



**LUNDS**  
UNIVERSITET

# Scrum ur ett SWOT-perspektiv

*En analys av Scrum-metodikens,  
styrkor, svagheter, möjligheter och hot.*

Jesper Sahlman  
Teja Trenkler

Ekonomihögskolan  
Institutionen för Informatik  
2014

Vi vill rikta ett särskilt tack till:

Vår handledare: Magnus Wärja

Och till våra intervjupersoner:

Jannika Björkholm

Fredrik Grenson

Magnus Månsson

Daniel Terborn

# Abstrakt

<b>Titel</b>	Scrum ur ett SWOT-Perspektiv
<b>Författare</b>	Jesper Sahlman Teja Trenkler
<b>Utgivare</b>	Institutionen för informatik
<b>Handledare</b>	Magnus Wärja
<b>Examinator</b>	Markus Lahtinen Umberto Fiaccadori
<b>Publiceringsår</b>	2014
<b>Uppsatsnivå</b>	Kandidat
<b>Nyckelord</b>	Projektledning, Scrum, Systemutveckling, Samarbete
<b>Abstrakt</b>	<p>Agil systemutveckling har utvecklats till en stark trend de senaste åren. Scrum är den mest tillämpade Agila projektledningsmetodiken idag, Scrum hyllas från håll och kanter. Syftet med vår uppsats är att skapa en bättre helhetsbild och ta reda på vilken bild Scrum som projektledningsmetod får genom en aktuell SWOT analys. Vilka är egentligen Scrums styrkor? Svagheter? Möjligheter? och Hot? För att bättre fastställa dessa startade vi vår forskning utifrån tidigare publicerad forskning. Med denna forskning som grund intervjuade vi fyra stycken projektledare med Scrum erfarenhet. Uppsatsens slutsats är en samlad helhetsbild avseende Scrum som projektledningsmetodik som inkluderar Styrkor, Svagheter, Möjligheter liksom Hot.</p>

# Innehållsförteckning

<b>1. INTRODUKTION .....</b>	<b>8</b>
1.1 Problemformulering .....	9
1.2 Syfte .....	9
1.3 Forskningsfråga.....	9
1.4 Avgränsningar.....	9
<b>2. LITTERATURGENOMGÅNG OCH TEORETISKA UTGÅNGSPUNKTER .....</b>	<b>11</b>
<b>2.1 Scrum .....</b>	<b>11</b>
2.1.1 Sprintar.....	13
2.1.2 Scrum flödet, User Stories och Product backlog .....	13
2.1.3 Scrum Roller .....	14
2.1.3.1 Produktägaren .....	14
2.1.3.2 The Team .....	15
2.1.3.3 ScrumMaster .....	15
2.1.4 Scrum evenemang.....	16
2.1.4.1 Sprint planning meeting .....	16
2.1.4.2 Daily Scrum .....	16
2.1.4.3 Sprint Review.....	16
2.1.4.4 Sprint Retrospective .....	16
2.1.5 Scrum Artifacts .....	17
2.1.5.1 Product Backlog .....	17
2.1.5.2 Burndown Chart .....	17
2.1.5.3 Sprint Backlog .....	17
2.1.5.4 Sprint Taskboard.....	17
<b>2.2 Tidigare publicerad forskning.....</b>	<b>18</b>
2.2.1 Samarbete .....	18
2.2.1.1 Utmaningar i utförandet av arbetsuppgifter i grupp.....	19
2.2.1.2 Teamprestationer .....	20
2.2.1.3 Det delade beslutsfattande .....	20
2.2.1.4 Förändrade krav för tanke och beslut .....	20
2.2.2 Kravhantering.....	21
2.2.2.1 Förändring av kravspecifikationen.....	21
2.2.2.2 Resursallokering .....	21
2.2.3 Kommunikation .....	22
2.2.3.1 Generell Kommunikation.....	22
2.2.3.2 Kommunikation mellan Projektledare, Scrum Master och teamet.....	22
2.2.3.3 Kommunikation och kravspecifikation .....	22
2.2.3.4 Kommunikation och planering .....	22
2.2.3.5 Kommunikation med andra team, och kunden.....	23
2.2.3.6 Kommunikation och distribuerade team.....	23

2.2.3.7	Sociokulturella skillnader.....	24
2.2.3.8	Dokumentation.....	24
2.2.4	Störningar.....	25
2.2.4.1	Kunder .....	25
2.2.4.2	Management .....	25
2.2.4.3	Externa källor.....	25
2.2.4.4	Processer .....	26
2.2.4.5	Kommunikation .....	26
2.2.4.6	Kundinblandning.....	26
<b>2.3</b>	<b>Sammanfattning .....</b>	<b>26</b>
2.3.1	Styrkor .....	26
2.3.2	Svagheter.....	27
2.3.3	Möjligheter.....	27
2.3.4	Hot.....	28
<b>3.</b>	<b>METOD.....</b>	<b>30</b>
3.1	Källor .....	30
3.2	Intervjuer.....	30
3.3	Semistrukturerade intervjuer.....	31
3.4	Etik .....	32
3.5	Urval av företag och intervjupersoner.....	33
3.6	Kvalitativ Analys av intervjuer.....	34
3.7	Kvalitetsaspekter på genomförd forskning.....	34
3.8	Kritik till valet av metod.....	35
3.9	SWOT.....	35
3.9.1	SWOT-Analys applicerad på Scrum .....	37
3.10	Frågor för intervju.....	37
<b>4.</b>	<b>EMPIRI.....</b>	<b>39</b>
4.1	Styrkor.....	40
4.1.1	Enkelhet och snabbriklighet .....	40
4.1.2	Kravhantering utforskande och lärande.....	40
4.1.3	Bättre planering och förståelse .....	41
4.1.4	Initiativ och gemensamt ansvar .....	41
4.1.5	Det bättre samarbetet.....	41
4.1.6	Mindre dokumentation och ökad kunskapspridning.....	42
4.1.7	Effektiv kommunikation .....	42
4.1.8	Mötesstrukturen och rollerna .....	42

<b>4.2</b>	<b>Svagheter .....</b>	<b>43</b>
4.2.1	Situationsberoende .....	43
4.2.2	Att arbeta iterativt med Scrum. ....	43
4.2.3	Teamet och gruppdynamik.....	43
4.2.4	ScrumMaster och samordning .....	44
4.2.5	Product Owner och kunden .....	44
4.2.6	Inbyggda mekanismer .....	44
<b>4.3</b>	<b>Möjligheter.....</b>	<b>45</b>
4.3.1	Kravhantering och förankring.....	45
4.3.2	Tillgänglighet, samarbete och engagemang.....	45
<b>4.4</b>	<b>Hot .....</b>	<b>46</b>
4.4.1	Externa krav, missförstånd och förväntningar .....	46
4.4.2	Erfarenheter och roller.....	46
4.4.3	Kultur och överblickbarhet.....	46
<b>4.5</b>	<b>Empiri tabell sammanfattning.....</b>	<b>47</b>
<b>5.</b>	<b>ANALYS.....</b>	<b>51</b>
<b>5.1</b>	<b>Styrkor.....</b>	<b>52</b>
5.1.1	Kundtillfredsställelse .....	52
5.1.2	Det bättre samarbetet.....	53
5.1.3	Motivation, snabbhet och teamets gemensamma segrar .....	54
5.1.4	Utforskande, lärande och kunskapspridning .....	54
5.1.5	Produktionsflödet.....	55
5.1.6	Scrum struktur och roller .....	55
<b>5.2</b>	<b>Svagheter .....</b>	<b>56</b>
5.2.1	Väldigt kommunikationsberoende .....	56
5.2.2	Sprintar.....	56
5.2.3	Ledarskap .....	57
5.2.4	The Team.....	57
5.2.5	Flera teams, och distribuerade team. ....	57
5.2.6	Resurser.....	58
5.2.7	Product Owner och kund .....	58
5.2.8	Övriga aspekter .....	59
<b>5.3</b>	<b>Möjligheter.....</b>	<b>59</b>
5.3.1	Kundtillfredsställelse .....	59
5.3.2	Engagemang och förankring.....	59
<b>5.4</b>	<b>Hot .....</b>	<b>60</b>
5.4.1	Kundens förväntningar och andra externa inflytanden .....	60
5.4.2	Kunden & produktägare.....	60
5.4.3	Missförstånd & Samordning.....	61
5.4.4	Team och partnersamarbeten.....	61
5.4.5	Kultur.....	61
5.4.6	Organisation och övriga hot .....	62

---

<b>6. SLUTSATS</b> .....	<b>64</b>
6.1 Styrkor .....	64
6.2 Svagheter .....	65
6.3 Möjligheter .....	66
6.4 Hot .....	66
6.5 Begränsningar .....	67
6.6 Förslag till fortsatt forskning .....	67
<b>7. BILAGOR</b> .....	<b>68</b>
Bilaga A1. Affecto Intervjutrascript .....	68
Bilaga A2. Cybercom Intervjutrascript .....	75
Bilaga A3. Eniro Intervjutrascript .....	82
Bilaga A4. Stretch Intervjutrascript .....	93
Bilaga B – Frågor för intevju .....	104
Referenser .....	106

# 1. Introduktion

Agile är ett samlingsnamn för ett antal olika metoder för att utveckla mjukvara. Metoderna har namn som eXtreme Programming, DSDM, Crystal Clear och Scrum. Agile är inte helt enkelt översatt men det betyder bland annat snabb, flexibel, lätttrölig, vig och välkoordinerad. Gemensamt för Agile är fyra värderingar som Agile-metoderna vilar på nämligen; värdera individer och interaktion högre än processer och verktyg, värdera fungerande programvara högre än omfattande dokumentation, värdera samarbete med kunden högre än att förhandla om kontrakt, värdera att anpassa sig till förändringar högre än att följa en uppgjord plan (Björkholm & Brattberg, 2010).

Problem i IT-utveckling uppstår ofta. IT utveckling kännetecknas av sju stycken drag som gör att de löper en större risk för misslyckande än andra projekt, varav tre av dem är; de omfattas av abstrakta begränsningar som ofta resulterar i orealistiska och överambitiösa projekt. IT projekten är svårare att visualisera, de är svåra att presentera på ett begripligt sätt för intressenter som ett led av förväntningar och överambitiösa mål, vilket leder till att problem upptäcks sent. Och för det tredje så omfattas IT projekt av en stor osäkerhet i kravspecificeringen och i de oförutsägbara problem som uppstår i samband med implementationen av det specificerade systemet (Al-Ahmad et al, 2009).

Enligt Standish Group, som sedan 1985 sammanställt utgången av 90 000 IT projekt världen över (60 % i USA, 25 % i Europa och kvarvarande 15 % resten av världen). I deras rapport ”Chaos Manifesto 2013” visades att 39 procent av projekten 2012 var lyckade. Medan 43 procent av projekten utmanades, de levererades försent, över budget eller med bristande funktionalitet. Och 18 procent misslyckades; de avslutades innan projektets färdigställdes eller levererades utan att senare komma till användning av kunden (The Standish Group, 2013). Även ifall statistiken ovan är skrämmande så har utvecklingen gått framåt och är bättre än tidigare år. The Standish Group rekommenderar ”Think Big, Act Small”, små projekt som genomförs i små team har en mycket hög sannolikhet till att lyckas, medan stora projekt så gott som alltid misslyckas. Stora projekt bör därför brytas ner till små. Agila processer bör användas istället för vattenfallsmetodiken genom att systemutvecklingsprojekten delas in i små systemutvecklingsprojekt. Där mjukvaran utvecklas i små team genom iterativa processer med kontinuerlig feedback på ”Steppingstones”, en liten men signifikant leverans som kan utvärderas visuellt eller ”hands on”. Agila processer är den sjätte viktigaste faktorn som har störst inflytande för framgång av projekt. Den andra viktigaste faktorn är interaktionen med slutanvändarna som är central i Agila metoder (The Standish Group, 2013).

Den mest kända och använda agila metoden är **Scrum** (Björkholm & Brattberg, 2010).



## 1.1 Problemformulering

Första gången vi kunde läsa om Scrum var 1986 då Harvard Business Review skrev in Scrum i historien genom sin artikel ”The New New Product Development Game”. Artikeln beskriver hur Honda, Canon och Fujitsu-Xerox 1986 skapade historia genom sin teamfokuserade ”All-at-once product development”. Metoden betonar vikten av ”Empowered”, självstyrande team och tar även upp managements ansvar i utvecklingsprocessen. Och sedan Ken Schwaber 1995 publicerade sin första skrift om Scrum 1995 har Scrum vuxit och blivit hyllad för sina fördelar med att bidra till en mjukvaruutveckling som befriar team från det konstanta sorlet av kundmissnöje och andra egenskaper som kännetecknar misslyckade projekt. Tillämpningen av Scrum har istället fått team att uppleva en ny verklighet med projekt som genererar det som kunder vill ha, med bättre avkastning, minskade kostnader, snabba resultat, självförtroende i en komplex utvecklingsvärld och inte minst mera glädje (Rubin, 2013).

Det finns mycket skrivet om alla fördelar som Scrum för med sig men mindre forskning som ger en rättvis bild av Scrum i sin helhet. Vid en sökning via Lubsearch på ”Swot Scrum” så framkommer en träff gällande en fallstudie gjord 2010. Vi finner vidare få vetenskapliga träffar när vi söker på ämnet via Google och inte någon utförlig rapport.

## 1.2 Syfte

Vårt syfte är att göra ett bidrag till forskningen som finns inom Scrum genom att vi dels använder tidigare kunskap i sökandet efter ny för att skapa en bättre helhet kring kunskapen som finns gällande Scrum med hänsyn till Scrums Styrkor, Svagheter, Möjligheter och Hot.

## 1.3 Forskningsfråga

Vilken bild får Scrum som metod genom ett aktuellt SWOT perspektiv? Vilka är Scrums Styrkor? Svagheter? Möjligheter? och Hot egentligen? Vi ämnar belysa denna SWOT matris genom att studera tidigare teori och forskning som skrivits inom ämnet samt intervjua fyra stycken verksamma projektledare med Scrum erfarenhet.

## 1.4 Avgränsningar

Vi kommer i vår undersökning att rikta oss mot fyra stycken projektledare med Scrumerfarenhet för att få deras bild av Scrums effekter på systemutvecklingen. Vi kommer till följd av antalet intervjuade inte rangordna Styrkorna, Svagheter, Möjligheterna och

Hoten. Detta eftersom antalet intervjuade som undersökning medger inte skulle göra en sådan rangordning giltig.

## 2. Litteraturgenomgång och teoretiska utgångspunkter

Detta kapitel är indelat i tre delar. I första delen beskrivs och förklaras Scrum på en övergripande nivå, terminologin och grundprinciper som är centrala för att få bättre förståelse för vår bedrivna forskning och dess resultat. För den som redan är insatt i Scrum så går det bra att gå direkt till den andra delen. Den andra delen (2.2) består av tidigare publicerad forskning i form av artiklar som beskriver kunskaper och erfarenheter som erhållits genom tidigare tillämpning av Scrum, både fördelaktiga och icke-fördelaktiga som är ett viktigt led för den tredje sammanfattande delen (2.3) där bilden av Scrum med hänsyn till Styrkor, Svagheter, Möjligheter och Hot presenteras genom en sammanfattande SWOT modell som finns som stöd för forskningen.

### 2.1 Scrum

Ordet Scrum härstammar från sporten Rugby. Det syftar på situationen under spelavbrott, med andra ord då bollen har gått ut ur spel. "Du har förmodligen sett en Scrum där två grupper av forwards klumpar ihop sig runt rugbybollen, med låsta armar och huvudet nere, kämpande för att vinna bollinnehavet" (Rubin, 2008 s 3).



Scrum vilar på det Agila manifestet vilket definierar de riktlinjer och principer (Agile Manifesto) som publicerades 2001. Scrum vilar på följande principer:

1. Kundtillfredsställelse genom att snabbt leverera användbar mjukvara
2. Välkomna förändrade krav, även sent i utvecklingen
3. Fungerande mjukvara levereras ofta (hellre veckor än månader)
4. Fungerande mjukvara är den principiella måttstocken för framsteg
5. Hållbar utveckling. Att vara kapabel att bibehålla ett konstant tempo.
6. Nära, dagligt samarbete mellan affärssidan och utvecklare
7. Ansikte-mot-ansikte konversation är den bästa formen av kommunikation
8. Projekt är konstruerade kring motiverade individer, som bör litas på.
9. Kontinuerlig uppmärksamhet för teknisk förträfflighet och god design
10. Simplicitet, konsten att maximera mängden arbete som är avslutat, är grundläggande.
11. Självorganiserande team
12. Regelbundet anpassande till förändrande omständigheter (Holzman och Panizel, 2013).

Scrum är ett sätt att snabbt utforska nya tillvägagångsätt, nya idéer och att snabbt lära sig vilka lösningar som är framgångsrika och vilka som inte är det. Denna process sker genom en kontinuerlig feedback av pågående arbete och pågående loop-lärande, vilka är centrala för Scrum (Rubin, 2008). Förändring är normen inom Scrum, det är inte möjligt att "förutsäga iväg" osäkerhet, vilket försöks göras när det läggs ner avsevärda resurser och tid på planering innan starten av projektet. Scrum ger möjligheten att tillfredsställa kunden genom att ge kunden vad den verkligen vill ha, inte enbart vad kunden har specificerat vid en tidpunkt då den visste minst om de verkliga behoven (Rubin, 2008). Det är därför bättre att dela in arbetet i mindre delar som ger många fördelar: (1) minskad cykeltid, (2) minskad variabilitetsflöde, (3) påskyndad feedback, (4) minskad risk, (5) minskat "overhead", (6) ökad motivation och snabbhet, (7) minskat kostnads och tidsåtgång (Rubin, 2008).

Heje och Heje, (2011) skriver i sin forskning att det finns nio svar till frågan om varför scrumprojekt fungerar:

1. Scrum kan bygga upp relationer och nätverk både inom och utanför projektteamet,
2. Scrum har mekanismer för att bygga förtroende, även på distans
3. Scrum ger projektteamet ett gemensamt språk och ett gemensamt mål att sträva efter
4. Scrum är mycket användbart för att koordinera arbetet inom projektgruppen
5. Scrum möjliggör ett antal "Boundary objects", och "Boundary Spanning Roles", för insamling av extern information.
6. Scrum har en struktur för möten som fungerar väl för kommunikation inom projektgruppen
7. Scrum tillämpar enkla men effektiva mekanismer för att övervaka projektets framgång
8. Scrum har inbyggda mekanismer för att säkerställa kvalitet
9. Användningen av Scrum motiverar gruppen och ger gruppen energi

Rubin (2008) lyfter vidare att Scrum kan utmärkt användas i situationer som kännetecknas av komplexitet; eftersom en sådan miljö kräver kreativitet och innovativa tillvägagångssätt genom en säker testmiljö för experimentering, genom sökandet och användandet av viktig information. I dessa miljöer är interaktioner och effektiv kommunikation självklar. Scrum medför de möjligheter som behövs för att upptäcka dysfunktioner och slöseri som förhindrar att organisationer kan nå sina verkliga potentialer (Rubin, 2008).

### 2.1.1 Sprintar

Sprintar är centrala i Scrum för scrumprojekt. Sprintar är sättet som projektarbetet är organiserat på. De har en viss varaktighet, de har ett mål som inte bör ändras när en sprint väl har påbörjats. Vilket är den grundläggande förutsättningen för det ömsesidiga engagemanget som utgörs av teamet och produktägaren. Teamet förpliktar sig till att uppnå målet mot slutet av sprinten och produktägaren förpliktar sig till att inte förändra målet under sprintens gång. Enbart ytterligare förtydligande är tillåtet, ifall omständigheterna verkligen kräver det (Rubin, 2008). Sprintars slutstadium är ett "done" som har definierats av teamet som är en checklista innehållande arbetet som teamet förväntas att färdigställa.

Varje sprintcykel är kopplad till en timebox som är upp till en månad lång. Timeboxing är viktigt till av flera anledningar: (1) det begränsar mängden pågående arbete, vilket reducerar risken för finansiell förlust. (2) de tvingar fram prioritering, så att det viktigaste arbetet utförs först. (3) ökar motivation till att färdigställa och validera viktiga delar av arbetet innan datumet för sprintens slut. (4) onödigt perfektionism undviks. (5) Motiverar till avslut, aktiviteter får en känsla av "urgency" och färdigställs snabbare än när de har en känd tidsfrist. (6) medför även en förbättrad förutsägbarhet. Korta sprintar resulterar även i (a) enklare planering; snabb återkoppling, (b) förbättrad avkastning på investeringar, (c) begränsar fel; hur mycket fel kan ett team orsaka under en två veckor lång sprint? (d) ökad spänning; det ligger i människans natur att entusiasm och spänning minskar desto längre vi måste vänta på tillfredsställelse, (e) frekventa kontrollpunkter; att tillåta flertalet kontrollpunkter för projektets framgång (f) förenklar planering, teamet blir van vid att en viss press gäller för tid och leverans (Rubin, 2008).

I slutet av en sprint cykel, en iteration så bör teamet ha en potentiellt lanserbar produkt eller del av en produkt som kan användas ifall det är lämpligt. I fallet att en iteration inte är tillräcklig, så kan en uppsättning av funktioner från flera iterationer släppas tillsammans. När varje iteration avslutas, så påbörjas hela processen av att planera nästa iteration (Rubin, 2008).

### 2.1.2 Scrum flödet, User Stories och Product backlog

Ett Scrum-projekt startar med en vision av systemet som ska utvecklas. Produktägaren är ansvarig för processen som innebär att de som tillhandahåller projektets medel får leverera sin vision på ett sätt som maximerar intressenternas ROI. Produktägaren formulerar en plan, så att

den inkluderar en Product Backlog. Product Backlog är en lista av funktionella och icke-funktionella krav som när de omvandlas till funktionalitet kommer att leverera visionen. Product Backloggen är prioriterad så att de objekt, som med högsta sannolikhet kommer att generera nytta, är högst upp placerade på listan och att objekten är indelade i föreslagna utgåvor. Den prioriterade Product Backloggen är en startpunkt; innehållen, prioriteringarna och grupperingen av Product Backloggen ändras vanligtvis samma ögonblick som projektet påbörjas. Förändringar i Product Backloggen reflekterar förändrade krav i verksamheten som kommer fram och hur fort eller långsamt teamet kan omvandla Product Backloggen till funktionalitet (Schwaber, 2004).

Detaljerna om projektkraven kommer från kontinuerligt pågående diskussioner och förhandlingar. Scrum bemöter krav som en viktig grad av frihet som kan förändras för att nå verksamhetens mål. Om ny information tyder på att fördelarna i relation till kostnaderna har blivit ogynnsamma, är det möjligt att ta bort de kraven från produkten. Inom Scrum, så istället för att skapa en stor lista med olika krav innan projektets början, skapas istället objekt i en Product backlog. Varje objekt i en Product backlog representerar önskad nytta för verksamheten. I början finns det väldigt få detaljer kopplade till dessa objekt (Rubin, 2008).

Produktutvecklingsobjektet är resultatet av konversationer med intressenter, produktägaren och utvecklingsteamet. User stories används ofta som ett praktiskt sätt att uttrycka den önskade nyttan för verksamheten genom de olika typer av objekt som finns i Product backlogen som exempelvis funktioner. Kraven kommer från en pågående dialog och konversationer som senare översätts till detaljerade krav. Stories som används för sprintar behöver vara små, testbara och ha genomtänkta kriterier för acceptans (Rubin, 2008).

