bakom det och säga det: åh nä, det är ju teamet som har det gem samma ansvaret och så, ja och det har dem men tanken med det är ju att man ska få en, alltså det är ju ett ledarskaps tänk att man liksom ska jobba tillsammans att man ska känna att vi gem samt, det här laget, ska sträva mot det här målet och Kom igen nu! vi hjälps åt, vi kämpar liksom, för man blir ju inte bättre än den... alltså det går ju inte fortare än den sista spelaren liksom, så det är väl mer det, så man ska se det tycker jag.

H – Ja, det som också kom upp där som jag tänkte på var just produktägare, han hade upplevt problem med att hitta en produktägare som liksom var rätt för rollen och var engagerad.

P3 – Ja, det är en bra poäng. Det är väldigt svårt eftersom produktägaren skulle vi egentligen vilja ha hos kunden ju, och det funkar ju inte heller fullt ut så där får vi också ha nån hybridmodell där vi har en... alltså så som vi jobbar, vi kan ju prata lite om det, så har vi en projektledare, en projektledare hos oss är den administrativa projektledaren som har hand om kundkontakten, budgeten, budgetuppföljning, se till att funktionsspecen efterföljs, att vi

implementerar det vi har lovat och ingenting annat, håller koll på tillägg, alltså det är den traditionella projektledarrollen så, den personen blir också lite produktägare eftersom det är han eller hon som har den mesta kontakten med kunden, om man säger att kunden egentligen är den riktiga produktägaren så blir det ändå den här projektledaren som får den här rollen internt, som ska vara någon slags tolkning av vad kunden har sagt.

Det är ofta svårt att ha en person hos kunden också, de är ofta några stycken. Då är det projektledarens jobb här hos oss då att liksom se till att prioriteringen faktiskt är i rätt ordning och kanske hjälpa kunden att göra det också, och förmedla det då till teamet, så det blir någon slags semiroll där. Och så har vi teamet då så har teamet en teamledare som är någon form av scrummaster då som ska vara den som faciliterar teamet, håller morgonmötena, ser till att allting flyter på, går till projektledaren och säger att jag behöver det här och det här, det här funkar inte, eller gå till mig som chef och säga att vi behöver ändra på ditten och datten, Kalle behöver en ny dator eller vad det kan vara liksom, så att det flyter på i det dagliga.

H – Men scrummastern är också utvecklare eller?

P3 – Lite beroende på hur stort teamet är men många gånger ja men inte alltid, det varierar lite beroende på vem det är och vad det är för projekt, hur stort det är och så. Och det funkar olika bra beroende på vem det är och vilket projekt det är också, många tycker att det är lite lurigt, men ofta har vi inte så stora projekt eller team heller så att vi har råd att ha en person som bara är scrummaster också, så ofta får man göra både och.

H – Produktägaren, jag gissar att det blir lite så där att projektledaren får översätta liksom då från...

P3 – Ja, precis han blir nåt slags tolk eller språkrör in i vår organisation liksom och han kan ju inte vara en renodlad produktägare heller för han måste ju också ta hänsyn till [Företaget 1] intressen, alltså med ekonomi och så.

J – Bara för att lite snabbt återgå till det här hur det lärs ut, ert arbetsätt/arbetsmetoden, det skedde informellt sa du? Ni har ingen handbok för hur ni ska arbeta?

P3 – Ja, alltså, det finns beskrivet någonstans, men vad jag skulle säga är att vi inte fokuserar så mycket på det utan jag tycker det är viktigare att folk har förstått, jag lägger hellre tid på att gå runt och prata med folk och kolla hur det funkar och ställa lite kontrollfrågor och liksom så än att skriva en handbok och dela ut till folk, för jag tror ändå ingen läser den.

J – Nä, ok. Jag tycker vi har täckt in det som vi har här. Du prickade rätt direkt när du pratade, rakt på sak, det var jättebra.

P3 – Men vad kommer uppsatsen mer handla om i...?

J – Vår forskningsfråga handlar om, eller den lyder hur agila utvecklingsmetoder anpassas eller tolkas för att för organisationens del då ge en effektiv arbetsmetod och sen också om hur den här arbetsmetoden sprids vidare till övriga inom organisationen som kan komma att arbeta med den också.

H – Så det är lite två steg, först översättningen från teori till praktik och sen alltså från praktik till liksom gemensam praktik.

P3 – Nu har ju vi ett väldigt agilvänligt företag, så det har inte varit svårt att liksom börja jobba på det här sättet utan alla har varit med på det, både chefer och utvecklare liksom, så det har varit väldigt väldigt enkelt att det har inte varit något motstånd överhuvudtaget liksom, det kan jag tänka mig att det finns på andra bolag som, där man kanske har mycket projektledare och folk som har jobbat i projekt i många år och har jobbat enligt vattenfallsmetoden och tycker det att det här är ju jättemärkligt så, men vi har en ganska ung organisation och de flesta tycker att det här är fullt naturligt för dem har kanske aldrig sett nånting annat så den biten har gått jättelätt här, vi har inte behövt anstränga oss alls för det.

H – Just det, tänkte på det också, ni har inte tagit in några konsulter, det har bara kommit folk inifrån organisationen?

P3 – Nä, alltså vi har väldigt mycket, jag är själv certifierad både product owner och scrum master och har jobbat i agila projekt, och vår VD var, jobbade på sitt förra jobb som produktutvecklingsansvarig och införde motsvarande grejer där liksom så vi har rätt mycket den kompetensen själva, så vi har aldrig känt att vi har behövt ta in någon extern.

J – Nä, för vi har förstått att det växer lite grand underifrån, det är inte så att det kommer från ledningen som säger att nu ska vi jobba efter den här metoden utan det kan lika gärna komma från utvecklarna som då nästan föreslår en metod...

