

Som handen i handsken

**En flerfallsstudie av den praktiska användningen
av den agila metoden Scrum**

Andersson, Samuel
Lindgren, Edvard

Handledare: Mikael Olsson

Examinator: Ann-Sofie Köping

Södertörns högskola | Institutionen för samhällsvetenskaper

Kandidatuppsats 15 hp

Företagsekonomi | Vårterminen 2017



Förord

Vi skulle vilja inleda med att rikta ett stort tack till er som har bidragit, stöttat och hjälpt oss i arbetet med den här studien. Vi vill också tacka vår handledare Mikael Olsson, våra opponenter och andra som hjälpt oss. Tack till Johnny Slätt och Stefan Olofsson för att ni delat med er av erfarenheter, råd och insikt. Framför allt vill vi rikta en hommage till de respondenter som ställde upp på intervjuer.

Samuel Andersson

Edvard Lindgren

2017-05-26

Abstract

Software development is a unique business area. The rate of change is higher than in any other market and the products are difficult to visualize before the actual making of the product has begun. The traditional methods of software development have been thought to lack the adjustability that the changing climate require. These methods and models were considered far too static for the software development business and as a response The Agile methods was created.

The agile methods are based on flexibility, adjustability and incremental software development. These methods have been widely accepted in the IT and software development business. The most common of this collection of methods is Scrum. Scrum is said to be applicable to any type of project and any type of software development. Despite this, research tells that the method is rarely used as it is described.

The purpose of the study was to investigate why a method that is said to fit all types of software development is adapted to the individual organization and how this adaption is done. The result of the study shows that Scrum is adapted by removing certain roles, aspects and elements that are perceived as time-wasting and complicates the maintenance work of existing software. Scrum is often combined with Kanban, another agile method, that fits the purpose of solving critical malfunctions.

Keywords: Scrum, Agile methods, Adaption, Software development

Sammanfattning

Mjukvaruutveckling är ett unikt svårt område. Dels på grund av den höga förändringstakten inom branschen, men även då en slutprodukt är svår att visualisera innan utvecklingen av produkten har påbörjats. Traditionellt har utvecklingen av mjukvara skett i projektformer som ansetts sakna den anpassningsförmåga som krävs i det föränderliga klimatet. Som svar på dessa svårstyrda metoder skapades de agila metoderna.

De agila metoderna bygger på lättörlighet, anpassningsbarhet samt inkrementell utveckling av mjukvara och har anammats på bred front inom IT- och mjukvarubranschen. Scrum är den vanligast förekommande agila metoden och förespråkas vara tillämplig på alla typer mjukvaruutveckling. Trots detta finns forskning som visar att metoden sällan används i sin helhet utan anpassas.

Studien hade som syfte att undersöka varför en metod som passar för alla typer av mjukvaruutveckling anpassas till den enskilda organisationen och hur denna anpassning ser ut. Studiens resultat är att Scrum anpassas genom att välja bort vissa roller, aspekter och moment. Detta för att dessa ses som överflödiga och försvårar underhållsarbetet med en befintlig mjukvaruprodukt. Scrum kombineras även med Kanban för att lösa akuthantering av problem vilket skapar en metod som påminner om båda metoderna.

Nyckelord: Scrum, Agila metoder, Anpassning, Mjukvaruutveckling

Innehållsförteckning

1 Inledning	1
1.1 Bakgrund	1
1.1.1 Traditionella och agila projektmetoder	1
1.1.2 De agila metoderna	2
1.2 Problemformulering	3
1.3 Syfte	5
1.4 Frågeställningar	5
1.5 Avgränsningar	5
2 Scrum	6
2.1 Metoden i översikt	6
2.2 Scrums roller	7
2.2.1 Produktägare	7
2.2.2 Scrum Master	7
2.2.3 Utvecklingsteam	7
2.3 Product Backlog	8
2.4 Sprintplanering	9
2.4.1 Sprintmål	9
2.4.2 Estimeringar och velocity	9
2.5 Sprintar	9
2.5.1 Daily Scrum	10
2.5.2 Definition of Done	10
2.6 Sprintavslut	10
2.6.1 Sprint Review	10
2.6.2 Sprint Retrospective	11
3 Tidigare forskning	12
3.1 Avvikelser från Scrum	12
3.1.1 Diffusa roller	12
3.1.2 Team	13
3.1.3 Självorganisering i team	13
3.1.4 Semi-funktionella team	13
3.1.5 Sprintar	13
3.1.6 Estimeringar och velocity	14
3.1.7 Sprint reviews och retrospectives	14
3.2 Att anpassa metoder	15
3.2.1 Contingency factors	16
3.2.2 Method engineering och metodfragment	16
3.2.3 Kombinationer med Scrum	17
3.3 Agila metoder i organisationer	18
3.3.1 Projektmetoder	18
3.3.2 Kritik mot agila metoder	19
3.3.3 Agila metoders uppkomst ur ett annat perspektiv	19
3.3.4 Agila metoder som organisationsrecept	20
4 Metod	21
4.1 Metodval	21
4.2 Undersökningsdesign	21
4.3 Urval	21
4.4 Intervjuer	23
4.4.1 Genomförande	23
4.4.2 Etik	24
4.4.3 Intervjuguide	24

4.5 Analys av empiri.....	25
4.6 Trovärdighet & pålitlighet	25
5 Resultat och analys	27
5.1 Respondenterna.....	27
5.1.1 Respondent AF	27
5.1.2 Respondent HZ	27
5.1.3 Respondent HÅ.....	27
5.1.4 Respondent MT	28
5.2 Team.....	28
5.2.1 Storlek.....	28
5.2.2 Roller	28
5.2.3 Självorganisering och arbetsuppgifter	29
5.2.4 Tvärfunktionalitet i teamen.....	29
5.2.5 Produktägarens involvering.....	29
5.2.6 Analys	30
5.3 Sprintplanering	31
5.3.1 Sprintmål.....	32
5.3.2 Estimeringar.....	32
5.3.3 Analys	33
5.4 Sprintar	33
5.4.1 Daily Scrum	34
5.4.2 Förändringar i sprinten	34
5.4.3 Definition of Done	35
5.4.4 Analys	36
5.5 Sprintavslut.....	36
5.5.1 Sprint Review	36
5.5.2 Sprint Retrospective.....	37
5.5.3 Analys	38
5.6 Användningen av Scrum	39
5.6.1 Anpassningen av metoden	39
5.6.2 Metodens applicerbarhet.....	40
5.6.3 Övergången till Kanban	40
5.6.4 En styrande metod	41
5.6.5 Synen på Scrum	42
5.6.6 Sammanfattning.....	42
6 Slutsatser.....	45
6.1 Undersökningens syfte	45
6.2 Hur Scrum används	45
6.3 Varför Scrum används och anpassas	46
7 Slutdiskussion.....	48
7.1 Kritik till metoden.....	48
7.2 Förslag till vidare studier	49
Referenser	50
Bilagor	54
Intervjuguide	54

Figurförteckning

Figur 1: Översikt av Scrum – Författarnas modell	6
Figur 2: Sprintprocessen – Författarnas modell.....	9

Tabellförteckning

Tabell 1: Respondentbeskrivning	22
Tabell 2: Sammanställning analys av Team	30
Tabell 3: Sammanställning analys av Sprintplanering.....	33
Tabell 4: Sammanställning analys av Sprintar.....	36
Tabell 5: Sammanställning analys av Sprintavslut	38
Tabell 6: Sammanfattning av analys.....	39

1 Inledning

I detta kapitel presenteras bakgrunden till ämnet vilket vidare mynnar ut i en problemformulering, syftet med uppsatsen samt dess frågeställningar och avgränsningar.

1.1 Bakgrund

IT-branschen växer och informationsteknik sträcker sig idag långt in i många olika branscher. Teknik och mjukvara har i många avseenden förändrat människors sätt att kommunicera, skaffa sig information och leva. Den ökade teknologiska förändringstakten innebär däremot även utmaningar idag för företag. En allt mer dynamisk företagsmiljö där krav hela tiden ändras gör att organisationer i högre grad måste anpassa sig till sin omgivning (Raval & Rathod 2014). Organisationers förmåga att reagera inom ett affärsmissigt landskap i ständig förändring, är viktigare idag än någonsin tidigare (Pries & Quigley 2011, s. 2).

Den teknologiska förändringen inom mjukvara är snabbare än inom många andra områden (Stepanek 2012, s. 8). Stepanek definierar mjukvara som både komplext och abstrakt, särskilt för andra än utvecklare själva. Detta innebär svårigheter med att efterfölja fasta krav när man utvecklar en produkt inom mjukvara då man inte kan visualisera produkten innan denna står klar. Denna aspekt gör att projekt inom mjukvaruutveckling blir unikt svårt. (ibid.)

1.1.1 Traditionella och agila projektmetoder

Traditionellt sett har projektmetoder såsom vattenfallsmodellen, code-and-fix och spiralmodellen använts för mjukvaruutveckling (Misra et al. 2012). Dessa metoder fick med tiden allt mer kritik av utvecklare då de ansågs vara restriktiva, tidskrävande och stagnera utvecklingsprocessen. Modellerna innefattade tunga processer, fasta krav och hade tydligt fokus på dokumentation. (Misra et al. 2012). Dessutom ansågs de vara svåra att implementera i praktiken (ibid.). De traditionella metoderna ansågs sakna den anpassningsförmåga som organisationer behövde på en alltmer föränderlig marknad (Stoica et al. 2016).

Som svar på de svårstyrda och komplexa traditionella projektmetoderna skapades agila metoder (Tonquist 2016, s. 33). Dessa metoder innefattar flera olika arbetssätt som bygger på flexibilitet och har som syfte att hantera ospecificerade och föränderliga krav för att

kunna utveckla mjukvara inom kort tid (Campanelli & Parreiras 2015). Flexibiliteten i metoderna är också anledningen till att grundarna valde att använda ordet Agile, vilket på svenska kan översättas till vig, rörlig eller smidig (Stepanek 2012, s. 133).

Agila metoder baseras på de värderingar och principer som presenteras i The Agile Manifesto, eller det agila manifestet (Misra et al. 2012; Williams 2012; Agile Alliance 2001). Fokus ligger på att reagera på ständiga förändringar och en balansgång mellan både flexibilitet och stabilitet, istället för att följa en i förväg bestämd plan (Tonnquist 2016, s. 33). Det agila manifestet lyder enligt följande:

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; and

Responding to change over following a plan.

That is, while there is value in the items below (not bold), we value the items above (bold) more.

(Agile Alliance 2001)

De agila metoderna har framförallt två saker gemensamt, de är lättroliga och iterativa vilket innebär att fokus framförallt flyttats från dokumentation och fasta krav till att fokusera på flexibilitet och samarbete inom team men även externt mot kund (Misra et.al. 2012; Williams 2012).

Traditionella utvecklingsprojekt hade fokuserat på att följa väldefinierade planer och detaljerad dokumentation (Misra et al. 2012). Det agila manifestets grundare ansåg att detta sätt att arbeta innebar att tiden som lades på planering blev bortkastad eftersom att de traditionella projektmetoderna inte innefattade erfarenheter och kunskap som uppstår under arbetets gång (Tonnquist 2016, s. 33).

1.1.2 De agila metoderna

Exempel på agila metoder är Scrum, Kanban, XP, Crystal och Feature-Driven Development (Dingsøy et al. 2012). Scrum är i dagsläget den mest använda varianten tillsammans med hybridvarianter som exempelvis Scrum/XP och Scrumban (VersionOne.com). Scrum utvecklades av Ken Schwaber och Jeff Sutherland och innebär

att arbetet delas upp i steg där fokus ligger på att inkrementellt sammanställa en färdig produkt (Schwaber & Sutherland 2013).

I traditionella mjukvaruutvecklingsprojekt är kundens roll begränsad till kravspecificering i början och feedback mot slutet av projektet. Begränsat samspel sker också mellan kunden och utvecklingsteamet. Agila metoders ramverk sätter i stället kundens samarbete i centrum för framgången för projektet. Här är kundens roll utvidgad genom hela processen till att omfatta involvering i själva arbetet med exempelvis funktioner, uppdatering av krav och snabb, regelbunden feedback till utvecklarna (Hoda et al. 2010).

Sedan det agila manifestet presenterades har IT- och mjukvaruindustrin anammat agila metoder (Campanelli & Parreiras 2015) och de har i det närmaste kommit att bli normen för hur arbete ska bedrivas inom utveckling av mjukvara (Danielsson 2016).

1.2 Problemformulering

Agila metoders fördelar presenteras ofta i termer av ökad kvalitet och produktivitet (Campanelli & Parreiras 2015). Enligt Papatheocharous & Andreou (2014) är den mest betydande fördelen möjligheten att hantera och prioritera när kundens krav ändras.

Däremot kan implementering av agil metodik innebära friktion inom organisationer på grund av den befintliga organisationskulturen (Campanelli & Parreiras 2015) eller förändringsmotstånd inom företaget (Papatheocharous & Andreou 2014). Samtidigt står och faller resultatet ofta på hur pass involverad beställaren är i projektet, vilket utretts av Hoda et al. (2010). De fann att bristande involvering av beställaren var en av de största orsakerna till att agila projekt misslyckades (Hoda et al. 2010). Andra orsaker kan härledas till brist på kunskap eller erfarenhet av agila metoder (Papatheocharous & Andreou 2014). Det finns även ett starkt behov av att hela organisationen anammar agilitet, även på managementnivå. (Campanelli & Parreiras 2015).

Papatheocharous & Andreou (2014) genomförde en större studie på det praktiska användandet av agila metoder på en global skala för att kartlägga spridningen av agila metoder och dess tillämpning. De fann att skräddarsydda och företagsspecifika agila metoder, utöver Scrum, var överrepresenterade (Papatheocharous & Andreou 2014).

Att metoderna anpassas kan bero på organisationers specifika behov och begränsningar. (Campanelli & Parreiras 2015). Företag saknar enligt Conboy & Fitzgerald (2010)

kunskaper om hur denna anpassning bör gå till. Många företag gör stora investeringar i att implementera agil metodik (Conboy & Fitzgerald 2010; Campanelli & Parreiras 2015) men genomförandet kan bli så sporadiskt och osammanhängande att det överges. Enligt Conboy & Fitzgerald (2010) finns det därför en efterfrågan om hur man bäst kan skräddarsy metoderna och hur man applicerar vald metod på företaget.

Tidigare forskning har i stor utsträckning fokuserat på själva införandet av agila metoder, viss forskning saknas därför på organisationer som använder sig av metoderna och som gjort det under en längre period (Dingsøy et al. 2012). Forskning kring hur företag antar, anpassar och använder de agila metoderna är ett forskningsområde som presenterats av Ågerfalk et al. (2009) som ett av de mest efterfrågade (Ågerfalk et al. 2009).

Det debatteras om huruvida agila metoder verkligen bör anpassas eller användas i sin helhet (Fitzgerald et al. 2006; Conboy & Fitzgerald 2010). Conboy & Fitzgerald (2010) beskriver att förespråkare menar att agila metoder behöver följas på ett strikt sätt för att de ska anses fungera. Andra menar att delar av metoderna kan tas bort, kombineras eller ersättas utefter eget tycke och behov (Fitzgerald et al. 2006). Conboy & Fitzgerald (2010) menar att de agila metoderna borde vara flexibla i sig. En agil metod ska för det första kunna hantera förändrade krav men bör också i sig själv vara flexibel för att kunna utstå anpassning. Tidigare forskning har i mindre mån fokuserat på just anpassningen (Conboy & Fitzgerald 2010).