### **2.1.3 Scrum Roller**

Det finns tre roller inom Scrum: Product Owner, The Team, och ScrumMaster. Allt ledningsansvar är delat mellan dessa tre roller. Människorna som ingår i dessa roller är de som har förpliktat sig till att utföra projektet. Det finns möjligen en andra grupp som har intresse i projektet men som inte är direkt knutna till det. Scrum ser en tydlig skillnad mellan dessa två grupper och säkerställer att de som har ansvaret för projektet även har auktoriteten att göra vad som krävs för projektets framgång. De som inte bär ansvaret har inte heller möjligheten att påverka projektet i onödan (Schwaber, 2004).

#### **2.1.3.1 Produktägaren**

Produktägaren bär ansvaret för att representera intressen av alla de som bidrar till projektets utveckling och dess resulterande slutprodukt. Produktägaren ansvarar för de inledande och pågående aktiviteterna för projektet sker genom skapandet av de inledande, övergripande kraven, avkastningen på investeringen (ROI), mål och lanseringsplaner. Listan med kraven kallas för Product Backlog och produktägaren är ansvarig för att utforma denna för att säkerställa att den mest värdefulla funktionaliteten produceras först och sedan byggs på. Detta

uppnås genom att prioritering av Product Backlog görs, de mest värdefulla kraven placeras först i kön inför nästa iteration. (Schwaber, 2004)

### **2.1.3.2 The Team**

Team är små, upp till 7 personer självstyrande, självorganiserande och tvärfunktionella team som är ansvariga för att utveckla funktionaliteten. Teamet är även ansvarigt för att lista ut hur Product Backloggen blir tillväxt av funktionalitet inom en iteration samt att styra sitt egna arbete för att åstadkomma detta. Medlemmarna delar kollektivt på ansvaret för framgången av varje iteration samt ansvaret för hela projektet (Schwaber, 2004). I självstyrande team, har teammedlemmarna ansvaret för inte att enbart leverera resultat men också att hantera, styra och överblicka sina egna prestationer. I självstyrande team blir beslutsfattandet som en konsekvens av detta gjort av en person med nyckel-kunskaper, färdigheter och förmågor som är relaterade till ett specifikt problem som möter teamet i olika sammanhang och i arbetet de möter (Moe et al, 2011). Självstyre stöds och främjs genom att fem faktorer finns för att främja och stödja arbetet: (1) ett klart och motiverande mål, (2) en struktur som främjar prestation, (3) En organisatorisk stödjande kontext, (4) Expert kunskap för coachning vid behov, (5) tillgång till rätt resurser. Anpassningsbarhet är också viktigt eftersom teamet behöver löpande genomföra operationella beslut medan strategiska beslut ska tas så sent som möjligt, för att teamet ska kunna hålla sig flexibla i en marknad som ständigt förändras. I ett självstyrande team är teamet slutligt ansvarigt för den delade beslutsprocessen på den operationella och den taktiska nivån men Produktägaren ska också vara involverad eftersom han eller hon är den som äger affärsperspektivet (Moe et al, 2011). I icke-Scrum team så är det traditionellt projektledaren som talar om för teamet hur de ska utföra arbetet (Schwaber, 2004).

### **2.1.3.3 ScrumMaster**

Är den som är ansvarig för Scrum-processen dvs. (1) för undervisning och tillämpningen av Scrum till alla som är inblandade i projektet vilket också innebär implementering av Scrum så att den passar in i organisationens kultur. Vidare (2) Att scrum levererar den förväntade nyttan för verksamheten. Och (3) för att säkerställa att alla följer Scrums regler och praxis.

ScrumMastern fyller positionen som i normala fall hade tagits av projektledaren. Till skillnad från den traditionella projektledaren som är ansvarig för att definiera och leda arbetet, så är ScrumMastern ansvarig för att leda Scrum-processen. ScrumMasterns auktoritet är till största del indirekt, då den bygger huvudsakligen på ScrumMasterns kunskap av Scrums regler och praxis och hans arbete för att säkerställa att dessa följs. ScrumMastern är ansvarig för projektets framgång och han hjälper till att öka sannolikheten för framgång genom att hjälpa produktägaren att välja ut det mest värdefulla ur Product Backlog. ScrumMastern hjälper teamet att omvandla Product Backlog till funktionalitet ScrumMastern är enbart en facilitator (Schwaber, 2004).

## **2.1.4 Scrum evenemang**

### **2.1.4.1 Sprint planning meeting**

Allt arbete utförs genom Sprintar. Varje Sprint är en iteration som varar upp till 30 dagar. Varje Sprint börjas med ett Sprint Planning Meeting, där produktägaren och teamet möts för att samarbeta och bestämma vad som kommer göras under nästa Sprint. För att säkerställa att teamet får tid till att genomföra själva arbetet och inte lägger all tid till att fundera på arbetet, får ett Sprint Planning Meeting inte vara längre än åtta timmar. Under första halvan av mötet väljs det som prioriteras högst ut ur Product Backloggen. Sedan berättar produktägaren för Teamet vad som är önskat och Teamet i sin tur berättar hur mycket av det önskade de tror att de kan omvandla till funktionalitet under kommande Sprint. Under den andra halvan av mötet planerar teamet själva Sprinten. På grund av att Teamet är ansvarigt för att övervaka och styra sitt eget arbete så behöver teamet en preliminär plan för att starta Sprinten. Uppgifterna som planen består av finns inskrivna i en Sprint Backlog. Starten av andra delen av planeringsmötet är även den officiella starten på själva Sprinten (Schwaber, 2004)

### **2.1.4.2 Daily Scrum**

Nästa punkt är dagliga Scrum-mötet. Dessa 15-minuters dagliga möten där Teamet samlas och där varje medlem ska besvara tre frågor. Frågorna är:

1. Vad har du gjort sedan förra Daily Scrum?
2. Vad planerar du att göra mellan detta möte och nästa?
3. Vilka potentiella utmaningar är i vägen för dig för att du ska uppfylla din förpliktelse till denna Sprint och detta projekt?

Syftet med detta möte är att dagligen synkronisera alla medlemmars arbete samt att schemalägga eventuella möten som Teamet behöver för att göra framsteg (Schwaber, 2004)

### **2.1.4.3 Sprint Review**

Vid slutet på varje Sprint så hålls ett Sprint Review-möte. Detta är ett fyra timmar långt (time-boxed) möte där Teamet presenterar vad som har utvecklats under Sprinten för produktägaren och de eventuella intressenterna som vill delta i mötet. Syftet med detta möte är att föra samman människor och att hjälpa dem att gemensamt bestämma vad som Teamet bör göra härnäst (Schwaber, 2004).

### **2.1.4.4 Sprint Retrospective**

Efter Sprint Review och innan nästa Sprint Planning Meeting, håller ScrumMastern ett Sprint Retrospective-möte med Teamet. Under detta tre timmar långa (time-boxed) möte, uppmuntrar ScrumMastern teamet att revidera, med hjälp av Scrum-processens ramverk och



praxis sin utvecklingsprocess för att göra denna mer effektiv och angenäm inför nästa Sprint (Schwaber, 2004)

## **2.1.5 Scrum Artifacts**

### **2.1.5.1 Product Backlog**

Kraven för systemet eller produkten som utvecklas i projektet bli samlade i en Product Backlog. Produktägaren är ansvarig för innehållet, prioritering och tillgängligheten av Product Backloggen. Product Backlog får aldrig ses som komplett. När den används i en projektplan så är det enbart för att göra preliminära uppskattningar av kraven. Product Backlog utvecklas i samma takt som produkten och i miljön som den kommer att användas i. Product Backlog är dynamisk med andra ord så ändras den hela tiden av ledningen för att identifiera vad produkten behöver för att vara lämplig, konkurrenskraftig och användbar. Så länge som produkten existerar så existerar även dess Product Backlog (Schwaber, 2004).

### **2.1.5.2 Burndown Chart**

Detta diagram visar mängden arbete som återstår i relation till tid. Burndown diagrammet är ett utmärkt sätt att visualisera ett samband mellan mängden arbete som återstår vid en slumpmässig tidpunkt samt teamets framsteg när det gäller att reducera detta arbete. Skärningspunkten mellan trendlinjen av återstående arbete och den horisontella axeln indikerar den mest sannolika tidpunkten då arbetet kommer att vara avslutat. Burndown diagrammet är en bild mellan verklighet och vad man har planerat eller hoppats på (Schwaber, 2004)

### **2.1.5.3 Sprint Backlog**

Denna Backlog definierar arbetet eller uppgifterna som Teamet har definierat för att omvandla Product Backloggen för Sprinten till ett increment av en potentiellt lanserbar produktfunktionalitet. Teamet sätter samman en preliminär lista av dessa uppgifter under andra halvan av Sprint Planingsmötet. Uppgifter bör delas, så att varje uppgift inte tar mer än 4-16 timmar att färdigställas. Uppgifter som tar längre tid än så bör betraktas endast som platshållare för uppgifter som inte har definierats på ett tillräckligt lämpligt sätt. Endast Teamet kan genomföra ändringar i Sprint Backloggen. Sprint Backloggen är mycket synlig, realtids bild av arbetet som Teamet planera att utföra under en Sprint (Schwaber, 2004)

### **2.1.5.4 Sprint Taskboard**

En av agiles principer som mycket ofta återkommer är att visualisering av problem och status. Visualiseringen av dessa aspekter har syftet att underlätta processen av att komma överens om vad nästa steg är. Att fysiskt synliggöra problem och status kan också påverka teamets effektivitet då teamet inte behöver störas av utomstående när dessa vill informera sig om projektets status (Brattberg och Björkholm, 2010).

## 2.2 Tidigare publicerad forskning

Nedan presenteras tidigare publicerad forskning i form av artiklar som beskriver kunskaper och erfarenheter som erhållits genom tidigare tillämpning av Scrum, både fördelaktiga och icke-fördelaktiga. Den tidigare forskningen har delats in under lämpliga rubriker och ett fåtal ytterligare underrubriker som varit gemensamt för artiklarna.

Författare	Samarbete	Kravhantering	Kommunikation	Störningar
Cho (2008)	X		X	X
Heje och Heje (2011)			X	
Holmstöm et al (2006)			X	
Holzman och Panizel (2013)			X	
Lee (2012)	X		X	
Moe et al (2009)	X	X		
Moe et al (2011)	X			
Tanner et al (2013)				X
Vlanderer et al (2011)	X		X	
Williams (2011)	X			

Tabell 1 - Författare

### 2.2.1 Samarbete

Moe et al (2009) skriver i sin forskning om det delade beslutsfattandet som gör att den agila mjukvaruutveckling står i kontrast mot traditionell mjukvaruutvecklingen. Agil utveckling kräver att teamet klarar av att anpassa sig till att hantera ett samarbetande arbetssätt och genomgå snabba beslutsprocesser. Ledarskap blir en delad funktion, viktiga beslut kring vad som behöver göras och hur saker och ting ska göras sköts genom ett samarbete av många personer som påverkar varandra, inte enbart av en person. Denna miljö av samarbete är mer komplicerad jämfört med traditionellt beslutstagande. Agil management betonar vikten av beslutsfattande som en delad beslutsprocess och självstyre ”self-management” (Moe et al, 2011). Samarbete ska alltid uppmuntras eftersom möjligheter till återanvändning och integration ökar. Sådana möjligheter kan upptäckas redan på ett tidigt stadium när managers och utvecklare pratar med varandra, vilket i sin tur kan skapa mjukvaruprodukter av högre

kvalité (Vlaanderen et al 2010). Cho (2008) fann även i sin forskning att regelbundna möten är ett av de starkaste argumenten för användningen av Scrum. Trots fördelarna så finns även baksidor; tidsåtgången, bokning, planering av möten och inte minst all den tid som krävs i möten för att planera och gå genom arbetet. Dessutom krävs ofta fungerande tekniska lösningar och kommunikationsplattformar för att hålla igång mötena. Miljön med mycket samarbete gör även inslag på mindre möjligheter till en privat sfär för arbete, team medlemmarna är konstant utsatta för ett brus av kommunikation även när de fokuserar på egna arbetet (Cho, 2008).

Kollegors förtroende för varandra är en central del i Scrum i möjliggörandet att uppnå en effektiv; kommunikation, feedback och återkoppling på varandras arbete samt möjligheten att uppnå ett framgångsrikt ledarskap. Scrum ger tyvärr inte möjligheten till några goda råd för hur detta ska uppnås och för hur team ska sätta det delade ledarskapet till verket (Moe et al, 2009).

Moe et al (2009) fortsätter i sin forskning och visar även att en isomorph projekt struktur; specifik arbetsuppgiftsfokus från början till slut hindrar samarbete, eftersom detta innebär att det blir svårare för teamet att dela förtroende, mentala modeller, kommunikation, koordinera arbete och att backa upp varandra. Teamet behöver utbildas och tränas i varandras arbetsuppgifter och självstyre ska alltid uppmuntras, ”alla blir en manager”. Utvecklingsprocessen ska anpassas för att möjliggöra effektivt arbete, genom att ge utrymme för lärande och reflektion (Moe et al, 2009).

Williams (2011) undersökning av tre separata team på Microsoft visade att team som använde Scrum i kombination med Microsoft Engineering metoder visade sig uppnå bättre kvalitet, produktivitet, och tidsestimering i jämförelse med liknande team som inte använder Scrum (Williams 2011).

### **2.2.1.1 Utmaningar i utförandet av arbetsuppgifter i grupp**

Moe et al (2009) fann flera utmaningar i deras forskning gällande utförandet av arbetet i team. (1) Det fanns ofta en utmaning i det delade-beslutsfattandet på den operationella nivån och den huvudsakliga anledningen var att teamet inte konfronterade varandra, de ville inte argumentera och ifrågasätta varandra utan fann det viktigare att vara överens och som en konsekvens så blev beslutsfattandet lidande. (2) Teamet saknade teamkänsla och kunskap om teamet; vilket resulterade i orealistiska planer. (3) Beslutsuteslutande (Decision hijacking); vissa medlemmar i teamet gjorde beslut utan att informera de andra i teamet. (4) Brist på kunskap och technocracy vilket innebar att när teamet inte hade rätt kunskap så kunde de inte ge råd och feedback till andra när de diskuterade problem eller hur en uppgift ska utföras. (5) Konsekvenserna av technocracy; vilket innebar att expertkunskap gjorde att super beslutstagande tog form, vissa kunniga utvecklare gjorde besluten och uteslöt andra från den delade beslutsprocessen. Visserligen så innebär inte delat beslutsfattande att alla i teamet var involverade i beslutstagandet men alla viktiga kommande beslut ska kommuniceras till hela

teamet och teamet måste avgöra om vilka beslut som behöver tas tillsammans (Moe et al, 2009).

### **2.2.1.2 Teamprestationer**

Team prestationer är en komplicerad process och prestationer beror inte endast på kompetensen som finns i teamet för att styra och utföra arbetet utan även den organisatoriska kontexten som kommer från management (Moe et al, 2009). Lee (2012) skriver om det effektiva teamet i sin forskning, vilken utförs av självstyrda team som är beroende av ett effektivt ledarskap med starka interpersonella och grupp planeringsfärdigheter till exempel externa förmågor som informationssökande och informationsgivande. De behöver även i hög grad möjligheten att påverka och förhandla – ”vilket är en hörnsten av vad som gör Scrum så framgångsrik” (Lee, 2012). Moe et al (2009) påvisade också att team ofta hade problem med att slutföra backlogen, följa sprintplanen, uppleva oproduktiva möten, utvecklare hade också en tendens av att rapportera till Scrum Mastern om problem som inte var relaterade till det planerade arbetet. Vidare visade deras studie att team medlemmar ofta prioriterade individuella mål över teamets mål, teamen hade inte enbart problem med kommunikation men också feedback, koordinering av uppgifter och samarbete (Moe et al, 2009).

### **2.2.1.3 Det delade beslutsfattande**

Moe et al (2009) forskade djupare in i beslutstagandeprocessen och fann tre olika motgångar för delat beslutsfattande (1) uppdelning av den strategiska projektplanen till handlingsplan; även ifall Scrum stödjer det så var det svårt att sätta det i verket till följd av att medlemmar hade olika mentala modeller över vad som är menat att bli levererat. Teamet och Produktägaren förstod inte komplexiteten i de kraven som projektet hade. (2) Det fanns inget forum för att lösa konflikter för beslut på strategisk, taktisk och operationella projekt nivåer. Det fanns vidare en konstant konflikt mellan behovet för långsiktiga mål på strategisk nivå och kortsiktiga mål, de kortsiktiga målen var alltid prioriterade (behovet av att visa framskridande) vilket resulterade i att kvalitén blev lidande. (3) Organisatoriska missanpassningar, Scrum teamet blev inte inbjudna att vara involverade i de strategiska besluten (Moe et al, 2009).

### **2.2.1.4 Förändrade krav för tanke och beslut**

I jämförelse med traditionell ”kommando och kontroll” orienterade miljöer så fungerar Scrum på ett nytt sätt för att ta beslutstagande gällande operationella problem och osäkerheter. ”Teamet är berättigad full auktoritet till att ta de beslut de finner nödvändiga och behöver göra för att nå sina mål” (Moe et al, 2009). För att möjligheten till att arbeta agilt med självstyre skall finnas så kan det ibland vara nödvändigt att förändra organisationers kulturer, strategi, och strukturer. Sådana förändringar kan ibland kräva flera års arbete för att genomföras (Moe et al, 2009). Att förändra arbetssättet är ibland en svår process och när det (1) omfattar en omvandling från en specialiserad kompetens till en redundant organisering av funktioner. Vidare (2) att gå från en rationell till en naturalistisk beslutsprocess (beslutsfattande baserad på erfarenhet), krävs ett nytt tillvägagångssätt inte bara av utvecklarna men också av

management. Om management inte är kapabla att ge en tydlig riktning till teamet då är sannolikheten stor att individuella mål kommer att prioriteras framför gruppens mål (Moe et al, 2009).

## 2.2.2 Kravhantering

Projektledare bör lägga mycket uppmärksamhet på hur de ska dela ner projektet i arbetsuppgifter, delegera arbetsuppgifterna till rätt personer, erkänna behovet av att koordinera och fördela arbetet. För att koordinationen ska bli framgångsrik är det därför viktigt att lägga mycket uppmärksamhet och arbete kring Product backlog, sprint backlog, Scrum board och de dagliga Scrum mötena (Heje och Heje, 2011).

Moe et al (2009) fann att sprintarna ofta saknade en tydlig definition av ”Done”, och Scrum Masters hade en benägenhet att vilja få teamet att stå i bättre dager genom att markera sprintarna som färdiga för tidigt. De fann vidare en effekt av orealistisk backlogs, utvecklarna fann inte möjligheten för teamet att engagera sig i teamets mål och började därför att helt fokusera på sina egna prioriteringar och mål som en konsekvens av en saknad tydlig riktlinje (Moe et al, 2009).

### 2.2.2.1 Förändring av kravspecifikationen

Moe et al (2009) fann vidare att mjukvarututvecklingsprojekten inte följde Scrum metoden som initialt menat. Enligt Scrum ska mål och strategier som förändras mitt i en sprint innebära att sprinten avbryts. Detta gjordes emellertid aldrig även ifall förändringar kom med i bilden. Moe et al (2009) fann vidare i sin forskning att teamen ofta var utsatta för konflikter gällande intressen, vilket innebar ett behov av att visa kortsiktiga resultat för de kortsiktiga målen (att visa framsteg) och de långsiktiga målen som behövs för att säkerställa kvalitet i lösningarna. Partnersamarbeten var också ofta ett problem eftersom partners kunde ses som en konkurrent och planer och mål var därför inte exponerade helt eller för sent, vilket ibland ledde till stora problem i utvecklingen och i kommunikationen (Moe et al 2009).

Moe et al (2009) fann vidare i deras fall studie att det ofta fanns brister i definitionen ”Done” och att Scrum Mastern regelbundet ville få teamet att stå i bättre dager än vad det verkligen gjorde. Det fanns också konflikter i intressen gällande prioriteringar inom företaget, vilket innebar att projektet ofta förlorade resurser vilket gjorde teamets självstyre omöjligt. Och slutligen fann Moe et al (2009) effekten av orealistisk sprint backloggar, att utvecklarna inte prioriterade teamets mål utan fokuserade istället på individuella mål. Ett resultat av att en klar styrning saknades (Moe et al 2009).

### 2.2.2.2 Resursallokering

Resursallokering kan ibland vara problematiskt för att nå kravspecifikationen. Moe et al (2009) fann in sin forskning att sprintarna var orealistiskt planerade, vilket orsakade problem i att planera resurser, eftersom ingen vara säker på att det var genomförbara mål för teamet. De huvudsakliga orsakerna för orealistiska backlogs var; viktiga beslut var inte koordinerade med

underleverantörer, vilket innebar att de hade fått för mycket ansvar för leveransen av funktionaliteten. Vidare fann de ofta brister i tydligheten av prioriteringen av arbetet som en konsekvens så innebar detta att sprint planeringsmötena togs upp av alltför mycket tid åt att diskutera omfattningen av projektet, kundbehov, funktioner och slutligen så fanns det inte tid till att planera den kommande sprinten. Moe et al (2009) fann vidare konflikter gällande prioriteringen inom företaget och projektet, projektet förlorade resurser till följd av att företagets övriga verksamhets prioriteringar gjorde sig gällande. Teamet saknade tillräckligt stöd från sin organisation och tillräckliga resurser för att genomföra projektet, vilket gjorde att teamet hade svårt att arbeta efter det önskade självstyret ”self-mangement” (Moe et al, 2009).

## **2.2.3 Kommunikation**

### **2.2.3.1 Generell Kommunikation**

Holzmann och Panizel (2013) forskning visar att utvecklingen av en effektiv kommunikation är central för teamet. Scrum team måste uppnå effektivitet i kommunikation med varandra för att nå framgångsrika resultat i projekten. Projektledare måste av den anledningen ha kommunikativa färdigheter för att se svagheter i kommunikationen och att öka sannolikheten till framgångsrika projekt (Holzman och Panizel, 2013).

### **2.2.3.2 Kommunikation mellan Projektledare, Scrum Master och teamet.**

För att kommunikationen ska vara framgångsrik är principen kring 15-minuters Scrum möten viktiga även ifall det är nödvändigt att använda e- eller telefonmöten för att uppnå mötena. Product Owners och Scrum Masters måste kunna verkställa ”broar för kommunikation” för att förbättra kvalitén och effektivitet i kommunikationen. Vidare bör kontrollen av projekten göras genom projekt milstolpar (sprintar), regelbunden leverans, snabb feedback, regelbundna möten och regelbundna rapporter om prestationer (Heje och Heje, 2011).

### **2.2.3.3 Kommunikation och kravspecifikation**

Vid uppbyggnaden av kravspecifikationen är en rik kommunikation med intressenter, Product Owner och team medlemmar viktiga för insamlingen av kunskap gällande (1) kund perspektivet, där definitionen av kvalité (Quality of service) är definierad som rör kundens uppfattning om vad som är hög kvalité för denne, (2) Affärsperspektivet; där kraven som gäller strukturen av affärsprocesserna är definierade; (3) de anställdas perspektiv, där kraven gällande kunskaper och upplevelser blir definierade och också kunskapen som behövs enligt de tidigare processerna definieras, (4) produkt perspektivet, där kraven gällande specifikationen av den slutliga produkten är definierad (Lee, 2012).

### **2.2.3.4 Kommunikation och planering**

För att skapa en känsla av ett ”team” så är det viktigt med en väl genomförd Scrum planering. Denna planering kan behöva göras med hjälp av en hemsida för att publicera projektets

framgång och för att underlätta en ”Vi” känsla och förbättra koordinationen och kommunikationen (Holmström et al, 2006).

### **2.2.3.5 Kommunikationen med andra team, och kunden**

I fall av större projekt finns det också ett behov av kommunikation mellan team, eftersom en brist på kommunikation kan leda till dubbelt arbete. Potentiella problem kan uppstå när kommunikationen är fokuserad kring möten. Detta kan resultera i att det blir svårare att involvera kunden, eftersom kunden kan ha andra åtaganden och möten som kunden måste prioritera (Cho, 2008).