P3 – Ja, här har det nog kommit från båda håll tror jag, vi har nog alla varit överens om att... men många gånger är det nog utvecklarna som tycker det, det går fortare där, det ligger lite grand dem närmare om hjärtat liksom. Ja, men då som sagt, lite traditionella ledare på företaget som kanske är lite äldre och har jobbat vattenfall, de vill ju fortfarande ha sina liksom mätpunkter och nyckeltal och så, det var det jag pratade om innan att, där måste man fortfarande ha det här lagret liksom, med att kanske filtrera bort det agila för dem som inte vill se det, så kan man se det. För dem vill ju ha svar på sina frågor ändå, kommer vi leverera i tid? kommer vi leverera inom budget?

H – Men så det blir lite så här som att projektledning för att skapa en kontext för ett projekt och så inom det kör man scrum?

P3 – Ja, så kan man se det.

J – Den kombineringen funkar bra eller?

P3 – Ja, det gör det, det måste det, jag ser inte att vi har nåt val liksom.

H – Det känns inte som att dem går emot varandra?

P3 – Ibland gör dem ju det, alltså det klickar ju inte perfekt, det gör det ju inte liksom, men det är för att det är lite grand äpplen och päron, alltså... det passar inte ihop för det är ju inte gjort riktigt för varandra, men det är den verkligheten vi lever i som sagt. Hade vi vart ett produktutvecklingsföretag, har ni pratat med nåt sånt också? För det är väldigt stor skillnad.

J – Nä

P3 – Gör det, det kan jag rekommendera. För det lever i en helt annan verklighet, alltså vi har ju projekt, nya kunder, början och slut, och så är vi leverantör och så är det inte mer med det, dem har ju en helt annan, de har ju liksom en roadmap, det är egentligen gjort och tänkt för den typen av bolag, alltså de har en evighets roadmap, de har en produkt som de vill göra till världens bästa, och det finns inget början och inget slut, det är inget projekt i den bemärkelsen liksom och då är ju detta väldigt naturligt och dem har kanske produktägare och så.

H – Mmm...

P3 – Det tycker jag ni kan göra, jag kan rekommendera det faktiskt.

H – Det var några vi snackade med som hade varit på sånt företag innan och det är ju gjort för det liksom då kan man använda vissa metoder som inte funkar annars

Ja, för du har ingen kund alltså allting är ju den samma, det uppstår motsvarande problem där också, för man har olika avdelningar på företaget, alltså så man har olika intressen och så där men det är i alla fall, vi har en gemensam budget och gemensamt tak och man sitter där med kontrakten och dem sakerna, man byter sällan leverantör inom produktbolaget, alltså det finns en utvecklingsavdelning och det är den man använder, det skulle rätt mycket till om man ringde ett annat företag och säger: Hej! Vill ni offerera vår produkt? Det gör man ju inte, så det blir väldigt så...

J – Vi känner oss nöjda.

H – Tack så mycket!

Bilaga 5, Intervjuprotokoll, P4

H – Vi tänkte börja med lite bakgrundsfrågor. Vad har du för utbildning?

P4 – Civilingenjör, elektroteknik, LTH.

H – Och din nuvarande tjänst är?

P4 – Projektledare kan man kalla det.

H – Vad har du för erfarenhet av agila projekt då, är det som projektledare?

P4 – Ja, jag har jobbat med agila projektmetoder de senaste tre åren, fyra, tre år kan man säga. Innan dess så var det mer vattenfalls...

H – Har det varit en agil metod, eller flera?

P4 – Nja, det är väl lite olika, det blir ju alltid lite så att man är inne på kustomiserat för olika företag. Sen är det olika faser på projekten också. Ett projekt där man har en road map och en feature-tillväxt som så, så kan man köra Scrum till exempel. Om man jobbar där det är, mer med maintainence där det är issues som kommer in som man ska hantera hela tiden så är det inte så lämpligt att köra Scrum för att man vet inte vad som kommer att ha kommit in och är av hög prioritet när man börjar en sprint.

H – Då funkar det inte att göra en backlogg?

P4 – Nä. Nä? En backlogg kan man ju ha, men man kan inte köra ostörda sprintar av en viss storlek. För att kommer det in prioriterade buggar under sprinten så måste man påbörja dom och jobba med dom. Det kan inte vänta till nästa sprint.

H – Så det är egentligen inte agilt nog?

P4 – Nä, exakt. Så är det. Jag vet inte om ni känner till Kanban. Det är ju lite mer flexibelt på det sättet att man behöver inte ha sprintar utan man kan köra på sin backlogg och olika faser för varje aktivitet, task, utan att man är bunden till sprintar, som sagt. Så det funkar lite bättre när man kör där det är superagilt där det kan ändras från dag till dag.

H – Är det huvudsakligen Scrum och Kanban som inspirerat er till ert arbetsätt?

P4 – Mmm...

J – Har ni tagit delar ifrån dom båda två och satt ihop till en helhet?

P4 – Just nu håller vi faktiskt på och... För det har varit lite olika projekt som man har kört. Vissa har kört Scrum och vissa vet jag inte vad dom har kört. Så, vi håller på att fundera lite nu. Då är det väll Kanban som är den minsta gemensamma nämnaren känns det som. Det är ganska så fritt. Man kan anpassa det rätt mycket. Man kan ju mer eller mindre göra Scrum av Kanban om man vill.

J – Men har ni en sån inställning att ni tittar på det, hur ni skulle vilja anpassa det till er egen verksamhet, så ni köper det inte rakt av bara?

P4 – Nä, det är ju inspiration, eller förenklat: hur har man löst det i Kanban. Det är sådär. Det kanske kan funka för oss också. Jag tror inte det finns nått företag, det hade varit konstigt med ett företag som skulle köpa en process rakt av utan att fundera igenom och gallra, eller lägga till. Jag har jobbat på [Företag 1] tidigare, för några år sedan. Under tiden som jag jobbade där

så införde dom nånting som dom kallade ”new ways of working”, som var nått Scrum-baserat men där dom ändå lite höll kvar dom här vattenfallsprojekt delar med PROPS och sådär. Så det blir... Det blev säkert bra till slut, men i början blev det både... det skulle både vara agilt inte agilt.