Scrum är den agila metod som har fått störst fotfäste (VersionOne.com) och har nått en så pass stor utbredning att empiriskt underlag med fördel kan samlas in (Eloranta et al. 2015). Metoden är konkret beskriven med dess tekniker, aktiviteter och roller. Dessutom anses Scrum kunna användas på alla typer av projekt och produkter, enligt dess förespråkare (ScrumAlliance.org). Däremot används Scrum inte alltid på samma sätt som det finns beskrivet (Eloranta et al. 2015) och har visat sig kunna skräddarsys (Fitzgerald et al. 2006) vilket enligt Scrums upphovsmän innebär att det i så fall inte alls är Scrum man använder (Schwaber & Sutherland 2013, s. 16).

Bennett & Rajlich (2000) beskriver att all mjukvara går igenom en livscykel. Från nyutveckling som efter en initial lansering går över till att handla om vidareutveckling, vidare till service, utfasning och slutligen avveckling. De agila metoderna har inte bara

anammats inom nyutveckling av mjukvara, utan har även fått spridning inom vidareutveckling och underhållsarbetet av befintliga mjukvaruprodukter (Heeager & Rose 2015). Scrum som metod är avsedd för utveckling av nya produkter men går även att dra nytta av vid användning under vidareutvecklingen. Dock finns skillnader mellan ny- och vidareutveckling vilket gör att metoden kan komma att behöva anpassas utefter den befintliga situationen (ibid.).

Förespråkare av Scrum menar att metoden går att applicera på alla typer av kontexter, situationer och projekt. Trots detta finns fenomenet med anpassade agila metoder. Tidigare forskning har till stor del kretsat kring implementation, men desto mindre kring anpassning av agila metoder (Conboy & Fitzgerald 2010). Undersökningen ämnar därför att studera hur den agila metoden Scrum anpassas till företags kontext och varför det sker.

1.3 Syfte

Utifrån problemdiskussionen är det av intresse att undersöka vilka aspekter som skiljer hur Scrum används i praktiken mot hur metoden beskrivs i teorin. Utöver detta ämnar studien att undersöka hur företagen ser på Scrum som en användbar metod och varför den behöver förändras för att passa dem. Syftet är således att ge författarna en fördjupad kunskap för hur och varför IT-företag väljer att använda och anpassa den agila metoden Scrum.

1.4 Frågeställningar

- Hur används den agila metoden Scrum i praktiken?
- Varför väljer de undersökta företagen att använda samt anpassa Scrum?

1.5 Avgränsningar

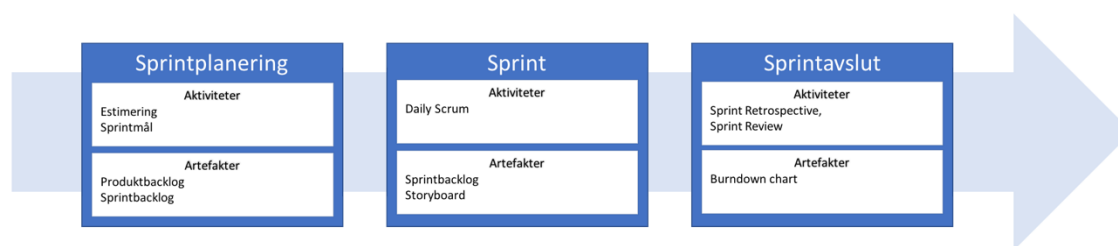
Undersökningen är avgränsad till företag i Sverige som utvecklar och underhåller en befintlig mjukvaruprodukt och på uppdrag av den egna organisationen. Vidare avgränsas studien till företag som arbetar med en egen variant av Scrum. Undersökningen strävar därför mot att studera hur team inom företag arbetar med en egen variant av Scrum som en ”in-house-metod” och varför den formats på det sätt den gjorts utan större extern påverkan.

2 Scrum

I avsnittet presenteras de roller, aktiviteter, artefakter och regler som används i Scrum med utgångspunkt i Schwaber & Sutherlands beskrivning av metoden.

2.1 Metoden i översikt

Ken Schwaber och Jeff Sutherland (2013) har skapat Scrum och de var även närvarande vid det agila manifestets grundande år 2001 (Williams 2012). Metoden hade vid denna tid fått fotfäste sedan Sutherland utvecklat metoden 1993 under sin tid på Easel Company (Lei et al. 2015). De har fortsatt att utveckla konceptet sedan dess men menar att de grundläggande beståndsdelarna är oförändrade (Schwaber & Sutherland 2013). Scrum är idag den mest utbredda agila metoden (VersionOne.com) men är däremot ett relativt outforskat område rent empiriskt jämfört med agil mjukvaruutveckling i övrigt (Lei et al. 2015).



Figur 1: Översikt av Scrum – Författarnas modell

Scrum bygger på att arbete sker i upprepningar för att kunna dra nytta av lärdomar under projektets gång så snabbt som möjligt. (Schwaber & Sutherland 2013). Gemensamt för agila metoder är att utveckling av mjukvara sker i iterationer (Stepanek 2012, s. 68). Dessa iterationer kan ses som delprojekt vilka inom Scrum kallas för sprintar och är ett bestämt tidsspann i vilket en del av produkten färdigställs och ska kunna levereras (Schwaber & Sutherland 2013). Antalet krav som utvecklas i en sprint är upp till teamet. Detta bestäms under den sprintplanering som hålls inför respektive sprint. När sprinten nått sitt slut hålls sedan två aktiviteter för att utvärdera sprinten, Sprint Retrospective samt Sprint review. I och med dessa aktiviteter utvärderar teamet inte bara sprinten utan även sig själva och kan på så sätt hitta förbättringsområden. (ibid.). Processen kan delas upp i tre delar vilket illustreras i figur 1.

2.2 Scrums roller

Ett team ska enligt Scrum vara tvärfunktionellt genom sin uppbyggnad. Med detta menas att kunskaper och kompetenser inom olika områden används i teamet (ibid.). Ett Scrum-team är dessutom självorganiserade. Med detta menas att teamet bestämmer över arbetsgång och uppgifter. Således styrs inte teamet av några utomstående parter. (ibid.) Scrum utgörs av ett antal roller vilka består av en Produktägare, en Scrum master och ett Utvecklingsteam.

2.2.1 Produktägare

Denna person är antingen en extern kund eller en beställare från det egna företaget (Softhouse 2014) och ansvarar för kravspecifikationer (Permana 2015). Produktägaren är ensamt ansvarig för att hantera Product Backlog, en lista på krav och funktioner för produkten. Viktigt i Scrum är att produktägaren alltid ska utgöras av en enda person (Schwaber & Sutherland 2013). Det är endast denna person som styr vilket arbete som utförs av utvecklarna och ingen annan än produktägaren får ge direktiv till dessa (ibid.).

2.2.2 Scrum Master

En Scrum Master har som uppgift att dels se till att de praktiker och regler som presenteras i Scrum efterlevs och förstås av hela teamet, dels att sköta kommunikationen mellan team och externa parter (ibid.). Rollen innebär även att leda utvecklingsteamet och att skapa så goda förutsättningar för arbetets fortgång som möjligt (Schwaber & Sutherland 2013; Softhouse 2014). Detta kan innefatta ett medlande vid konflikter i teamet samt ett ansvar för att anskaffa de resurser som teamet behöver (Pries & Quigley 2011, s. 51).

2.2.3 Utvecklingsteam

Utvecklingsteamet består av utvecklare med olika specialistkunskaper och kan beskrivas som själva arbetslaget. (Schwaber & Sutherland 2013). Medlemmarna i teamet bestämmer själva hur arbetet ska bedrivas men fördelar även arbetsuppgifter mellan varandra (Softhouse 2014). Detta går så långt som att inte ens Scrum Master har befogenhet att bestämma hur arbetet utförs (Schwaber & Sutherland 2013).

Trots att teamet är tvärfunktionellt undantas individuella titlar i Scrum. Rollerna är inte fasta utan alla benämns som Utvecklare oavsett arbetsuppgifter (ibid.). Tanken är här att alla medlemmar ska kunna byta arbetsuppgifter med andra i teamet på ett smidigt sätt (Softhouse 2014).

Enligt Schwaber & Sutherland (2013) ska mängden utvecklare i ett team ligga mellan tre till nio personer för att teamet ska vara så smidigt som möjligt samtidigt som alldeles för få personer innebär att arbetsbördan blir för tung. I Scrum är därför detta den optimala storleken. Utvecklingsteamet är dessutom kollektivt ansvariga och ska arbeta som en helhet. (ibid.).

2.3 Product Backlog

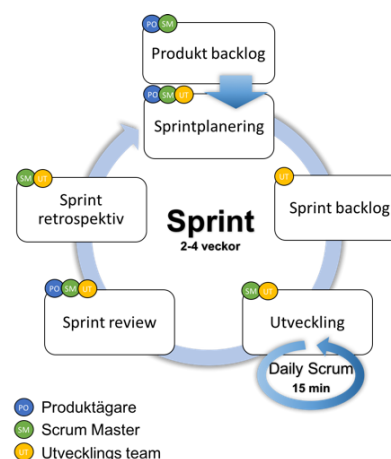
Alla funktioner och krav för produkten men även förbättringar och förändringar som ska göras samlas i Produktbackloggen. Således fungerar denna som en loggbok eller lista med arbetsuppgifter för Utvecklingsteamet och kan även beskrivas som en kravlista för funktionalitet. (ibid.). Den fungerar även som en prioriterad uppställning över mål som ska åstadkommas (Softhouse 2014) och fungerar därmed även som en vision för teamet för vad som ska uppnås. Produktägaren är ansvarig för backloggen vilket inkluderar dess innehåll, ordning och hur tillgänglig den är för andra (Schwaber & Sutherland 2013). Detta för att produktägaren företräder projektet och kommunikationen mellan teamet och projektets utomstående intressenter och kunder (Kautz et al. 2014).

Produktbackloggen är från början en grov estimering av kända krav och utvecklas allt eftersom i takt med utvecklingen av produkten. Backloggen är på detta sätt dynamisk (Lei et al. 2015) och ska kunna förändras hela tiden utifrån hur marknaden förändras och andra påverkningar, så att den ska kunna användas så effektivt som möjligt (Schwaber & Sutherland 2013). Produktbackloggen tas därför fram både utefter tekniska krav men även efter produktägarens vision och de affärsmässiga mål som finns för produkten (Lei et al. 2015).

Inom Scrum representeras krav på funktionalitet som *items* (Schwaber & Sutherland 2013) eller *user stories* (Pries & Quigley 2011, 18). Dessa fungerar som en kort beskrivning av hur objektet används (Cohn 2010, s. 239). En user story ska vara tillräckligt enkelt beskriven för att få rum på en post-it-lapp vilka ofta används för att fysiskt visualisera dessa krav (Pries & Quigley 2011, 18). Utvecklingen av varje user story visualiseras sedan med hjälp av en *story board* för att följa deras olika steg i utvecklingen. På så sätt skapas en enkel överblick över projektets status och vad som är klart respektive återstående arbete (Pries & Quigley 2011, 18).

2.4 Sprintplanering

Scrum Master håller detta möte tillsammans med hela teamet och utarbetar en plan över arbetet i sprinten. Planeringen består av att bestämma hur mycket från backloggen som kan avklaras inom sprintens tid. Detta blir till en Sprint backlog där det tydligt kan ses vad som ska avklaras inom sprinten. Hur mycket som kan åstadkommas baseras på en estimering över teamets kapacitet utifrån erfarenheter från tidigare sprintar. (Schwaber & Sutherland 2013).



Figur 2: Sprintprocessen – Författarnas modell

2.4.1 Sprintmål

Utifrån sprintplaneringen utformas ett sprintmål vilket utgörs av vad som ska göras under sprinten för att ge teamet ett gemensamt mål och syfte (ibid.). Sprintmålet innebär samtidigt att förändringar inte kan ske i sprinten eftersom detta då påverkar om målet uppnås eller ej (ibid.). På så sätt förpliktigas teamet att genomföra de arbetsuppgifter som fastställs i sprintplaneringen. (Cohn 2010, s. 280).

2.4.2 Estimeringar och velocity

Estimeringen sker genom att dela ut poäng till varje arbetsuppgift för att på så sätt kunna skilja dessa i storleksordning. Detta sker av utvecklarna kring hur dessa funktioner eller krav ska estimeras (Pomar et al. 2014). Denna estimering ska ske av utvecklarna själva (Schwaber & Sutherland 2013) och ligger till grund för hur många arbetsuppgifter som kan åstadkommas i en sprint (Pries & Quigley 2011, s. 29).

Hur många estimeringspoäng som teamet klarar av att slutföra under en sprint mäts vanligen i *velocity* (Pomar et al. 2014). Teamets hastighet kan användas både som prognos inför hur mycket arbete som kan åstadkommas i framtida sprintar och för att upptäcka om teamet ligger i fas med hur snabbt man tidigare arbetat (ibid.).

2.5 Sprintar

Viktigt inom Scrum är att sprintar ska vara konsekventa och inte får fluktuera och dessutom inte sträcka sig längre än en månad (Schwaber & Sutherland 2013). Ett vanligt tidsspann för en sprint är mellan två och fyra veckor enligt Eloranta et al. (2015). Upplägget för en sprint illustreras i figur 2.

För varje sprint väljs ett antal objekt från produktbackloggen för den specifika sprinten och blir till en lista med uppgifter som ska genomföras. Detta kallas då för en sprintbacklogg och formar en plan för hur detta steg i utvecklingen kommer att se ut (ibid.). Dessa arbetsuppgifter ska inte ändras under sprintens gång utan fastställs i planeringen av sprinten. Förändringar som påverkar sprinten påverkar också om sprintens mål uppfylls eller ej och teamet bör därför undvika sådana rubbningar (Schwaber & Sutherland 2013).

2.5.1 Daily Scrum

Daily Scrum är ett dagligt möte och är tidsbegränsat till 15 minuter (Schwaber & Sutherland 2013). Medlemmarna i teamet samlas för att gemensamt gå igenom sitt tidigare arbete och vad som ska göras för dagen (Kautz et al. 2014). Utvecklingsteamet synkroniserar således sina arbetsuppgifter för att säkerställa att sprintens utveckling går åt rätt håll och att sprintmålet kommer att nås (Schwaber & Sutherland 2013). Medlemmarna tar även upp eventuella problem som hittills inte fått någon lösning, för att arbetet ska kunna fortgå obehindrat (Kautz et al. 2014). Syftet är också att förbättra kommunikation och samarbete inom teamet. Mötet ska hållas på samma tid varje dag vilket Utvecklings-teamet sköter på egen hand men med Scrum masters överseende. (Schwaber & Sutherland 2013).

2.5.2 Definition of Done

I Scrum ska teamet ha en gemensam definition av en färdigställd arbetsuppgift. När ett objekt beskrivs som ”Done”, det vill säga klar, måste konsensus finnas inom teamet vad detta innebär (ibid.). Denna delade uppfattning avgör när arbetsuppgifter kan klassificeras som klara (ibid.). Enligt Pomar et al. (2014, s. 778) innebär definitionen att arbetsuppgiften är ”kodad, testad och godkänd av produktägaren”.

2.6 Sprintavslut

Sprintavslutet delas upp på två möten nämligen Sprint Review och Sprint Retrospective. Dessa möten är till för att presentera, utvärdera samt reflektera över det arbete som har utförts.

2.6.1 Sprint Review

Syftet med Sprint Review är att utvärdera det arbete som utförts (Schwaber & Sutherland 2013). Mötet innefattar ofta en form av demonstration för produktägaren och eventuella externa intressenter (Lei et al. 2015). Dessa kan innefatta exempelvis en extern kund eller

ledning och användare (Softhouse 2014). Produktägaren styr mötet och Utvecklingsteamet presenterar vad som gjorts och demonstrerar en körbar version av produkten. Syftet med Sprint Review är också att bestämma och prioritera arbete för nästkommande sprint. Detta sker bland annat genom att produktägaren uppdaterar teamet med eventuella förändringar för produkten (Pham & Pham 2012, s. 10).