### **2.2.3.6 Kommunikation och distribuerade team**

När projekten är gjorda på distans, mellan olika länder, tror de anställda ofta att deras arbeten kan gå förlorade, de upplever ofta en känsla av förlorad kontroll och de räds ofta att bli tvingade till att flytta för att behålla arbetet. Vidare finns det kulturella problem som kan uppstå, som attityder till hierarkier, känsla av tid, kommunikationsstil och behovet för struktur skiljer sig mellan olika länder, vilket kan leda till missförstånd (Holmström et al, 2006). Hultström et al (2006) skriver om samarbetet i team som är separerade från varandra och betonar svårigheten att behålla en god kontroll, koordination och kommunikation. En klar nackdel var teamets påverkan av de tidsskillnader som olika länder kan påverkas av, vilket ofta innebär att delar av teamet får en känsla av att vara ”efter” eller ”stå om intet” om viktig information gällande samarbete, utveckling och framgång (Holmström et al, 2006).

Agila metoder kan användas för att minska de negativa inflytande som distansarbeten kan orsaka gällande kommunikation, koordination och kontroll. Tidsmässiga skillnader kan ofta vara utmanande i aspekterna av att kontrollera projekt och möjligheten att skapa förtroende och en känsla av ett ”team” i projekt som sträcker sig mellan olika platser. Ofta blir det så sätt att det uppstår goda relationer på de högre nivåerna i organisationen men utvecklarna får sällan en chans till att mötas. Trots tekniska hjälpmedel så uppstår ofta en känsla av att det är två team och inte ett (Holmström et al, 2006).

De agila problemen med distribuerade team är ofta associerade med kommunikation och relationer mellan människor, ifrån perspektivet av Project Management Institute (PMI); (1) är det ibland svårt att bilda en delad förståelse för projektets omfattning, (2) Det är svårare att erkänna din kollega och ge denne beröm när han eller hon gör ett utmärkt arbete. (3) det är svårare att kommunicera på distans och att bygga förtroende. (4) Mera fokus behövs för att dela ner projektet i en struktur som lämpar sig åt ett distribuerat team. (5) kvalitén kan ibland vara ett problem eftersom förväntningarna och uppfattning på kvalitet kan skiljas åt mellan olika kulturer och länder, och kan således göra kvalitetsaspekten svårare att hantera (Heje och Heje, 2011).

Vidare från perspektivet från International Project Management Association (IPMA) så är det i huvudsak de beteendemässiga aspekterna som bör tas i aktning. Framförallt (1) sprida

positiv energi och entusiasm på avstånd kan vara svårt. Vidare att (2) utveckla teamet, utveckla en känsla och team spirit när människor inte träffas. Och att (3) försäkra att kommunikationen inom projekt teamet är effektiv eftersom det är mera utmanande när kommunikationen sker på distans mellan olika kulturer (Heje och Heje, 2011).

### **2.2.3.7 Sociokulturella skillnader**

Detta handlar om de komplicerade mönster som omfattar kultur, nationella kulturer, språk, politik, och människors individuella ambitioner och arbetsetik som kan även de vara ett hinder för teamets framgång (Holmström et al, 2006). Holmström et al (2006) understryker även i sin forskning de problem som olika språk kan innebära genom missförstånd och förvirring. Och kultur kan vara ett problem, framförallt i länder där det är oförskämt att säga ”nej” när ett ”ja” är betydligt mera lämpligt svar. Religiösa skillnader kan också innebära svårigheter om de kommer in och utmanar människor i det dagliga arbetet (Holmström et al, 2006).

Det kan vara möjligt att underlätta och kringgå den sociala dimensionen av de problem som arbete i distribuerade team medför genom all hand meetings och att låta människor med speciella Scrum roller till exempel Scrum masters träffas och bilda relationer mellan de olika delarna av teamen kan underlätta förtroende och samarbete (Heje och Heje, 2011).

### **2.2.3.8 Dokumentation**

Scrum, precis som andra Agila mjukvaruutvecklingsmetoder reducerar signifikant mängden dokumentation. Ofta är dokumentation placerad som kommentarer i koden i mjukvaruutvecklingen istället för i riktig dokumentation. Bristen på dokumentation kan bli ett påtagligt problem när nya utvecklare ansluter sig till teamet eller när utvecklare som arbetar på nya delar som de inte har någon tidigare erfarenhet av. En utvecklare uttryckte detta som en betydande svaghet i Cho (2008) forskning. Agila metoder medför snabbare och effektivare hantering av arbete så länge som den som utvecklar har en mycket bra förståelse för vad kunden behöver, annars blir det långsamma metoder eftersom de kräver omfattande merarbete till följd av behovet av förtydligande. En annan svaghet vid bristande kommunikation är att viktig kunskap stannar hos vissa individer och om individerna av någon anledning skulle lämna projektet så kan detta innebära att viktig kunskap går förlorad. Syftet med att reducera dokumentationen är att göra projektmedlemmar lika genom att de delar kunskaper med varandra men detta sätt att hindra att kunskap går förlorad på är ofta inte tillämplig i verkligheten (Cho, 2008)

Vlanderer et al (2010), visade i sina fallstudier att det ofta fanns problem med att täcka kravspecifikationen och att hålla alla uppgifter uppdaterade i sprint loggen i den detaljerade dokumentationen. Visserligen är en av Scrums principer att individer och interaktion går före viktiga processer och administration, vilket i sin tur innebär att så länge som projektarbetet blir gjort, blir projektadministrationen mindre viktig. Den administrativa processen för överblick är ändå viktig och produkt management sprinten, vilken hanterar definitionen kring kravspecifikationen som ligger som grund för resursallokeringen. Denna bör



påbörjas vid ungefär halva tiden innan den ”verkliga” produktutvecklingssprinten väl påbörjas, för att med större säkerhet inte några sena förändringar kommer att ske i sprinten när den väl startats (Vlaanderen et al, 2010).

## **2.2.4 Störningar**

Tanner och Maureen (2013) forskning rör de förändringar som sker i mitten av en sprint, Scrum Masters är inte benägna att välkomna förändringar som sker i mitten av en sprint eftersom sprintar ska inte ha någon extern påverkan när de väl påbörjats (Tanner och Maureen, 2013). Sådana förändringar har en tendens till att orsaka osäkerhet och rastlöshet för projektteamet. Trots detta så sker de ändå och då som ett led i bristande planering, integration i teamet och i förståelsen för vad projektgruppen verkligen är ansvarig för (Tanner och Maureen, 2013). Vidare identifierade de sex olika störningar i deras forskning; kunder, management, extern påverkan, processer, kommunikation och erfarenhet (Tanner och Maureen, 2013).

### **2.2.4.1 Kunder**

Kunder vet ofta vilket mål de har med uppdraget men de är ofta osäkra på kraven för hur de ska nå uppdragets mål. Till följd av detta behöver utvecklare, förändra, göra om och förändra planeringen av utvecklingen, vilket orsakar stora förluster av viktig tid och resurser i projektet. Det händer också att den överliggande sprint ”historien” är vag som gäller för sprintarna vilket innebär att förändringar behöver göras i mitten av en sprint. Därmed går viktig arbetstid och resurser förlorade (Tanner och Maureen, 2013).

### **2.2.4.2 Management**

Ledningen har ibland en bristande förståelse och har en tendens till att störa projektets gång. En bristande förståelse sker då management inte fullt förstår Scrum teamet och processerna som omfattar projektet. Medan en störning sker när management avbryter Scrum teamet i sitt arbete (Tanner och Maureen, 2013).

### **2.2.4.3 Externa källor**

Dessa källor av störningar sker utanför Scrum teamet. Dessa bör hanteras av Scrum Mastern, för att försäkra att dessa störningar inte sker i mitten av en sprint. Ibland är det omöjligt att hindra dessa externa störningar eftersom de ligger utanför Scrum Masterns kontroll och inkluderar störningar från miljön, psykologiska, sociala, och organisatoriska influenser. Ett exempel på en extern störning kan vara oförutsägbara buggar som behöver lösas i utvecklingen (Tanner och Maureen, 2013).

#### **2.2.4.4 Processer**

Det omfattar processer inom Scrum teamet och inom företaget i stort. Källor av störningar i mitten av en sprint kan vara förlusten av resurser som viktig personal kan medföra som lämnar projektet och företaget. Sådana förluster kan ofta innebära stora störningar för utvecklingen av sprintarna. Denna typ av störningar är de vanligaste förekommande i mitten av en sprint (Tanner och Maureen, 2013).

#### **2.2.4.5 Kommunikation**

Scrum team och management måste göra sig säkra på att de har de relevanta verktyg som behövs för att kommunicera effektivt. Ofta uppstår problem till följd av ineffektiv kommunikation som ibland är associerade med skillnader i språk kultur och olika tidszoner (Tanner och Maureen, 2013).

#### **2.2.4.6 Kundinblandning**

Kundens inblandning ses ofta som en nyckel till framgång i projekt. Inte bara i Scrum utan även i agil utveckling generellt. Cho (2008) skriver att utvecklare ofta finner detta problematiskt eftersom kunder ofta blir involverade i utvecklingen för sent. Problemen uppstår då ofta som följd av att kunderna har problem med att finna utrymme i kalendern till följd av andra åtaganden. Andra problem med kund inblandning är bristen på specifikation och osäkra kund krav, vilket innebär omfattande merarbete (Cho, 2008).

## **2.3 Sammanfattning**

Nedan följer den sammanfattande delen av litteraturkapitlet. Innehållet är indelat i SWOT-kategorierna; Styrkor, Svagheter, Möjligheter och Hot. Slutligen presenteras detta i en modell.

### **2.3.1 Styrkor**

Scrum har flera styrkor; Scrum stödjer människor och interaktionen mellan dessa, Scrum: (1) Ger kundtillfredsställelse genom att snabbt leverera användbar mjukvara. (2) Välkomnar förändrade krav, även sent i utvecklingen, (3) Nära dagligt samarbete mellan affärssidan och utvecklare. (4) Uppmuntrar användandet av de bästa kommunikationsvägarna "ansikte mot ansikte" (Holzman och Panizel). (5) Regelbunden effektiv mötesstruktur (Cho, 2008) (Heje och Heje, 2011). (6) Minskar de negativa inflytande som distansarbete kan orsaka gällande kommunikation, koordination och kontroll (Holmström et al, 2006). (7) medför snabbare och effektivare hantering av arbete (Cho, 2008). (8) Samarbete uppmuntras och återanvändning och integration ökar (Vlanderer et al, 2010). (9) Medför mekanismer för att bygga förtroende,

även på distans. (10) motiverar och ger energi till teamet. (11) ger projektteamet ett gemensamt språk och ett gemensamt mål att sträva efter. Scrum är iterativt, och arbetet sker i mindre omgångar medför. (12) reducerad cykeltid. (13) reducerat variabilitetsflöde. (14) snabbare, och mer frekvent feedback. (15) reducerad risk. (16) reducerad overhead. (17) reducerad tids- och kostnadsökning (Rubin, 2008). Vidare medför Scrum (18) möjligheter att snabbt utforska nya tillvägagångsätt och idéer i effektivt lärande (kontinuerlig loop-lärande) (Heje och Heje, 2011). Scrum använder sig utav Timeboxing vilket i sin tur medför vissa interna fördelar så som: (19) det begränsar mängden pågående arbete, vilket bidrar till att (20) reducera risken av finansiell förlust, (21) tvingar prioritering av arbete, vilket innebär att det viktigaste arbetet utförs först (22) undviker onödig perfektionism.

### 2.3.2 Svagheter

Scrumprojekten har visats sårbara och har problem att hantera brister i följande: (1) Kommunikationen; (a) är av stor vikt för att skapa ett fungerande, grupparbetande självstyrande ”team”, (b) i formuleringen kravspecifikation, mellan team och kund för utvecklingen av projektet, (c) i distribuerade team för att öka ”team känslan”, sprida positiv energi, skapa förtroende och kommunicera teamets prestationer (Lee, 2012). Vidare (2) det delade beslutfattandet; fungerar inte detta så kan det medföra brister i leveransen (Moe et al, 2009). (3) Avsaknaden av ett forum för att lösa konflikter (Holmström et al, 2006) (4) En bristande definition av ”done”; teamen vet inte när en sprint är färdig ”done”; vilket skapat förvirring och osäkerhet (Moe et al 2009). (5) Beroende av ett effektivt informationsorienterat ledarskap; individuella mål kan prioriteras om inte klara direktiv till teamet ges (Lee, 2012). (6) De frekventa mötena; kan innebära att stor tidsåtgång och ibland har de även visats oproduktiva (Cho, 2008). (7) Kravspecificeringen; är central för utvecklingen och kan medföra allvarliga problem för projektet om den inte är genomtänkt (Vlaanderen et al, 2010). (8) Product Backlogs; planeringen av resurser görs gentemot denna och brister kan givetvis ställa till förödande konsekvenser (Moe et al 2009). (9) Med nära samarbete begränsas möjligheten till en privat sfär för arbete och teammedlemmar är konstant utsatta för ett brus av kommunikation (Cho, 2008). Vidare så kan (10) bristande kommunikation leda till dubbelt arbete (Cho, 2008). (11) Konflikter kan uppstå på grund av behovet av att visa kortsiktiga resultat för de kortsiktiga målen och att visa de långsiktiga målen som behövs för att säkerställa kvalitet i lösningarna (Moe et al, 2009). (12) I distribuerade team, trots tekniska hjälpmedel, uppstår ofta en känsla av att det är två team och inte ett (Holmström et al, 2006). (13) Om utvecklare inte har en mycket bra förståelse för kundens behov, är det en långsam metod eftersom den kräver omfattande merarbete, till följd av behovet av förtydligande (Cho, 2008).

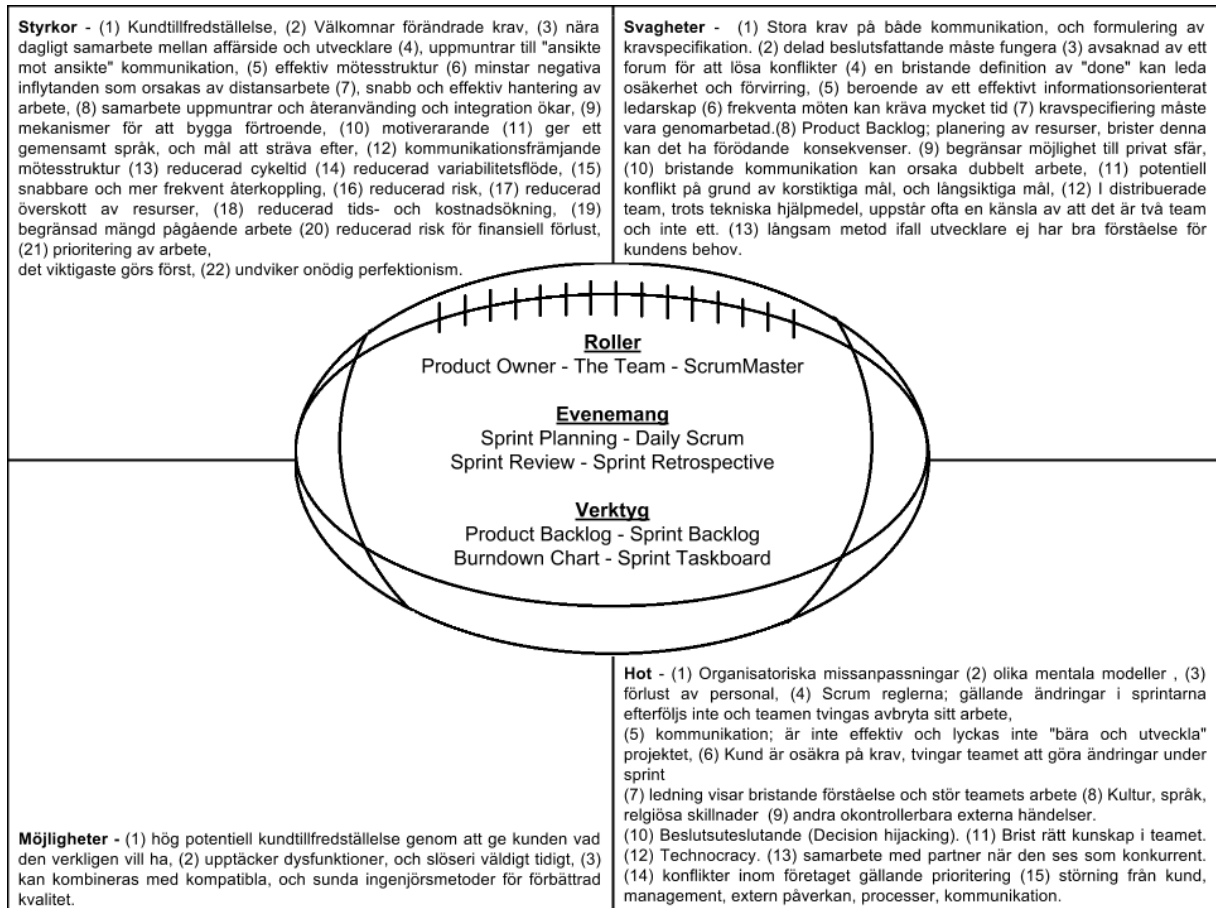
### 2.3.3 Möjligheter

Användningen av Scrum visade sig ha flera positiva möjligheter som skulle kunna utnyttjas. Då Scrum involverar kunden i hög utsträckning, genom kontinuerlig kommunikation och återkoppling kan nya förutsättningar snabbt anpassas till utvecklingen, detta har som följd att;

(1) hög potentiell kundtillfredsställelse uppnås genom att ge kunden vad den verkligen vill ha (Rubin, 2008).-(2) Scrum kan hjälpa att upptäcka dysfunktioner och slöseri tidigt (Rubin, 2008) och åtgärda fel. (3) Om Scrum används i kombination med sunda ingenjörsmetoder visade det sig också frambringa bättre resultat för densiteten av fel (Williams, 2011)

### 2.3.4 Hot

Scrum projektens framgång har visats i teorin kunna hotas av flera faktorer: (1) Organisatoriska missanpassningar; Scrum teamet blev inte inbjudna att vara involverade i de strategiska besluten. (Moe et al, 2009). Vidare (2) den strategiska planen; ofta är det svårt att dela ner denna i en handlingsplan som projektgruppen ska arbeta efter (Moe et al, 2009).(3) Förlust av personal (Cho, 2008). (4) Scrum reglerna; gällande ändringar i sprintarna efterföljs inte och teamen tvingas avbryta sitt arbete och planera om eller göra om sitt arbete (Vlaanderen et al, 2010). (5) Kommunikationen; är inte effektiv och lyckas inte ”bära och utveckla” scrumprojekten (Heje och Heje, 2011). (6) Kunder; är osäkra på kraven för att nå sina mål, teamet blir tvingade att göra förändringar mitt i sprinten (Tanner och Maureen, 2013). (7) Management; visar en bristande förståelse för teamet och stör eller avbryter det pågående arbetet (Tanner och Maureen, 2013). (8) Kulturer, språk och religiösa skillnader kan ibland även ha en negativ inverkan (Holmström et al, 2006). (9) Andra okontrollerbara externa händelser; som inte är möjliga att planeras för (Tanner och Maureen, 2013). (10) Beslutsuteslutande (Decision hijacking); vissa medlemmar i teamet gjorde beslut utan att informera de andra i teamet (Moe et al, 2009). (11) Brist på kunskap när teamet inte hade rätt kunskap så kunde de inte ge råd och feedback till andra när de diskuterade problem eller hur en uppgift ska utföras (Moe et al, 2009). (12) Konsekvenserna av technocracy; vilket innebar att expertkunskap gjorde att superbeslutstagande tog form; vissa kunniga utvecklare gjorde besluten, uteslöt andra från den delade beslutsprocessen (Moe et al, 2009). (13) Partnersamarbeten, partners kunde ses som en konkurrent och planer och mål var därför inte exponerade helt eller för sent, vilket ibland ledde till stora problem i utvecklingen och i kommunikationen (Moe et al, 2009). (14) konflikter i intressen gällande prioriteringar inom företaget, vilket innebar att projektet ofta förlorade resurser, vilket gjorde teamets självstyre omöjligt (Moe et al, 2009). (15) Ytterligare en störning kan vara från processer,(Tanner och 2013).



Figur 1 – Sammanfattande SWOT-Matris

## 3. Metod

I kapitlet nedan presenteras aspekter, tillvägagångssätt och metodik som använts för vår forskning samt en beskrivning av de intervjupersoner och företag vi valt för vår forskning.

### 3.1 Källor

Vi har varit noggranna med att läsa genom tidigare forskning för gemensamma mönster och olikheter. Då den tidigare forskningen har haft stor betydelse för vår forskningsfråga så har vi lagt stor vikt vid att försäkra oss om att den använda teorin har uppfyllt kraven för reliabilitet; källan mäter det den utger sig för att mäta, valid dvs. den är väsentlig för vår frågeställning och att den är reliabel dvs. fri från systematiska fel (Eriksson, 2001). Källans kvalitet har varit central för oss att avgöra. Vi har fastställt källans kvalitet genom en bedömning av vilken kunskap och kompetens som författaren till källan har. Detta tillvägagångssätt rekommenderas av Jacobsen (2002). Vidare har urvalet av teori och artiklar varit ganska stort, vilket gjort att de olika källorna har balanserat varandra. Bortsett från vetenskapliga artiklar har vi även använt oss av två böcker författade av Schwaber (2003) och Rubin (2013). Dessa författare är erkända och författare som har stora erfarenheter av Scrum och balanserar varandra bra. Balansering av använda källor är vidare rekommenderat av Jacobsen (2002).

### 3.2 Intervjuer

Vi ville med vår forskning få en djupare förståelse, för att få en ökad möjlighet till att beskriva detaljerna och helheten i sammanhanget, därför var den kvalitativa metoden lämpligast. Kvalitativa metoder är lämpliga för att utforska områden, där lite är känt eller mycket är känt för att skapa en mer rik förståelse (Straus och Corbin, 1998). Kvalitativ forskning kännetecknas i hög grad av närhet, tillit och förståelse mellan forskare och undersökta personer (Holme och Solvang, 1997). Kvalitativa analyser ger vidare möjlighet att öka förståelsen av komplexa sammanhang, vilka blir möjliga att förstå och utifrån en analys av enskilda fall, går det att bygga upp modeller som belyser fenomenet från ett nytt perspektiv (Lanz, 2007).

Det finns flera olika sätt att genomföra intervjuer på per post, per telefon, via ett personligt möte eller i en kombination. För vår forskning var besöksintervjuer lämpligast eftersom besöksintervjuer innebär fördelar då intervjun rör komplicerade frågor och medför fördelar för kontroll över intervjusituationer; för att lättare följa upp på varandras frågor, för att skapa förtroende och för möjligheten att använda kroppsspråket för att följa ansiktsuttryck och svar. Nackdelen med besöksintervjuer är att de kan påverkas av intervju-effekten d.v.s. att vi påverkar varandra (Eriksson, 2001). Detta har vi försökt kringgå genom att hålla oss till frågorna som skickats innan intervjun såvida frågorna i intervjun behövt förtydligande eller motiverade följdfrågor. Eftersom vissa av våra frågor var av en mer komplicerad art så skickade vi frågorna en tid innan intervjun. Vilket är ett bra sätt att förbereda intervjupersoner på komplicerade frågor som kräver reflektion och eftertanke (Eriksson, 2011).