H – Jag tänkte, när ni kom fram till er arbetsmetod, var kom det ifrån. Kom det uppifrån för hela organisationen, kommer det från teamen själva att bestämma sitt arbetsätt?

P4 – Nu jobbar jag på [Företag 2] så vi är ett företag med... 80-100 anställda, nånting sånt. Vi är väl sju-åtta projektledare och kanske 40-50 utvecklare och där har det väl blivit mest att vi själva hjälps åt allihopa, alltså vi projektledare, att komma fram till hur vi vill arbeta tillsammans då. Vår chef då som gärna vill vara med då och bestämma processen, men vi har alla jobbat här ett tag med olika projekt så vi har väl kommit fram till vad som funkar och vad som inte funkar. Så har vi väl nu hittat ett gemensamt sätt att driva det.

H – Så ni kör nån slags kontinuerlig utvärdering?

P4 – Ja. Ja, det kan man säga. Jag vet inte. Just nu vi mer bara speca upp en process, både abstrakt hur vi jobbar, och så konkret med vilka verktyg vi använder. Jag vet inte om ni känner till jira atlassian.

J – Nä.

H – Nä.

P4 – Det är i alla fall ett verktyg för issue management och som då också har ett tilläggspaket som är... greenhopper kallas det, som då är ett verktyg för att jobba med agila projekt, där man kan göra sin backlogg och planera olika releaser och sprintar, burndown charts och sånt. Och så är det då kopplat till det här issue management-systemet, där man hanterar buggar och change requests och andra... releaser. Det finns lite olika delar i det. Först försöker man hitta på nått sätt, helst då minsta gemensamma nämnare som alla projekten kör för att man ska känna igen sig lite om man behöver vara stand in för nån eller man ska ta över nåns projekt eller om man vill få insyn i ett projekt. Vad dom pysslar med liksom, och kunna få reda på leveranser.

Man kanske är beroende av ett annat projekts leverabler eller sånt, och då är det ju bra att kunna få insyn i det projektet, och fatta vad dom pysslar med och hur dom ligger till med sina olika tasks. Det vi håller på med just nu är att hitta minsta gemensamma... vad vi behöver åstadkomma. Sen är då nästa steg som vi har börjat på, vilket verktyg vi ska användas för att åstadkomma det då är det här atlassians jira... har vi använt en del tidigare, men alla tycker inte att det är jättesmidigt. Som många andra verktyg så är det webbaserat, måste hela tiden kommunicera med en server, och det innebär att varje klick jag då gör... fem sekunder, och man blir rätt frustrerad av att skriva in sina grejor och så är det fem sekunder varje gång man ska klicka på nånting, även om servern sitter inne på vårt intranät.

Det är inte responsivt helt enkelt. Och så är det alltid lite för många fält, och så ska man helst kunna... fält som är tvingande. Det är bra att istället ha verktyg som inte är så tvingande. Om man ska lägga in en ny task till exempel så kanske... I det här fallet ska man fylla i 15 fält och så tänker man skit i det för det tar så lång tid att fylla i det. Då har man misslyckats med

vektyget, för att det dröjer en minut att fylla i det när man helst bara vill att det skulle dröja fem sekunder.

J – Jag tänkte på det du tala om nu med att alla skulle veta vad som sker i olika projekt så man kan hoppa omkring lite så. Har ni nån praxis för det? Hur man förmedlar den här informationen till allihopa.

P4 – Nä, det har vi inte. Eller menar du hur man förmedlar...

J – Blir det nån informell kommunikation eller är det formell kommunikation?

P4 – Vi har ju statusmöten en gång i veckan där alla berättar om sina olika projekt, vad statusen är och framförallt om man har några problem. Då lyfter man det då till andra och chefen som är med där. Men. Vi har inte nån formell kommunikation hur vi jobbar i dom olika projekten. T.ex. att nån jobbar enligt Scrum och nån jobbar enligt Kanban och nån jobbar enligt nånting annat. Det har vi inte. Det är mer för att berätta innehållet i projektet. Hur det går med processen.

J – Så mer informell kommunikation då egentligen?

P4 – Ja. Mellan projektledare då. Sen inom projektet så måste man ju berätta för projektdeltagarna hur det körs för annars så förstår dom ju inte vad dom ska jobba med om man inte exakt berättar det hela tiden. Det är inte så effektivt, och ganska oinspirerande för dom som sitter... utvecklarna som sitter jobbar. Att bli micromanage:ade hela tiden, där jag berättar exakt vad dom ska göra. Det är ju mycket bättre att ha en backlogg med färdig... så behöver man inte gå och fråga projektledaren vad man ska göra utan då kollar man ju backloggen så man vet vad man ska göra.

H – Hur funkar den kommunikationen?

P4 – Ja... vi kör ju... i dom flesta projekt så kör vi ju möten åtminstone några gånger i veckan, och oftast så har man ju morgonmöten.

H – Dagliga?

P4 – Mmm, och så sitter vi ju nära varandra så det blir ju ofta flera gånger om dagen som man pratar inom projekten, just för att alla är... det blir lätt så att man inte har dom här även om jag går runt och pratar med projektdeltagarna många gånger, så måste man ändå ha dom här morgonmötena för att alla ska samlas och den ena ska... Kalle vet vad Nisse gör också

J – Det ska finnas en gemensam förståelse?

P4 – Ja, så alla får lite helikopterperspektiv.

J – Kan det komma in andra projektdeltagare in i ditt projekt, till exempel, för att hjälpa till?

P4 – Ja, absolut.

J – Hur lär ni upp honom då så att säga, hur ni arbetar?