2.6.2 Sprint Retrospective

Till skillnad från en Sprint Review innebär Sprint Retrospective att teamet möts för att diskutera vad som kan förbättras vad gäller samarbete och för att granska det arbete som har gjorts. Detta för att lägga upp en plan inför uppstarten och planeringen av nästa sprint över vad som kan förbättras. (Schwaber & Sutherland 2013). Syftet är framförallt att effektivisera lagarbetet genom att överlägga vad som gått bra under sprinten, respektive mindre bra (Pham & Pham 2012, s. 11).

3 Tidigare forskning

I avsnittet beskrivs den forskning som tidigare har bedrivits inom ämnet och som ligger till grund för analysen av studiens insamlade data.

3.1 Avvikelser från Scrum

Avvikelser i Scrum har i tidigare forskning kallats för ScrumButs. Detta kan beskrivas som orsaker för att inte använda Scrum på det sätt som är beskrivet (Pham & Pham 2012, s. 146). Begreppet får en enklare översättning när det sätts in i ett sammanhang som exempelvis kan vara ”Vi arbetar i sprintar *men...*”. Denna typ av avvikelser leder till negativa konsekvenser enligt Pham & Pham (ibid.). Eloranta et al. (2015) har beskrivit dessa typer av avvikelser i Scrum som *anti-patterns*. Att inte följa Scrums verktyg och tillvägagångssätt förklaras som att detta ses som lämpligt för företaget. Eloranta et al. (2015) fann däremot att dessa avvikelser i längden får en negativ effekt. Avvikelserna gör att det finns en risk i att målen och syftet med Scrum inte uppfylls.

3.1.1 Diffusa roller

I den forskning som gjorts inom avvikelser i Scrum är framförallt rollerna i metoden utbytta eller utgör en form av flexibel roll. I vissa fall är rollerna som Scrum Master eller produktägare helt borttagna ur metoden och ersatta med en annan form av titel (ibid.). Även Permana (2015) fann att Scrum Master är en mer flytande roll som i vissa fall hanteras av vad man kallar en variant av Product manager (ibid.). Eloranta et al. (2015) beskriver också hur många företag inte använder sig av en produktägare alls, att rollen utgjordes av flera personer eller att samma produktägare delades mellan flera team.

Pham & Pham (2012, s. 149) beskriver hur man kan anpassa rollen som Scrum Master till att delas mellan individer i teamet men att det då krävs att dessa personer är väl införstådda med Scrum. I de fall som organisationer försöker kombinera produktägare och Scrum Master till en enda roll krävs att denna person separerar sitt agerande i de olika rollerna. Pham & Pham (ibid., s. 152) beskriver denna avvikelse som en av de mest riskfyllda då det lätt kan skapa förvirring i teamet. En Scrum Master ska enligt Scrum coacha teamet medan en produktägare står för kravspecifikationer (Schwaber & Sutherland 2013).

3.1.2 Team

Scrum Masters roll innebär bland annat att skydda Utvecklingsteamet från externa störningar eller avbrott under sprintens gång (Schwaber & Sutherland 2013). Enligt Eloranta et al. (2015) är det däremot vanligt förekommande att detta ändå sker. De fann att i många fall ingrep utomstående i projektet vilket gjorde att teamet hade svårt att utföra sitt arbete. (ibid.).

3.1.3 Självorganisering i team

I den undersökning som Eloranta et al. (ibid.) utförde framkom att endast hälften av de undersökta teamen var självorganiserade på ett sådant sätt att varje utvecklare själv väljer arbetsuppgifter och att de inte delegeras ut genom teamet. Den andra hälften utgjorde teams som blev tilldelade arbetsuppgifter från en product manager.

3.1.4 Semi-funktionella team

I Scrum bör teams vara tvärfunktionella vilket innebär att alla kompetenser som behövs för att utveckla produkten finns i teamet. Avvikelser från denna aspekt innebär bland annat att företag väljer att låta olika team utveckla vissa delar av produkten. Dessa teams är speciellt utformade för att hantera en specifik del av produkten utifrån expertis. Detta innebär enligt Eloranta et al. (ibid.) att utvecklingen av produkten avstannar. Detta på grund av att testning sker av annan personal än utvecklarna själva. Om expertis inom testning och design finns i andra team innebär detta samtidigt att mycket tid läggs på kommunikation mellan dessa teams (ibid.).

3.1.5 Sprintar

Vissa företag experimenterar med sprintarnas längd för att sedan hamna på en längd som bäst passar teamet (ibid.). Eloranta et.al. (ibid.) fann att korta sprintar på under två veckor gjorde att sprintplanering och sprint reviews tog upp alldeles för mycket av sprinten för att något värdefullt arbete skulle kunna utföras. Istället var det 2-veckors-sprintar som både var mest vanliga och dessutom visade sig åstadkomma bäst resultat (ibid.).

Trots Scrums rekommendation på att all mjukvara ska testas inom sprinten använder vissa företag 4-veckors sprintar som delas till två sprintar. Den första sprinten innefattar utveckling och den andra sprinten utgör en testfas som utförs av ett separat team (Eloranta et al. 2015). Pham & Pham (2012, s. 151) menar att detta sätt att utföra testning innebär att man går ifrån den agila principen om att testning ska ske fortlöpande. Eloranta et al. (ibid.)

menar att detta innebär problem då eventuella buggar och andra problem måste åtgärdas långt senare än när koden skrevs. Detta kräver extra arbete för utvecklarna att åtgärda då en tid gått sedan funktionaliteten implementerades samt att inget nytt värde kan ges till produkten medan man inväntar testresultatet (ibid.).

3.1.6 Estimeringar och velocity

Pomar et.al. (2014) utförde en fallstudie på implementation av Scrum och dess effekter på sprintarnas variation i genomförandegrad baserat på hur snabbt de genomförs, så kallad *sprint velocity*. De fann att teamets frihet kring arbetsuppgifterna gjorde att de konstant misslyckades att leverera deras åtaganden. Endast 15 % av sprintarna utgjordes av sprintar där det planerade arbetet åstadkoms. Detta kan förklaras med en allt för stor mängd arbetsuppgifter samtidigt vilket sker på grund av att arbetsuppgifter är alltför stora för att åstadkommas i en enda sprint (ibid). De flesta av de team som Eloranta et al. (2015) undersökte hade inte någon uttalad hastighet i sina sprintar och visste då inte hur lång tid deras arbetsuppgifter tog att genomföra. Detta leder till svårigheter att få en överblick över i vilken takt produkten utvecklas (ibid).

Arbetsbördan leder till att genomförandet försämras samt att flera av arbetsuppgifterna inte blir klara i sprintens slut (Pomar et al. 2014). Vad Pomar et al. (ibid.) dock fann som roten till problemet med missade åtaganden till uppgifter var att komplexiteten inte togs i beaktning i planeringen av sprintar. Denna komplexitet uppstod på grund av projektet innefattade teknologier som inte bemästrades av alla utvecklare samt att krav inte alltid var tydliga (ibid).

3.1.7 Sprint reviews och retrospectives

Eloranta et al. (2015) menar att avsaknad av konsekvent feedback från beställare eller produktägare innebär att teamet gör egna beslut för produkten, eftersom att de inte vet vad som efterfrågas. Detta kan leda till att mycket arbete behöver göras om. De menar därför att Sprint reviews är essentiellt för att utveckla produkter som är av värde för slutkunden.

De fann även att vissa team hade väldigt korta Retrospective eller inget möte alls vilket enligt Eloranta et al. (ibid.) innebär att det inte skapas någon förbättring i teamet. På så sätt kan teamet inte utvecklas och förbättra sin produktivitet när det inte finns någon diskussion eller feedback efter avslutad sprint. Detta är enligt dem ”kärnan i Scrum” och Retrospective bör hållas så ofta som möjligt. (ibid.).

3.2 Att anpassa metoder

Conboy & Fitzgerald (2010) har undersökt hur anpassning av agil metodik kan göras mer effektiv. Metoden som i deras forskning stod i fokus var XP men författarna menar att studien även är applicerbar även på andra agila metoder. I deras undersökning av irländska företag framkom att även om metoden i fråga användes av utvecklarna fanns inte fullständig kunskap om metoden. Conboy & Fitzgerald (ibid.) fann att utvecklare oftast inte hade kunskap om alla tillämpningar i en metod vilket gjorde att metoden skraddarsyddes på grund av att de delar som bedömdes som mer utmanande förbisågs. (ibid.). Således beror anpassning enligt forskarna till viss del på en brist på förståelse för metodens alla delar och processer.

Conboy & Fitzgerald (2010) har även slagit fast att brist på kontroll över hur väl metoden i fråga efterlevs av teamet påverkar anpassningen. Detta kan innebära att metoden formas utefter hur utvecklarna bestämmer hur denna ska användas. I andra fall leder denna brist dock till att metoden överges. Conboy & Fitzgerald (ibid.) menar därför att det finns ett värde i att utvärdera hur väl metoden efterlevs. Om delar av metoden skall behållas eller inte beror då på hur väl den passar teamet och inte på grund av försummelse.

“Lättrörligheten” hos agila metoder har förklarats med att de är mindre regelstyrda och formellt beskrivna än traditionella metoder (Kniberg & Skarin 2009, s. 8). Agila metoders fokus på att vara lättrörliga har gjort att de till viss del setts som improviserade och att utvecklarna fått fria händer (Conboy & Fitzgerald 2010). Andra menar att det snarare är omöjligt att lyckas med agila metoder om dessa inte efterföljs fullt ut och att det krävs disciplin för att de ska fungera i praktiken. (ibid.). Henderson-Sellers & Serour (2005) menar dock att agila metoder ofta blir ett kontrollerande verktyg på grund av sina tydliga regler som är tvungna att följas och tar upp XP som exempel. Då metoden i sig är flexibel och mottaglig för förändringar i omvärlden innebär avsteg från mallen att det inte är lämpligt att benämna detta som XP, enligt dess förespråkare (ibid.). Scrum är mindre deskriptiv än XP men innebär fortfarande fler restriktioner än vissa andra metoder (Kniberg & Skarin 2009, s. 9).

Detta är något som även återfinns i Schwaber & Sutherlands beskrivning av hur Scrum bör användas. De menar att avsteg från metodens beskrivning innebär att det inte är går att benämna detta som Scrum.

“Scrum’s roles, artifacts, events, and rules are immutable and although implementing only parts of Scrum is possible, the result is not Scrum. Scrum exists only in its entirety”
(Schwaber & Sutherland 2013, s. 16).

Henderson-Sellers & Serour (2005) menar att eftersom agila metoder är uppbyggda på detta sätt kan de inte implementeras om man inte följer metodens värderingar och principer. Detta gör att metoderna inte är särskilt flexibla. Conboy & Fitzgerald (2010) menar dock att agila metoder i sig bör vara flexibla för att utstå anpassning och således går att förändra för att passa organisationen.

Anpassning av agila metoder sker genom framförallt två tillvägagångssätt; *contingency factors* och *method engineering* (Fitzgerald et al. 2006).

3.2.1 Contingency factors

Ett tillvägagångssätt för att anpassa agila metoder till företagets kontext innebär att utse en metod utefter ett antal kriterier, så kallade ”contingency factors” (Fitzgerald et al. 2003) och därmed väljs den metod som bäst passar ett projekt (Fitzgerald et al. 2006). Nivån av osäkerhet styr därför i mångt och mycket vilken metod som används och tanken är att denna metod ska kunna plockas fram och användas på ett enkelt sätt (Campanelli & Parreiras 2015). Detta innebär dock att utvecklarna behöver kunna hantera och förstå en rad olika metoder (ibid.), såsom XP eller Scrum. Både Fitzgerald et al. (2003) och Fitzgerald et al. (2006) pekar dock på att detta är svårt då utvecklare ofta inte alltid är fullt insatta i ens en metod.

3.2.2 Method engineering och metodfragment

Medan tillvägagångssättet ur *contingency factors*-teorin innebär att välja en metod bland många möjliga innebär Method engineering att en ny metod byggs utifrån de olika metoderna. Detta sker genom att använda ”metodfragment” från olika typer av projektmetoder för att skapa en metod speciellt framtagen för företagets kontext och utmaningar. (Campanelli & Parreiras 2015; Qumer & Henderson-Sellers 2008).

Fragmenten representerar delar av metoden, såsom Scrums Backlog eller Daily Scrum, och kan väljas ut och kombineras med andra metodfragment från till exempel XP (Qumer & Henderson-Sellers 2008). För detta krävs en metodingenjör (Campanelli & Parreiras 2015; Fitzgerald et al. 2006) som väljer ut lämpliga fragment och sammanställer dessa till en metod som är anpassad (Qumer & Henderson-Sellers 2008).

Meningen är att Method engineering ska skapa en flexibilitet till organisationen genom att välja ut de bästa delarna för att kunna anpassas till kontexter (Fitzgerald et al. 2006). Det är dock komplext att sammanställa en sådan metod (Henderson-Sellers & Serour 2005). Skapandet av en metod genom Method engineering gör organisationen till ägare av vad Campanelli & Parreiras (2015) kallar för en ”in-house”-metod.

3.2.3 Kombinationer med Scrum

Scrum har visat sig kunna kombineras med andra metoder för att skapa hybridvarianter. Sådana kombinationer är till exempel Scrum och XP men även Scrum och Kanban (Fitzgerald et al. 2006; Nikitina et al. 2012; VersionOne.com).

Kanban bygger på principerna av Lean production (Al-Baik & Miller 2015) men kan även klassas som en agil metod enligt Kniberg & Skarin (2009, s. 29). Lean utvecklades initialt inom produktionsverksamhet av Toyota och målet med Lean är att minimera slöseri och göra produktionskedjan så nedbantad som möjligt. Detta ska kunna ”leverera värde till kund mer effektivt”. (Al-Baik & Miller 2015). Kanban är en av Leans mest populära metoder och fungerar som ett verktyg för att kontrollera arbetsprocessen (ibid.). Fokus inom Kanbanteori är att prioritera arbetsuppgifter genom att visualisera arbetsflödet och på så sätt definiera vad som behöver göras och när detta behöver göras. På detta sätt identifieras de viktigaste arbetsuppgifterna utefter prioritet, vilket minimerar risk. (Lei et al. 2015). David Anderson menar att fördelarna med Kanban bland annat är att metoden tydligt visualiserar arbetsprocessen, minimerar pågående arbete (Work in progress) samt att man underlättar hanteringen av arbetsflödet (Anderson 2010, s. 15).

Kniberg & Skarin (2009) beskriver Scrum och Kanban som anpassningsbara metoder. De menar dock att Scrum innebär fler restriktioner än Kanban och att detta innebär färre valmöjligheter (ibid., s. 8). Metoden delar vissa likheter med Scrum, till exempel en visuell bild över arbetsprocessen och möjligheten till mätbart arbete. Trots vissa likheter skiljer sig metoden dock avsevärt. Kanban saknar sprintar, strukturerade möten såsom retrospekt och daily Scrums samt saknar definierade roller. Således är Kanban ”*desto mindre preskriptiv*” än Scrum. (Nikitina et al. 2012).

De likheter som finns mellan Scrum och Kanban gör att metoderna ibland kombineras och att deras metoder ”blandas” och detta kallas då för Scrumban. (Nikitina et al. 2012; Wang et al. 2012). Scrumban beskrivs som en anpassad metod för underhåll av produkter (Wang

et al. 2012). Arbetsuppgifter och fel i programvaran kan åtgärdas fort eftersom att tidsåtgången minimeras. I dessa fall utgör sprintar ett hinder för att snabba ändringar eller fel i programvaran ska åtgärdas (ibid.). Vissa aspekter av Scrum kan dock användas såsom dess möten (ibid.) och tvärfunktionella team (Kniberg & Skarin 2009, s. 24).