En väl genomförd intervju kännetecknas av att den ger information som är tillförlitlig och ett giltigt resultat som är användbart för andra (Lanz, 2007). Vi har i våra intervjuer varit måna om att skapa en atmosfär där den intervjuade känt sig trygg att tänka fritt och uttrycka sina tankar så att svaren inte skulle avvika från en annan situation. Detta tillvägagångssätt underlättar för oss att få reda på det subjektiva, vilken mening en individ ger ett fenomen (Lanz, 2007). Vi har använt oss av den riktade öppna intervjuformen. Eftersom när vi använder denna intervjuform så får vi en bra grund för att söka efter en persons upplevelser av ett fenomenets kvaliteter (Lanz, 2007). Vi genomförde den större delen av intervjun genom att en vid fråga ställdes; som respondenten senare fick fördjupa sig i och reflektera sina tankar och åsikter inom.

Våra intervjuer har varit öppna eftersom vi varit intresserade av att få den intervjuade att fritt utveckla sina tankar, sätt att uppfatta ett fenomen, resonera med sig själv och beskriva det sammanhang som han eller hon anser är betydelsefulla för beskrivning av ett fenomen. Vilket öppna intervjuer är lämpade för (Lanz, 2007).

### **3.3 Semistrukturerade intervjuer**

Forskningens intervjuer genomfördes på intervjupersonernas arbetsplats och vi använde oss av ett intervjuformulär som vi sammanställt efter att vi genomgått tidigare relevant forskning inom vårt forskningsområde av intresse. Våra intervjuformulär vi använde oss av var gjorda i förväg. På så sätt kan forskaren i hög grad undvika styrning av samtalet från forskarens sida (Holme och Solvang, 2001).

Vi började våra frågor med några öppna generella frågor om företaget och om personens bakgrund. Vi ville på så sätt göra en ”mjuk” start och bygga upp ett förtroende för den intervjuade som skulle spegla av sig på resten av intervjun, så att den intervjuade skulle känna

sig bekväm att svara öppet och reflektera fritt i fortsättningen av intervjun, då våra frågor krävde mera eftertanke. Den inledande fasen är mycket betydelsefull för den kommande intervjun eftersom det är under denna som arbetsrelationen mellan intervjuare och respondenten upprättas (Lanz, 2007).

Intervjuerna strukturerades efter den intervjumall som vi tagit fram innan intervjun med speciellt fokus på SWOT modellen; Styrkor, Svagheter, Möjligheter och Hot. Eftersom vårt mål var att den intervjuade skulle börja med att fritt reflektera över sina erfarenheter och subjektiva inställning till användning av Scrum. Därefter använde vi de mer specifika frågorna som vi sammanställt, genom att vi läst in oss på tidigare publicerad forskning och teori inom ämnet. Frågorna finns representerade i bilaga 2 och om omständigheterna motiverade det så ställde vi även en följdfråga för vårt intresse av att ta del av den påbörjade reflektionen som ännu inte blivit avslutad.

Våra intervjuer spelades in eftersom vi var benägna över möjligheten att kunna göra en noggrann analys och bearbetning av intervjun i efterhand. En risk med inspelad intervju är att intervjupersonen kan bli mer förtegen eller att hon eller han känner sig obekvämt i situationen (Lanz, 2007) Vi fann inte några sådana reaktioner när vi spelade in intervjun utan snarare tvärtom verkade det upplevas förtroendeingivande. Det kan även vara problematiskt att enbart föra anteckningar då detta innebär att informationen filtreras bort på ett osystematiskt sätt. Och i en öppen intervju blir det allt besvärligare eftersom det endast kommer att finnas summariska anteckningar av vad som sagts, vilket minskar möjligheten att bedöma principerna för de anteckningar som gjorts. Då slutsatser kan också endast dras från det som finns antecknat (Lanz 2007).

### 3.4 Etik

Ett etiskt dilemma uppstår när våra avsikter döljs som en del av vår metod i undersökningen. Oftast sker detta till följd av att forskaren vill undvika att den undersökta döljer eller uppträder på ett sätt som är annorlunda än det ”normala” som skulle gjorts i en vanlig situation, då de vet att de är observerade. Etiska dilemman måste handskas med balans genom insikten om undersökningens syfte och vi har inte funnit tillräckligt med motiverade skäl till att dölja våra avsikter som kan uppnås genom att dels dölja avsikten och dels mot de problem som uppgiftslämnaren kan uppleva om forskaren gör det (Jacobsen, 2012). Vi har i vår undersökning varit tydliga och beaktat det Informerade samtycket som är centralt för etiken, vilken tar hänsyn till de risker och vinster som deltagandet i undersökningen kan innebära. Informerat samtycke inkluderar Kompetens, Frivillighet, Full information och Förståelse (Jacobsen, 2002). Vi har varit noga med att följa Jacobsens etiska riktlinjer. Den intervjuade har varit i stånd att bestämma det eventuella deltagandet själv, den undersökta har fått fritt välja om deltagande utan påtryckningar om deltagande. Vidare har den undersökta



fått full information med hänsyn till den gyllene regeln om tillräcklig informationen och slutligen har vi försäkrat att den undersökte har förstått hur resultaten av undersökningen kommer att användas. Vi har även erbjudit full anonymitet som en del av vårt fokus på att ta hänsyn till den undersökta personens situation avseende risker i deltagandet. Vår undersökning har inte behandlat den privata individens sfär och våra resultat är inte av någon risk för den intervjuades fortsatta karriär.

### 3.5 Urval av företag och intervjupersoner

Undersökningarna genomfördes på fyra företags kontor och de är alla verksamma inom IT branschen. En utförlig beskrivning finns i Empirin.

Kriterierna för urvalet var att projektledarna på företagen skulle vara verksamma inom området för Systemutveckling, de skulle samtliga vara projektledare med en längre tids erfarenhet och expertis av projektledning, systemutveckling och Scrum. Företagen som intervjuades var medelstora till stora företag som tillämpade Scrum i sin dagliga verksamhet. Skillnaderna på storleken av företagen ser vi inte som ett problem för svaret av vår forskningsfråga eftersom vi är intresserade av Scrum som projektledningsmetod och en varietet i storleken på företaget medför snarare, om någon skillnad en berikad bild av tillämpningen av Scrum i sin helhet.

Tre av de fyra företagen var konsultföretag som genomförde regelbundna uppdrag åt mindre och större organisationer och företag och det fjärde företaget var ett stort företag som använde Scrum i sin dagliga utveckling. Valet av intervjuperson baserades på deras tidigare projektledare erfarenheter i kombination med Scrum erfarenhet och deras tillgänglighet för intervju.

Namn	Jannika Björkholm	Fredrik Grenson	Magnus Månsson	Daniel Terborn
Position	Teamchef	Projektledare	Projektledare	VD
Erfarenhet av projektledning/ Scrum	6 år	10 år varav två års erfarenhet av Scrum	10 år varav 7- 8 års Scrum erfarenhet	14 varav 7 års erfarenhet av Scrum
Företag	Eniro	Affecto	Cybercom	Stretch
Verksamhetsområde	Search	Business Intelligence	Business Intelligence	Business Intelligence
Grundat	2000	2003	1995	2002
Antal anställda	2816	150	1564	170

Tabell 2 - Intervjupersoner

## 3.6 Kvalitativ Analys av intervjuer

All information som kommer utifrån intervjuerna är inte föremål för analys utan endast information som är relevant för frågeställningen (Lanz, 2007). Vi har reducerat informationen för analysen, vilket innebär att vi systematiskt valt ut och valt bort information, inför den fortsatta analysen samt förenklat rådata. Viktigt i denna process är att det är nödvändigt att bestämma vilket innehåll som skall vara som underlag för analysen (Lanz 2007). Analysen av kvalitativ data genomfördes genom en iterativ process av de tre faserna; Beskrivning, Systematisering och kategoriseringsfasen och Kombinationsfasen (Jacobsen, 2002).

Beskrivningsfasen; innebär att intervjuer och samtal skrivs ner så noggrant som möjligt genom s.k. tjocka beskrivningar som är rika på detaljer, analyser och variationer (Jacobsen, 2002). Vi genomförde transkriberingen genom att vi skrev ner ord för ord vad intervjuerna återgav strax efter att de genomfördes. Vi lyssnade i denna process genom båda våra bandupptagningar för att säkerställa att ev. brus inte störde vår transkribering och skrev ner innehållet av dem, som finns representerade i bilagorna A1, A2, A3 och A4.

Systematisering och kategoriseringsfasen; innebär att systematisera och välja bort information för att göra den samlade informationen överskådlig före analysen (Jacobsen, 2002). I denna process gick vi genom det samlade materialet med färgpenna för att identifiera dimensioner/kategorier som strukturerades med hjälp av SWOTs områden; Styrkor (grön färgpenna), Svagheter (orange färgpenna), Möjligheter (blå färgpenna) och slutligen Hot (röd färgpenna). Lanz (2007) beskriver denna process som då helheten bryts ner i delar och dessa får ett sammanfattande namn i syftet att fördjupa förståelsen av detaljerna, som bildar helheten (Lanz, 2007) När detta var gjort så sammanställdes information i en lista som sedan låg som material för analysen av Empirin.

Kombinationsfasen; innebär att informationen börjar tolkas dvs. att förståelse, orsaker och samband söks i informationen (Jacobsen, 2002). I denna process sammanställde vi informationen för forskningen i vars en fristående SWOT analys för empirin och en SWOT analys för teorin som vi sedan ställde bredvid varandra för att på så sätt få en ökad förståelse för samband och olikheter mellan vår empiri och teori som valts ut för forskningsfrågan. Dessa finns presenterade i början av Analys kapitlet.

## 3.7 Kvalitetsaspekter på genomförd forskning

En viktig del i analysen är att kontinuerligt förhålla sig kritisk gällande kvaliteten på den genomförda forskningen (Repstad, 2007). Denna bedömning genomförde vi kontinuerligt under hela forskningsprocessen även i slutet. Reliabilitet används vanligen synonymt med pålitlighet dvs. hur pålitlig och precis informationen som tagits fram är och om analysen har genomförts utan brister och felaktigheter. Validitet är vanligen synonymt med giltighet dvs.

har forskningen lyckats mäta det som varit avsett att mäta eller något mer generellt. Datinsamlingen, bearbetningen, analysen och tolkning bör göras på ett självkritiskt sätt (Repstad, 2007). Vi har i vår forskning varit kontinuerligt kritiska gentemot de informationskällor och analyser vi använt oss för att nå våra svar på forskningsfrågan. Vi har varit medvetna vid intervjuerna om att den intervjuade kan ha anledning att dölja information. Under tre av de fyra intervjuerna har båda forskarna varit närvarande och vi har gjort två inspelningar på samtliga intervjuerna för att säkerställa att den kommande analysen inte haft några brister eller felaktigheter. Trots att endast en av forskarna hade möjlighet att närvara vid en av intervjuerna så bör det inte fått några konsekvenser för våra slutsatser eftersom samtliga samtalen bandades via två bandare som vi båda senare lyssnade på och genomförde en transkription som sedan kontrollerades också denna transskript via kontakt med den aktuella respondenten.

### **3.8 Kritik till valet av metod**

En förutsättning för en professionellt genomförd analys är en inbyggd motvilja mot att dra för snabba slutsatser. Det yttersta målet för kvalitativ forskning är att den skall bidra till kunskapsutvecklingen (Lanz, 2007).

Våra intervjuer genomfördes mot fyra stycken verksamma projektledare och med tanke på antalet genomförda intervjuer, är det riskabelt att dra för snabba generella slutsatser. Våra resultat är ändå användbara och kan vara värdefulla för andra som är intresserade av Scrum eller som ämnar ta vid en ny undersökning varifrån denna forskning avslutas.

SWOTs vida popularitet kommer från metodens flexibilitet och enkelhet. Den är enkel att förstå och användningen av SWOT kräver inte någon teknisk kunskap eller speciell träning. SWOT har möjligheten att belysa ett stort antal situationsbaserade faktorer till en hanterbar mängd men SWOT har en tendens till att överförenkla situationen, intressenter kan vidare ha olika uppfattningar om vad som är svaghet och styrka. Det kan även vara nödvändigt att prioritera faktorerna i de fyra kategorierna; annars finns det en risk för att möjligheter och hot balansera ut varandra, när det faktiskt är så att hoten är av större risk för organisationen än vad möjligheterna kan bidra med (Team five, 2013).

Vi har i vår forskning riktat oss till projektledare och av den anledningen skulle vår SWOT analys kunna vara annorlunda om vi genomfört våra intervjuer mot andra intressenter, enskilda projektmedlemmar eller kunder.

### **3.9 SWOT**

Begreppet SWOT är en förkortning för styrkor=S, svagheter=W, möjligheter=O, och hot=T. Swot-analysen är en verksamhetsanalysmetod som en organisation kan tillämpa för produkter, tjänster, och marknader inför beslut i vilken riktning som bäst resulterar i framtida tillväxt.

Processen involverar identifikation av organisationens styrkor och svagheter, samt möjligheter och hot som existerar i miljön den är verksam i (Team FME, 2013).

	Fördelaktiga	Ofördelaktiga
Interna	Styrkor	Svagheter
Externa	Möjligheter	Hot

Figur 2 - SWOT-Matrisen

**Styrkor** - Interna faktorer som är fördelaktiga för att uppnå organisationens mål.

En styrka är någonting som har en positiv konsekvens. Den har ett mervärde, eller erbjuder organisationen en konkurrensfördel. (Team FME, 2013)

**Svagheter** - Interna faktorer som är ofördelaktiga för att uppnå organisationens mål.

Svagheter är de konsekvenser som medför en nackdel i jämförelse med konkurrenter, eller avleder värde från det som organisationen erbjuder. (Team FME, 2013)

**Möjligheter** - Externa faktorer som är fördelaktiga för att uppnå organisationens mål.

Möjligheter kan uppstå till följd av en rad olika anledningar, kan vara resultatet av externa faktorer som till exempel kunden, teknologi, eller nya produktionsmetoder. (Team FME, 2013)

**Hot** - Externa faktorer som är ofördelaktiga för att uppnå organisationens mål.

Hot som är den sista delen i SWOT-processen, involverar att göra en uppskattning av externa risker som organisationen står inför. Dessa risker refereras till hot som består av externa faktorer som inte går att kontrolleras. (Team FME, 2013)

SWOT-analysens styrka beror på att den kan appliceras i många olika organisatoriska scenarion men svagheten är att det kräver klart tänkande och god bedömningsförmåga för att kunna dra någon verklig nytta av den. Processerna för att tydligt identifiera verksamhetens mål och att kategorisera SWOT faktorerna är båda lika viktiga eftersom de är beroende av varandra. Detta ömsesidiga beroende har som följd att SWOT-analysen ofta är en iterativ process där resultaten medför att målet omdefinieras och ny analys görs. Resultatet av en analys är inte nödvändigtvis slutgiltig. (Team FME, 2013)

SWOT är både enkel och flexibel att använda. Det är enkelt för alla att begripa SWOT och användandet kräver ingen teknisk kunskap eller specialiserad utbildning. Med SWOT så kan ett stort antal faktorer koncentreras till ett hanterbart antal men SWOT har även en tendens till att uppmuntra och grovt förenkla en situation. Ett annat problem med SWOT är att det inte finns några prioriterande vikter för vad som är relevant och vad som inte är relevant att tänka på för företagets situation. (Team FME, 2013)

### 3.9.1 SWOT-Analys applicerad på Scrum

I vår forskning är syftet med SWOT analysen att identifiera så många unika faktorer som möjligt utifrån tillgängligt material för att skapa en bättre helhet gällande kunskapen som finns gällande Scrum.

För att avgöra om något är en Styrka, Svaghet, Möjlighet eller Hot så måste det finnas en tydlig indikation gällande vilken kategori som en åsikt ska tillfalla dvs. om det är en Styrka, Svaghet, Möjlighet eller Hot. Vidare för att kunna applicera SWOT på Scrum är det viktigt att fastställa om faktorn är intern, eller extern. Rollerna ScrumMaster, Team och Produktägare är integrerade i Scrum. Dock så utövas produktägarrollen i de flesta fall av kunden och därför kan till exempel något negativt gällande produktägaren tolkas som både hot eller svaghet och en enskild bedömning måste göras.

I detta arbete analyseras intervjuerna och externa studier i litteraturkapitlet på det ovan nämnda viset. I fallet att en faktor hittas mer än en gång i olika intervjuer noteras detta men varje faktor nämns enbart en gång i den resulterande SWOT-Matrisen. Den samlade SWOT-Matrisens är av nytta för läsare oberoende av Scrum erfarenhetsnivå.

## 3.10 Frågor för intervju

En Intervju består av fyra delar;

§

Intervjuformuläret är uppdelat i fyra delar. Den första delen handlar om individen som intervjuas och dennes erfarenhet med Scrum. Andra delen handlar om företagets användande av Scrum. Andra delen informerar om, varför, när och hur intervjupersonens nuvarande organisation använder Scrum. I svaren som ges i denna del finns möjligheten att redan i detta tidiga stadium få svar av respondenten som kan användas och tolkas utifrån SWOT. Del ett,

och del två har vidare effekten av att få igång konversationen kring Scrum. Genom att intervjupersonen ”blir varm i kläderna” och bekvämt kan prata öppet vilket är viktigt då del tre har de mest öppna frågor i intervjun.

Tredje delen är den centrala och samtidigt den mest öppna delen i intervjun. Intervjupersonen ges här möjligheten att säga vad den själv tycker är viktigast med Scrum. Dessa frågor har ingen definierad tidsbegränsning för att optimera svaren av faktorer som kan vara av nytta för vår forskning. Mer strukturerade frågor riskerar att leda till kortare svar som även skulle ha större sannolikhet att likna svaren i andra intervjuer. Då intervjupersonen behöver ta hänsyn till strukturen.

Fjärde och sista delen innehåller kompletterande följdfrågor som bygger på vår litteraturstudie. Urvalet bygger på områdets relevans i Scrum. En fråga kan vara intern eller extern. Vilket syftar på balansen mellan styrkor-svagheter och möjligheter-hot. Ifall intervjupersonen har svårigheter att besvara frågan eller om vi uppfattar svaret som inte fullkomligt så finns i vissa frågor möjligheten till att utveckla frågan med ett exempel. Exempelvis: “-Regeln att inte ändra målet under en Sprint”. Frågorna har vidare konstruerats utifrån den använda teorin för mera ingående detaljer se bilaga B.

## 4. Empiri

I kapitlet nedan presenteras de fyra projektledarna som vi träffade och företagen som de är verksamma på för att ta reda på projektledarnas åsikter, tankar och erfarenhet kring Scrum. Kapitlet följer även med en presentation av intervjuerna som har kategoriserats enligt tidigare presentation för ökad tydlighet. Fulla intervjuer finns presenterade i bilagorna.

Fredrik Grenson intervjuades på Affectos kontor i Malmö. Fredrik har varit verksam som projektledare sedan cirka tio år tillbaka. På Affectos kontor i Malmö har Fredrik haft tjänsten som projektledare sedan 4 år tillbaka och han har där praktiserat Scrum sedan ett par år tillbaka (se bilaga A1, s1). Affecto erbjuder sina kunder implementation av tekniska lösningar, utbildning, support, förvaltning, och affärsrådgivning. Affecto är verksam inom beslutsstöd, eller Business Intelligence (BI). Företagets Vision är att 2013 vara det ledande Enterprise information management företaget med hänsyn till intäkter, marginal, kund- och medarbetarnöjdhet. Affecto arbetar för att varje person ska ha informationen som gör hans/hennes arbete effektivt och inspirerande. Företaget har mer än 1100 anställda i Finland, Sverige, Norge, Danmark, Estland, Litauen, Lettland och Polen ([www.Affecto.se](http://www.Affecto.se) , 2014).

Magnus Månsson intervjuades på Cybercom i Malmö. Magnus har varit verksam som projektledare sedan cirka åtta år tillbaka med Scrum. På Cybercom har han arbetat som projektledare, produktägare, utredare och kravinsamlare (Se bilaga A2, s 1). Cybercoms expertisområden täcker hela ekosystemet inom kommunikationstjänster, det vill säga Connectivity. Cybercoms vision är "att vara kundens mest betrodda affärspartner i den uppkopplade världen". Cybercoms affärsidé är att "stärka företags och organisationers verksamheter och affärer genom att ta tillvara på den uppkopplade världens möjligheter". Cybercoms erbjudande är indelat i fyra områden: Digital Solutions, Connected Engineering, Connectivity Management och Secure Connectivity. Företaget har ca 1260 anställda i Sverige, Finland, Singapore, Dubai, Polen, Danmark, Indien ([www.Cybercom.se](http://www.Cybercom.se), 2014).

Jannika Björkholm intervjuades på Eniros huvudkontor i Stockholm där hon varit verksam som teamchef och projektledare inom Scrum sedan 2008 (se bilaga A3). Eniro är ett ledande svenskt företag som är verksam inom såväl Skandinavien och Polen. Eniros vision är att vara en symbol för "local search" och att vara den ledande leverantören i Skandinavien, Finland och Polen i "mobile local sök". Eniros tjänst fungerar som ett gränssnitt mellan kunder och företaget, i den digitala värdekedjan. Eniro filtrerar, organiserar, samlar, organiserar och

presenterar lokal information relaterat till privatpersoner och företag för att dessa ska vara tillgängliga 24/7. Företaget har idag 2 816 anställda ([www.eniro.se](http://www.eniro.se)).

Daniel Terborn intervjuades på Stretchs huvudkontor i Malmö. Daniel är VD för Stretch men har även varit verksam som utvecklingsarkitekt på Capgemini, där han tillhörde en av Capgeminis viktigaste resurser för globala utvecklingsprojekt. Daniel har varit verksam som utvecklingsarkitekt och projektledare sedan 2007 (se bilaga A1, s 1 och s 6). Stretch är det stora lilla konsultbolaget som grundades 2002 och är verksam i Stockholm, Malmö, Göteborg, Karlstad, Köpenhamn och Oslo. Stretch erbjuder sina kunder expertis inom processtöd, beslutsstöd, arkitektur, integration, säkerhet och användarvänlighet genom enhetliga och effektiva system. Stretchs vision är att alla våra kunder ska driva fantastiska företag där resurserna går till affären. Deras affärsidé är att förenkla företags processer så att maximalt av resurserna går till affären. Med den senaste tekniken utvecklar de kunders verksamhet så att invanda mönster utmanas med marknadens ledande system och morgondagens processtöd. Stretch har idag ca 170 anställda ([www.stretch.se](http://www.stretch.se)).

Nedan presenteras intervjuerna i en samlad helhet med hänsyn till Styrkor, Svagheter och Möjligheter. För att för att öka tydligheten så är innehållet i varje kategori även indelad in lämpliga underrubriker beroende på innehåll.

## **4.1 Styrkor**

### **4.1.1 Enkelhet och snabbriklighet**

En av intervjupersonerna upplevde att Scrums projektstruktur och projektdokumentation var enkel, vilket innebär att själva skapandeprocessen går fort (Se bilaga A1, s 3). Vidare ansåg två av intervjupersonerna att tillämpningen av Scrum medför att något levereras inom en kort tid, det går extremt mycket snabbare att leverera inom betydligt kortare tid (se bilaga A1 och A3, s 4 och s 1). Vidare så framkom det att detta ger ett tillfälle till att fira segarna oftare och ökar motivationen genom de kontinuerliga leveranserna (se bilaga A3, s1). Scrum gör också att arbetet blir extremt mycket mer snabbrikligt (se bilaga A3, s1).

### **4.1.2 Kravhantering utforskande och lärande**

Vidare så lyfte två av de intervjupersonerna att de naturliga avbrotten i Scrum ger möjligheten att hantera förändringar och kravhantering bättre. Avbrotten är bra för förtydliganden och att förbättra kraven i efterhand. I vattenfall så skrivs kraven från början vilket är det största problemet med denna metodik (se bilaga A2s 2). En av intervjupersonerna underströk även att Scrum ger ökade möjligheter till utforskandet på vägen tills projektet är färdigt (se bilaga A1, s 3). Scrum ger vidare möjligheten att upptäcka fel tidigare genom att diskussioner



kontinuerligt hålls med kund och att kundbehov kan fångas med upp till 100 % (se bilaga A4, s 4). Ytterligare betonas styrkan med det ökande lärandet, man lär sig något längs vägen d.v.s. det är både möjligt att ta bort och lägga till under resans gång utan att det behöver bli kostnader associerade med förändringarna (se bilaga A4, s 2). User stories är även bra eftersom de fångar både rollen och nyttan på ett bra sätt (se bilaga A1, s 7).