P4 – Det är egentligen genom att vara med på dom här dagliga mötena. Är det nått väldigt kort så... Någon som hoppar in och ska hjälpa till några dagar så... ja det blir mer eller mindre att man micromanage:ar den personen. Säger att det är exakt det här du ska jobba med, och så gör han eller hon det. Är det lite längre tid så får man sätta upp lite mer processer.

Som sagt det är inte sådär jätte komplicerat precis, vid planeringen till exempel. Om man har mycket planering. Dom flesta har ju jobbat med Scrum här nån gång tidigare.

J – Då är det ganska lätt att känna igen sig?

P4 – Ja.

H – Jag tänkte på Scrum. Var kommer produktägare in för er? Var kommer det perspektivet ifrån?

P4 – Vi har ju på [Företag 2]... det är lite olika. Jag jobbar mycket med kundprojekt. Vi har ju dels kundprojekt som vi kör mot kunder. Vi gör då produkter som integreras hos kunder, och den integrationen blir ett kundprojekt. Produktägarna för våra produkter, dom sitter ju här på kontoret, och läser av marknaden och vad som är lämpligt att vi gör härnäst. Vilka produkter som bör uppgraderas. Vilka nya produkter vi bör göra och så. Sen så sätter dom upp en road map så som dom tror att det skulle kunna se ut. Sen försöker projektledarna planera enligt den här road map:en, och exekvera då för att vi ska få ut det dom önskar sig.

H – Så det är typ en tvåstegsprocess, först marknadsanalys och sen projektledarna som översätter det till något genomförbart?

P4 – Ja, kan man säga. Det är ofta så att det kommer high level requirements från produktägaren då. Jag vet inte om ni vet riktigt vad vi pysslar med, men vi pysslar med bilder och kamera i mobiltelefoner. Till exempel: Nu vill vi ha ett nytt cartoon-filter där alla ser ut som seriegubbar om man har tagit en bild. Det är kanske på den nivån eller lite mer detaljerat som produktägaren tar det, sen blir det upp till respektive projektteam att bryta ner det här till nånting som är mer definierat, och planera in när det skulle kunna genomföras i tiden. Sen andra delen som jag då jobbar mer med är kundprojekten. Där är det ju kunderna som är kravställare och därmed produktägare.

H – Hur funkar det då att få backlogg från dom?

P4 – Nä, utan det får man själv försöka översätta deras önskemål till nån slags backlogg. Det finns nog dom kunder som har lite koll och skulle kunna jobba så men dom flesta skickar mest ett mail. Man har kommit fram till ett scope. Jag vet inte om ni vet vad en statment of work är, men det är egentligen ett dokument som beskriver vad som ska göras i projektet. Däri står vad som ska göras. Oftast så står det inte nån prioritet men ibland dom olika leverablerna därinom, så det får man själv försöka leta ut. En prioritetsordning är ju vilken ordning grejorna ska levereras i, men det behöver ju inte vara det som är hela sanningen. Det kan ju vara andra delar med som bestämmer vilken ordning det ska prioriteras i. Då får man som projektledare försöka... då blir man lite produktägare som projektledare.

H – Så produktägaren blir lite uppdelad mellan kunden och produktägaren?

P4 – Ja, precis.

[Därefter tackade vi för oss och sa hejdå]

Bilaga 6, Intervjuprotokoll, P5

J – Vi börjar med lite bakgrundsfrågor om dig.**H – Vad har du för utbildning?**

P5 – Jag har läst teknisk matematik på LTH, så det är en ganska ny linje, civilingenjör, så mycket fokus mot bildbehandling och datorseende och sådana grejer.

H – Vad är din tjänst här?

P5 – Min tjänst nu, jag är ansvarig för all roadmaps, jag är director of products, så jag har hand om planen så att säga, lite mer långsiktigt, noll till ett år är min, men det blir ofta att man tittar lite mot ett, två år sådär. Men mycket fokus ligger ju på noll till tre månader, det är där execution är liksom.

H – Om man utgår från Scrum, så är du lite av en produktägare där då eller?

P5 – Nej, alltså jag har produktägare under mig som rapporterar till mig då så att de är mer... om man tittar i scrum, så har vi just nu inte scrum rent av utan vi jobbar med produktprojekt, så vi har mer projektform, och de då är ju någon form av product owner eller sådär motsvarande då för det här projektet, sen om dem jobbar med scrum eller kanban i projektet är upp till dem, de får göra vad de vill, de kan jobba med ren projektmetod metodik så att säga men det viktiga där är vad som kommer ut så att säga i slutmålet, sen ansvarar jag då för hur den här produkten ska tas vidare och vad händer och så där. Så jag är inte på detaljnivå utan jag är mer på en strategisk nivå. Jag gör highlevel-krav i princip, verkligen på stor nivå – vad är huvudfeaturen i den här produkten, den är ju baserad på teamets diskussioner men, sen att bryta ner den till små features, små tasks, det är det teamet som gör.

H – Har du någon erfarenhet personligen av agila projekt?

P5 – Ja, jag har varit produktägare i renodlad scrum, vi körde den för 1,5 år sedan, vi körde verkligen i princip, alltså... vi körde den 99 % scrum skulle jag säga, vi hade maintenance i teamet som skiljde sig och det körde vi lite på sidan, så vi räknade med viss lack, vi har alltid varit lite bortskämda med produkter som har väldigt förutsägbar maintenance här. Så det har väldigt sällan varit spikar i maintenance utan vi har haft liksom 10 % av våra utvecklare har jobbat med maintenance och det har varit väldigt lätt att förutsäga, så det var så vi gjorde då, vi gjorde scrum team, jag var produktägare och så hade vi en teamleader...

H – Ni körde liksom benhårt, möte varje dag?

P5 – Dem hade möte varje dag, jag var med... ja, vi körde by the book i princip. Vi hade även scrum-kurser här med, vad heter dem nu? Med [Företag 1] som gjorde lite scrum-kurser.