Sjøberg et al. (2012) menar att begränsningar i Scrum kan lösas med ett införande av Kanban. Företaget i undersökningen övergick till Kanban därför att Scrum upplevdes som alltför stelbent. Metoden passade inte företagets behov av att underhålla produkten och möten sågs som ”slöseri” som inte behövdes (ibid.). Genom att arbeta med Kanban kunde man hantera de avbrott som orsakades av plötsliga felrapporter och problem. Att Kanban föredrogs var också ett resultat av att arbetsuppgifter underskattades i samband med estimeringar. Detta ledde till att arbete sköts på nästkommande sprint eller inte gjordes alls (ibid.). Detta problem har uppdagats av Nikitina et al. (2012) som en av anledningarna till en övergång från Scrum till Scrumban. Arbetet med Kanban gjorde istället att utvecklarna kunde fokusera på färre uppgifter samtidigt, vilket ökade kvaliteten (Nikitina et al. 2012; Sjøberg et al. 2012).

3.3 Agila metoder i organisationer

3.3.1 Projektmetoder

Christensen & Kreiner (1997 s. 29) menar att projekt ”byråkratiseras” när organisationer strävar efter effektivitet. De menar att en standardisering och teoretisering av projekt leder till den tröghet som byråkratin kan kännetecknas av. På ena sidan finns byråkrati med dess standarder och rutin och att andra sidan behovet av att svara på föränderliga omständigheter. Christensen & Kreiner (1997 s. 29) menar dock att det innebär svårigheter när byråkrati inte snabbt nog kan svara på den turbulens som finns i omvärlden.

Blomberg (2013, s. 54) argumenterar för att välplanerade projekt inte per definition innebär ett lyckat projekt. Istället kännetecknas lyckade projekt enligt Blombergs undersökningar av förhållandevis lite planering och en anpassningsbarhet utefter rådande omständigheter. Planeringen sker då i takt med att projektet pågår (Blomberg 2013, 56). Christensen & Kreiner (1997, s. 70) argumenterar på ett liknande sätt då de menar att projekt är i behov av kontinuerligt lärande under själva processen. De menar att det behövs en ”dynamisk anpassning” av aktiviteterna under projektets gång (Christensen & Kreiner 1997, s. 70).

Projekt sker under osäkerhet vilket gör att det finns en risk med att ”kursen stakas ut” under projektets tidigaste fas (Christensen & Kreiner 1997, s. 33).

3.3.2 Kritik mot agila metoder

Agila metoder skapades för att vara lättroliga och motverka den stelbenthet som traditionella projektmetoder stod för. Enligt Ericksson et al. (2005) definieras agilitet av att undanröja så mycket som möjligt av de tunga processer som de traditionella metoderna stod för. Således bygger agila metoder på att kunna svara på förändrade krav och omständigheter genom att avlägsna det som anses belasta arbetet.

Blomberg (2013, s. 64) framför kritik mot synen på de agila metoderna och menar att de har utvecklats till att alltmer likna de metoder som de skulle ersätta.

De korta flexibla självorganiserade processerna har alltmer kommit att bli formellt strukturerade processer och omfattar alltmer dokumentation och regler.

(Blomberg 2013, s. 64).

Blomberg (2013, s. 62) förklarar detta med att alla projektmetoder bygger på en ”byråkratisk logik”. Detta är paradoxalt eftersom att projektorganisering skulle motverka den stelbenthet som annars fanns inom organisationer och förbättra innovationskraft och flexibilitet. Här återfinns den dragkamp mellan organisering och projekt som Christensen & Kreiner (1997) beskriver. Blomberg (2013, s. 62) menar att organisationer behöver stabilitet och är därmed inte skapta för att ständigt reagera på förändringar.

De agila metodernas överlägsenhet är enligt flera forskare också ifrågasatt då mycket av metodernas påstådda fördelar är anekdotiska (Serrador & Pinto 2015; Dybå & Dingsøy 2008). Serrador & Pinto (2015) menar att det inte finns tillräckligt med forskning som visar på att agila projekt är lyckade i större utsträckning än traditionella projekt. På samma sätt stöds inte påståendet om metodernas applicerbarhet på alla projekt och situationer av empirisk forskning, enligt Dybå & Dingsøy (2008).

3.3.3 Agila metoders uppkomst ur ett annat perspektiv

Blomberg (2013, s. 63) menar att agila metoder skapades ur en maktkamp mellan programmerare och arbetsgivare. Företag ville effektivisera arbetsprocessen genom att standardisera denna och utvecklarna ville behålla sin kompetens och nyckelpositioner. Detta är något som även Görling (2009, s. 67) tar upp som menar att den agila rörelsen delvis består av denna ideologiska åsikt där utvecklarna bör vara direkt involverade i

projektet. Utvecklare har i och med det agila manifestet fått en mer central roll i projekt där man menar att dessa bör ges mer makt. Detta för att det är utvecklarna som sitter på kompetensen att genomföra projektet och därför blir arbetet lidande av att utomstående, exempelvis projektledare, utan sådan kompetens ger utvecklarna en kravspecifikation. Då agil utveckling bygger på experiment och förändringar växer produkten fram organiskt snarare än att det går att fastställa resultatet i för tid (ibid, s. 67). Något som även Stepanek (2012, s. 8) tar upp och menar att mjukvara är svårt att visualisera för utomstående innan denna står klar. Utvecklare har på så sätt fått en större roll i projekt och makten har delvis förflyttats till de som utför arbetet.

3.3.4 Agila metoder som organisationsrecept

Agila metoder och Scrum kan ses som organisationsrecept. Røvik (2000, s. 13) beskriver detta begrepp som en institutionaliserad standard för ”hur man bör utforma delar av en organisation”. Dessa recept ses som en förebild för organisationer då de anses som effektiva och moderna av samtiden (Røvik 2000, s. 13). När organisationer anammar recept erhåller de samtidigt legitimitet i förhållande till den omvärld de befinner sig i. På så sätt kan organisationsrecept ses som bärare av ett symboliskt värde då de inte endast används för att effektivisera företag och organisationer. (Røvik 2000, s. 35).

Recept kan av denna anledning adopteras av organisationer för att de ska upplevas som framåtsträvande och moderna trots att receptet inte alltid är passande för verksamheten. Recept som är typiska för dess tid används då för att ge organisationen legitimitet eftersom att dessa känner ett tryck från omvärlden att inlemma receptet i organisationen. Således skapas en dragkamp mellan att behålla ett, inom organisationen välkänt arbetssätt och att anamma ett nytt som uppfattas som modernt och att på så sätt uppnå legitimitet. (Røvik 2000).

Lösningen på dilemmat kallar Røvik (2000, s. 146) för en *frikoppling* mellan ”prat och praxis” där organisationen anammar receptet men att detta inte påverkar aktiviteterna i någon större mån. Med andra ord upplevs organisationen utåt sett arbeta enligt receptet men organisationen är internt mer eller mindre opåverkad av de nya processerna. Organisationer kan på så sätt fortsätta arbeta enligt inarbetade lösningar och receptet uppnår endast en svag effekt på hur man exempelvis utvecklar produkter och tar beslut. (Røvik 2000, s. 146).

4 Metod

I följande avsnitt presenteras den metod som har använts och de val som har gjorts för studiens tillvägagångssätt.

4.1 Metodval

Studien har genomförts med ett kvalitativt tillvägagångssätt. Kvalitativa metoder har som syfte att bringa förståelse för den sociala verkligheten då metoden är tolkningsinriktad (Bryman & Bell 2005). Detta passar undersökningens syfte då författarna vill nå en fördjupad förståelse för fenomenet anpassning av den agila metoden Scrum. En kvalitativ forskningsansats innebär att fenomenet kan belysas, inte endast med hur metoden anpassas men även varför, utefter respondenternas egna uppfattningar av verkligheten.

4.2 Undersökningsdesign

Undersökningen har genomförts som en flerfallstudie av fyra företag, två i Stockholm och två i Göteborg. Fokus har varit att förstå den sociala verkligheten inom företagen och i synnerhet hur olika team i olika företag arbetar med metoden. Detta för att syftet var att undersöka fenomenet med anpassade versioner av Scrum och inte ett specifikt företag. Insamlingen av data har skett genom kvalitativa intervjuer med personer som har djup inblick i hur ett eller flera team arbetar. Detta för att uppnå en förståelse för hur metoden används i praktiken men även varför de använder den på det sätt som de gör.

4.3 Urval

Valet av respondenterna skedde genom ett subjektivt urval. Respondenterna valdes strategiskt efter i vilken mån de ansågs kunna ge relevant data till undersökningen (Denscombe 2016). Inledningsvis eftersöktes personer i rollen som Scrum Master, dock fanns svårigheter att hitta personer som endast agerade som Scrum Master vilket ledde till att urvalet breddades. Det viktigaste var att respondenterna skulle ha djup inblick i hur ett eller flera team praktiskt arbetar med Scrum. Teamen skulle arbeta med vidareutveckling av en befintlig produkt samt med en intern beställare eller produktägare. Detta var av stor vikt då uppsatsen har som syfte att undersöka varför Scrum anpassas i en organisation och inte bara hur det sker. Företagen har därmed anpassat metoden utefter egna preferenser utan att ha blivit påverkade av en extern kund. På så sätt kunde fokus ligga på vilket sätt metoden har formats och skapats utifrån teamens utformning och förutsättningar. Respondenterna

kommer således från företag som skapat en egen ”in-house”-metod som var av intresse för undersökningen.

Genom personliga kontakter nåddes tre av intervjupersonerna som efter viss korrespondens valde att ställa upp på en personlig intervju. Kontakt skedde inledningsvis via email där uppsatsens ämne och syfte presenterades. Det var även genom denna kontakt som författarna kunde säkerställa personens roll inom företaget samt huruvida de arbetade med Scrum. De svar författarna fick från dessa respondenter var att de alla arbetade med en variant eller en form av Scrum vilket ansågs kunna bidra till studien eftersom det just var denna typ av skräddarsydda metod som ligger till grund för undersökningen. Således söktes inte specifikt efter företag som sade sig arbeta med Scrum och som ansåg sig göra det utan respondenterna har från början varit medvetna om att de arbetar med en anpassad version av Scrum.

Förfrågningar skickades även ut genom LinkedIn.com. Detta skedde genom sidans sökfunktion och genom de filtreringar som kan göras. Personer söktes utefter filter-kriterierna Scrum och Sverige och författarna gjorde sedan en bedömning på hur relevanta de personer som söktes fram var för undersökningen. Framförallt bedömdes om personens företag var passande för studiens relevans och i relation med resterande undersökta företag. En respondent valdes ut på detta sätt.

Nedanstående tabell presenterar kort respondenterna, vilken roll de har, var de är placerade, vilken bransch¹ de verkar inom samt hur lång tid respektive intervju tog. Respondenternas namn har bytts ut mot initialer för att anonymisera dessa.

Respondent	Roll	Placering	Bransch	Intervjulängd
AF	IT-projektledare	Göteborg	Konsultverksamhet avseende företags org.	47 min
HZ	Agil coach	Göteborg	Datorer & kringutrustning, tillverkning	1 h 6 min
HÅ	Engineering manager	Stockholm	Dataprogrammering	53 min
MT	Product manager	Stockholm	Ljudinspelning & fonogramutgivning	40 min

Tabell 1: Respondentbeskrivning

¹ Branschdefinitioner hämtad från <https://www.retriever.se>

4.4 Intervjuer

Datainsamlingen har genomförts genom semistrukturerade djupintervjuer. Detta innebär en färdig lista med ämnen men att forskaren samtidigt är flexibel gällande ämnenas ordningsföljd (Denscombe 2016). Författarna valde denna typ av intervju för att kunna få mer utförliga och djupgående svar från respondenterna. Då varje teams förutsättningar skiljer sig har denna öppenhet passat för att ge intervjuerna ”luft” för att således få ut mer uttömmande svar. På så sätt har författarna även kunnat undersöka både vilka avsteg som görs från Scrum utifrån teorin men även fått respondentens svar på varför det gjorts.

Samtliga intervjuer har genomförts på ett personligt sätt, det vill säga ansikte mot ansikte. Detta för att det ger fördelar i och med att interaktionen blir direkt, samt undviker eventuella missförstånd. Intervjuer är komplexa interaktioner (ibid.) och uppsatsens författare har därför tagit i beaktning att missuppfattningar kan uppstå i ett sådant samtal. Fördelen med personliga intervjuer är att detta minimerar felaktiga tolkningar som kan uppstå vid en mer bristfällig kontakt såsom vid telefonintervjuer eller mail-kontakt.

Dessutom har beaktning tagits att vi i mångt och mycket kan påverka intervjupersonens svar. Den så kallade intervjuareffekten har kunnat ta sig uttryck i att respondenten gett svar som denne tror att vi som intervjuare velat få. Vi har i mångt och mycket försökt undvika detta genom att ha en neutral och professionell intervjuteknik. Intervjuguiden har i största möjliga mån utformats utan ledande frågor för att inte påverka respondenten. Ledande frågor kan dock ha uppkommit under intervjuerna i form av följdfrågor där vi ville få svar på frågan om varför det ser ut på det sätt det gör.

4.4.1 Genomförande

Samtliga intervjuer har spelats in och därefter transkriberats i sin helhet. På så sätt har det kunnat säkerställas att allt som tas upp i intervjuerna också har gått att analysera. Inspelning av intervjuerna har också eliminerat behovet av att föra anteckningar som kan påverka uppmärksamheten hos intervjuledarna (Bryman & Bell 2005).

Två av intervjuerna genomfördes i Stockholm på kontoren hos de undersökta företagen. Resterande intervjuer tog plats i Göteborg. Majoriteten av intervjuerna har genomförts på plats hos de undersökta företagen och då på en avskild plats. Detta innebar att miljön för intervjuerna var optimal.

Intervjun med HZ genomfördes på ett kafé i Göteborg. Detta var nödvändigt på grund av den geografiska placeringen av företagets kontor vilket innebar att intervjun istället tog plats i centrala Göteborg. Intervjumiljön innebar vissa svårigheter med transkribering och bearbetning av empirin men författarna anser trots detta att detta inte utgjort ett avgörande problem för studiens resultat.

4.4.2 Etik

Till varje intervju hade säkerställts att intervjupersonen förstod undersökningens fokus och ämnet som intervjuerna skulle komma att behandla. Dessutom säkerställdes att vi hade intervjupersonernas tillåtelse att spela in intervjuerna. Innan intervjuerna inleddes gavs ett klagörande för intervjupersonen att de själva fick bestämma i vilken mån de ville vara anonyma men att det var behjälpligt för intervjuerna om personen kunde klargöra vad dennes roll innebar. Det klargjordes också att intervjupersonen kunde avbryta intervjun om denne önskade det, oavsett anledning. Detta för att intervjupersonen i största möjliga mån skulle känna sig bekväm med att förmedla information om hur företaget arbetar och utvecklar produkter. Empirin som framkom under intervjuerna har endast hanterats av författarna som underlag till denna kandidatuppsats och inte för något annat ändamål, vilket även lovades till respondenterna.

4.4.3 Intervjuguide

Intervjuguiden (Se bilaga Intervjuguide) utformades utifrån teorins beskrivning av Scrum för att undersöka i vilka avseenden teamen avviker från metoden. Frågorna fungerade som en mall utefter vilken intervjun kunde förlöpa. I och med att intervjuerna utfördes på ett semi-strukturerat sätt utgjordes frågorna i intervjuguiden som en ritning över intervjuernas riktning där följdfrågor lätt kunde ställas till varje fråga. Framförallt eftersträvades att till varje fråga kunna ställa följdfrågan om varför man arbetade på ett visst sätt. Frågorna följde en uppsatt ordningsföljd efter Scrums tillvägagångssätt. Intervjuguiden konstruerades för att börja med att undersöka den tillfrågade respondenten om dennes ansvarsområden och sin roll.