### **4.1.3 Bättre planering och förståelse**

Scrum ger fördelar för produktionsstyrningen genom att Scrum är bra för planering, stötta och styra vad som ska utvecklas under varje iteration (se bilaga A4, s 2). Genom att kunder arbetar nära med projektet och är aktiva i diskussioner så innebär det även att transparensen ökar (se bilaga A4, s 7). Vidare lyfte en av intervjupersonerna att genom Scrum så får team verkligen en förståelse för att de arbetar rätt vilket gör att teamen kan arbeta effektivare och att rätt saker utvecklas och levereras (se bilaga A2, s 2-3).

### **4.1.4 Initiativ och gemensamt ansvar**

Ytterligare styrka är associerad med team målen; det blir teamets gemensamma ansvar att nå team målen, eftersom teamen levererar något tillsammans genom ett gemensamt mål även när arbetet sker på distans (se bilaga A1, s 2-3). Vidare lyfte två av intervjupersonerna styrkan med att Scrum motiverar team medlemmarna att ta eget initiativ genom att de kan välja fritt sin arbetsuppgift, eftersom principen är att varje teammedlem kan ta den uppgiften den vill från den prioriterade backlogen (se bilaga A2, s 6) (se bilaga A3, s 1-2).

### **4.1.5 Det bättre samarbetet**

Scrum medför att projekten kommer bort från Micro management och istället motiverar till ett nära samarbete. Att jobba nära varandra förebygger missförstånd som sker när man är utspridd. Går något fel så är det hela teamets fel vilket förebygger konflikter (se bilaga A2, s 3). Ytterligare lyfte tre av de fyra intervjupersonerna att det nära samarbetet där utvecklare helst sitter bredvid varandra ger effektivitetsvinster men arbete på distans med Scrum går också bra eftersom samarbete och kommunikation främjas (se bilaga A3, s 2-3) (se bilaga A2, s 7)(se bilaga A4, s 9). Scrum är en fantastisk metodik när teamet verkligen känner varandra, vad de är duktiga och är dåliga på och på så sätt kan arbeta i informella roller (se bilaga A1, s 4). Ytterligare styrka är den medvetenhet som Scrum medför om varandras utförda arbete vilket resulterar i mindre risker om en medlem skulle gå förlorad under projektets gång (se bilaga A4, s 8).

#### **4.1.6 Mindre dokumentation och ökad kunskapsspridning**

Vidare betonar två av intervjupersonerna att den visuella dokumentationen som finns i Scrum är bättre, det behövs inte längre skrivas flera hundra sidor dokumentation som aldrig läses utan det är betydligt mindre dokumentation, istället används mera visuell dokumentation som fungerar som en central kunskapsspridare (A4, s9). Kunskapsspridning ger även kraftfullare team framöver genom att flera kan göra varandras arbetsuppgifter (se bilaga A2, s 5) (se bilaga A3, s 7). Kunskapsspridningen ger ytterligare styrkor genom en ökad teamkänsla då bättre samarbete, öppenhet, tydlighet blir till samtidigt som team medlemmarna motiveras av varandra och har en större tendens att samarbeta gällande idéer och utveckling (se bilaga A1, s 9) (se bilaga A1, s 7). Vidare ger dokumentationen i Scrum styrkor eftersom det är mindre tryck på att ”dokumentera ihjäl sig”, utan istället att dokumentera tillräckligt och att dokumentationen ändå fungerar som en kunskapsspridare (se bilaga A1, s 7).

#### **4.1.7 Effektiv kommunikation**

Ytterligare stryka med Scrum är det nära samarbetet som sker genom projekten, vilket förebygger missförstånd (se bilaga A2, s 4). Två av intervjupersonerna berättade även att Scrum skapar en effektivare kommunikation, inom Scrum blir mycket av kommunikationen ansikte mot ansikte. Denna kommunikation stödjer det nära samarbetet, vilket ger effektivitetsvinster och bättre samarbeten (se bilaga A3, s9) (se bilaga A2, s 6). Vidare uttryckte en av intervjupersonerna att sprintarna är jättebra eftersom de skapar väldigt mycket struktur för utvecklingen av projekten även ifall de kan behövas ändras ibland, när saker och ting händer (se bilaga A4, s5). Scrum medför även styrkor genom “Gain all the man brainpower” genom samarbete och tvärfunktionella team, vilket gör att olika avdelningar har en större tendens till att samarbeta i produktutvecklingen gällande idéer och utveckling (se bilaga A3, s 6). Den ökade kunskapsspridningen medför även att kraftfullare team skapas framöver genom att flera kan göra varandras arbetsuppgifter (se bilaga A2, s 6).

#### **4.1.8 Mötesstrukturen och rollerna**

Tre intervjupersoner tyckte att mötesstrukturen i Scrum var bra eftersom de är effektiva och väl informerande speciellt med hänsyn till frågorna; vad jag gjorde igår, vad jag gjorde idag, och impediment (se bilaga A1, s 7) (se A3, s7). Dessa best practices är bra för kommunikationen och planering av nuvarande och kommande arbete och ger möjligheten att verkligen lyfta förbättringspunkter för att förbättra det fortsatta arbetet (se bilaga A2, s 5) (se bilaga A1, s 7). Vidare tyckte två av intervjupersonerna att rollerna var en styrka eftersom de ger en tydlighet och var jättebra att arbeta med även ifall Scrum Mastern ofta får för mycket uppmärksamhet i hänsyn till viktigt ansvar i jämförelse med Produktägaren (se bilaga A4, s 5) (se bilaga A2,s 3).

## 4.2 Svagheter

### 4.2.1 Situationsberoende

Scrum är inte i bra för alla situationer utan det handlar mycket om förhandlingen med kunden, hur man kommer att få in uppdraget och hur man vidareutvecklar affären och någonstans under början görs bedömningen om hur man kan angripa projektet på ett bra sätt (se bilaga A1, s 1). Rör det sig om traditionell upphandling där kunden verkligen vill veta exakt vad den kommer att få i slutet, till vilket pris då är det tämligen värdelöst att arbeta med Scrum (A1, s1).

### 4.2.2 Att arbeta iterativt med Scrum.

Tre intervjupersoner lyfte att Scrum inte är en helhetslösning, fast det ofta ses som det. (Se bilaga A1,s 5)(se bilaga A2, s 2)(se bilaga A4,s 1). En intervjuperson berättar att när man bygger stora system så måste man ta hänsyn till arkitekturen. Arkitektur är en långsiktig aspekt, och en hel del viktiga beslut måste fattas i början av projektet. Man måste skapa sig en långsiktig bild över många sprintar, eventuellt många år framöver, och arkitekturen kan inte ändras efter ett halvår, då kostnaden för ändring kan vara enorm. Enligt samma intervjuperson kan man inte köra totalt agilt i branschen, det krävs en hybrid av agilt och vattenfall (se bilaga A1,s 2). Men Scrum kan även vara olämpligt vid väldigt kortsiktig arbete. En intervjuperson berättade att organisationens mindre team behövde kontinuerligt leverera utveckling i högre hastighet än vad som gjordes i tre-veckors sprintar, och att planera detta enligt Scrum var inte möjligt (se bilaga A3,s 2). En annan intervjuperson lyfte att det inte alltid är möjligt att leverera något efter varje Sprint som kan presenteras för intressenter. Inom Business Intelligence behövs front-end för att kunna presentera funktionalitet och utvecklingen av back-end tar väldigt, väldigt lång tid (Bilaga A1,s 5).

### 4.2.3 Teamet och gruppdynamik

Enligt en intervjuperson finns det grupper där produktiviteten sjunker och frustration uppstår när ingen har ledarskapet och ansvarar för välmående, effektivitet och lösning av konflikter (se bilaga A1,s 4) Vidare så tar Scrum inte hänsyn till individer som trivs bäst med att enbart fokusera på arbetet. En intervjuperson säger att det inte är rimligt att ställa kravet på alla medlemmar att vara jättesociala, ta mycket ansvar och att tänka på saker de inte är intresserade av, eller duktiga på (se bilaga A1,s 4). Även om Scrum uppmuntrar till att medlemmar väljer arbete själva efter prioritet, så löser Scrum inte de olikheter som finns i en arbetsgrupp gällande kompetens (se bilaga A3,s 8). En annan aspekt gällande teamet då Scrum inte tar hänsyn är gällande gruppsammansättning påpekar en annan intervjuperson;

Man måste vara medveten om vilka styrkor och svagheter man har på både personlig och kompetensnivå och sedan anpassa arbetsuppgifter efter det (se bilaga A1, sid). Ytterligare svagheter finns i att Scrum inte helt kan hantera de kompetenser som finns i en grupp med avseende på om någon medlem skulle gå förlorad. Olika individer har olika kompetenser och tvärfunktionella grupper kan inte kompensera för förlust av specialkompetens (se bilaga A4,s 8).

#### **4.2.4 ScrumMaster och samordning**

En intervjuperson ansåg att en svaghet finns i samordning när flera team jobbar tillsammans mot ett och samma projekt. Att samordna och få till möten för att få överblick i vad de olika teamen gör. Visserligen är inte problemet unikt för Scrum men inte heller löst med Scrum (se bilaga A3,s 2). Vidare så kan prioritering av arbete i mindre team vara ett problem (se bilaga A3,s 4) En intervjuperson lyfte att vid konflikthantering kan auktoritet, som i Scrum är informell, vara ett problem ifall en junior utvecklare är ScrumMaster och någon senior utvecklare inte har respekt. Det kan vara tufft när det verkligen är en konflikt som teamet själv inte löser (se bilaga A2,s 4).

#### **4.2.5 Product Owner och kunden**

I Scrum så läggs för stor tonvikt på ScrumMastern och den viktiga rollen som Produktägaren har kommer i skymundan som ändå är den viktigaste rollen enligt en intervjuperson (se bilaga, A4,s 5). Vidare så lyfte en intervjuperson att Produktägaren har en viktig och central roll med stort ansvar i Scrum och behöver ofta avlastas för få hjälp med de prioriteringsproblem som uppstår med alla intressenter som Produktägaren ansvarar för (se bilaga A1,s 4). För ett optimalt resultat så måste kunden ta på sig rollen som Produktägare för att uttolka kraven (se bilaga A1, sid6). Scrum är en jättebra metodik om kunden är van vid att arbeta med Scrum men det är ingen metodik som går att använda om kunden inte har vanan (se bilaga A4,s 2). För att hela konceptet, kommunikation med verksamhet ska fungera, så krävs en kontinuerlig dialog med kunden (se bilaga A4,s 5)

#### **4.2.6 Inbyggda mekanismer**

En intervjuperson lyfte att dokumentation egentligen bör vara en leverabel som allt annat i Scrum (se bilaga, A4,s 8). Vidare så saknar Scrum en bra metodik för den ekonomiska styrningen samt är Scrum inte är anpassad för delen i projektet som relaterar till den tekniska arkitekturen (se bilaga A4.s 1-2) En annan intervjuperson påpekade att planering av resurser kan vara ett problem i Scrum. Ibland har man halvtidsresurser och då krävs det ett annat sätt av styrning, då krävs kontroll, och då krävs ledning (se bilaga A1,s 8). Ytterligare en intervjuperson tog upp att vid allokering av resurser kan kunden ibland vilja ha mer kontroll på att man rapporterar allokering, beläggning, och debitering. Det går dock utanför Scrum och

får hanteras av en projektledare (se bilaga A2,s 6). User stories är väldigt bra men de måste kompletteras i en processbeskrivning. Userstories fångar inte upp vad en traditionell process bild fångar upp. I en sådan kan man exakt påpeka vem som gör vad, i vilket steg och var nyttan genereras (se bilaga A1,s 7). Vidare så lyfte en intervjuperson att det kan hända att något som utvecklas inte fungerar i drift och ibland behövs mer fokus på operations och driften (se bilaga A2,s 6). Slutligen så finns det svagheter gällande tidsuppföljning, vilket inte är inbyggd i Scrum-modellen (se bilaga A4, s3)

## **4.3 Möjligheter**

### **4.3.1 Kravhantering och förankring**

Scrum ger möjligheter när en budget finns men ingen riktigt vet hur den ska användas utan man vill utforska efterhand (se bilaga A1, s 1). Vidare berättar samma intervjuperson att Scrum kommer till användning när det är osäkert vad som ska göras och under vilka former, då kunden inte vet inte hur den ska arbeta i projektformen d.v.s. kunden har inte full förståelse för hur en förhandling bör göras och hur han/hon ska organisera sig och planera för kravhandling (se bilaga A1,s 2). Om Scrum tillämpas bra så kommer kunden att vara till stor hjälp för optimering och sammanställningen av kravspecifikationen (se bilaga A4, s 2). Genom kundens kontinuerliga involvering i projektet så kommer även en automatisk förankring av projektet (se bilaga A4, s 4).

### **4.3.2 Tillgänglighet, samarbete och engagemang**

Vidare berättar två av intervjupersonerna att Scrum kan bli en enabler via det förbättrade samarbetet och det tidiga engagemanget i projektet, med en utarbetad metodik för att skapa dialog och få intressenterna engagerade under projektets gång (se bilaga A2,s 6) (se bilaga A4, s 4). Scrum kräver av kunden att det måste finnas en produktägare och att kunden måste vara tillgänglig och närvarande vid demonstrationer och retrospectives. Kunden blir väldigt involverad, 1-3 dagar i veckan måste kunden vara tillgänglig (se bilaga A1,s 6). Scrum engagerar till diskussioner och till delaktighet i ett syfte för kunder att få sina fulla kundbehov levererade (se bilaga A4, s 4). När kunder väl lärt sig att arbeta med Scrum så älskar de det och vill inte jobba på något annat sätt (se bilaga A4 s 7). Kunder blir mer engagerade och nöjdare när de arbetar med Scrum (Se bilaga A2, s 6).

## 4.4 Hot

### 4.4.1 Externa krav, missförstånd och förväntningar

Externa krav och intressen kan utgöra ett hot, kunder har genom Scrum makt och inflytande att påverka, vilket kan ibland vara problematiskt om det utnyttjas fel under utvecklingen, vilket Scrum inte kan hantera (se bilaga A4, s 2). Kontakt med kund kan vidare vara problematiskt även ifall det aldrig är fel. Det är inte självklart att lösningar hittas enbart för att man tillämpar Scrum (se bilaga A2, s 3). Två av intervjupersonerna berättar att Hot ofta är förknippade med förväntningarna, missförstånd kan uppkomma och det handlar hela tiden om att hantera förväntningarna för att undvika besvikelser (se bilaga A2, s 3). Vidare så måste förväntningarna på projektet klargöras så att det stämmer med projektet hos projektets intressenter och medlemmar (se bilaga A4, s 9) Vissa kunder tror att allt löser sig med hjälp av Scrum (se bilaga A4, s 4).

### 4.4.2 Erfarenheter och roller

Det finns kunder som har tidigare dåliga erfarenheter av att arbeta med Scrum, då de sett Scrum som en helhetslösning (se bilaga A4, s2). Vissa kunder har även olika uppfattningar om hur Scrum ska tillämpas vilket i sig kan utgöra ett hot för projektet (se bilaga A4, s 2). Vidare måste en lämplig produktägare hittas på kundens sida och det kan vara problematiskt att hitta någon som är beredd att ta på sig rollen och ansvaret att samla ihop kraven (se bilaga A2, s 3-4). Det finns även å andra sidan ett hot med att produktägaren engagerar sig för mycket och kommer med ändringar på löpande band eller att kunden har ett extremt kontrollbehov och kan därmed störa projektet eller påverka det tekniska väldigt mycket (se bilaga, A2, s 3-4) (se bilaga A1, s 6). Tre av intervjupersonerna instämmer att produktägaren har ett mycket stort ansvar, vilket innebär att han/hon kan vara svår att nå, ibland måste till och med Scrum Mastern gå in och agera Produktägare, vilket inte är optimalt och den önskade interaktionen mellan produktägare och team uteblir (Se bilaga A1, s 6) (se bilaga A2, s 3-4) (s bilaga A4, s 5). Citat från Daniel Terborn ”*Product Ownern var väldigt, alltså väldigt viktig men jag tror det är väldigt så ofta man bara pratar om ScrumMaster men man har inte så mycket koll på vem som är Product Owner om man säger så. Nånstans så är ju Product Owner det viktigaste*” (se bilaga A4, s 5).

### 4.4.3 Kultur och överblickbarhet

Bristande kommunikation och överblickbarhet mellan team kan vara ett hot eftersom ett av teamen inte vet vad det andra teamet gör, och ändringar som teamet gör påverkar det andra (se bilaga A3, s 3-4) . Olika kulturer och förhållningssätt till arbete kan även utgöra ett hot för

projektet, det är därför viktigt med förståelse för olika kulturer så åtgärder kan vidtas för att skapa ett bättre team (se bilaga A4,s 9).

## 4.5 Empiri tabell sammanfattning

Tabellen nedan ger en överblick över samtliga punkter som ingår i SWOT-analysen. Numrering visar kopplingen mellan tabellen nedan och SWOT-matrisen i det senare analyskapitlet. Numren är inte av någon vikt dvs. inte någon rangordning av något slag. I varje kolumn återfinns bokstäver i parantes för varje styrka, svaghet, möjlighet och hot vilket visar vilken, eller vilka intervjuer som kolumnen är kopplad till.

(A= Affecto, C= Cybercom, E= Eniro, S= Stretch)

Nr	Styrkor	Svagheter	Möjligheter	Hot
1	Enkel projektstruktur och projektdokumentation (A)	Ingen helhetslösning. Scrum (A, C, S)	Lämplig när det inte finns kravbild men en budget finns. (A)	Externa krav och intressen. (S)
2	Kort tidshorisont, och leverera inom denna, (A, E)	Arkitektur kräver långsiktig planering. Kan ej ändras en gång per halvår (A)	Kunden vet ej hur man ska definiera kraven. (A)	Kontakt med kund kan vara utmanande. Ingen garanti att en lösning hittas enbart för att Scrum används. (C)
3	Motiverar genom att fira segrar regelbundet (E)	ibland är sprintplanering inte effektiv när något behöver vara ännu snabbare, ett jämt flöde av gjort arbete hela tiden. (E)	Kunden kan optimera kravsammanställningen. (S)	Missförstånd kan uppstå och det handlar om att hantera förväntningarna och undvika besvikelser. (C)
4	Det är extremt mycket mera snabbriktigt (E)	Det går inte alltid att demonstrera utfört arbete i korta sprintar. (A)	Förankring hos kunden (S)	Kund har tidigare haft dåliga upplevelser med Scrum. (C, S)
5	Naturliga avbrott gör det möjligt att hantera förändringar och kravhantering (C, S)	Tydlig ledarroll saknas. Ingen ansvarar för välmående, effektivitet och att lösa konflikter. (A)	Förbättrade samarbeten, naturliga möten och engagerade kunder. (C, S)	Kunder arbetar med Scrum på olika sätt. (S)
6	Utforskande (A)	Tar ej hänsyn till individuell motivation till socialt engagemang. (A)	Engagerar till kundens delaktighet (A, S)	En lämplig Product Owner kan vara svår att hitta på

				mottagarens, kundens sida. (C)
7	Upptäcker fel tidigare (S)	Tar ej hänsyn till individuella kompetenser (E)	Främjar att kundens behov levereras genom kontinuerliga diskussioner med kund. (S)	Produktägare är för engagerad och därmed stör teamet i arbetet. (A, C)
8	Man lär sig längs vägen (S)	Tar ej hänsyn till grupp sammansättning. (A)	Kunderna blir mer nöjda när de arbetar med Scrum. (C)	Produktägare har ett stort ansvar och kan bli otillgänglig. (A, E, S)
9	Bra på produktionsstyrning (S)	Tvärfunktionella team ersätter inte förlorad special kompetens. (S)	När väl kunder lärt sig Scrum vill de inte jobba på något annat sätt (S)	Att ej veta vad det andra teamet gör. (E)
10	Hjälper att arbeta effektivare, och att arbeta rätt. (C)	Beroende av kontinuerlig kommunikation och dialog. (S)		Olika kulturer. (S)
11	Fokus på att leverera rätt saker (C)	Samordning mellan flera team i ett och samma projekt kan vara krävande. (E)		Kommunikationen. (E)
12	Stöttar planering och vad som ska utvecklas (S)	Prioritering av arbete i mindre team av utvecklare. (E)		
13	User stories fångar både rollen och nyttan (A)	Eventuellt problematisk för Scrum masters att hantera auktoriteten i ett seniort utvecklar team. (C)		
14	team mål, teamets gemensamma ansvar för att nå dit. (E)	Större fokus på ScrumMaster, och mindre fokus på Product Owner. (S)		
15	Principen att team medlemmarna kan om det är motiverat välja fritt sin arbetsuppgift ur sprint loggen är bra (C, E)	Product Owner har mycket på sitt bord. (E)		
16	Kommer ifrån Micro management samtidigt som det främjar ett nära samarbete. (C)	För optimalt resultat bör kunden agera Product Owner. (A)		
17	Öppet och nära samarbete förebygger konflikter (C)	Kunden behöver vara redo för att arbeta Scrumorienterat. (S)		
18	Tighta samarbetet där utvecklare helst sitter	Kräver mycket av kunden. (S)		



	bredvid varandra ger effektivitetsvinster men arbete på distans går också bra. (C, E, S)			
19	Fantastisk när teamet verkligen känner varandra. (A)	För lite fokus på dokumentation. (S)		
20	Medvetenhet om varandras utförda arbete (S)	Scrum hjälper inte någon med ekonomisk styrning, riskhantering och att bygga tekniska lösningar. (S)		
21	“Gain all the man brainpower” genom samarbete och tvärfunktionella team (E)	Ibland finns halvtidsresurser som ska planeras mot backloggen vilket kan vara lite omständligt. (A)		
22	kunskapsspridning ger kraftfullare team framöver (C)	Scrum ger inte tillräckligt underlag för allokering, beläggning och debitering, hur mycket som debiteras kund. (C)		
23	Ökad teamkänsla och underlättar kunskapsspridning (E)	Userstories behöver kompletteras med en processbeskrivning. (A)		
24	Visuella dokumentationen, som är den centrala, fungerar som kunskapsspridare (E, S)	Ibland behövs mer fokus på operations och driften. (C)		
25	Dokumentera så lite som möjligt men ändå tillräckligt. (E)	Ingen tidsuppföljning. (S)		
26	Ökad transparens (S)	Situationsberoende projektform. (A)		
27	Tydlighet och undviker missförstånd. (C)	Kunden vet exakt vad den vill, samt vill veta exakt vad det kommer kosta. (A)		
28	Effektivare kommunikation (C,E)			
30	Sprintar är jättebra, det skapar väldigt mycket struktur. (S)			
31	Bra med Retrospectives, problem kommer upp till ytan och teamet			

	förbättras (C,E)			
31	Bra mötesstruktur (A,C,E,S)			
32	Rollerna är bra, det ger en tydlighet. (C, S)			

*Tabell 3 – Sammanfattning av SWOT i empirin.*

## 5. Analys

Nedan inleds kapitlet med två stycken SWOT-matriser, en för uppsatsens teori och en för uppsatsens empiri. Analysen som följer behandlar teorins och empirins styrkor, svagheter, möjligheter och hot som finns presenterade i Swot-matriserna nedan. Numrering visar kopplingen mellan SWOT-matrisen och den tidigare presenterade tabellen från kapitlet i Empirin samt Sammanfattning av teorin. Numren är inte av någon vikt dvs. inte någon rangordning av något slag. Slutligen sammanfattas resultatet av analysen genom en enhetlig SWOT, en sammansatt SWOT-Matris bestående av de gemensamma likheterna eller olikheterna från empirin och teorin. För att för att öka tydligheten så är innehållet i varje kategori även indelad in lämpliga underrubriker beroende på innehåll.