H – Ja, de låter bekanta. Men din projektroll var alltså produktägare?

P5 – Ja, det var min roll då ja, jag var även utvecklare i ett halvscrum produktteam för tre år sedan. Det var, då var vi inte riktigt renodlade.

H – Var det teamen själva som valde vilka metoder de ville använda?

P5 – Just nu ja, vi kan väl säga att vi är en reorg så vi har gått från ganska renodlad scrum som inte fungerade för att vi, pga. andra organisationsförändringar, vi hade en supportavdelning på sidan så att säga, sen hade vi en produktutvecklingsavdelning som körde scrum, det funkade

så länge, men problemet var att våra projekt började under förra året ta väldigt mycket plats, så det gjorde det liksom, det döda teamen på nåt sätt i den meningen att vi inte kunde ha kvar den här scrumpart-biten och det gjorde att det blev svårt att driva scrum, så att, sen gick vi tillbaka till ritbordet och började från början – hur får vi ihop de här två organisationerna? Och då var det projektmodellen, och så ville vi fortfarande ha kvar teamen, så istället för att teamet tar ansvar för en task, som de får av en produktägare så tar nu teamen ansvar för ett projekt eller flera projekt där varje projekt såklart har en tasklist, så egentligen handlar det mer om att alla team här nu ska kunna jobba med både produktprojekt och kundprojekt beroende på vad som kommer in, innan var det väldigt behårt runt roadmap och sen kundorganisationen jobbade helt latjo lajban, det var inte alls nån metodik utan det var - om kunden sa hoppa och rosa hattar så gjorde vi det.

H – Ni hade alltså då flera team när ni körde scrum? Hur fick ni liksom ihop dem?

P5 – Det fick vi bra, vi var otroligt effektiva med scrum, jag skulle säga att det som var det bästa med scrum var att vi var otroligt effektiva på de produkter vi redan hade släppt, för att göra patcher, maintenance, förbättra dem, lägga in lite nya features men ändå inte ändra designen, så att säga arkitekturen. Det var otroligt effektivt, jag tror aldrig nånsin att vi kommer vara så effektiva här med en sådan metod, däremot att göra helt nya features eller helt nya funktioner för scratch var inte så smidigt för det gör det svårt eftersom du inte har det här en månadswindowet du ska göra userstories, du har en början och ett avslut, det är väldigt svårt när du jobbar med nånting som är lite semiresearch, det är liksom så här bara en månad, du kan inte ens göra en tasklist. Så det är det som inte funkade på oss, att göra nya features och kunduppdrag i scrum.

H – Så på ett mer långsiktigt arbete funkar det bra?

P5 – Ja, att göra en ny Ipad liksom från scratch hade nog inte funkade i scrum, men att göra från en Ipad 2 till Ipad 3 hade funkade alldeles utmärkt, det är lite det som vi upplevde. Jag hoppas nu att den här kombinationen av ett projektform och... vi kommer nog gå tillbaka till motsvarande scrum, vi har ju fortfarande dagliga möten tex. varje dag, vi har requirement list och vi har liksom del av det kvar men vi har inte tvingat, ett tag hade vi det påtvingat så att säga: nu är det scrum som gäller, så då körde vi allt med planeringskort och sådär, det gör vi inte nu.

H – Hur ser den formella strukturen ut för metodval? Var ligger ansvaret, är det på projektledare då eller?

P5 – Ja, så projektledaren är en motsvarande product owner och scrummaster i scrum-metodik skulle jag säga, sen har vi inte startmöten och slutmöten, vi har inte integrationen, det är möjligt att det kommer snart men just nu har vi inte det. Får se, jag tror det kommer komma tillbaka, jag tror vi kommer en backlog, jag tror vi kommer gå tillbaka till en lite mer scrumlik typ av sätt att jobba på. För det visade sig vara väldigt effektivt.

J – Jag tänkte på det du sa att ni hade flera scrumteam, hur integrerade ni med varandra, alltså hur var kommunikationen där emellan? Hade ni nån speciell metodik för det?

P5 – Nej, alltså utvecklarna kände ju varandra mellan teamen och teamen fick tydliga ansvar

på kodmoduler och så hade vi då arkitekter som låg lite åt sidan sådär, som var lite cross, som såg till att strukturen i kodbasis såg likadan ut beroende på vilket team man var i, sen hände det då ibland att teamen, folk bytte plats i teamen, det var lite rotation om en väldigt liten men det var väldigt lätt att byta team, alltså dem hade exakt samma form så det funkade väldigt bra, det enda problemet var ju återigen då när man har team, att vissa team jobbade med ganska mycket prio 1 saker, ett annat team kanske hade prio 3 saker som man jobbade på för att det helt enkelt inte gick att flytta över, det var ju inte bara att flytta över prio 1 saker till det teamet för det var ett annat ansvar.

Så där, egentligen skulle vi göra det idag med samma sätt så skulle vi lastbalansera dem teamen lite mer dynamiskt än vad vi gjorde då, vi hade dem väldigt statiskt, det var liksom 10 pers här, och sen försökte vi fylla backloggen så att säga snarare än att man skulle säga: ok, den här backloggen kändes inte riktigt prioriterat så det här teamet, dess existens kanske ska ifrågasättas så att säga, det är en väldigt dynamisk bransch liksom, vi jobbar ju med mobilmjukvara till kameran som ni kanske vet, så det händer ju mycket hela tiden. Det är väl inte helt trivialt att liksom planera, inte ens för [Företag 2] så att säga, så tycker vi lyckas ganska bra men det är ändå, jag skulle nog säga att varje kvartal får man nog tänka på att lastbalansera teamen i den här branschen, det är helt omöjligt att förutse.

H – Som jag förstår det nu, i dagsläget är det mer projektledare som är produktägare? I kundprojekten då, alltså når kundens åsikter fram?