Därefter efterfrågades teamets uppbyggnad med syfte att undersöka i vilken utsträckning man använder sig av fasta team som är tvärfunktionella och självorganiserade. I Scrum finns inga titlar inom teamet förutom Scrum Master, produktägare och utvecklingsteam och därför ville vi få reda på om så var fallet även här eller om man kombinerar detta med andra typer av roller. I Scrum förespråkas att alla kompetenser som behövs ska finnas i teamet.

Dessutom så ska teammedlemmarna själva, utan extern påverkan, bestämma hur arbetet bäst bedrivs.

Detta mynnade sedan ut i frågor kring hur teamet arbetar en ”vanlig arbetsdag”. Detta i syfte för att undersöka om det fanns någon skillnad hos teamets sätt att arbeta och Scrums beskrivning. Därefter efterfrågades i vilken mån man följer Scrums instruktioner om fasta sprintar och hur man arbetar med dessa, hur de planeras och avslutas. Dessutom tillfrågades respondenterna hur teamet ser på förändringar under sprintens gång.

Intervjuguiden avrundades med att fråga huruvida teamet har experimenterat med andra metoder, om anpassningen har tagits fram medvetet samt om respondenten upplevde att metoden anpassats ur organisationens kontext.

4.5 Analys av empiri

I och med att intervjuerna har transkriberats i sin helhet har författarna kunnat kategorisera svaren, vilket har gjorts primärt utefter de frågor som ställdes under intervjuerna. Citat som har varit karaktäristiska har tagits fram och jämförts med de andra intervjuerna. Vidare har författarna tolkat intervjuerna och jämfört med den teori som används i studien.

4.6 Trovärdighet & pålitlighet

Kvalitativ forskning innebär svårigheter i att dels generalisera, men även att validera resultatet av studien då en social miljö är svår att återskapa (Denscombe 2016). Det går därför inte att kontrollera resultatet genom att upprepa en kvalitativ studie med samma lätthet som en kvantitativ studie. Istället bedöms kvalitativ forskning efter dess trovärdighet, det vill säga insamlad datas träffsäkerhet, och pålitlighet som rör huruvida studiens resultat skulle vara samma om studien hade genomförts av någon annan (ibid.).

De genomförda intervjuerna var djupgående och omfattande samtidigt som respondenternas svar ofta var uttömmande och detaljerade. För att öka studiens trovärdighet har författarna återkopplat till respondenterna för att säkerställa att respondenternas svar har tolkats korrekt vid de tillfällen då svaren har varit otydliga eller svåra att tolka.

Gällande pålitligheten har författarna beskrivit det som ansetts vara relevant för studien gällande den kontext som de undersökta företagen befinner sig i. De val som gjorts för forskningsprocessen och tillvägagångssättet har redogjorts i största möjliga utsträckning. Författarna har försökt att inte låta personliga värderingar påverka utförande och analys av studiens resultat, även om det är svårt att komma ifrån att författarnas identitet de facto har en inverkan på studien. Författarna anser att de har kunnat hantera empirin på ett opartiskt sätt samt att studien har skett under god praxis.

5 Resultat och analys

Avsnittet inleds med en kort beskrivning av varje respondent, vidare presenteras empirin som hämtats från intervjuerna tillsammans med författarnas analys av resultatet gentemot tidigare forskning.

5.1 Respondenterna

Empirin bygger på intervjuer med fyra personer vilka alla innehar en roll med övergripande ansvar för minst ett team och därmed även en god förståelse för teamets eller teamens arbetssätt. Samtliga team arbetar med underhåll och vidareutveckling av existerande produkter.

5.1.1 Respondent AF

AF arbetar som IT-projektledare för två utvecklingsteam på företagets kontor i Göteborg. Företaget har kontor i Göteborg och Stockholm där IT- respektive ekonomiavdelningen befinner sig samt små kundhanteringskontor i de länder där kunderna befinner sig. Företaget tillhandahåller en mjukvaruprodukt för andra företag som skräddarsys för varje kund. AF beskriver uppgiften i sin roll som IT-projektledare är att optimera för sina team och se till att få ut så mycket som möjligt av dessa.

5.1.2 Respondent HZ

HZ är Agil coach hos ett företag vika nyligen har påbörjat införandet av agila arbetssätt i organisationen. Företaget tillverkar och underhåller maskiner med tillhörande mjukvara för industri. HZ har ansvar för övergången till ett agilt arbetssätt och således även att vänja både ledning och teamet vid de nya metoderna. HZ uppgav att det är ett tämligen litet kontor och att han därför arbetar både nära teamet och ledningen. Teamet är uppdelat i tre kompetensområden: Mjukvara, hårdvara samt mekanik.

5.1.3 Respondent HÅ

Företaget där HÅ arbetar tillhandahåller en mjukvaruprodukt som riktar sig mot både mot företag och konsumenter. HÅ arbetar som Engineering manager på företagets kontor i Stockholm. Han ansvarar för teamens långsiktiga planering men är inte direkt involverad i Scrum-teamen.

5.1.4 Respondent MT

MT är Product manager hos ett företag som tillhandahåller en mjukvaruprodukt främst riktad mot företag. MT ansvarar för processen, lanseringen och arbetssättet hos ett team. MT menar att de använder sig av en process med inslag från Scrum.

5.2 Team

5.2.1 Storlek

De undersökta teamens storlek skilde sig från varandra i viss utsträckning. HZ uppgav att hans team utgjorde 3 personer som arbetade med mjukvaran. Dessa utvecklare ingick i sin tur i ett större team som totalt uppgick till 7 personer, som tillsammans utvecklade produkten. Respondent HÅ uppgav att man inom företaget inte hade en ram för hur stora teamen skulle vara men att ett typiskt Scrum-team uppgick till mellan 6 och 7 personer. Teamet i respondent MT:s fall uppgick till 6 personer varav en av dessa var testare. AF hade det största teamet på 8 personer.

5.2.2 Roller

MT och HÅ använder sig av tech lead (sv. teknisk ledare), en form av gruppleddare med teknisk kompetens som hanterade kommunikation mellan respondenten och övriga teamet. Denna person fungerar även som en coach för teamet och i fallet med HÅ strukturerade även denna person backlog samt hade bestämmande över ny utveckling. Respondenten MT uppgav att han använde sig av en tech lead på grund av att MT själv *"inte är så teknisk"*. MT och denna tech lead hade gemensamt ansvar för hur backloggen skulle struktureras och prioriteras. Både MT och HÅ hade roller som kan liknas vid produktägare då de uppgav att deras arbete var att planera över lång sikt och sköta visionen av arbetet. Teamet rapporterade till dessa respondenter främst genom ovan nämnda tech lead.

En formellt utsedd Scrum Master saknades hos alla respondenter. MT uppgav att han delade rollen med tech lead och att dessa tillsammans hade en Scrum Masters ansvar över teamet. HÅ beskrev att teamen själva fått tilldela respektive tech lead rollen som Scrum Master men att detta varit helt upp till teamen att välja själva. AF och HZ uppgav att de på olika sätt hade ansvaret som både projektledare och Scrum Master men att rollen inte var en uttalad titel hos respondenterna. AF menade att det inte fanns en bestämd Scrum master i teamen som såg till att Scrum *"hålls ihop"*. HZ uppgav att: *"Mitt fokus är att vara där"*

som, inte projektledare, inte riktigt Scrum Master, men någonstans i den häraden som sysslar med de frågorna då som de olika rollerna innebär”.

5.2.3 Självorganisering och arbetsuppgifter

Viss variation kring teamens självorganisering fanns bland respondenterna. HZ och AF uppgav att deras team i princip var självgående. Teamen fick mycket eget ansvar och organiserade arbetsuppgifterna mellan varandra i så stor utsträckning som möjligt. AF menade att han i största möjliga mån lät utvecklarna själva välja arbetsuppgifter eftersom att de kunde optimera på ett mer effektivt sätt. Inga regler kring detta fanns och att det var helt upp till varje team hur man bäst organiserade arbetet. På grund av prioritetsskäl menade dock AF att utvecklarna ibland kunde få arbetsuppgifter tilldelade: *”Jag vill att de plockar själva. Sen finns det alltid surdegar som ingen plockar och då får jag gå in och hjälpa till där och säga att: ’Du får ta den här, nu är den här viktig’, och det är inga konstigheter”.*

”Teamet bestämmer vad man ska göra, när man ska göra det och framför allt hur man ska göra saker.” – HZ

I fallet med HÅ och MT fanns viss självorganisering kring hur arbetet skulle bedrivas men desto mindre kring arbetsuppgifterna i sig. Båda respondenterna beskrev att detta skedde som en diskussion mellan tech lead och övriga teamet utefter förslag från utvecklarna. Tech lead hade dock det formella ansvaret.

5.2.4 Tvärfunktionalitet i teamen

Alla respondenter uppgav att man arbetade med fasta team. Dessa är vanligen uppdelade utefter kompetensområden med undantag för HZ vars teams arbete utgjorde hela produkten. I de övriga fallen arbetade teams med specifika delar av produkten och ansågs av respondenterna vara tvärfunktionella inom sitt område. Alla respondenter uppgav att resurser i form av personal delades frekvent mellan teams utefter vad som krävdes. Respondent AF uppgav att: *”Vi tar in arbete från andra. Andra team kan behöva någonting [...] då tar vi den uppgiften av dem.”*

5.2.5 Produktägarens involvering

Båda respondenterna MT och HÅ agerar i rollen som produktägare. MT uppger att det är han och tech lead som gemensamt hanterar backlog och gemensamt med teamet planerar sprintarna. I övrigt lämnas teamet att arbeta på egen hand medan MT *”sitter i massa möten men det är också för att jag har olika roller”* men menar att detta krävs för att driva andra

projekt och synkronisera arbetet med andra managers. MT menar också att hans ansvar främst innebär att driva visionen för produkten och se över arbetsprocessen för teamet.

HÅ är vad han kallar för Engineering Manager och agerar vad som kan liknas vid en produktägare till flera team. Han beskriver hur organisationen tidigare använt sig av Product Managers men att de gått över till Engineering Managers med ett större fokus på en långsiktig produktplanering. HÅ beskriver att han låter teamen sköta sina dagliga möten på egen hand men att han ”som manager brukar väl gå och lyssna in mig lite, kanske har lite kommentarer eller så [...] om jag har någon input på andra grejer”. Han menar även att Engineering Managers har mer direktkontakt med utvecklarna medan de tidigare Product Managers inte deltagit i sådana möten. Samtidigt har teamens tech leads enligt HÅ mer gått över till att ha produktansvar och att det är denna person som strukturerat om backloggen och skött planering.

AF och HZ uppgav att arbetet med produktägaren sker på ett involverande sätt. En produktägare till AF satt med i samma rum som teamet och var dagligt involverad i projektet. Produktägarens roll innebär här att prioritera backlog samt styra projektet medan AF konkretiserar arbetet till arbetsuppgifter

HZ nämner också prioritering som produktägarens största ansvar men att denna också lyssnar till teamets åsikter. Detta är viktigt då arbete lätt kan bli svårt om produktägaren ställer krav utan att vara involverad och har en förståelse för hur processen ser ut. HZ beskriver också att produktägaren har ansvar utåt och att HZ styr ”alla andra kanaler utifrån via honom (produktägare) så att teamet får lugn och ro”.

5.2.6 Analys

Team	AF	HZ	HÅ	MT
Storlek	8	3 (7)	6-7	5
Scrum Master	AF	HZ	Tech lead	MT + tech lead
Certifierad inom Scrum	Ja	Ja	Ja	Nej
Produktägare	Ja	Ja	Ja	Ja
Produktägare involverad	Ja	Ja	Ja	Ja
Tvärfunktionella	Ja men inom specifika områden	Ja	Ja men inom specifika områden	Ja men inom specifika områden
Självorganiserade	I arbetet men inte vilka arbetsuppgifter	Ja	I arbetet men inte vilka arbetsuppgifter	I arbetet men inte vilka arbetsuppgifter

Tabell 2: Sammanställning analys av Team

I intervjuerna framkom att samtliga undersökta team hade en storlek som överensstämde med Scrums förslag på mellan 3 och 9 personer i utvecklingsteamet. Dessa var självorganiserade på ett sådant sätt att arbete organiseras av teamet själva.

Hos respondenterna MT och HÅ fanns dock viss kontroll över hur arbetsuppgifter vilket enligt Eloranta et al. (2015) utgör en *anti-pattern* då teamet blir tilldelade arbetsuppgifter från en product manager.

De undersökta teamen är alla tvärfunktionella men inom specifika områden. Detta sätt att tilldela teams specifika arbetsuppgifter innebär enligt Eloranta et al. (2015) att problem uppstår i och med att resurser behöver lånas mellan teamen. Detta innebär att utvecklingen av produkten avstannar.

En formellt utsedd Scrum Master saknades hos alla undersökta respondenters team och utgjordes istället ofta av antingen projektledare eller produktägare vilket i två av fallen var respondenten själv. Denna form av avvikelser är något som visat sig i tidigare forskning. Både Permana (2015) och Eloranta et al. (2015) har beskrivit att roller inom Scrum används på detta mer flytande sätt. En av respondenterna, MT, uppgav att han agerade i båda rollerna som produktägare och Scrum master, vilket enligt Pham & Pham (2012) kan innebära risker. Detta på grund av att det kan finnas svårigheter med att separera vad rollerna innebär.

5.3 Sprintplanering

AF uppgav att han själv i kontakt med produktägaren tar fram vad som är viktigt inför nästa sprint. Vidare upprättar AF en grovplanering efter vad han anser vara rimligt för den kommande sprinten.

HZ beskriver sin sprintplanering som en kort aktivitet där de säkerställer att uppgifterna är tillräckligt nedbrutna innan de lyfts in i sprinten. Sedan beslutas vad som ska göras i sprinten tillsammans i teamet genom en diskussion om vad man tror att man klarar av. MT berättade att deras sprintplanering går ut på att sälla bland de önskemål och problem som dykt upp under tidigare sprintar. I intervjun med HÅ framgick att hans team inte använt sig av sprintplanering sedan en längre tid.

5.3.1 Sprintmål

I flera av de undersökta teamen används inte sprintmål i den mån som beskrivs enligt Scrum. Istället för att ha ett gemensamt mål i den enskilda sprinten har AF istället ett fokus som stäcker sig över flera sprintar och baseras på det huvudprojekt som sprinten är en del av. I intervjun med HÅ framgick att de sätter högre prioritet på ett antal uppgifter som de måste vara klara med vid sprintens slut men har inte konkreta sprintmål. MT uppgav att de tidigare använt sprintmål men att detta försvunnit. Han menade att de istället har en underförstådd prioritering.

HZ uppgav att han försöker sätta upp sprintmål men att dessa kan störas av akuta problem som uppstår under sprintens gång. HZ menar dock att sprintmål fungerar som en vision och underlättar beslutsfattande.

”Det är lite grann som en vision. Sätter man en vision har man en ledstjärna [...] Har man inget gemensamt mål blir det individens beslut varje gång, inte teamets beslut.” – HZ

5.3.2 Estimeringar

I intervjuerna framkom att ingen estimerade arbetsuppgifterna inför kommande sprint. Samtliga respondenter menade istället att de utgår ifrån tidigare erfarenheter och uppskattar hur många uppgifter de kommer klara av att slutföra.