<p><b>Styrkor</b></p> <p>1. Kundtillfredställelse - 2. Välkomnar förändrade krav - 3. Nära dagligt samarbete mellan affärsidé och utvecklare. - 4. Uppmuntrar till "ansikte mot ansikte" kommunikation. - 5. Effektiv mötesstruktur - 6. Minskar negativa inflytanden som orsakas av distansarbete. - 7. Snabb och effektiv hantering av arbete. - 8. Samarbete uppmuntrar och återanvändning och integration ökar. - 9. Mekanismer för att bygga förtroende. - 10. Motiverande - 11. Ger ett gemensamt språk, och mål att sträva efter. - 12. reducerad cykeltid. - 13. Reducerad variabilitetsflöde - 14. Snabbare och mer frekvent återkoppling. - 15. Reducerad risk - 16. Reducerad överskott av resurser. - 17. Reducerad tids- och kostnadsökning. - 18. Loop-lärande 19. begränsad mängd pågående arbete. - 20. Reducerad risk för finansiell förlust. - 21. Prioritering av arbete, det viktigaste görs först. - 22. Undviker onödig perfektionism.</p>	<p><b>Svagheter</b></p> <p>1. Stora krav på både kommunikation, och formulering av kravspecifikation. 2. Delad beslutsfattande måste fungera. - 3. Avsaknad av ett forum för att lösa konflikter. - 4. En bristande definition av "done" kan leda osäkerhet och förvirring. - 5. Beroende av ett effektivt informationsorienterat ledarskap. 6. Frekventa möten kan kräva mycket tid. - 7. Kravspecifiering måste vara genomarbetad. - 8. Product Backlog; planering av resurser, brister denna kan det ha förödande konsekvenser. - 9. Begränsar möjlighet till privat sfär 10. Bristande kommunikation kan orsaka dubbelt arbete. - 11. Potentiell konflikt på grund av kortsiktiga mål och långsiktiga mål. - 12. I distribuerade team, trots tekniska hjälpmedel, uppstår ofta en känsla av att det är två team och inte ett. - 13. Långsam metod ifall utvecklare ej har bra förståelse för kundens behov.</p>
<p><b>Möjligheter</b></p> <p>1. Hög potentiell kundtillfredställelse genom att ge kunden vad den verkligen vill ha. 2. Upptäcker dysfunktioner, och slöseri väldigt tidigt. 3. kan kombineras med kompatibla, och sunda ingenjörsmetoder för förbättrad kvalitet.</p>	<p><b>Hot</b></p> <p>1. Organisatoriska missanpassningar - 2. Konflikt mellan strategisk plan och handlingsplan. - 3. Förlust av personal 4. Scrum reglerna; gällande ändringar i sprintarna efterföljs inte och teamen tvingas avbryta sitt arbete. - 5. Kommunikation; är inte effektiv och lyckas inte "bära och utveckla" projektet. - 6. Kund är osäkra på krav, tvingar teamet att göra ändringar under sprint. - 7. Ledning visar bristande förståelse och stör teamets arbete. - 8. Kultur, språk, religiösa skillnader - 9. Andra okontrollerbara externa händelser. 10. Beslutsuteslutande (Decision hijacking). - 11. Brist rätt kunskap i teamet. - 12. Technocracy. - 13. Samarbete med partner när den ses som konkurrent. - 14. Konflikter inom företaget gällande prioritering. 15. Störning från kund, management, extern påverkan, processer, kommunikation.</p>

Figur 2 – SWOT-Matris av teori

<p><b>Styrkor</b></p> <ol style="list-style-type: none"> <li>1. Projektstruktur - 2. Tidshorison - 3. Motivation - 4. Snabbrörigt</li> <li>5. Förändring och kravhantering 6. Utforskande - 7. Upptäcker fel</li> <li>8. Loop-lärande - 9. Produktionsstyrning - 10. Arbeta rätt &amp; effektiv</li> <li>11. Leverera rätt - 12. Planering - 13. User stories - 14. Team mål</li> <li>15. Eget arbetsinitiativ - 16. Samarbeta - 17. Konfliktförebyggande</li> <li>18. Nära- och distanssamarbeta - 19. Fantastiskt - 20. Medvetenhet</li> <li>21. Tvärfunktionalitet - 22. Kunskapspridning - 23. Teamkänsla</li> <li>24. Visuell dokumentation - 25. Tillräcklig Dokumentation - 26. Transparens</li> <li>27. Tydlighet - 28. Effektivare kommunikation - 29. Sprintar</li> <li>30. Retrospectives - 31. Mötesstruktur - 32. Roller</li> </ol>	<p><b>Svagheter</b></p> <ol style="list-style-type: none"> <li>1. Ingen helhetslösning - 2. Arkitektur, långsiktighetsbeslut</li> <li>3. Kvicka problemåtgärder - 4. Vissa demonstrationer</li> <li>5. Ingen tydlig ledarroll - 6. Individuell motivation - 7. Individuella kompetenser</li> <li>8. Gruppsammansättning - 9. Förlorad kompetens</li> <li>10. Kommunikationsberoende - 11. Samordning mellan flera team</li> <li>12. Prioritering av arbete - 13. Juniöra Scrum Master</li> <li>14. Mindre fokus på produktägaren - 15. Produktägarens stora ansvar</li> <li>16. Kunden bör vara produktägare - 17. Kunden måste vara anpassad</li> <li>18. Kräver mycket av kunden - 19. Lite Dokumentation</li> <li>20. Ekonomistyrning, risker och tekniska lösningar</li> <li>21. Halvtidsresurser 22. Allokering, beläggning och debitering</li> <li>23. Processbeskrivning - 24. Operations och drift - 25. Tidsuppföljning</li> <li>26. Situationsberoende - 27. Traditionell upphandling</li> </ol>
<p><b>Möjligheter</b></p> <ol style="list-style-type: none"> <li>1. Vid oklar kravbild - 2. Kunden kan ej definiera krav</li> <li>3. Optimering av krav - 4. Förankring hos kund</li> <li>5. Engagerade kunder - 6. Delaktiga kunder - 7. Kundens behov</li> <li>8. Nöjdare kunder - 9. Kund älskar Scrum</li> </ol>	<p><b>Hot</b></p> <ol style="list-style-type: none"> <li>1. Externa krav och intressen - 2. Kontakt med kund kan vara utmanande</li> <li>3. Missförstånd och förväntningar - 4. Tidiga dåliga Scrum upplevelser</li> <li>5. Olika Scrum uppfattningar - 6. Olämpliga Produktägare</li> <li>7. För engagerade Produktägare - 8. Otillgängliga Produktägare</li> <li>9. Samordning av flera team - 10. Olika kultur - 13. Kommunikationen</li> </ol>

Figur 3 – SWOT-Matris av empiri

## 5.1 Styrkor

### 5.1.1 Kundtillfredsställelse

Genom Scrum så ökar kundtillfredsställelsen och möjligheten till förändringar och välkomnande av krav, även sent i utvecklingen (Holzman och Panizel, 2013). Intervjuerna belyser detta genom att Scrum har styrkan att leverera rätt lösningar åt kunder (se bilaga A2, s2). Scrum involverar kunden i hög utsträckning genom kontinuerlig kommunikation och återkoppling och på så sätt kan nya förutsättningar snabbt anpassas till utvecklingen, vilket bland annat har en stor betydelse för kundtillfredsställelsen (Rubin, 2008). Liknande belyser en av intervjupersonerna att Scrum gör att team får en stark förståelse för att de arbetar rätt, vilket gör att arbetet blir effektivare och att rätt saker utvecklas och levereras (se bilaga A2, s 2-3). Genom att kunder arbetar nära med projektet och är aktiva i diskussioner så innebär det även att transparensen ökar, fortsätter en annan intervjuperson (se bilaga A4, s 7). Scrum ser

positivt på och välkomnar förändring av krav i projekten, som är ett av de viktiga Agila principerna som Scrum vilar på (Holzman och Panizel, 2013). Detta belyses positivt i empirin i dialogen om de naturliga avbrotten som sker i Scrum, vilka gör det möjligt att genomföra förändringar och hantera kravförändring under arbetets gång (se bilaga A2, s 3 ).

### 5.1.2 Det bättre samarbetet

Teorin för Scrum belyser styrkor i det nära dagliga samarbetet mellan affärsidé och utvecklare (Holzman och Panizel, 2013) som uppmuntrar till användningen av den bästa formen av kommunikation, ansikte-mot-ansikte (Holzman och Panizel, 2013). Samarbetet ger även möjligheten till enklare planering och snabb återkoppling (Rubin, 2008). Intervjupersonerna beskriver på liknande sätt att ansikte-mot-ansikte kommunikationen blir mer använd som en följd av Scrum och fungerar som ett stöd i det bättre samarbetet (se bilaga A3, s9) (se bilaga A2, s 6). Empirin visar många observationer som är förknippade med positiva aspekter till Scrums sätt att främja samarbete. En av intervjupersonerna belyser detta genom, att jobba nära varandra förebyggs missförstånd som sker när man är utspridd. Går något fel så är det hela teamets fel vilket förebygger konflikter (se bilaga A2, s 3-4) och det nära samarbetet som sker genom projekten förebygger missförstånd (se bilaga A2, s 3). En annan intervjuperson lägger till följande om samarbetet, Scrum är en fantastisk metodik när teamet verkligen känner varandra, vad de är duktiga och är dåliga på och på så sätt kan arbeta i informella roller (se bilaga A1, s 4). Ytterligare belyser en intervjuperson den medvetenhet som Scrum medför om varandras utförda arbete vilket resulterar i mindre risker om en medlem skulle gå förlorad under projektets gång (se bilaga A4, s 8). Samarbetet främjar till bildandet av tvärfunktionella team ”Gain all the man brainpower” belyser en annan intervjuperson (se bilaga A3, s 6). Vidare belyser litteraturen den effektiva mötesstrukturen som är ett av de starkaste argumenten bakom tillämpningen av Scrum (Cho, 2008). Vilket belyses även i intervjuerna eftersom mötena är effektiva, välinformerade och ger bra förutsättningar för att förbättra det kommande arbetet (se bilaga A1, s 7) (se A3, s7). Dessa best practices är bra för kommunikationen och planering av nuvarande och kommande arbete och ger möjligheten att verkligen lyfta förbättringspunkter för att förbättra det fortsatta arbetet, belyser empirin (se bilaga A2, s 5) (se bilaga A1, s 7). Två av intervjupersonerna belyser styrkan med Scrum samarbetet genom att det motiverar team medlemmarna att ta eget initiativ, de kan välja fritt sin arbetsuppgift, eftersom principen är att varje teammedlem kan ta den uppgiften den vill från den prioriterade backlogen (se tabell 15).

Scrum har också mekanismer för att bygga förtroende, även på distans (Heje och Heje, 2011). Scrum medför styrkor i samarbetet som sker distans, eftersom det minskar det negativa inflytandet som distansarbete kan innebära gällande kommunikation, koordination och kontroll (Holmström et al). Detta belyses även empirin, genom att det nära samarbetet, även ger effektivitetsvinster under arbete på distans, även ifall samarbete görs bäst då utvecklare sitter bredvid varandra (se bilaga A3, s 2-3) (se bilaga A2, s 7)(se bilaga A4, s 9).

Vidare lyfter Moe et al (2011) och Vlaanderern et al (2010) fördelar med samarbetet eftersom möjligheter till återanvändning och integration ökar, vilket uppstår då managers och utvecklare pratar med varandra, som i sin tur kan skapa mjukvaruprodukter av högre kvalitet.

### **5.1.3 Motivation, snabbhet och teamets gemensamma segrar**

Scrum ökar motivationen, arbetet blir teamets gemensamma ansvar, vilket ökar motivationen och tidsåtgången i att färdigställa och validera viktiga delar av arbetet innan datumet för sprinten gått ut (Rubin, 2008). Detta belyser även två av intervjuerna, att tillämpningen av Scrum medför att något levereras inom en mycket kort tid (se bilaga A1, s 4) (se bilaga A3, s1). Arbetet blir extremt mycket mer snabbriktigt (se bilaga A3, s1). Och även att Scrums projektstruktur och dokumentation är vidare enkel, vilket innebar att själva skapandeprocessen utmärks av snabbhet (Se bilaga A1, s 3). En intervju fortsätter belysa Robin (2008) genom att Scrum medför att olika avdelningar har en större tendens till att samarbeta i produktutvecklingen, gällande idéer och utveckling (se bilaga A3, s1). Vidare säger samma intervjuperson att teamen får en ökad motivation, genom att de regelbundet kan fira de gemensamma segrarna oftare, genom de snabbare leveranserna (se bilaga A3, s1).

Heje och Heje (2011) lyfter styrkan med att Scrum ger ett gemensamt språk och ett mål för teamet att sträva efter. Detta belyser även en av intervjupersonerna som menar att Scrum medför att Micro Management undviks och det blir ett nära samarbete som gör att missförstånd undviks i högre utsträckning (se bilaga A2,s 3). Det blir också teamets gemensamma ansvar att nå team målen då teamet levererar något tillsammans genom ett gemensamt mål även då arbetet sker på distans (se bilaga A1,s 2-3).

### **5.1.4 Utforskande, lärande och kunskapsspridning**

Scrum ger möjligheten att snabbt utforska nya tillvägagångssätt, nya idéer och att snabbt lära sig vilka lösningar som är framgångsrika och vilka som inte är det. Detta sker genom att en kontinuerlig feedback av pågående arbete görs och pågående loop-lärande vilket är centralt för Scrum (Rubin, 2008). Moe et al (2009) skriver utvecklingsprocessen ska anpassas för att möjliggöra effektivt arbete, genom att ge utrymme för lärande och reflektion. I intervjuerna belyses lärandet, teamet får ett ökat lärande genom arbetets gång, det är möjligt att både ta bort och lägga till under resans gång, utan att det behöver bli kostnader associerade med förändringarna (se bilaga A4,s 2). Scrum ger vidare möjligheten att upptäcka fel tidigare genom att diskussioner kontinuerligt hålls med kund och att kundbehov kan fångas med upp till 100 % fortsätter samma intervjuperson (se bilaga A4,s 2). En annan av intervjupersonerna underströk även att Scrum ger ökade möjligheter till utforskandet på vägen tills projektet är färdigt.

Cho (2008) fick fram i sin forskning att bristen på dokumentation kan bli ett allvarligt problem när nya utvecklare ansluter sig till teamet, eller när utvecklare arbetar på nya delar som de tidigare inte har någon erfarenhet av. Intervjuerna däremot visar inte att teamen har några problem som Cho (2008) funnit i sin forskning utan tvärtom. En av intervjupersonerna understryker att dokumentera tillräckligt räcker gott och väl och det finns inga behov av att dokumentera mera eftersom kunskapsspridningen fungerar och främjar en ökad teamkänsla (se bilaga A2, s 5) (se bilaga A3, s 7). Kunskapsspridning ger även kraftfullare team framöver fortsätter annan intervjuperson genom att flera kan göra varandras arbetsuppgifter (se bilaga A2, s 5) (se bilaga A3, s 7). Kunskapsspridningen ger ytterligare styrkor genom en ökad teamkänsla som ett led av bättre samarbete, öppenhet, tydlighet blir till samtidigt som team medlemmarna motiveras av varandra och har en större tendens att samarbeta gällande idéer och utveckling (se bilaga A1, s 9) (se bilaga A1, s 7). Vidare betonar två av intervjupersonerna att Scrums visuella dokumentationen är bättre, det behövs inte längre skrivas flera hundra sidor dokumentation som aldrig läses istället används mera visuell dokumentation som fungerar som en central kunskapsspridare (A4, s9). Vad skillnaderna mellan Cho (2008) och våra observationer beror på är svårt att fastställa och det kan behövas fler studier innan ett säkert svar kan ges men våra observationer tyder på att den mindre dokumentationen är en styrka.

### **5.1.5 Produktionsflödet**

En Intervju belyser fördelar för produktionsstyrningen genom att Scrum är bra för planering, stötta och styra vad som ska utvecklas under varje iteration (se bilaga A4, s 2). Vidare fortsätter en annan person; User stories är även bra eftersom de fångar både rollen och nyttan på ett bra sätt (se bilaga A1, s 7). Teorin lyfter mera detaljerat genom Rubin (2008) styrkorna som projektutveckling med Scrum för med sig, genom kontinuerlig utveckling av projektet i mindre delar minskar risken i projektet. Detta har följder även för resurserna, det blir ett mindre överskott av resurser som inte används. Det begränsar även mängden pågående arbeten med fördelar som: (1) minskad cykeltid, (2) minskat variabilitetsflöde, (3) påskyndad feedback, (4) minskad risk, (5) minskat "overhead", (6) ökad motivation och snabbhet, (7) och minskad kostnads och tidsökning (Rubin, 2008) Skriver vidare om Timeboxings viktiga fördelar (1) det begränsar mängden pågående arbete, vilket reducerar risken för finansiell förlust (2) tvingar fram till prioritering, så att det viktigaste arbetet utförs först. Vidare styrkor kommer (3) genom ökad motivation till att färdigställa och validera viktiga delar av arbetet innan datumet för sprintens slut. De resulterar även i att (4) onödig perfektionism undviks (Rubin, 2008).

### **5.1.6 Scrum struktur och roller**

Heje och Heje (2011) skriver att scrumprojekt har en struktur för möten som fungerar väl för kommunikation inom projektgruppen. Liknande stödjer Empirin detta resonemang genom att

en av intervjupersonerna belyser att sprintarna är jättebra eftersom de skapar väldigt mycket struktur för utvecklingen av projekten även ifall de kan behövas ändras ibland, när saker och ting händer (se bilaga A4, s4). Vidare belyser två av intervjupersoner att retrospectives fungerar mycket bra eftersom de ger möjlighet att förbättra det kommande arbetet (se bilaga A2, s 5) (se bilaga A1, s 7). Vidare belyste två av intervjupersonerna att rollerna var en styrka eftersom de ger en tydlighet och var mycket bra att arbeta med även ifall Scrum Mastern ofta får för mycket uppmärksamhet i hänsyn till viktigt ansvar i jämförelse med Produktägaren (se bilaga A4, s 5) (se bilaga A2,s 3). Även ifall svagheter har kommit fram gällande roller så tyder mycket av empirin på att Scrums uppbyggnad är en styrka som uppskattas att tillämpas.

## 5.2 Svagheter

### 5.2.1 Väldigt kommunikationsberoende

Kommunikation mellan utveckling och verksamhet är väldigt krävande för båda parter. Lee (2012) skriver att vid uppbyggnaden av kravspecifikationen är en rik kommunikation med intressenter, Produktägare och team medlemmar viktig för insamling av kunskap vid definition av kvalitet, affärskrav, användarkrav, produktkrav. Vidare framgår det ur empirin att för att hela konceptet ska fungera så måste det finnas en kontinuerlig dialog med verksamheten under hela projektets gång (se bilaga A4,s 5).

### 5.2.2 Sprintar

Vår studie visade att jobba iterativt inom Scrum, det vill säga i Sprintar, hade betydligt mer svagheter än vad vår litteraturstudie pekade på. I teorin kunde det stå en konflikt mellan behovet av att visa kortsiktiga resultat för de kortsiktiga målen och de långsiktiga målen som behövs för att säkerställa kvalitet i lösningarna, vilket ses som en svaghet av iterativt arbete (Moe et al, 2009). Utvärdering av intervjuerna, visade dock att det finns fler faktorer gällande Sprintar som medvetenhet bör finnas om, Arkitektur för stora system kräver långsiktiga beslut och det är problematiskt att genomföra förändringar i den (se bilaga A1,s 2). Även vid spektrumets andra ände kan planering i Sprintar vara olämplig; när det är nödvändigt att kontinuerligt leverera utveckling i högre hastighet än vad som är möjligt i Scrum (bilaga A3,s 2). Vidare kan det vara så att det inom Business Intelligence inte är möjligt att demonstrera sprinten vid sprintens slut då utvecklingen av back-end tar lång tid och det kan ta över ett år innan front-end delen som har verktygen börjar utvecklas (Bilaga A1,s 5).



### 5.2.3 Ledarskap

Viss överensstämmelse finns mellan vad vi har funnit i vår studie och andra studier. Problematiken med hantering av ledarskap visade sig vara ett återkommande ämne. Lee (2012) skriver att Scrum är beroende av ett effektivt fungerande informationsbaserat ledarskap, annars finns risken att medlemmar prioriterar individuella mål om inte klara direktiv inte ges. Ytterligare det bristande delade beslutfattandet, fungerar inte detta så kan det medföra brister i leveransen (Moe et al, 2009). Vidare så skriver Moe et al (2009) att förvirring och osäkerhet kan uppstå som följd av att en ScrumMasters definition av "done" är bristande. Vår studie visar att när ingen har ledarskapet, och inte ansvarar för välmående, effektivitet och lösning av konflikter så sjunker produktivitet och frustration uppstår (se bilaga A1,s 4). Vidare så kan prioritering av arbete i mindre team vara ett problem (se bilaga A3,s 4) En intervjuperson lyfte att vid konflikthantering kan auktoritet, som i Scrum är informell, vara ett problem ifall en junior utvecklare är ScrumMaster och någon senior utvecklare inte respekterar ScrumMasters auktoritet. Det kan vara tufft när det verkligen är en konflikt som teamet själv inte löser (se bilaga A2,s 4). Liknande stödjer Holmström et al (2006) detta då lösning av konflikter inom Scrum kan vara ett problem på grund av avsaknaden av ett forum för att lösa dessa (Holmström et al 2006).

### 5.2.4 The Team

Gruppdynamik visade sig vara ett återkommande tema i vår studie till skillnad från andra studier. Cho (2008) förklarar att med det nära samarbetet, som är viktig inom Scrum, så begränsas möjligheten till en privat sfär för arbete, medlemmar är konstant utsatta för ett brus av kommunikation. I vår studie fann vi att det fanns en ytterligare nackdel med närsamarbetet; det tar inte hänsyn till individer som trivs bäst med att enbart fokusera på arbetet. Det är inte rimligt att ställa kravet på dessa att vara jättesociala, ta mycket ansvar och tänka på saker de är inte duktiga på eller intresserade av (se bilaga A1,s 4). Vidare så tar Scrum inte hänsyn till grupp sammansättning; Man måste vara medveten om vilka styrkor och svagheter man har på både personlig och kompetensnivå och sedan anpassa arbetsuppgifter efter det (se bilaga A1, sid). Ytterligare så kan Scrum inte hantera de kompetenser som finns i en grupp med avseende på om någon medlem med viss kompetens skulle gå förlorad. Olika individer har olika kompetenser och tvärfunktionella grupper kan inte kompensera för förlust av specialkompetens (se bilaga A4,s 8).

### 5.2.5 Flera teams, och distribuerade team.

Vår studie visar att samordning av flera team, som arbetar på samma projekt kan vara problematisk i Scrum. Det kan vara svårt att samordna och få till möten för att få överblick i vad de olika teamen gör (se bilaga A3,s 3). Cho (2008) skriver att ett annat problem med flera team inom Scrum är att bristande kommunikation kan leda till dubbelt arbete. Vidare, så visas

i en annan studie att i fallet av distribuerade team, att det ofta uppstår en känsla av att man arbetar i två olika teams och inte ett (Holmström et al, 2006).