P5 – Ja, då har vi kundprojektledaren, det är motsvarande en projektledare, P4 är ju ett sånt exempel, så det finns ju liksom projekt som han driver som är direkta kundprojekt, där han även checkar in kod så att säga eller gör en produktutveckling i nån mening så att säga, det är där jag kopplas in för då blir på nått sätt, jag blir ju ytterst ansvarig för det han checkar in i produkten som sedan vi kan leverera till andra kunder förutom hans kund som han jobbar med så att säga, sen är det klart det finns ju andra saker som han jobbar med som bara är kundspecifikt mot den kunden och det skiter jag väl i i nån mening, alltså det är inte mitt ansvar i alla fall, det är klart att det ska bli bra men jag litar väldigt mycket på, vi litar ganska mycket på våra produktägare så att säga eller våra projektledare då, de är duktiga som ni kanske märker, de vet vad...

H – De är duktiga för att liksom översätta från faktiska kunder till reella?

P5 – Ja, det har inte varit nåt problem, vi har ju utvecklare som i Taiwan tex. vi jobbar med [Företag 3] då har vi kundteam i Taiwan som accessar kunden först, det är de som sitter, de som svettas lite ibland med vissa kundrequests, sen pratar de med en mottagare som vi har här i Sverige, som sänder kundprojekten här inne, så det är aldrig så att de går rakt in i en kunds, en request går aldrig in direkt i backloggen, utan det vänds och vrids lite, särskilt i och med att vi har en produkt, en baseline som vi själva tillhandahåller våra kunder.

H – Alltså, sker det nån dialog där som kollar har jag uppfattat rätt eller är det bara att kunden kommer med åsikter sen översätts det in i backloggen?

P5 – Nej nej, det blir ofta mycket bollande, sen kommer vi fram till liksom en tradeoff. Som nu tex, jag kommer precis från ett möte där vi, kunden hade kommentarer och då tittade vi på vad vi kan göra och så lämnar vi ett förslag och sen får vi se lite vad dem kommer med, vi är

ju fortfarande ett svenskt företag så vi gillar ju att prata, men nej där har vi stenkoll på i alla fall vad som kommer in i produkten. Alltså, jag vet inte, det är frågan vart vi går, antingen går, vi är just nu i ett vägval, vi kan gå mot kanban liksom motsvarande, kanske lite mer löst men jag tror att vi kommer ha nån form av scrum, det är det vi är vana vid.

H – Kör ni nån utvärdering av det här då? Eller mer allmän diskussion?

P5 – Ja, mer allmän diskussion.

H – Ok

P5 – Anledningen till att vi valde scrum från början var väl att vi hade en chef för vår mjukvaruutveckling som var väldigt prosmant, vi hade en diskussion sen insåg vi att det här passar oss så vi kör på det, så det finns ju, det är återigen ett modeord. Det känns lite mer som färgval eller jackor och lite sånt.

H – Jag tänkte eftersom det kom lite uppifrån så att säga, funkar det att grunda det ända ner liksom, hängde alla med?

P5 – Ja, det gör det. Det var ju väldigt annorlunda mot hur vi jobbade innan, innan jobbade vi verkligen... nån form av väldigt ostrukturerat sätt att jobba, det var väl innan scrum som var det verkligen, då hade vi ett team som i princip fick tasks lite av dem som hade kundprojekt eller dem som var technical account manager, så hade vi en mer start techleader-roll, där techleadern i princip var produktägare också, men det funkade när vi var små så att säga, då fick han in en massa request och så tittade han lite på den sen bestämde han ihop med teamet vad vi gjorde så var det lite, ja vi satte en deadline för i princip varje CR, problemet är att det blev extremt svårplanerat när vi växte och fick många kunder, vi kunde inte sätta en deadline för varje CR, då blev det väldigt smidigt att ha en scrum-metodik för att säga: nä, men den kommer in i nästa sprint. Eller den kommer förmodligen inte in i nästa sprint utan den kommer nog att... och sen hade vi kvartalsreleaser och fick en puls, så vi gick från kaos till ordning i nån mening, så att det var ganska lätt att genomföra, folk var ganska frustrerade.

J – Men ni hade då en extern utbildare, [Företag 1]?

P5 – Ja, det var [Företag 1]. Så vi hade en utbildning för alla en dag och vi hade även uppföljningar och vi gick igenom... vi började med ett intranät eller ett buggrapporteringsystem, det här klassiska Jira, så vi hade en IT-kille som var väldigt duktig på att göra flöden så han gjorde väldigt bra scrumigt flöde så att säga i Jira så vi kunde flytta produktkrav och bryta ner smidigt och så här i... så det var riktigt smidigt. Så jag kunde gå som produktägare ner i varje checkin av kod och se exakt vad som hade gjorts med kommentarer och så där. Det var väldigt översiktligt och väldigt bra.

H – Hur var det då, förde ni in scrum i komponenter eller som helhet?

P5 – Hur menar du?

H – Alltså, vissa väljer att bara ha tex dagligt möte och så slänger man in nåt annat.

P5 – Nej, vi körde bara en night and day, det var efter jul tvåusen... ja nåt år, efter jul så sa vi: nu byter vi! Och vi sa i princip att vi räknar med en månads waste, alltså vi räknar med failure och vi sa liksom att: ok, nu pratar vi, skitmycket pratar vi liksom och så fort vi inte

fattar nåt så är vi accessible, vi försökte vara nåbara vi försökte prata, faktum är att den första månaden blev okej, det vi missade var ju tidsestimeringarna som var åt pipsvängen, vi kanske inte lyckades få en riktigt bra kvalitet med kod innan en sprints avslut och så där, det var den typen av tweaks vi blev bättre på.

H – Hade ni fortsatt stöd av [Företag 1] under den här inlärningsperioden?