Både HÅ och MT beskrev estimering som alltför tidskrävande aktiviteter och deras team har på grund av det här valt att inte använda det. Istället går utvecklarna på rutin och erfarenhet. HZ gav en liknande beskrivning: *”Vi tittar mer på vad tror vi att vi är kapabla att göra den här veckan [...] Vi har en känsla för hur stora de är. Vi har inte gjort någon detaljestimering, inte heller säkerställt att de är lika stora.”*

AF berättade att de tidigare estimerade arbetet men att teamet inte upplevde detta som tillräckligt konkret och aktiviteten var därmed svår att motivera.

”Folk hade svårt att förstå hur de skulle kunna estimera det här [...] Det bet inte, det kändes inte vettigt att göra. Det kändes som att det var på låtsas.” – AF

Bristen på estimerat arbete gjorde att inget undersökt team använde velocity som mätinstrument.

5.3.3 Analys

Sprintplanering	AF	HZ	HÅ	MT
Deltagare	AF & produktägare	HZ, produktägare & utvecklingsteam	-	MT, tech lead & utvecklingsteam
Estimeringar	Nej	Nej	Nej	Nej
Sprintmål	Nej	Ja	Nej	Nej

Tabell 3: Sammanställning analys av Sprintplanering

Som framgår ur tabellen sker inte sprintplanering enligt Scrum. Respondenterna HZ och MT följer Scrum på så sätt att hela teamet är involverat i planeringen. Samtidigt går det ur empirin att utläsa att sprintplanering är en aktivitet som det inte läggs särskilt stor vikt vid. Istället har teamen kommit fram till ett arbetssätt som går i linje med hur Blomberg (2013) menar att lyckade projekt inte är alltför välplanerade. Estimeringar över arbetet ses inte som något viktigt utan att man istället ”går på magkänslan”. På grund av detta kan teamen inte heller beräkna hur snabbt de kan slutföra arbete.

Bristen på estimeringar innebär att arbetet i sprinten inte alltid hinner färdigställas och att teamet i så fall missar sina åtaganden (Pomar et al. 2014). Bristen på estimeringar och velocity innebär att teamet inte vet hur lång tid deras arbetsuppgifter tar att genomföra (Eloranta et al. 2015). Detta verkar dock inte vara ett problem för de undersökta teamen då flera av respondenterna menar att detta beror på erfarenhet. Bristen på estimeringar och velocity kan också förklaras med att teamen saknar externa beställare som ställer krav på att arbete ska slutföras i tid. Uppsatta mål för sprinten saknades dock hos många undersökta, vilket förklaras med att planeringen utgörs genom att prioritera arbetsuppgifter snarare än att detaljplanera. Snarare framkommer en bild av att det inte spelar så stor roll hur mycket som åstadkoms i sprinten.

”Ofta brukar det inte komma in nya grejer, utan vi brukar skjuta nya features på nästa sprint.” – MT

”Det vi gör är inte så hårt efterfrågat för ett visst datum [...] och då är det kanske inte lika viktigt om det är klart i en viss sprint eller om det går över till nästa sprint.” – HÅ

5.4 Sprintar

Alla undersökta team hade som utgångspunkt att arbeta med fasta sprintar, däremot varierade sprintarnas längd mellan de undersökta teamen. HZ använde endast en vecka för

varje sprint medan MT hade en månad. De andra respondenterna hade tvåveckorssprintar, även om AF uppgav att de i extremfall kan förlänga sprinten.

”Det finns inget heligt med två veckor utan vi tar det som passar, men två veckor tycker jag är lagom.” – AF

AF beskrev ett specifikt fall som teamet nyligen hade varit med om där de hade en bestämd deadline. Teamet visste vad de skulle göra och när allt skulle vara klart vilket gjorde att det planeringsarbete som sker inför nya sprintar kändes överflödigt.

MT beskrev att deras fyraveckorssprint delades upp i två faser. Två veckor ägnar teamet åt utveckling och två veckor används för test och release av det som utvecklats. Testningen sker externt av ett annat företag som skickar tillbaka feedback kring buggar och fel som hittats. Efter att detta åtgärdats skickas detta åter igen till testning.

5.4.1 Daily Scrum

Samtliga av de undersökta teamen håller dagliga Scrum-möten á 15 minuter med undantag från MT som uppgav att deras Daily Scrum hålls under 10 minuter. Under detta möte diskuteras vad som gjorts, vad som ska göras och eventuella problem som har uppstått i arbetet vilket var genomgående hos alla team. Genomgående är att respondenterna ser mötet som nyttigt för teamet men att det även fungerar för att dagligen planera arbete. HZ beskriver Daily Scrum som ett omplaneringsmöte där en revision görs av sprintplaneringen.

”Nu till exempel, när vi kör Kanban med mitt team så diskuterar vi mer på Daily vad vi tar in för någonting när de är klara med något.” – HÅ

5.4.2 Förändringar i sprinten

Både HÅ och MT menar att det endast är buggar vilka behöver åtgärdas akut som läggs in i sprinten, något som MT kallar för ”Brandsläckning”. Även HZ uppgav att det som kräver akut tillsyn läggs in i sprinten men även om något är till fördel för att nå sprintmålet kan förändringar ske. I övrigt håller gärna HZ kraven så fasta som möjligt, men det är samtidigt upp till teamet om något ska ändras.

HÅ beskrev att de tidigare, när organisationen precis hade börjat arbeta med Scrum, hade hårda krav vilka inte fick ändras. När organisationen nu har gått mer mot att använda

Kanban har det förändrats och alla team har någon gång behövt byta ut något i sprinten. Framför allt gällde detta buggar som behöver prioriteras.

Till skillnad från de tre andra tillfrågade respondenterna berättar AF att de är tämligen öppna för att lägga till nya saker. AF menar att så länge teamet är med på det och alla tycker att det är rimligt är det inget problem att ändra innehållet i sprinten. Detta beror på att teamet inte har fasta krav för vad som ska åstadkommas under sprinten menar AF.

5.4.3 Definition of Done

Inget av de undersökta teamen hade en klar definition av färdigställt arbete. I Scrum innebär Definition of Done att det finns en gemensam förståelse för vad färdigställt arbete innebär för alla i teamet. AF och HZ uppger att det idag inte finns en utarbetad definition som finns nerskriven. Istället utgörs den av vissa kriterier.

”Det skulle vi definitivt kunna jobba på [...] det är väldigt, väldigt vaga saker där. Det har varit uppe på tapeten flera gånger. Vad som ska vara klart, när det är klart och så.” – AF

”Den är ganska så tunn, men det finns vissa kriterier som vi ska ha testat av då.” – HZ

Istället anses arbetsuppgifter vara färdiga när de har testats av utvecklarna själva. MT ger en liknande beskrivning där mjukvaran testas antingen internt eller externt. *”Och sen när den har provats så är den Done”*. Däremot finns ingen konsensus bland utvecklarna vad det innebär att vara färdig med en arbetsuppgift.

HÅ beskriver ett bortfall av Definition of Done i samband med att Scrum i sig inte efterlevs då det inte längre är ”påtvingat” inom organisationen och att avsaknaden av product managers bidragit till detta. Man har istället gett teamet eget ansvar och inte påverkat deras beslut så länge det fungerar. HÅ beskriver att definitionerna upplevdes som självklara och därför inte behövts. Respondenten beskriver däremot att detta kan leda till missförstånd inom teamet och att problem kan uppstå när utomstående parter vill få insikt i teamets arbete då det saknas transparens i hur arbetet genomförs.

5.4.4 Analys

Sprintar	AF	HZ	HÅ	MT
Längd	2 veckor	1 vecka	2 veckor	4 veckor
Daily Scrum	15 min	15 min	15 min	10 min
Tillåter ändringar	Det är upp till teamet	Vid akuta behov och till fördel för sprintmål	Vid akuta behov	Vid akuta behov
Definition of Done	”Väldigt vaga saker”	”Ganska så tunn”	”Har fallerat”	Genom extern testning

Tabell 4: Sammanställning analys av Sprintar

Ur empirin framkommer att de undersökta teamen följer Scrums rekommendation om sprintar mellan 2–4 veckor. Uppdelningen av sprintar med extern testning i MT’s fall återfinns som anti-pattern hos Eloranta et al. (2015) som menar att detta innebär problem med att fel upptäcks sent. Detta är något som MT också nämner som ett problem: *”Då kan det hända att vi får en kritisk bugg här som är väldigt svår att förutse, vilket kan förstöra ganska mycket. [...] Det blir liksom att i nästa sprint förkortas den här tiden.”* Pham & Pham (2012) menar att det är fördelaktigt om testningen sker i takt med utvecklingen.

Daily Scrum efterlevs hos alla undersökta team. Denna praxis är relativt enkel att efterleva vilket också kan utgöra en förklaring. En annan förklaring är att Daily Scrum verkar i till viss del ha ersatt sprintplanering och istället hålls kortare möten varje dag.

I Scrum ska det arbete som ska utföras i sprinten fastställas i sprintplaneringen och därmed görs inga ändringar i detta under sprinten. Ur empirin framkommer dock att bristen på sprintmål kan vara en av anledningarna till tillåtna förändringar under sprintarna. Då inget sprintmål definieras av teamet kan därför ändringar göras utan att påverka detta. HZ utgör här undantag då hans team använder sprintmål och tillåter förändringar i sprinten om dessa går i riktning mot målet.

I Schwaber & Sutherlands (2013) guide till Scrum innebär Definition of Done att det finns en gemensam förståelse för vad färdigställt arbete innebär och att detta finns dokumenterat. Hos respondenterna framkommer istället en bild av att definitionen är underförstådd inom teamet. Genomgående är också att testning i sig utgör en definition av färdigt arbete.

5.5 Sprintavslut

5.5.1 Sprint Review

De tillfrågade respondenterna hade olika synsätt på sprintens avslut. Sprint review var i sig inget vedertaget begrepp utan de tillfrågade använde oftast demonstrationer eller ”demos”

som begrepp för detta möte vid sprintens avslut. MT uppgav att det fanns en avsaknad av uppföljning efter avslutad sprint och vad som genomförts. Han menade att de detta var något som kunde förbättras. Istället förs oavslutat arbete direkt över till nästa sprint och skjuts på framtiden. Detta var något som även stämde överens med AF's beskrivning av hur sprintens ofärdiga arbetsuppgifter förs över till nästa sprint, om de är av relevans.

I AF's team föregås däremot detta av en demonstration av produkten för produktägare där en genomgång görs av vad som är klart respektive ofärdigt.

HZ uppgav att det finns utrymme för förbättring vad gäller Sprint Reviews i respondentens organisation. I dagsläget presenteras produkten efter varje avslutad sprint men detta sker mer som en intern demonstration inom teamet. HZ menar samtidigt att detta är något som ska förbättras framöver genom publika presentationer av arbetet för att få in feedback från produktens alla intressenter.

HÅ uppger att de tidigare haft strukturerade sprint review-möten men att dessa har blivit mer och mer ovanliga inom organisationen. Detta förklarar HÅ med att den funktionalitet som hans teams arbetade med inte alltid var något som kunde utvärderas av utomstående. Detta på grund av att det som arbetades med i detta team rörde justeringar som inte syntes utåt på produkten i sig. Avsaknaden av organiserade review-möten förklarades därför med att det "inte kändes relevant" eftersom att arbetet inte gick att presentera.

5.5.2 Sprint Retrospective

På ett likartat sätt har även Retrospective, team-möten som hålls för att diskutera förbättringar inom teamet, med tiden fasats ut ur HÅ's organisation. Han uppger att: *"Efter ett tag så kändes det som att det var mycket samma saker som kom upp och inte riktigt hände så mycket på den delen så att det var väl en av de grejerna man, åtminstone i mina team, slutade med först. Det gav inte lika mycket."*

Enligt HÅ har man inte använt Retrospective på länge men att det har hänt att man haft möten då nya teammedlemmar velat ta upp problem. En annan förklaring som gavs var att personer i teamet inte kände sig bekväma med att *"bli utvärderade för vad de klagade på"*. Många teams hade därför avstått från att hålla detta möte. I andra fall hade dessa istället krävt att HÅ och andra managers inte skulle medverka på mötet, för att diskussionen skulle kunna vara så öppen som möjligt.

MT och AF hade strukturerade Retrospective-möten och respondenternas sätt att hålla dessa liknade varandra där teamen adresserar både problem och beröm. MT beskrev hur hans team hade ett format som innebar att teamet skulle antingen börja, sluta eller fortsätta med olika aspekter i sitt arbetssätt. AF menar att de sällan fokuserar på arbetssättet i själva sprinten utan att det istället kan vara hur teamet arbetar på ett allmänt plan.

Gemensamt för MT och AF var att Retrospective sker som en diskussion där problem tas upp och lösningar röstas fram gemensamt. Dessa lösningar, som båda respondenter beskriver som ”actions”, är genomgående för hela teamet och tas med till framtida arbete. AF beskriver dock hur hans team försöker hålla dessa lösningar till en minimal nivå: *”Vi kommer ändå köra det igen om två veckor [...] Man kan komma fram till 10 saker som är bra att göra men de kommer ändå aldrig göras.”*

HZ använder Retrospective efter varje sprint men uppger att dessa är korta. Istället hålls längre möten var fjärde vecka. Detta större Retrospective innebär att mål sätts upp vilket testas under kommande fyra veckor. HZ beskriver detta som ett experiment som sedan mäts och testas kontinuerligt hela tiden genom avstämningar. På så sätt kan resultat synliggöras.

”Sen när man då gör en slutgiltig utvärdering av de här experimenten, då inser man att det antingen var en bra idé eller inte och då för vi in det i vår process.” – HZ

5.5.3 Analys

Sprintavslut	AF	HZ	HÅ	MT
Review	Demo	Intern demo	Ovanliga	Nej
Retrospective	Ja	Ja	Nej	Ja

Tabell 5: Sammanställning analys av Sprintavslut

Ur tabellen ovan kan utläsas att en stor variation kring Sprint review existerar bland de tillfrågade. Både om de hålls, men även hur de hålls. I alla de undersökta fallen innebär momentet ett avsteg från Scrum. AF är den respondent som efterliknar Sprint review närmast enligt teorin då teamet har en demonstration för utomstående och planerar inför nästkommande sprint.

Avsaknaden av en formell utvärdering av sprinten hos de andra respondenterna tyder på att de inte är beroende av andra team eller utomstående intressenter. Det kan också enligt Eloranta et al. (2015) tyda på att teamet lider av en avsaknad av feedback från utomstående

parter vilket leder till att mycket arbete behöver göras om. Detta kan kopplas till den tidigare beskrivna akuthantering av problem som uppstår under sprintens gång.

Sprint Retrospective efterlevs hos tre av de undersökta respondenterna. Att denna aspekt av Scrum efterlevs kan tolkas som att mötet ses som viktigt för teamen. Förbättringar i teamet främjas genom dessa diskussioner (Eloranta et al. 2015) vilket tyder på att det finns ett värde för teamen att utvärdera sin arbetsgång och sitt samarbete.

5.6 Användningen av Scrum

Sammanfattning	AF	HZ	HÅ	MT
Teamstorlek	8	3 (7)	6-7	5
Scrum Master ²	AF	HZ	Tech lead	Tech lead + MT
Sprintar	2 veckor	1 vecka	2 veckor	4 veckor
Tvärfunktionalitet	Ja men inom specifika områden	Ja	Ja men inom specifika områden	Ja men inom specifika områden
Självorganisering	Ja	Ja	I arbetet men inte vilka arbetsuppgifter	I arbetet men inte vilka arbetsuppgifter
Estimeringar	Nej	Nej	Nej	Nej
Daily Scrum	15 min	15 min	15 min	10 min
Definition of Done	”Väldigt vaga saker”	”Ganska så tunn”	”Har fallerat”	Genom extern testning
Sprint Review	Demo	Intern demo	Ovanliga	Nej
Sprint Retrospective	Ja	Ja	Nej	Ja

Tabell 6: Sammanfattning av analys

5.6.1 Anpassningen av metoden

Gemensamt för respondenterna var att Scrum inte sågs som ett absolut tvång. Istället poängteras att de saker som inte behövs kan plockas bort fritt ur metoden utan att detta påverkar arbetet. HÅ menar att följa Scrum inte ses som viktigt längre inom organisationen. AF beskrev samtidigt att:

”Vi har en kultur av att försöka vara så slimmade som möjligt hela tiden. Det ingår i vår affärsidé att slösa med så lite tid som möjligt. Jag tror det är därför också att allt som inte är nödvändigt tar vi bort. Det ligger lite i kulturen här. [...] Saker och ting som inte behövs, dom tenderar att försvinna” – AF

² Den person som respondenterna uppgett inneha en liknande roll då ingen formell Scrum Master finns utsedd hos något av de undersökta teamen.