### 5.2.6 Resurser

I både vår studie och andra studier fann vi sammantaget ett fåtal brister gällande hantering av resurser i Scrum. Arbetet av Cho (2008) berättar att de frekventa mötena i Scrum kan innebära stor tidsåtgång och ibland har de även visats oproduktiva. Moe et al (2009) skriver om att Product Backlogs; planeringen av resurser görs gentemot denna och brister kan givetvis ställa till förödande konsekvenser. I vår studie fann vi att planering av halvtidsresurser kan vara ett problem i Scrum, då krävs det ett annat sätt av styrning, då krävs kontroll, och då krävs ledning (se bilaga A1, s 8). Ytterligare visar den att vid allokering av resurser kan kunden ibland vilja ha mer kontroll på att man rapporterar allokering, beläggning, och debitering. Vilket går utanför Scrum och hanteras av projektledare (se bilaga A2, s 6).

### 5.2.7 Product Owner och kund

Vid jämförelse av vår studie och andra studier fann vi att aspekten av produktägaren och hur denna hanterades i Scrum visade sig ha fler svagheter i vårt arbete. Scrum är inte i bra för alla situationer utan det handlar mycket om förhandlingen med kunden, hur man kommer att få in uppdraget, och hur man vidareutvecklar affären och någonstans under början görs bedömningen om hur man kan angripa projektet på ett bra sätt (se bilaga A1, s 1). Rör det sig om traditionell upphandling där kunden verkligen vill veta exakt vad den kommer att få i slutet, till vilket pris då är det tämligen värdelöst att arbeta med Scrum (A1, s1). Enligt Cho (2008) så är Scrum en mycket långsam metod när utvecklare inte har mycket bra förståelse för kundens behov. Finns inte förståelse för kunden så krävs det omfattande merarbete till följd av förtydligande. Vidare så fann vår studie att för optimalt resultat, så krävs det att enbart kunden kan agera produktägare (se bilaga A1, s 6). Produktägaren har en viktigt och central roll med mycket stort ansvar i Scrum (se bilaga A1, s 4), men trots detta så visar resultat av en intervju ur denna studie att i Scrum så läggs för stor tonvikt på ScrumMastern och rollen som Produktägaren har hamnar i skymundan trots att den kan ses som den viktigare av de två (se bilaga A4, s 5). Och en annan intervju återgav att Produktägaren har en tendens att bli överbelastad (se bilaga A3, s 5). Vidare så fann vi i vår studie att Scrum, fast att det är en jättebra metodik, så kräver den att kunden måste ha vanan att arbeta på detta vis (se bilaga A4, s 2), detta inte minst på grund av att hela konceptet, kommunikation med verksamhet ska fungera, så krävs en kontinuerlig dialog med kunden (se bilaga A4, s 5). Vlaanderen et al (2010) belyser liknande att betänkandet av kraven som produktägarens ansvar för, är central i utvecklingen och om denna har brister så kan detta medföra allvarliga problem för projektet (Vlaanderen et al, 2010).

## 5.2.8 Övriga aspekter

Slutligen finner vi i vår studie ett antal svagheter som berör flera olika aspekter av Scrum, många kan kategoriseras som verktyg eller mekanismer. När det gäller dokumentationen i Scrum, så belyste en intervju person, att det skulle vara bättre om dokumentationen vore en leverabel som allt annat i metodiken (se bilaga A4,s 8). Vidare så anses User Stories vara mycket bra men otillräcklig och i behov av en komplettering av en processbeskrivning för att fånga upp vad en traditionell process bild fångar upp (se bilaga A1,s 7). Scrum saknar även en bra metodik för ekonomisk styrning och är inte anpassad för den delen i projekt som relaterar till teknisk arkitektur (se bilaga A4,s 1-2). Vidare så visar vår studie att Scrum saknade en inbyggd metodik för tidsuppföljning (se bilaga A4,s 3) och slutligen så lyfte en intervju person att med Scrum, så kan det hända att något som utvecklas inte fungerar i drift. Det kan därför behövas mer fokus på operations och drift (se bilaga A4,s 3)

## 5.3 Möjligheter

### 5.3.1 Kundtillfredsställelse

Rubin (2008) skriver om de möjligheter som Scrum medför genom att nya förutsättningarna snabbt anpassas till utvecklingen. Detta har som följd att en högre kundtillfredsställelse blir möjlig genom att kunden kan få det den verkligen vill ha (Rubin, 2008). I likhet stödjer empirin Rubin (2008) med att Scrum ger ökade möjligheter för engagemang, förbättrade samarbete som engagerar kunden. Kunden blir nöjdare när Scrum tillämpas (Se bilaga A2,s 6). Vidare belyser en annan intervju person att kunder älskar att arbeta med Scrum, när de väl har lärt sig att arbeta med Scrum (se bilaga A4,s 7). Rubin (2008) fortsätter att på en mer detaljerad nivå beskriva att Scrum medför möjligheter som behövs för att upptäcka dysfunktioner och slöseri som förhindrar att organisationer når sin verkliga potential (Rubin, 2008). Detta står i likhet med innehållet i en av intervjuerna, där intervju personen berättar att alla gör fel ibland och kommer alltid att göra fel men med Scrum så finns det en god chans att man upptäcker felen tidigare (se bilaga A4, s4). Williams (2011) instämmer i resonemanget i teorin och lägger till att Scrum förbättrar resultatet i kombinationen med andra metoder och minskar antalet fel.

### 5.3.2 Engagemang och förankring

Empirin belyser i större utsträckning de utvidgade möjligheterna än teorin. Vi fann i Empirin att flera möjligheter ges när en budget finns men kunskapen inte finns fullt ut finns för hur den ska användas, utan det är istället nödvändigt att utforska efter hand för att få fram en kravbild (Se bilaga A1,s 1). Vidare lyfter empirin att Scrum ger mer möjligheter, då det råder

osäkerhet i hur man bäst bör organisera sig och planera för kravhandling. Samma intervjuperson lyfter vidare att genom att kunden får möjligheten att bjudas in i verksamheten så sker en automatisk förankring av projektet hos kunden (se bilaga A4,s 7). Vi fann även tydligt i Empirin att Scrum ger vidare ökade möjligheter för ett djupare engagemang från kunden sida då kunden är kontinuerligt inblandade i projektet och i diskussioner i ett tidigt skede för att få kundbehoven levererade (Se bilaga A2,s 6).(se bilaga A4,s 4). Visserligen fann vi även i Empirin att det finns en investering för kunder att lära sig Scrum men när kunder väl har lärt sig att arbeta med Scrum så älskar de det och de vill då inte jobba på något annat sätt (se bilaga A4,s 7).

## **5.4 Hot**

### **5.4.1 Kundens förväntningar och andra externa inflytanden**

Resultaten ur vår studie visade sig ha en betydande skillnader i fokus vid fastställandet av hot. I vår studie fann vi att mycket av fokusen hos våra intervjupersoner låg på kunden. De sammanfattade hoten av andra studier rör vid några fler aspekter än vår studie. Tanner och Maureen (2013) belyser att det finns ett flertal olika externa hot som till exempel störningar från miljön, psykologiska, sociala, och organisatoriska influenser. Ett annat exempel på störning kan vara oförutsägbara buggar som behöver lösas i utvecklingen (Tanner och Maureen, 2013). En annan extern faktor är kunden vilket vår studie belyser; enligt en intervju så har kunder genom Scrum inflytande och makt att påverka, vilket kan ibland vara problematiskt om det utnyttjas fel under utvecklingen, vilket Scrum inte kan hantera (se bilaga A4, s 2). Vi finner koppling till detta i en annan studie som belyser att kunden kan vara osäker på krav och tvinga teamet att göra förändringar mitt under sprinten (Tanner och Maureen, 2013). Vidare så kan samarbete med kunden hotas av att denne tidigare har haft dåliga erfarenheter av att arbeta med Scrum (se bilaga A4, s2). I vår studie fann vi även kontakten med kunden, kan hotas av felaktiga förväntningar; det är inte självklart att lösningar hittas enbart för att Scrum tillämpas (se bilaga A2, s 3). Likande resultat visade sig återkomma i en annan intervju, förväntningarna på projektet måste klargöras så att det stämmer med projektet hos projektets intressenter och medlemmar (se bilaga A4, s 9) Vissa kunder tror att allt löser sig med hjälp av Scrum (se bilaga A4, s 4).

### **5.4.2 Kunden och produktägare**

Gällande kunden och rollen som produktägare fann vi i vår studie att finna en lämplig produktägare på kundens sida kan vara problematisk. Individen måste vara beredd att ta på sig rollen och ansvaret att samla ihop kraven (se bilaga A2, s 3-4). Vidare så var en återkommande punkt i våra intervjuer att produktägaren har ett mycket stort ansvar, vilket

innebär att han/hon kan vara svår att nå, ibland måste till och med Scrum Mastern gå in och agera Produktägare, vilket inte är optimalt och den önskade interaktionen mellan produktägare och team uteblir (Se bilaga A1, s 6) (se bilaga A2, s 3-4) (s bilaga A4, s 5). Motsatsen till detta scenario är dock inte heller alltid önskvärt, då det finns ett hot med en produktägare som engagerar sig för mycket och kommer med ändringar på löpande band eller att kunden har ett extremt kontrollbehov och därmed kan störa projektet eller påverka det tekniska väldigt mycket (se bilaga, A2, s 3-4) (se bilaga A1, s 6). Slutligen visade vår studie att vissa kunder har helt olika uppfattningar om hur Scrum ska tillämpas vilket i sig kan utgöra ett ytterligare hot för projektet (se bilaga A4, s 2).

### **5.4.3 Missförstånd och Samordning**

I teorin fann vi att i Scrum så kunde uppdelning av den strategiska projektplanen till handlingsplan vara problematisk till följd av att medlemmar hade olika mentala modeller över vad som är menat att bli levererat. Teamet och Produktägaren förstod inte komplexiteten i de kraven som projektet hade (Moe et al, 2009). Ett annat hot fann vi är brist på kunskap och technocracy vilket innebar att när teamet inte hade rätt kunskap så kunde de inte ge råd och feedback till andra när de diskuterade problem eller hur en uppgift ska utföras (Moe et al, 2009). Konsekvenserna av Technocracy vilket innebar att expertkunskap medförde att super beslutstagande tog form; vissa kunniga utvecklare gjorde besluten, uteslöt andra från den delade beslutsprocessen. Vidare så skriver Moe et al (2009) om beslutsuteslutande (Decision hijacking); vissa medlemmar i teamet gjorde beslut utan att informera de andra i teamet.

### **5.4.4 Team och partnersamarbeten**

I vår studie fann vi att bristande kommunikation och överblickbarhet mellan team vara ett hot eftersom ett av teamen inte vet vad det andra teamet gör och ändringar som teamet gör påverkar det andra (se bilaga A3, s 3-4). Vidare så fann vi i en studie i teorin att Partnersamarbeten kunde vara ett hot, partners kunde ses som en konkurrent och planer och mål var därför inte exponerade helt eller för sent, vilket ibland ledde till stora problem i utvecklingen och i kommunikationen (Moe et al, 2009).

### **5.4.5 Kultur**

Enligt en intervjuperson så framkom att skillnader i kulturer och förhållningsätt till arbete kan utgöra ett hot för projektet (se bilaga A4,s 9). Detta är något som även belyses av Holmström et al (2006) som tar upp problematiken i attityder till hierarkier, känsla av tid, kommunikationens stil samt behovet av struktur, vilket kan skilja sig mellan länder. Heje och Heje (2011), skriver att kvalitén är en aspekt som kan påverkas av detta eftersom förväntningarna och uppfattning på kvalitet kan skilja sig mellan kulturer och länder.

### 5.4.6 Organisation och övriga hot

Resultat i andra studier visade att det finns vissa organisatoriska hot som bör tas hänsyn till. Enligt Tanner och Maureen (2013) så har management ibland en bristande förståelse och en tendens till att störa projektets gång. Detta till följd av att ledningen inte fullt förstår Scrum teamet och processerna som omfattar projektet. Ett annat hot är de organisatoriska missanpassningarna, till följd av att Scrum teamet inte blir inbjudna att vara involverade i de strategiska besluten. (Moe et al, 2009). Ytterligare hot utgörs utav att konflikter i intressen gällande prioriteringar inom företaget, vilket innebar att projektet ofta förlorade resurser, vilket gjorde teamets självstyre omöjligt (Moe et al, 2009). Slutligen finns hotet gällande ändringar i sprintarna inte efterföljs och teamen tvingas avbryta sitt arbete och planera om eller göra om sitt arbete (Moe et al 2009).

<p><b>Styrkor</b></p> <p><b>Kundtillfredsställelse</b> Kundtillfredställa - Leverera rätt lösningar åt kunder Involverar kunden i hög utsträckning - Arbeta effektivt och rätt Transparens - Välkomnar förändring - Naturliga avbrott Hantera kravförändring - dagligt, nära samarbete mellan affärsida utvecklare</p> <p><b>Det bättre samarbetet</b> Bästa formen av kommunikation, ansikte-ansikte - Enklare planering, snabb återkoppling - Missförstånd förebyggs - Kollektivt ansvar Ökad medvetenhet om pågående, och utfört arbete Tvärfunktionella team - Effektiv mötesstruktur Motiverar till initiativtagande - Bygger förtroende, även på distans Distanssamarbete - Återanvändning och integration ökar</p> <p><b>Motivation, snabbhet och teamets gemensamma segrar</b> Snabbriktigt - Projektstruktur - Enkel dokumentation Fira gemensamma segrar oftare - Gemensamt språk - Gemensamt mål</p> <p><b>Utforskande, lärande och kunskapspridning</b> Utforskande - Loop-lärande - Uptäcker fel tidigare Kunskapsspridning - Ökad teamkänsla - Visuellt dokumentation Dokumentation för kunskapspridning</p> <p><b>Produktionsflödet</b> Produktionsstyrning - Planering, stötta och styra - User stories Minskar risken i projekt - Minimerar slöseri av resurser Minskad cykeltid - Minskad variabilitetsflöde - Påskyndad feedback Minskad overhead - Minskad kostnads- och tidsökning Reducerad risk för finansiell förlust - Prioriterar viktigaste arbete först Underviker onödig perfektionism</p> <p><b>Scrum struktur och roller</b> Sprintar skapar struktur - Retrospectives - Roller ger tydlighet</p>	<p><b>Svagheter</b></p> <p><b>Väldigt kommunikationsberoende</b> Krävande kommunikation - Kommunikationsberoende - Situationsberoende</p> <p><b>Sprintar</b> Konflikt mellan kortsiktiga och långsiktiga mål - Långsiktiga beslut Kontinuerlig levera utveckling i högre hastighet - Demonstration</p> <p><b>Ledarskap</b> Beroende av informationsbaserat ledarskap -Definition av "done" kan vara bristande - Ingen tydlig ledarroll - Prioritering av arbete i mindre team - ScrumMasterns auktoritet - Konflikthantering</p> <p><b>The Team</b> Begränsad privat sfär - Tar ej hänsyn till individuell motivation till social engagemang. - Tar ej hänsyn till gruppsammansättning Kompensation för förlorad kompetens.</p> <p><b>Flera teams, och distribuerade team.</b> Samordning av flera team - Kommunikation mellan teams Teamkänsla vid distansarbete</p> <p><b>Resurser</b> Tidskrävande mötesstruktur - Product Backlog och resursplanering Planering av halvtidsresurser - Ekonomisk styrning</p> <p><b>Produktägare och Kund</b> Långsam metod vid otillräcklig förståelse för kundens behov Kund måste agera produktägare - Stort kritiskt ansvar för produktägaren - För mycket fokus på ScrumMaster som roll - Scrum är krävande för kunden - Kräver kontinuerlig dialog med kund</p> <p><b>Övriga aspekter</b> Dokumentation kan upplevas som bristande - Processbeskrivning saknas - Saknar metodik för tidsuppföljning - Mer fokus på operations och drift</p>
<p><b>Möjligheter</b></p> <p><b>Kundtillfredsställelse</b> Kundtillfredställelse - Engagerade kunden - Kunden kan älska Scrum Upptäcka fel och dysfunktioner tidigt - Förbättrade resultat</p> <p><b>Engagemang och förankring</b> Lämplig när det inte finns kravbild men en budget finns. När det råder osäkerhet i organisering, planering av kravhandling Automatisk förankring av projekt hos kund</p>	<p><b>Hot</b></p> <p><b>Kundens förväntningar och andra externa inflytanden</b> Miljö - Psykologi - Sociala - Kundens makt Störningar under pågående sprint - Kund har tidigare dåliga erfarenheter Kund har felaktiga förväntningar</p> <p><b>Kund och produktägare</b> Hitta lämplig produktägare - Otillgänglig produktägare För engagerad produktägare - Skilljande uppfattningar av Scrum</p> <p><b>Missförstånd och samordning</b> Olika mentala modeller - Technocracy, brist på kunskap Superbeslutstagan pga Technocracy -Beslutsuteslutande (Decision hijacking)</p> <p><b>Team och partnersamarbete</b> Bristande kommunikation och överblickbarhet mellan team Partnersamarbete, ses som konkurrent</p> <p><b>Kultur</b> Olika kulturer - Olika språk - Religiösa skillnader</p> <p><b>Organisatoriska</b> Management, bristande förståelse - Organisatoriska missanpassningar Prioriteringar inom företaget - Teamet tvingas till avbrott under sprint</p>

Figur 4 – Sammanfattande SWOT-Analys

## 6. Slutsats

Syftet med vår studie har varit att belysa Scrum dels genom tidigare teori och genom genomförandet av intervjuer med fyra stycken projektledare med erfarenhet av Scrum, för att på så sätt få en bättre beskrivande, genomarbetad helhet kring kunskapen som finns gällande Scrum avseende; Styrkor, Möjligheter, Svagheter och Hot. Nedan presenteras våra slutsatser under dessa kategorier.

### 6.1 Styrkor

Empirin lyfter att Scrum är en utmärkt projektmetodik att tillämpa när medlemmarna i teamet verkligen känner varandra, vad de är duktiga på och är dåliga på och projektmedlemmarna på så sätt kan arbeta i informella roller. I empirin och i såväl som teorin är den ökade kundtillfredsställelsen central som uppnås genom de positiva effekter som kundens nära involvering innebär för projektet, i utvecklingen av lösningarna samt den kontinuerliga öppenheten för förändring. Vidare betonas att det bättre samarbetet främjas genom ansikte-ansikte kommunikation. I empirin ser vi i större utsträckning de positiva effekter som ett nära samarbete genom Scrum innebär, som resulterar i bättre medvetenhet, motivation, initiativ och kunskapsspridning. Vidare betonas styrkan med teamets gemensamma ansvar för att nå mål, vilket även förebygger och medför bättre konflikthantering. Scrum har även visat sig ha en mycket effektiv mötesstruktur för samarbetet. Och även ifall empirin visar att utveckling fungerar bäst när utvecklarna sitter bredvid varandra så är Scrum också lämplig och fungerar bra för distansarbete då kommunikation och samarbete främjas. Ytterligare är det centralt i såväl teori som empiri att leveransen, att utvecklingen går fortare, att motivation och tydlighet stiger och att arbetet blir mycket mer snabbt. Detta som en följd också av att dokumentationen och projektstrukturen är enkel.

Vidare finner våra resultat i teorin och såväl empirin att utforskandet är centralt, vilket betonar Scrums styrka i utforskandet i att snabbt finna nya tillvägagångssätt, nya idéer och att snabbt lära sig vilka lösningar som är framgångsrika. Det kontinuerliga lärandet är centralt och möjligheten att fånga upp det fulla kundbehovet. Dokumentationen upplevs även som en klar styrka i Empirin och fungerar som en kunskapsspridare och att därmed fler kan göra varandras arbeten, vilket även får effekten av en ökad teamkänsla, samarbete och kraftfullare team. Cho (2008) fann i sin forskning att dokumentationen kunde bli ett allvarligt problem när nya utvecklare ansluter sig eller när utvecklare arbetar på nya delar som de tidigare inte har någon erfarenhet av. Våra empiriska resultat visar däremot inte några sådana problem



gällande kunskapsspridning och mera forskning eller artikelstudier är motiverade för att nå en djupare förståelse för dokumentationens positiva sidor och avigsidor. Våra studier visar även att Scrum är bra för produktionsstyrning, planering och att stötta och styra vad som ska utvecklas under varje iteration. Teorin via Rubin (2008) belyser i större detalj styrkorna med den kontinuerliga utvecklingen av mindre delar i projektet. Vidare belyser empirin och utvecklar Heje och Heje (2011) att scrumprojekt har en struktur för möten som fungerar väl för kommunikation inom projektgruppen. Framförallt lyfts fördelarna med retrospectives och idén med sprintar för att skapa struktur. Även rollerna upplevdes positivt även ifall produktägaren står lite sidan om ”ljuset” som ScrumMastern befinner sig i med ett mycket viktigt omfattande och kritiskt ansvar.

## 6.2 Svagheter

Något som var återkommande i vår empiri var att Scrum inte är en helhetslösning, eller ”Silver Bullet”. Scrum är situationsberoende, och inte en optimal lösning för alla lägen. Rör det sig om traditionell upphandling där kunden verkligen vill veta exakt vad den kommer att få i slutet och till vilket pris då är Scrum en mer eller mindre värdelös metod att använda. Vidare finner studien att en av de mest grundläggande aspekter av både Scrum och agilt dvs. att arbeta iterativt inte alltid är möjligt. Inte allting kan göras i Sprintar, där det är möjligt att ändra riktning efter behov vid slutet av varje sprint. Ibland krävs ett mycket långsiktigare perspektiv och ibland även ett kortsiktigare. Detta kan ses som ett problem i det agila arbetssättet som Scrum vila på eftersom Scrum säger att en sprint ska vara 2-4 veckor. Vidare att det kontinuerliga samarbetet med kunden som kräver demonstration av funktionalitet efter varje sprint, inte är möjligt i alla situationer. Med andra ord så har Sprintar tydliga begränsningar.

Vidare så visade vår studie att Scrums hantering av ledarskap, det vill säga självorganiserade team har tydliga nackdelar. Konflikthantering, välmående, gruppammansättning, individers sociala förmåga är bland annat element som inte hanteras direkt i Scrum. Scrum kräver, väldigt mycket av gruppen, speciellt för en nybildad grupp där individer inte känner varandra.

Scrum ställer även väldigt stora krav på kunden, samarbetet med kunden och produktägaren. Scrum metodiken förmedlar ett intryck av att den viktigaste rollen är ScrumMaster men i vår empiri fann vi att så inte är fallet. Produktägaren har ett mycket stort ansvar för projektet och för optimalt resultat bör denna roll inte utövas av någon annan än själva kunden.

Ekonomistyrning, samordning av flera teams, dokumentation och processbeskrivning var ytterligare problemområden men sammanfattningsvis så visade sig att några av Scrums mest framstående aspekter; sprintar, självorganiserande teams och kundsamarbetet visade sig ha de flesta svagheter i vår studie.

## 6.3 Möjligheter

Empirin stödjer Rubin (2008) som belyser möjligheterna med en högre kundtillfredsställelse genom att nya förutsättningar snabbt anpassas till utvecklingen och möjligheterna med ett ökat engagemang, förbättrade samarbeten som engagerar kunden och att när väl kunder lärt sig att arbeta med Scrum så älskar de det. Vidare lyfter Rubin (2008) och Williams (2011) möjligheterna med få ett bättre resultat och att minska antalet och upptäckten av fel tidigare. Empirin belyser i större utsträckning de utvidgade möjligheterna. Möjligheter ges när en budget finns men kunskapen inte finns fullt ut för hur budget ska användas, då det är nödvändigt att utforska efter hand för att få en kravbild. Scrum ger ökade möjligheter när osäkerhet råder i hur man bäst organiserar sig och planerar för kravhandling. Möjligheterna med automatisk förankring belyses även i Empirin och möjligheterna att få kundens fulla behov levererade.