P5 – Nej, det hade vi inte, utan de var med i början och sen fick vi nåt sånt här kit liksom, lite kort och instruktioner, sen hade vi ju han den här [Namn 1] från [Företag 4], som ni säkert träffat då, det är [Namn 2] chef. Så han följde ju upp det själv så att säga, det var bra, så att det... men varje (ohörbart) har ju klart en liten diskussionsrunda minst sagt, men vi får se vart vi är på väg nu, jag tror vi kommer dit med mindre projekt, jag tror det blir bra, då vinner vi överskådligheten av, det kan man väl se lite med scrum liksom att... för att hålla ihop nånting som är lite större än bara business as usual som förbättras, scrum är ju väldigt bra för patchar, men ska du göra nåt lite större så är det rätt meckigt, vi försökte göra nya features i scrum och det var ganska kämpigt, tillslut sa vi bara att, la vi liksom in den här fyra veckor, vi hade fyra veckor sprint, fyra veckor research, liksom den här och en person, då var det lite oscrumigt liksom, men ibland fick man göra det för att det liksom inte gick att göra users service liksom.

J – Men du sa det, det är kanban ni funderar på?

P5 – Kanban är ju lite mer vilda västern i min värld, man kan ju lite scrum fast jag vet inte, allt är ju relativt. Vi tittar på det lite grand men det är väl en diskussion som vi kommer ha nu innan sommaren så får vi se hur vi gör, det är lite mer öppet, men det funkar ju som det är, vi har projektledare och ganska små team och vi jobbar ganska bra, vi gör ju små deadlines hela tiden, vi får räkna med att det finns vissa penalties med att jobba så som vi gör just nu men det är lite transition så att det funkar.

J – Ok, men det var lite skillnad från vad vi har hört förut, att ni då kör hela metoden på en gång...

P5 – Alltså, P4 var ju inte i produktorganisationen...

J – Nä, jag menar alltså inte från ert företag utan det vi hört från tidigare intervjuer, så det går att... alltså folk gör på olika sätt helt enkelt...

P5 – Ja, jo det har man ju förstått, verkligen alltså, det här med att folk skriver på sitt CV att de kan scrum är ju bara bullshit, alltså det känns liksom... det är så här... ja, jag har jobbat i ett team som får post-it-lappar.

H – Vi har fått lite upplevelsen om att många ställen kanske bara läser lite grovt om scrum och tänker att nu kör vi det och så tänker man inte så mycket mer på det.

P5 – Ja, men vi var ganska religiösa, det var bra, alltså det funkade verkligen, det gjorde verkligen att vår... det blev en jättestor skillnad från att köra lite så här källarkodning till scrum, visst det är klart, det kunde kännas löjligt att flytta post-it- och startmöten och slutmöten där man på nåt sätt... vissa team hade två veckor scrum, det var så här: ja, titta, här har vi gjort dem här tre sakerna, så visade man en demo så här och så gick man ut, sen hade man ett startmöte dagen efter: ja, nu ska vi göra dem här tre sakerna. Det är klart det känns, men det var ändå en ganska stor psykisk sak liksom att man måste presentera nåt.

H – Men det var väl också det att ni kom inte från vattenfall utan att ni kom från lite mer ad-hoc där ni kräver lite mer översikt som inte gick när ni växte, om jag har förstått det rätt?

P5 – Ja, precis. Det är många som ju fortfarande förespråkar vattenfall, det är ju det som är motargumentet till varför man inte ska jobba med scrum, det är ju vattenfall. Som i min roll, vattenfallsmetodik är ju som att leka ”Phu pinnar” eller vad man kallar det, när man slänger ner en pinne och så kollar man när den kommer fram, så får man, det är ju liksom den bästa estimeringen man har. Nån form av projektmodell eller scrum-modell är ju betydligt enklare att estimeras, då har du ju heartbeats, du har... du vet liksom dina, man kan kommunicera till en kund i mjukvaruindustrin att den kommer denna månad eller nästa, ofta om det inte är ett akut problem, är det ett akut problem så gör man ändå en quickfix eller nånting, så det...

H – Som det har först en del kritik mot är ju det lite att kunder ofta vill höra liksom att det är det här vi har beställt, så här mycket kommer det att kosta och så här lång tid kommer det ta liksom, att det inte skulle funka med scrum...

P5 – Nej, det håller jag inte med om. Min bild är att man hade väldigt bra kostnadskontroll, för att du vet ju i alla fall i teamet så kunde man ju lätt göra en tidsestimering, jag menar vi satt på startmöten och jag hade idéer på saker jag ville göra, så var det vissa saker som dem bara la i en hög siffror, då var det så här jag vänder tillbaka den här, det var ganska enkelt, då kunde man bara gå till kunden och säga: ledsen, vi kan inte göra det, liksom vårt team sa att det tar för lång tid, då blev det: men om vi gör så här och så här, så kunde nästa sprint gå dit liksom, vi lyckades tex. att få igenom... alltså öka priset på vissa saker och sådana grejer för att teamen estimerade, alltså för man hade koll. Annars är ju risken att man bara: det här låter kul, det kör vi på sen vet man inte riktigt, jobbas det jobbas det liksom.

H – Som vissa menar liksom att produktägaren ska vara en kund som bara slänger in vad som helst utan backlog utan den (produktägarrollen) är mer i två steg?

P5 – Ja, alltså vi hade ju en backlog men vi hade ju ändå förplanningsmöten innan startmötet där vi tittade lite på backloggen och så där och hade en dialog men sen planingsmötet satt jag i princip som lyssnare och då lät jag dem dra sitt sträck liksom efter backloggen, man kanske skippade nån, dem tog ganska fritt så att säga, där var vi nog ganska bra, det är klart man behövde planingsmöte för att kunna kommunicera hur man faktiskt tänkte göra, att läsa en userstory är ju inte... kan ju vara lite flummigt också... så det är typ så. Men det ska bli kul att se om vi kan få in det i projektmodeller och så där på sikt, det hade varit intressant.