Ericksson et al. (2005) menar att kärnan av agilitet innebär att avlägsna så mycket onödiga processer som möjligt för att kunna svara på förändringar. Detta är något som framkommer ur empirin då respondenterna uppger att deras team gjort sig av med de delar av Scrum vilka upplevs som överflödiga och tidsödande. Respondenterna påvisar uppfattningar om att det agila arbetssättet bedöms vara viktigare än att följa metoden i sin helhet. Scrum som metod har på flera sätt fått ge plats till fördel för en slimmad arbetsprocess.

5.6.2 Metodens applicerbarhet

HZ beskriver att teamet fortfarande arbetar med att närma sig Scrum som arbetssätt men att detta till viss del går långsamt. *”Målet är att börja med det och ta sig närmare Scrum. Någonstans på vägen kommer vi säkert hitta någonting som passar bra”*. HZ uppger samtidigt att Scrum inte passar organisationens alla behov och att detta framförallt beror på den akuthantering av problem som krävs inom arbetet med mjukvara. Likaså menar AF att metoden inte alltid passar dem.

”Vi hittade inget riktigt bra sätt att göra det på och dessutom har vi väldigt mycket saker som bara kommer in och behöver fixas.” – AF

Scrum beskrivs av dess förespråkare som en metod som kan användas för alla typer av projekt och situationer. Enligt Dybå & Dingsøy (2008) är detta påstående dock ifrågasatt och bevisen bristfälliga. De agila metodernas överlägsenhet är istället av anekdotisk karaktär och ifrågasatt av flera forskare (Serrador & Pinto 2015). Empirin i denna studie påvisar att respondenterna upplever att Scrum inte kan tillgodose alla teamets behov. Istället för att använda metoden i sin helhet behöver teamen anpassa delar av den för att effektivisera arbetet.

5.6.3 Övergången till Kanban

Alla respondenter uppgav ett förändrat arbetssätt som mer liknar Kanban. Anledningen till detta berodde på att fokus hos de undersökta teamen låg i att framförallt arbeta för underhåll av produkten snarare än att vara deadline-drivna. Det fanns inte ett fokus på att släppa ett visst datum. Detta beskrevs bland annat på följande sätt:

”Vi har inte supertydliga viktiga deadlines, vi har inte saker som ska lanseras om tre månader utan vi ska lansera när någonting är bra. Scrum är ju jättebra om man vill hitta sin

velocitet, hitta någon förutsägbarhet man måste ändå göra någon nedbrytning tidigt. Det är inte så viktigt, utan när det är tillräckligt bra för att släppa då släpper vi det.” – AF

AF menar att de, i och med dessa behov, lika gärna kan använda Kanban där estimeringar inte behövs i samma utsträckning. HÅ ger en liknande förklaring till att Kanban till viss del ger fördelar för hans team. Däremot har denna övergång gått långsamt i och med att Scrum från början var pådrivet från ledningen.

”Vi fick tillåtelse att gå över till Kanban men jag fick knappt berätta det för andra team för att de ville att det skulle hållas i Scrum så långt som möjligt. Men i och med att det luckrades upp så är det fler och fler team som har gått över lite mer till Kanban-mässigt” – HÅ

I och med teamens varierande behov fungerar inte Scrum alltid för dem. Då arbetet sällan sker mot deadlines fungerar Kanban bättre eftersom prioriteringar av de viktigaste funktionerna kan göras samtidigt som akuta problem kan åtgärdas. Scrum har enligt flera respondenter fungerat väl i tidigare projekt då man har haft en mer fokuserad uppgift. Däremot blir detta svårare när man enligt AF inte har ”hela problembilden”. Att använda Kanban innebär istället att man kan reagera på problem som uppdagas i sprinten. HZ uttrycker det på följande sätt:

Vi behöver ha de här snabba, reaktiva teknikerna som finns i Kanban, man tar det som ligger överst i listan. [...] I vissa fall kan vi inte planera en vecka framöver för det kommer avbrott. [...] I det läget passar Scrum inte speciellt bra.” – HZ

5.6.4 En styrande metod

Metoden upplevs av respondenterna även som stel och påtvingande. HÅ uppgav att Scrum som metod införts på ett sätt som upplevdes kontrollerande och styrt i organisationen och att metodens moment fallerat i takt med att ledningen bytts ut och product managers ersatts av engineering managers. MT uppgav att Scrum ses som ett styrmedel för teamet:

”Det finns ingen yrkesgrupp, känns det som, som är så kontrollerad av vad de ska göra som utvecklare. [...] Och det är inte så roligt att hålla på med, att styra på den nivån” – MT

Blomberg (2013) menar att agila metoder såsom Scrum inte längre står för den flexibilitet de är avsedda för och pekar på att agila metoder över tid blivit mer och mer strukturerade och reglerande. Detta återfinns i hur respondenterna beskriver metoden där bland annat HÅ beskrev hur metoden infördes på följande vis:

”De [företagsledningen] skulle verkligen se till att vi började använda Scrum och då sattes det väldigt hårda regler på vad Scrum var, och de fick man inte bryta mot” – HÅ

5.6.5 Synen på Scrum

Gemensamt för respondenterna är att de beskriver att de är medvetna om att deras teams arbetssätt inte följer Scrum till punkt och pricka. Samtidigt ses Scrum som en god utgångspunkt för att föra in agila arbetssätt och filosofi till organisationen. Det framkommer också en bild av att Scrum är något som är utbrett inom branschen samt ses som ett modernt arbetssätt inom organisationer.

”Det är en industristandard. Alla har jobbat med det, alla som kommer från något bolag har tittat på det. Vi tror ju på det, det är en agil metodik som är bra. Den finns alltid kvar, det är ändå något som är bevisat att det funkar.” – AF

”Vi vart uppköpta av ett investeringsbolag som ville strukturera bolaget till att bli mer modernt. Och då sågs Scrum som en ganska viktig grej.” – HÅ

5.6.6 Sammanfattning

Att Scrum inte används i sin helhet beror på att metoden inte kan tillgodose teamens alla behov. Lösningen på detta har då blivit att ta bort de moment och delar av metoden som ses som onödiga och ohjälpsamma. Ur Ericksson et. al (2005) definition av agilitet tyder detta på att Scrum inte ses som en tillräckligt lättviktig metod i respondenternas uppfattning. Genom att försöka magra arbetsprocessen görs metoden om till en ännu mer agil variant. Metoden är därmed inte direkt applicerbar på de undersökta teamens kontext, enligt respondenterna. Detta trots att Scrum är tänkt att passa till alla situationer och till alla typer av projekt.

Detta har också att göra med teamens behov av att svara på akuta problem och nya krav vilket har gjort att de börjat anamma Kanban till fördel för Scrum. Dessa akuta problem

som alla undersökta team arbetar med gör att Kanban bättre passar deras behov, vilket också förklarats av respondenterna själva. Den tidigare forskning som bedrivits av Sjøberg et al. (2012) pekar på likheter mellan de undersökta företagens behov av att hantera akuta problem och förändrade krav i sprintarna. Sjøberg et al. (2012) fastslog att problem framförallt fanns kring att estimerar arbetsuppgifter vilket även framkommit i undersökningen.

Kombinationen av Scrum och Kanban, det vill säga Scrumban, är därmed något som respondenterna har använt sig för att slimma arbetsprocessen. Precis som Sjøberg et al. (2012) skriver anser respondenterna att långa möten är tidsödande undviker därför dessa så vida de inte verkligen behövs. Att övergå till Scrumban blir ett sätt att anpassa Scrum efter företagets behov.

Detta hör även samman med Scrums styrande och formella struktur som Blomberg (2013) pekar på. Metoden ses helt enkelt som för reglerande om den skulle följas till punkt och pricka. Den flexibilitet som de undersökta teamen eftersträvar återfinns enligt respondenterna inte i metoden. Teamen har därför valt att utarbeta egna varianter av metoden vilket tyder på att Scrum enligt respondenternas uppfattningar speglar den byråkratisering av projektmetoder som Christensen & Kreiner (1997) beskriver. Standardiseringen av en metod som Scrum, med dess regler, roller och processer, tyder istället på att metoden i själva verket leder till tröghet och stagning av arbetsprocessen.

Att Scrum ändå anammats i någon mån förklaras ur empirin att metoden ses som en standard inom branschen. Att arbeta som andra företag gör innebär enligt Røvik (2000) ett sätt att framstå som legitim gentemot andra inom fältet. Scrum kan i ljuset av detta ses som en institutionaliserad standard som används av företagen för att framstå som moderna organisationer. På detta sätt finns en viss frikoppling mellan hur man beskriver sin arbetsform och hur teamen faktiskt arbetar. Man säger sig arbeta med Scrum trots att man endast arbetar med valda delar. Scrum kan i och med detta ses som en symbol i enlighet med hur Røvik (2000) beskriver organisationsrecept. Metoden är på så sätt synonym med att vara framåtsträvande eftersom att ”alla arbetar med det”.

Ur empirin kan det utläsas att teamen som undersökts alla besitter en form av tvångslöshet. De är i stor grad självstyrande och saknar utomstående krav i form av estimering och utvärdering av sitt arbete. Då Sprint Reviews ofta saknas hos teamen gör detta att utomstående saknar inblick för hur teamet arbetar och stor tillit läggs på utvecklarnas

kompetens. Detta kan kopplas till hur Blomberg (2013) och Görling (2009) beskriver utvecklarnas allt mer centrala roll och makt. Att utvecklarna fått mer makt återfinns även i hur Scrum upplevs som en styrande metod som till viss del övergetts. Ur analysen av empirin framkommer därför en bild av att det till stor del är medlemmarna i teamet som avgör i vilken mån metoden används.

6 Slutsatser

I avsnittet besvaras studiens frågeställningar och författarnas slutsatser utifrån analysen presenteras.

6.1 Undersökningens syfte

Undersökningen har haft som utgångspunkt att undersöka hur metoden Scrum skiljer sig i praktiken gentemot hur den beskrivs i teorin. Studiens syfte var att ge författarna en fördjupad kunskap för hur och varför IT-företag väljer att använda och anpassa den agila metoden Scrum. Detta genom att svara på följande frågeställningar:

- Hur används den agila metoden Scrum i praktiken?
- Varför väljer de undersökta företagen att använda samt anpassa Scrum?

6.2 Hur Scrum används

Ur empirin går att utläsa att alla de undersökta teamen anpassar den agila metoden Scrum. Denna anpassning sker till viss del genom de *anti-patterns* som beskrivs av Eloranta et al. (2015). Den mest framträdande *anti-pattern* är att en formell Scrum Master saknas. Liksom Eloranta et al. (2015) har även denna undersökning funnit att självorganiserade team inte alltid bestämmer arbetsuppgifter och att de i vissa fall är specialiserade inom särskilda områden. Dessa former av semifunktionella team (Eloranta et al. 2015) gör att personal måste arbeta med arbetsuppgifter även utanför sina team.

Vissa aspekter av Scrum efterlevs däremot. Alla undersökta team har en storlek som överensstämde med metodens förslag om mellan 3–9 personer. Sprintarnas längd överensstämmer även de med Scrums förslag på under en månad. Hos de undersökta teamen finns även en produktägare som styr arbetet enligt Schwaber & Sutherlands (2013) beskrivning, att styra vision och planering för produkten samt att hantera backlog. Möten som i hög grad efterlevs är Daily Scrum och Sprint Retrospective vilket kan förklaras av att dessa möten anses vara givande för samarbetet inom teamet.

Heeager & Rose (2015) menar att Scrum kan användas för vidareutveckling av mjukvara, men att metoden då måste anpassas. Detta är något som framkom i undersökningen. Då företagen arbetar med att underhålla och vidareutveckla befintliga produkter har samtliga

team öppnat för att svara på förändrade krav som inte sattes under sprintplaneringen samt att åtgärda akuta problem som uppstår eller framkommer under sprintens gång.

Förutom att plocka ut valda delar av metoden har anpassningen även skett genom att kombinera Scrum med Kanban. Alla respondenter beskrev en övergång till ett arbetssätt som innebar en blandning mellan Scrum och Kanban. Denna kombination av metoderna kan liknas vid teorins beskrivning av Scrumban (Nikitina et al. 2012; Wang et al. 2012). Metoderna kombineras genom att plocka ut valda delar, så kallade metodfragment, från de två metoderna. Däremot byggs inte ”in-house”-metod på ett sätt som beskrivs av Campanelli & Parreiras (2015) och Qumer & Henderson-Sellers (2008). Metoden skapas istället på ett omedvetet sätt där teamen provat sig fram. Vilka delar av metoderna som används av teamet bestäms inte av en person, det vill säga en metodingenjör, utan tillvägagångssättet bestäms snarare tillsammans av teamen. Dock finns en kunskap om hur de båda metodernas tekniker och principer kan användas.

6.3 Varför Scrum används och anpassas

Avsaknaden av en formellt utsedd Scrum master kan till viss del förklara varför Scrum inte efterlevs. I Schwaber & Sutherlands (2013) beskrivning av rollen som Scrum Master innebär detta framförallt att se till så att principerna i Scrum efterlevs och förstås av hela teamet. I och med att rollen saknas ger detta en indikation på att metoden anpassats utifrån denna aspekt. Detta kan kopplas till hur Conboy & Fitzgerald (2010) menar att det krävs att metoden regelbundet bör utvärderas för att den inte ska överges på grund av förbiseende. Att Scrum Master saknas hos alla de undersökta teamen innebär då att det inte finns en central roll som ser till att metoden följs.

Att Scrum inte används i sin helhet beror på att metoden inte är direkt applicerbar till den kontext som de undersökta teamen verkar i. Metoden kan därför inte tillgodose teamens alla behov, trots att Scrum sägs passa alla situationer och typer av projekt. Samtidigt ses Scrums regler och rigida beskrivning som reglerande. Lösningen på detta blir då att ta bort delar av metoden för att göra metoden mer lättviktig och flexibel vilket skapar ytterligare agilitet i arbetssättet.

Resultatet av detta blir en kombination mellan Kanban och Scrum. Detta på grund av teamens behov av att kunna ta in nytt arbete löpande i sprinten i takt med att nya krav och problem dyker upp. Således innebär kombinationen av de båda metoderna att teamen

använder sig av Scrumban, enligt Nikitina et al. (2012) och Wang et al. (2012). Scrums möten och planeringsarbete ses även som tidsödande och respondenterna undviker därför dessa så vida de inte verkligen behövs.

Att företagen väljer att använda Scrum kan ses som att metoden innebär en institutionaliserad standard. Scrum i sig är då inte användbart som arbetsmetod men det är fortfarande legitimt för teamen att tala i de termer som metoden förknippas med. På så sätt skapas en form av frikoppling likt den Røvik (2000) beskriver mellan ”prat och praxis”. Scrum bär på så sätt en form av symbolism för de undersökta teamen, trots att arbetssättet skiljer sig från dess beskrivning av Schwaber & Sutherland (2013).

De anpassningar som gjorts kan även kopplas till agila metoders ideologiska synsätt på utvecklarnas roll i projekt (Blomberg 2013; Görling 2009). Detta eftersom att teamen till stor del fungerar som självorganiserade och specialiserade enheter utan utomstående krav och utvärdering av arbetet. Således erhåller utvecklarna viss makt då de besitter kompetens som är värdefull för företaget och inte kan påverkas av utomstående parter.

7 Slutdiskussion

Författarna inledde denna studie med en viss förförståelse för Scrum. Vi har däremot mött en verklighet som är långt mer komplex än vad vi stött på utifrån de teorier som tidigare studerats. Att Scrum inte används enligt teorins anvisningar enligt punkt och pricka är det mest framträdande som går att utläsa av denna studie. Att Scrum är den mest populära agila metoden men samtidigt anpassas hos företag leder till en undran över hur många andra varianter som finns.

Något som framförallt varit intressant för undersökningen är respondenternas goda kunskap kring Scrum. Detta är tänkvärt då det motbevisar Conboy & Fitzgerald (2010) som menar på att en bristande förståelse för en agil metod till viss del kan förklara varför anpassning sker.

Utifrån resultatet i denna studie kan författarna fråga sig vilken roll metoder som Scrum spelar i dagens företag. Agila metoder ses än idag som frälsningen för många företags problem och utmaningar, trots att det agila manifestet togs fram för mer än 15 år sedan. Att metoder som Scrum inte används enligt dess beskrivning leder till reflektion kring hur relevanta de agila metoderna står sig idag. Snarare verkar det som att själva filosofin kring det agila arbetssättet är vad som lockar företag att anamma metoder såsom Scrum, istället för metoderna i sig. Möjligtvis behöver både Scrum och det agila manifestet en uppdatering för att besvara de förväntningar som idag ställs av organisationer.

7.1 Kritik till metoden

Studien baseras på intervjuer av fyra respondenter. Det hade varit önskvärt med ett större urval som kunnat styrka studiens resultat. Det är möjligt att resultatet även skilt sig vilket lett till en annan slutsats. De undersökta företagen verkar också till viss del i olika kontexter vilket kan ha påverkat studien. Trots detta var intervjuerna djupgående och respondenterna gav detaljerade svar. Författarna har i dessa intervjuer kunnat tyda ett mönster hos alla de undersökta teamen vilket bringar understöd för det fenomen som vi funnit hos alla respondenter.

Intervjuer är enligt författarna en bra datainsamlingsmetod till en undersökning som denna. Metoden innebär dock vissa begränsningar. Det kan finnas svårigheter med att endast intervjua den typ av respondenter som undersökningen studerat. Detta på grund av att

studien endast behandlar dessa respondenters verklighet, vilket kan bli missvisande. Genom enkäter och observationer hade ytterligare data kunnat samlas in för att undersöka även medlemmarna i teamets verklighet och uppfattning av anpassningen av Scrum.

7.2 Förslag till vidare studier

Författarnas förslag till vidare studier är att genomföra en observationsstudie för att på en djupare nivå få insikt i hur och varför metoden anpassas samt vilka effekter det medför. Flera andra faktorer kan på så sätt också träda fram. Vidare är det även av intresse att undersöka huruvida de anpassningar som görs av Scrum har något orsakssamband.

Författarna efterfrågar dessutom en undersökning kring företag vilka utvecklar en ny produkt i syfte att undersöka om även dessa är i behov av att anpassa Scrum till sina begränsningar och kontext.

Referenser

- Agile Alliance. (2015). Manifesto for Agile Software Development. <https://www.agilealliance.org/agile101/the-agile-manifesto/> [Hämtad 2017-04-12]
- Al-Baik, O. & Miller, J. (2015). The kanban approach, between agility and leanness: A systematic review. *Empirical Software Engineering*. Vol. 20(6), pp.1861-1897. DOI: 10.1007/s10664-014-9340-x [Hämtad 2017-05-08]
- Anderson, D. J. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.
- Bennett, K. & Rajlich, V. (2000). Software Maintenance and Evolution: A Roadmap. *Proceedings of the Conference on the future of software engineering*. May 2000, pp.73–87
DOI: 10.1145/336512.336534 [Hämtad 2017-05-10]
- Blomberg, J. (2013). *Myter om projekt*. 2 Uppl. Lund: Studentlitteratur.
- Bryman, A. & Bell, E. (2005). *Företagsekonomiska forskningsmetoder*. 2 Uppl. Stockholm: Liber.
- Campanelli, A.S. & Parreiras, F. S. (2015). Agile methods tailoring – A systematic literature review. *The Journal of Systems and Software*. Vol. 110, December 2015, pp.85–100.
DOI: 10.1016/j.jss.2015.08.035 [Hämtad 2017-02-03]
- Christensen, S & Kreiner, K. (1997). *Projektledning: Att leda och lära i en ofullständig värld*. 1 Uppl. Bjärred: Academia Adacta.
- Cohn, M. (2010). *Succeeding with agile: Software development using Scrum*. 1 Uppl. Addison-Wesley.
- Conboy, K. and Fitzgerald, B. 2010. Method and developer characteristics for effective agile method tailoring: A study of XP expert opinion. *ACM Transactions on Software Engineering and Methodology*. Vol. 20(1), June 2010, pp.1–30.
DOI: 10.1145/1767751.1767753 [Hämtad 2017-02-09]
- Danielsson, L. (2016). *15 tecken på att du är snett ute när du jobbar agilt*. Computer Sweden. 2016-06-05 <http://computersweden.idg.se/2.2683/1.659798/agilt-15-tecken> [Hämtad 2017-02-08]
- Denscombe, M. (2016). *Forskningshandboken - för småskaliga forskningsprojekt inom Samhällsvetenskaperna*. 3 uppl. Lund: Studentlitteratur.
- Dingsøyr, T., Nerur, S., Balijepally, V. & Moe, N.B. (2012). A decade of agile methodologies: Towards explaining agile software development. *The Journal of Systems & Software*. June 2012, Vol. 85(6), pp.1213–1221.
DOI: 10.1016/j.jss.2012.02.033 [Hämtad 2017-02-08]

- Dybå, T. & Dingsøy, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*. Vol. 50(9). pp.833-859.
DOI: <https://doi.org/10.1016/j.infsof.2008.01.006> [Hämtad 2017-06-13]
- Eloranta, V., Koskimies, K. & Mikkonen, T. (2015). Exploring ScrumBut – An empirical study of Scrum anti-patterns. *Information and Software Technology*. June 2016, Vol. 74, pp.194–203
DOI: 10.1016/j.infsof.2015.12.003 [Hämtad 2017-02-09]
- Erickson, J., Lyytinen, K. & Siau, K. (2005). Agile Modeling, Agile Software Development, and Extreme Programming: The State of Research. *Journal of Database Management*. 34/2005, Vol.16 (4), pp.88-100.
DOI: <http://dx.doi.org/10.4018/jdm.2005100105> [Hämtad 2017-06-13]
- Fitzgerald, B., Harnett, G. & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems*. Vol. 15(2), pp.200–213.
DOI: 10.1057/palgrave.ejis.3000605 [Hämtad 2017-02-08]
- Fitzgerald, B., Russo, N. & O'Kane, T. (2003). Software development method tailoring at Motorola. *Communications of the ACM*. 01 April 2003. Vol.46 (4), pp.64–70.
DOI: 10.1145/641205.641206 [Hämtad 2017-03-29]
- Görling, S. (2009). *Att arbeta med IT-projekt*. 1 Uppl. Lund: Studentlitteratur.
- Heeager, L. & Rose, J. (2015) Optimising agile development practices for the maintenance operation: nine heuristics. *Empirical Software Engineering*, Vol.20 (6), pp.1762-1784
DOI: 10.1007/s10664-014-9335-7 [Hämtad 2017-05-10]
- Henderson-Sellers, B. & Serour, M. (2005). Creating a Dual-Agility Method: The Value of Method Engineering. *Journal of Database Management (JDM)*. Vol.16 (4), pp.1-24
DOI: 10.4018/jdm.2005100101 [Hämtad 2017-05-08]
- Hoda, R., Noble, J. & Marshall, S. (2010). The impact of inadequate customer collaboration on self-organizing Agile teams. *Information and Software Technology*. Vol. 53, pp.521–534.
DOI: 10.1016/j.infsof.2010.10.009. [Hämtad 2017-02-08]
- Kautz, K., Heide Johanson, T. & Uldahl, A. (2014). The Perceived Impact of the Agile Development and Project Management Method Scrum on Information Systems and Software Development Productivity. *Australasian Journal of Information Systems*. 01 November 2014. Vol.18 (3).
DOI: 10.3127/ajis.v18i3.1095 [Hämtad 2017-04-08]
- Kniberg, H. & Skarin, M. (2009). Kanban vs Scrum: How to Make the Most of Both. C4media: <http://www.infoq.com/minibooks/kanban-Scrum-minibook>
[Hämtad 2017-05-16]

Lei, H., Ganjeizadeh, F., Jayachandran, P. K. & Ozcan, P. (2015). A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer Integrated Manufacturing*. February 2017. Vol. 43, pp.59-67.
DOI: 10.1016/j.rcim.2015.12.001 [Hämtad 2017-03-28]

Misra, S., Kumar, V., Kumar, U., Fantazy, K. & Akhter, M. (2012). Agile software development practices: evolution, principles, and criticisms. *International Journal of Quality & Reliability Management*. Vol. 29(9), pp.972–980.
DOI 10.1108/02656711211272863. [Hämtad 2017-02-08]

Nikitina, N., Kajko-Mattsson, M. & Stråle, M. (2012). From Scrum to Scrumban: A case study of a process transition. *International Conference on Software and System Process*. June 2012, pp. 140–149
DOI: 10.1109/ICSSP.2012.6225959 [Hämtad 2017-05-09]

Papatheocharous, E. & Andreou, A.S. (2014). Empirical evidence and state of practice of software agile teams. *Journal of software: Evolution and process*. Vol. 26(9), pp.855–866.
DOI: 10.1002/smr.1664. [Hämtad 2017-02-08]

Permana, P. (2015). Scrum Method Implementation in a Software Development Project Management. *International Journal of Advanced Computer Science and Applications*. 01 September 2015. Vol. 6(9), pp.198–204.
DOI: 10.14569/IJACSA.2015.060927 [Hämtad 2017-02-08]

Pham, A. & Pham, P-V. (2012). *Scrum in action: Agile software project management and development*. 1 Uppl. Boston: Course Technology Inc.

Pomar, F.A., Calvo-Manzano, J., Caballero, E. & Arcilla-Cobián, M. (2014). Understanding sprint velocity fluctuations for improved project plans with Scrum: a case study. *Journal of Software: Evolution and Process*. September 2014, Vol.26 (9), pp.776-783.
DOI: 10.1002/smr.1661 [Hämtad 2017-04-08]

Pries, K. H. & Quigley, J. M. (2011). *Scrum project management*. 1 Uppl. Florida: CRC Press.

Qumer, A. & Henderson-Sellers, B. (2008). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and Software Technology*. Vol.50 (4), pp.280–295.
DOI: 10.1016/j.infsof.2007.02.002 [Hämtad 2017-03-30]

Røvik, K.A. (2000). *Moderna organisationer: trender inom organisationstänkandet vid millennieskiftet*. 1 Uppl. Malmö: Liber.

Schwaber, K. & Sutherland, J. (2013). The Scrum guide - The Definitive Guide to Scrum: The Rules of the Game. <http://Scrum.org/>
[Hämtad 2017-02-08].

Scrum Alliance. (2016). Learn about Scrum.
<https://www.Scrumalliance.org/why-Scrum>
[Hämtad 2017-05-12]

Serrador, P. & Pinto, J.K. (2015). Does Agile work? - A quantitative analysis of agile project success. *International Journal of Project Management*. July 2015, Vol.33(5), pp.1040-1051
DOI: <http://dx.doi.org/10.1016/j.ijproman.2015.01.006>

Sjøberg, D. I. K., Johnsen, A., & Solberg, J. (2012). Quantifying the Effect of Using Kanban versus Scrum. *IEEE Software*. Vol. 29(5), pp.47–53.
DOI: 10.1109/MS.2012.110 [Hämtad 2017-05-23]

Softhouse. (2014). Scrum på fem minuter.
https://issuu.com/softhouse/docs/Scrum_broschyr_sve_ny_121029
[Hämtad 2017-04-12]

Stepanek, G. (2012) *Software Project Secrets: Why Projects Fail*. 1 Uppl. California: Apress.

Stoica, M., Ghilic-Micu, B., Mircea, M. & Uscatu, C. (2016). Analyzing Agile Development – from Waterfall Style to Scrumban. *Informatica Economica*. Vol.20 (4/2016). pp.5-14.
DOI: 10.12948/issn14531305/20.4.2016.02 [Hämtad 2017-04-13]

Tonnquist, B. (2016). *Projektleddning*. 6 uppl. Stockholm: Sanoma Utbildning.

VersionOne. (2013). 10th annual state of agile development survey.
<http://stateofagile.versionone.com/>
[Hämtad 2017-02-08]

Ågerfalk, P. Fitzgerald, B. Slaughter, S. (2009). Introduction to the Special Issue - Flexible and Distributed Information Systems Development: State of the art and Research Challenges. *Information Systems Research*, 2009, Vol. 20(3), pp. 317–328.
DOI: 10.1287/isre.1090.0244 [Hämtad 2017-02-12]

Wang, X., Conboy, K. & Cawley, O. (2012). "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development. *The Journal of Systems & Software*. June 2012, Vol.85 (6), pp.1287-1299
DOI: 10.1016/j.jss.2012.01.061 [Hämtad 2017-05-09]

Williams, L. (2012). What Agile Teams Think of Agile Principles. *Communications of the acm*. April. Vol. 55(4), pp.71–76.
DOI:10.1145/2133806.2133823 [Hämtad 2017-03-30]

Bilagor

Intervjuguide

Förklaring:

Vi kommer, med din tillåtelse, att spela in intervjun. Du får själv klargöra vilken grad av anonymitet du vill ha såsom företag och namn. Det skulle dock vara behjälpligt om du ville förmedla vilken roll du har i företaget. Du kan när som helst avbryta intervjun av vilken anledning som helst och utan förklaring. Empirin kommer endast att hanteras av Edvard Lindgren och Samuel Andersson som underlag till denna kandidatuppsats och inget annat.

1. Vilken är din roll och vilka ansvarsområden innefattar den?
 - a. Kan den benämnas som Scrum Master, eller motsvarande?
2. Är du utbildad eller certifierad inom Scrum?
3. Hur ser teamet ut och vilka definierade roller har ni?
 - a. Hur stora är de?
 - b. Vem bestämmer över hur arbete ska bedrivas?
 - c. Använder ni fasta team? Skulle du beskriva de som tvärfunktionella?
4. Hur sker samarbetet med produktägaren ut?
 - a. Hur involverad är denne i arbetet?
5. Kan du beskriva hur en vanlig arbetsdag ser ut i teamet?
 - a. Hur hanteras backlogen?
 - b. Hur delas arbetsuppgifter ut?
 - c. Hur definieras ”done” i teamet?
 - d. Använder ni Daily Scrum?
6. Hur ser upplägget för era sprintar ut?
 - a. Hur startas en sprint och hur planeras den?
 - b. Använder ni estimeringar och velocity?
 - c. Hur långa är era sprintar?
 - d. Använder ni sprintmål?
7. Hur avslutas sprintar?
 - a. Används Sprint review?
 - b. Sprint Retrospective?
8. Hur ser ni på förändringar under sprintens gång? Tillåts det?
9. Har ni arbetat med andra metoder och experimenterat med dessa?
10. Är anpassningen medveten eller har det arbetats fram på något sätt?
11. Tror du att det är något specifikt i er organisation som gör att det ser ut på detta sätt?