J – Jag tror vi har täckt in det som vi behöver ha.

[Därefter tackade vi för oss och sa hejdå]

8 Referenser

- Bhalerao, S., Puntambekar, D. (2009): Generalizing Agile Software Development Life Cycle, *International Journal on Computer Science and Engineering*, Nr: 1, s. 222-226.
- Cervone, H.F. (2011): Understanding agile project management methods using Scrum, *OCLC Systems & Services: International digital library perspectives*, Nr: 1, Vol: 27, s. 18-22.
- Christensen, S., Kreiner, K. (1997): *Projektledning - Att leda och lära i en ofullkomlig värld*. 2:a uppl. Köpenhamn: Jurist- og Ekonomförbundets Forlag.
- Deenen, R. (2007): Project Governance – Phases and Life Cycle, *Management & Marketing*, Nr: 1, s. 193-198.
- Elshamy, A., Elssamadisy, A. (2007): Applying Agile to Large Projects: New Agile Software Development Practices for Large Projects, *Lecture Notes in Computer Science*, Vol: 4536, s. 46 - 53.
- Gerami, M. (2010): Knowledge Management, *International Journal of Computer Science and Information Security*, Nr: 2, Vol: 7, s. 234-238.
- Harmon, P. (2007): *Business Process Change: A guide for Business Managers and BPM and Six Sigma Professionals*, 2:a uppl., Burlington: Morgan Kaufmann Publishers.
- Hayes, S., Richardson, I. (2008): Agile Processes in Software Engineering and Extreme Programming, *Scrum Implementation Using Kotter's Change Model*, Vol: 9, s. 161-171.
- Hneif, M., Hock Ow, S. (2009): Review of agile methodologies in software development, *International Journal of Research and Reviews in Applied Sciences*, Nr: 1, Vol: 1.
- Hunt J. (2006): *Agile Software Construction*, London: Springer-Verlag.
- Jacobsen, D. (2002): *Vad, hur och varför? Om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen*. 1:a uppl. Høyskoleforlaget AS – Norwegian Academic Press.
- Jacobsen, D., Thorsvik, J. (2008): *Hur moderna organisationer fungerar*. 3:e uppl. Bergen: Fagbokforlaget Vigmostad og Bjørke.
- Jyothi, V. E., Rao, K. N. (2012): Effective Implementation of Agile Practices – Incoordination with Lean Kanban, *International Journal on Computer Science and Engineering*, Nr: 1, Vol: 4, s. 87-91.
- Kettunen, P., Laanti, M. (2008): Combining Agile Software Projects and Large-scale Organizational Agility, *Software Process: Improvement and Practice*, Nr: 2, Vol: 13, s. 183-193.
- Lalsing, V., Kishnah S., Pudaruth S. (2012): People factors in agile software development and project management, *International Journal of Software Engineering & Applications*, Nr: 1, Vol: 3, s. 117-137.

Landy, F. J., Conte, J. M. (2007): *Work in the 21st Century*. 2:a uppl. Malden: Blackwell Publishing.

Macheridis, N. (2009): *Projektaspekter*. 3:e uppl. Lund: Studentlitteratur AB.

Magdaleno, A. M., Lima Wenera, C. M., Mendes de Araujo, R. (2011): Reconciling software development models: A quasi-systematic review, *Journal of Systems and Software*, Nr: 2, Vol: 85, s. 351-369.

Mishra, A., Mishra, D. (2011): Complex software project development: agile methods adoption, *Journal of Software Maintenance and Evolution: Research and Practice*, Nr: 8, Vol: 23, s. 549-564.

Moe, N. B., Dingsøyr, T., Dybå, T. (2010): A teamwork model for understanding an agile team: A case study of a Scrum project, *Information and Software Technology*, Nr: 5, Vol: 52, s. 480-491.

Petersen, K., Wohlin, C. (2011): Measuring the flow in lean software development, *Software: Practice and Experience*, Nr: 9, Vol: 41, s. 975-996.

Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., Still, J (2008): The impact of agile practices on communication in software development, *Empirical Software Engineering*, Nr: 3, Vol: 13, s. 303-337.

Rychlý, M., Tichá, P. (2008): *Balancing Agility and Formalism in Software Engineering*, New York: Springer Berlin Heidelberg.

Singh, A. (2012): Agile: Analysis of its Problems and their Solutions, *International Journal of Computer Applications*, Nr: 6, s. 32-35.

Softhouse. (2012): Hämtad: 2012-02-22 från:
<http://www.softhouse.se/index.php/ladda-ner-information/>

Stepanek, G. (2005): *Software Project Secrets: Why Software Projects Fail*, New York: Springer-Verlag.

Stober, T., Hansmann, U. (2010): *Agile Software Development*, London: Springer Berlin Heidelberg.

Suárez-Barraza, M. F., Ramis-Pujol, J. (2010): Implementation of Lean-Kaizen in the human resource service process - A case study in a Mexican public service organization, *Journal of Manufacturing Technology Management*, Nr: 3, Vol: 21, s. 388-410.

The Agile Manifesto. (2001): Hämtad: 2012-04-05 från:
<http://agilemanifesto.org/>

Turban, E., Aronson, J., Liang, T-P., Sharda, R. (2010): *Decision Support and Business Intelligence Systems*. 9:e uppl. Upper Saddle River: Pearson Education.

Turner, R., Ingold, D., Lane, J. A., Madachy, R., Anderson, D. (2012): Effectiveness of kanban approaches in systems engineering within rapid response environments, *Procedia Computer Science*, Vol: 8, s. 309-314.

Vlaanderen, K., Jansen, S., Brinkkemper, S., Jaspers, E. (2011): The agile requirements refinery: Applying SCRUM principles to software product management, *Information and Software Technology*, Nr: 1, Vol: 53, s. 58-70.