## 6.4 Hot

Det som var mest framträdande i vårt arbete gällande hot var skillnaden i fokus mellan empiri och teori. Intervjupersonerna hade mycket fokus på hot relaterat till kunden medan teorin gav en bredare bild. En av de viktigaste aspekterna i Scrum är det kontinuerliga nära samarbetet med kunden. Scrum är väldigt beroende av kunden och det gör samarbetet i vissa fall till ett hot. En för engagerad kund kan störa projektet samtidigt som en icke-engagerad kund som inte är tillgänglig eller ointresserad kan få hela projektet att falla. Kunder kan ha fel förväntningar gällande Scrum, de kan tro att eftersom Scrum tillämpas så löses allt. De kan även haft tidigare dåliga erfarenheter av Scrum och har därför en negativ inställning. Att kunden har en annan uppfattning om arbete och tillämpning av Scrum kan också vara ett hot.

I andra studier fann vi att organisatoriska hot, att en organisations prioriteringar av resurser kan påverka projektet negativt. Management har visats haft bristande förståelse för Scrum och kan störa projektet eller så har organisatoriska missanpassningar lett till att teamet inte blivit involverade i strategiska beslut. Ledning har visats genom sitt inflytande avbrutit sprintar. Vilket inte bör göras enligt Scrum, förändringar ska inte tvingas fram på ett sådant sätt.

Scrums självorganiserande team kan hotas av technocracy, vilket i sin tur kan leda till att utvecklare har expertkunskap och tar besluten i teamet. Beslutstagande i teamet kan vidare hotas av Decision-hijacking, det vill säga beslutsuteslutande där medlemmar tar beslut utan att informera de andra om dessa.

Utöver de nämnda hoten gällande kund och organisation så finns det ytterligare hot som berör partnersamarbete, samarbete mellan flera team, kulturella och religiösa skillnader, olika språk och miljöer.

## 6.5 Begränsningar

Vi har genomfört vår forskning genom fyra stycken intervjuer med projektledare som har betydande erfarenhet av projektledning och Scrum. Antalet intervjuer är för få för att dra några definitiva slutsatser. Flera intervjuer hade kunnat ge en större bild men vår forskning ger ändå återkommande punkter som visserligen skiljer sig men fortfarande rör samma aspekter. Vilket förmodligen beror på vår öppna intervjustruktur som tillåter mera variation i svaren gällande formulering och perspektiv.

En mer styrd intervju med fler frågor och mindre öppna frågor skulle kunna minska variationen men å andra sidan skulle viktiga erfarenheter och kunskaper kunna ha filterats bort. Vår begränsning är samtidigt en styrka som motiverar till fortsatt forskning inom områdena för våra resultat. Vår forskningsfråga är besvarad men då ämnet är stort så kan vi inte säga att svaret är komplett. Vilket som följd indikerar att det finns en giltighet i våra resultat och att en mer omfattande studie skulle kunna fastställa svaret i en större utsträckning.

## 6.6 Förslag till fortsatt forskning

Vidare forskning som vi rekommenderar är samma frågeställning men som omfattar flera intervjuer för att kunna säkerställa forskningsresultat med statistisk signifikans.

Vidare föreslår vi en fördjupad forskning inom områdena kring fördelar och problematik kring dokumentationen och Scrum rollerna med fokus på produktägarens ansvarsområden gällande fördelar, problematik och hur produktägarens roll bäst bör tillämpas i Scrum. Vi föreslår även fördjupad forskning med fokus på gruppdynamiken i Scrum.

# 7. Bilagor

## Bilaga A1. Affecto Intervjutrascript

### 1.3 I vilka länder är Affecto verksam?

Främst de nordiska länder, men även i Polen, Estland, Lettland möjligen något land till, de står nu på hemsidan.

### 1.4 Vad är Affectos vision?

Ja, alltså vision är ju att hjälpa företagen, personer, individer i företag att (ta) så bra beslut som möjligt, alltså hjälpa dom med deras arbete att få fram den informationen de behöver för att kunna fatta bra beslut, och för att samla ihop informationen. Det är egentligen det. Så är det en del nyttor kopplade till det men jag kan tyvärr (inte) rabbla alla. Har inte kollat de här på ett tag och de ändrats lite, också finns nu på webben nu också tror jag.

### 2.1 Vad heter du? / 2.2 Vad är din yrkestitel?

Jaha, jag skulle börjat med det. Fredrik Grenson jag har jobbat här i 4 år. Jag är specialiserad på projektledning och kvalitetssäkrare, min roll är alltså quality manager och projektledare. Jag gör revision på vårt arbetssätt framförallt när det gäller projektform det finns ju andra leveransformer också, och så ser jag till att utveckla det på rätt sätt och utveckla kvalitetsramverk på ledningssystem som vi har.

### 2.3 Hur länge har du jobbat som projektledare? 2.4 Hur länge har du jobbat med Scrum?

Ja, ett par år, projektledare har jag väl varit sen kanske 10 år de är ju också sedvanligt programmering, databasprogramering, och sen så någonstans fick man välja till slut själv då om man vill satsa på tekniken eller mjuka delarna, människor, utförande. Man kan inte göra både och så då valde jag någonstans där ledning, och människor.

### 3.1 Använder ni Scrum idag?

Ja.

För vissa typer av leveranser, uppdrag och upplägg så passar scrum bättre. Projekt kan ju utföras på en mängd olika sätt beroende på de här faktorerna. 1. beroende på hur styrd man är av budgetar, av ett specifikt slutresultat i änden, så kan man då angripa det här på olika sätt, och scrum lämpar sig väl när man har en budget men att man inte riktigt vet vad man vill göra med den

utan vill utforska de efter hand. Då lämpar de sig bra, men om de är traditionell upphandling, där kunden verkligen vill veta exakt vad de kommer få i slutet, och vill veta exakt vad de kommer kosta då de är de tämligen värdelöst att arbeta Scrum. Så de beror lite på situationen, de handlar mycket om förhandling med kunden, hur man kommer in i uppdraget, och hur vidareutvecklar affären man får kanske börja lite smått och utvecklar det sig man gör den här bedömningen nästan för hur man angriper detta på ett bra sätt. Besvarar de frågan?

### **3.2.2 När använder ni scrum?**

När de är väldigt osäkert, vad som ska göras, under vilka former, och kunden vet inte hur man skall arbeta i projektförmen. Vet inte hur man ska dra upp förhandling, förstår inte hur man ska organisera sig och planera för kravhandling. Dom vet inte själva hur man ska definiera kraven så där är det lämpligt

### **3.2.3 Hur använder ni scrum?**

Det är ju intressant, vi har ju ett kvalitetsramverk som är vattenfalls baserat. Så i för att tag sen hade vi ingen metod alls för scrum överhuvudtaget, men så ville jag testa på det för jag hade några uppdrag från början ett fram för allt där de var nödvändigt. Så då, försökte jag forma om vårt kvalitetsramverk, och tillämpa det, få mer agilt då, och scrum blev det i de här fallet, och de man måste veta med scrum är att de är fortfarande så att man måste utföra ett antal aktiviteter som aldrig kommer ifrån, och de måste göras i viss ordning de kvittar eller om de är scrum, vattenfall eller något annat sätt, man kommer aldrig ifrån att vissa saker måste göra innan dom andra. Försökte ta dom bitarna från vattenfall, och göra det mer iterativt, kortare iterationer, och ställa krav då på utförarna, att de skulle utföra de på ett visst sätt, och åstadkomma nånting inom dom här iterationerna på ett annat sätt än vad som är brukligt. Då började jag göra typ av semi-agil tillämpning och vårt vattenfall.

(Ett) Exempel är att om man ser det framför sig att man ska bygga nånting framöver, då måste man ha en arkitektur, då måste man fatta en hel del viktiga beslut i början, när man köper in dom här grejerna, och sätter hela arkitekturen och då kan man liksom inte efter ett halvår ändra det, det kostar enorma mängder pengar. Man måste ändå skapa sig en bild framöver över många sprinter, kanske över många år framöver, det måste man göra i början man kan inte bara sätta sig ner och börja pilla med grejerna och se vad som händer, de går liksom inte att köra helt total agilt i den här branschen. Lite hemsidor möjligen, men inte om man ska bygga ett helt system som kostar flera miljoner att bygga måste man liksom ha arkitekturen på plats. Ett exempel, där man inte bara kan snurra runt i iterationer hur som helst så därför måste det bli nån slags hybrid.

### **3.2.4 Kompletterar/kombinerar ni scrum med andra metoder?**

Nej nu kan jag inte kanban speciellt bra, något slags tankesätt. de är ju faktiskt, nån annan typ av kod som man tillämpar de på. det är mycket databasprogramering i BI och där kan det vara svårt att tillämpa alla typer av bra metoder.

## 4 SWOT frågor

### 4.1 Styrkor?

De är ju de, man kan ha förhållandevis kort horisont, och försöka leverera någonting inom den horisonten. Strukturera det på de sättet att man ska komma nån slags leverabel som är till gagn, på något sätt, så kan man utforska vart man är på väg på de sättet. Och chansen att stycka upp kravarbetet om kunden inte har möjlighet riktigt i ett längre tidsperspektiv. Och sen är det väldigt enkel metodik, det väldigt snabbt att rigga det liksom, och skapa det hela. Projektstrukturen, och projektdokumentationen relativt enkel också. Mer komplext vid vattenfall fler dokument, mer på djupet och sådär, så de är ju fördelarna med det kan man säga det finns ju både för och nackdelar med det. Jag brukar säga till alla, mina projektgrupper och projektmedlemmar: det finns ju några fundament som scrum vilar på så som jag ser det iaf, och finns inte dom så rasar allt som ett korthus.

### Styrkor (och Svagheter?)

En styrka är att kunden måste vara väldigt väldigt involverad och produktägare, en till tre dagar i veckan måste han vara avse för projektet, verkligen vara tillgänglig. Det är en styrka om kunden kan vara med så mycket men samtidigt, är dom inte med så faller allt, det bara kollapsar, och då blir det som vanligt att man levererar nånting, och så märker man efter att det var inte alls vad jag behövde. Så får man göra om den då, kräver väldigt mycket av kunden, och tydliga besked från den som tillämpar, utvecklar det här; så här måste ni gå till väga, och få till det.

Styrkor, det finns såna här tveeggade styrkor och svagheter. Svagheter de är ju de här man kan jämföra det som en bil, vill man köra på en landsväg bil snabbt behöver man en sån bil, och kör man på en skogsväg behöver man en terrängbil. Det samma med metodik, ibland fungerar agilt bättre och ibland vattenfall.

Jag har inte läst så mycket litteratur om scrum, jag har gått ett par kurser, men de slår mig liksom att en av de här fundamenten är att de ska vara självorganiserande, det är ingen som skall leva ledare där, leda teamet, de ska själv förstå vad de ska göra och fånga upp saker som inte görs, och så. Det är ju otroligt komplext, det görs inte över en dag.

Så de som slår mig, dom som har hittat på scrum gör det för att dom bara har haft dåligt ledarskap, dom har inte haft bra ledare som har kunnat göra det som en ledare ska göra, typ projektledare. Det är ren frustration. Vi får fan fixa det själv, gruppen får själv ta hand om dom här bitarna, och planera, ta sig an roller och förstå vad en roll innebär. Men nån behöver leda dom här testerna, hur gör jag det här? Men jag gillar det här med tester, jag är duktig med det att det är så det ska fungera, jag kan ju de här lite grann jag tar det, strukturerar upp det här, och de här ansvaret. det är ju bara ren dåligt ledarskap om man inte kan förstå som projektledare. Vi behöver en testledare här och testledare ska göra dittan och dattan, dom aktiviteterna som faller mellan stolarna, dom tillfaller testledaren och liknande. Jag tycker det jävligt lustigt, det slår mig, det är bara frustration reaktion på det.

Vi har ju grupper här där jag ser när ingen har ledarskapet, och ansvarar för välmående och effektivitet, så sjunker produktiviteten. Det är väldigt tydligt ansvar att en grupp fungerar tillsammans, och lösa konflikter, och ta ansvar för det. Gör ingen det så blir det frustration och då sjunker produktiviteten. Har sett en hel del exempel på, någon måste ta de ansvaret och det är uttalat att scrum att ingen har ledaransvaret men om alla sitter och retar sig på en i teamet. vad händer då? vem tar ansvar för det? De kan man ju se, om man sitter och funderar på såna saker när man arbetar. Det är klart det tar ju tid från de man egentligen ska göra. det är jättelätt att gå in i en grupp medvetet och förstöra skapa konflikter och se till folk blir distraherade, och bryter mot gruppens normer, och störa hela produktiviteten, det är hur enkelt som helst. Och det illustrerar ju att det motsatta är också möjligt att se till att en grupp binds samman, och får samma mål, värderingar och syften, att man löser konflikter och sätter regelverk, finns ju inte där, vem fan tar ansvar för det? Det måste nån göra det är en sån uttalad del som man inte hanterar, liksom dom här administrativa sakerna uppföljning, och dom bitarna. Det är ju ingen som tar ansvar för det, så i vissa fall rekommenderas det ju att man har en projektledare vid sidan om, då har man ju kringgått principerna tycker jag.

Jag tycker att det bara handlar om dåligt ledarskap då, det är också en svaghet; man kan inte sätta ihop ett gäng personer och ett syfte, och förvänta sig att nu tar ansvar för allting, ni får hålla alla bollarna, alla är ju inte såna. Vissa är ju såna; jag tycker om att få en arbetsuppgift på papper, skitduktiga programmerare. Jag vill ha exakt här vad kraven är, jag vill inte ha med andra att göra, bara i nödfall, ha möte, jag vill inte fika, jag vill bara sitta och göra det jag ska, och då trivs jag jättebra. Ska man då tvinga dom att vara med och vara jättesociala och ta massa ansvar, tänka på andra grejer som de inte är intresserade av, och inte duktiga på. Det är inte rimligt att kräva det här av alla människor. Det är såna saker man måste verkligen fundera över då, ta ansvar för dom bitarna och se till att det fungerar. Scrum team fungerar fantastisk när man väl kommit över det här när man har ett gäng som verkligen känner varandra, dom har tatt på sig mer informella roller. Vad dom andra är duktiga och dåliga på. Då funkar det ju bättre, men måste ju komma dit.

Det blir självklart bättre med tiden, om det nu funkar och det inte blir några konflikter på vägen, och teamet bryts upp hela tiden vilket också är ju vanligt, i branschen att man bara gör nya constellationer, så det här är viktiga bitar att tänka på.

### **Möjligheter/Hot**

Teamet inte tar ansvar, och för teamets effektivitet. Löser konflikter när man är projektledare så har man kanske 4 sätt att lösa det, och det är olika syften, olika tillvägagångsätt, olika effekter för hur man löser en konflikt. Är man tidsnöd till exempel så går man in kanske och bara bestämmer hur det ska vara man löser inte konflikten utan man bara fördröjer den medan man kan köra konsensus, då är det kanske en sämre lösning, tar längre tid men man löser konflikten mer. Nån måste ju fatta de här beslutet, nån måste ju lösa det här, och hur ska vi hantera det efter

Det handlar ju om gruppens sammansättning också, att sätta samman en grupp, så får man ju se vilka personlighetstyper det är, och att bara ha en grupp med såna här introverta personer, analytiker det är inte bra, för då ofta kommer man ingenstans, utan alla bara rotar ned sig och analyserar,

analyserar, analyserar. Sen finns det ju de här exekutiva som “vi kör, vi kör, vi kör” och är lite här uppe på fria höjder. Man kan inte bara ha såna heller man behöver en mix, så behöver man sociala personer, då måste ju nån ta ansvar först och sätta samman gruppen, se vad som fattas, vad som finns också lite mer relaterat till det, så måste man vara medveten också vilka styrkor och svagheter man har på både personlig och kompetensmässig nivå, anpassa arbetsuppgiften efter det.

### **Hot utifrån (kunden)**

Ofta är det bara en fördel att kunden är inblandad, det är mer om kunden är märklig, om individen är märklig på något vis, så blir det problem, extremt kontrollbehov eller påverkar det tekniska väldigt mycket. Det finns vissa som är så, är kunden inte inblandad så mycket så faller allt

Det är inte ovanligt heller nån får en uppgift den inte är speciellt intresserad av på kundsidan.

Silver bullet, jag tycker man märker dom som gillar scrum, dom är oerhört bitna, hallelujastämning, the scrum of scrums. Det funkar inte riktigt så, det är ett verktyg bland andra. Man får bedöma om man ska tillämpa det, och då är det jävligt bra. I andra är det helt värdelöst, det måste man vara medveten om. Har träffat man många kollegor och så via utbildning det är verkligen hallelujah-stämning. Jag aldrig varit med nånstans att det finns en silver bullet det är alltid så att man måste anpassa lite, kanske välja vilka verktyg man använder.

Sharepoint, collaboratives kör mer scrum, inte hybrida

### **Angående hur strikt man följer scrum, funktionalitet för presentation efter varje sprint:**

Man försöker få in den aspekten, i BI är det så; man lägger ut det som back-end och front-end back-end samlar man ju upp kraven på hur informationen ska, vilken information som ska in och vilka affärsregler som ska tillämpas, men då är det inte tillgänglig än och utan man via front-end via verktygen. Det är en extrem nytta med det man gör back-end, men det är ju ingen som kan använda och ett back-end projekt kan ta en väldigt väldigt lång tid innan man kommer till front-end. Okej hur ska man göra då om man utvecklat ett år, eller ett och ett halv om man inte ens har kommit till front-end. Scrum är kanske inte är riktigt lämpligt där, och om man nu försöker tillämpa det så kan man inte visa och då får man liksom jämföra med kraven och försöker göra något extrakt i Excel, det blir lite svårt.

### **Scrum möten**

Det är bra, de brukar jag också göra i vattenfall. Ibland dagliga möten, när vi kör scrum kör vi inte dagliga. Det är inte dagliga möten men det är ju sprint möten, daily scrum. Vi tillämpar det, så det går snabbt och att man ska göra det man säger, vad jag gjorde igår, vad jag gjorde idag och impediment. Snabbt och effektivt.

Det är viktigt att genomföra daily scrum enligt de vanliga riktlinjerna

### **Angående product owner / kunden**

Nej, det måste vara kunden, ibland är det så att kunden inte kunnat vara med riktigt, och då får nån ta den rollen så och uttolka kraven, men det blir inte riktigt bra tyvärr. Det ska vara kunden,



men man brukar inte kalla product ownern för kunden, det säger inte dom speciellt mycket, utan de är nån som är ansvarig där för att ge oss kraven och får godkänna lösningen

### **Angående ifall Affecto lär ut Scrum till kunden**

Visst lär vi ut vad en product backlog är men det går väldigt snabbt.

### **Dokumentation:**

Krav måste dokumenteras, så är det bara, det kvittar hur bra kompisar man är med kunder, och allt är och finemang, det blir alltid tjaffs. Det är alltid som nån glömt, nånting som nån har sagt, och varför är det så här? Men så här har vi inte sagt, och alla minns tydligt men jo så var det, men det går inte att bara köra upp det i nyllet på dom hur som helst. Trots allt det är en betalande kund, det är superviktigt att dokumentera krav

Sen så måste taktiska och strategiska bitarna också dokumenteras, som arkitekturen och så. Blir det någonting fel där så vill man inte stå med nåtning muntligt, ”varför köpte ni in dom här licenserna? För tio personer så kostar det 5000 men nu vill va ha det här för 2000 pers. Vi kommer få licenskostnader på 20 miljoner.” Såna diskussioner vill man inte, och det är en ganska vanlig situation, det måste dokumenteras.

### **Angående erfarenhet med distribuerade teams**

I vattenfall så har jag det, i scrum har jag också det kan man väl säga

alltid problem med dagliga möten, får det på telefonen.

Teamkänsla: nej inte mer än dom vanliga nackdelarna när teamet sitter placerat på olika platser man missar mycket, det blir mycket mer effektivt om man sitter i samma rum, om man har en fråga och kan knacka varandra i axeln. Det är väldigt värdefullt.

### **Internationella projekt:**

Skulle sätta upp en organisationer för en större kund. Skulle sätta upp deras, utvecklingsorganisation och förvaltningsorganisation, hade ett team i Danmark, ett i USA, och ett i Indien. Det var ganska intressanta möten, tidzoner, förstog inte vad indierna sa, amerikanerna tyckte allting lät bra i början, det hände inte så mycket. Danskarna var mest frustrerade och ville att saker skulle hända. Nån måste tar ansvar för det, och se till att det fungerar, tydliga regelverk, det går inte hoppas på att alla plockar upp bollen. funkar inte

I dom här fallen var det mer vattenfall-projekt, jag ser inte hur det skulle funka med Scrum faktiskt. Det är en inkörning också. Är man inte van vid Scrum måste man vara på plats, och se till att alla fattar hur det ska gå till. Dokumentationen är också en utmaning att få det att fungera. Försökte att utgå ifrån bara user stories, det är liksom kravingången. Det räcker inte man måste skapa en dokumentationsstruktur under det. Det räcker inte bara med user stories, då måste man få med alla på detta, och utbilda alla på det, vilka dokument som ska finnas, hur man ska fylla i dom, hur alla kravbeskrivningar relaterar till user stories, och man nu har user stories, som portalen kravet, så måste man ha en annan typ av beskrivningar längre ned i hierarkin hur hänger det ihop?

Men user stories gillar jag, jag tycker om det, man fångar både rollen och nyttan där. Det är faktiskt väldigt bra men det måste ofta kompletteras, om man nu är med i en processbeskrivning faktiskt. Det är bara i en processbeskrivning man kan se hela nyttan, kvantitativt, kvalitativt, exakt var genereras pengarna som motiverar eller den nyttan som motiverar hela kostnaden för det här. Exakt i vilket steg? Vem ska göra det? Och hur ska man säga till denna individen, rollen och få denna att förstå du måste nu göra på det här sättet, för annars är hela det här projektet meningslöst. Det är ju en förändring i någons arbetssätt, för annars kan man ju bara fortsätta som innan. Så någon måste ändra på något beteende ganska ordentligt för att man ska uppnå projektnytta, och då måste man veta exakt vad?, vem?, och när?, och hur? Och då måste finnas någon form plan för det, benefit realisation plan, måste finnas för det händer inte av sig själv att alla bara fattar "oj nu måste jag göra så här." Verkligen distinkt för då kommer dom där i hörnet där, direktörerna, de beslutet dom fattat, blir glada. Det funkar inte så riktigt. Såna bitar är viktiga, vet inte hur man tar ställning till det i Scrum. Det fångas ju inte i en user stories, det fångas bara i en traditionell process bild, där kan man exakt påpeka på vem som gör vad, i vilket steg och var den nyttan genereras.

Det är ju oftast så att man ej har någon medvetenhet om de här alls i projekt, hur man ska realisera nyttan, alltså planen för det, efteråt. Man bara tror att det händer av sig självt, det är inte alltid så. Särskilt när det är såna här kvalitativa mål, som inte går att mätas med pengar, att folk kommer jobba effektiva på något sätt som inte går att mäta. Kunna analysera mer och inte bara såna träiga grejer som att bara samla ihop informationen. Analysera mera och fatta bättre beslut, det måste någon säga till dom ofta.

### **Planering av resurser**

Det är också ett problem, det i Scrum är när man har resurser fulltid över en sprint, och sen bara fyller man upp deras tid från backloggen, men så enkelt är det ju inte. Utan ibland har man halvtidsresurser då måste man styra på ett helt annat sätt, då måste man ha kontroll på det. Då måste man timeboxa och det kräver ju ledning, alla tänker olika. De funkar bäst när man resurser fulltid, men verkligheten ser inte ut så alltid.

### **Samarbete med en tredje part**

Det händer att vi gör det men då är det våra regler som gäller. det kan vara så att man behöver blackbox leverans får dom sköta det själv då om dom har någon egen metodik. Men det är ju så ofta det är på det viset utan då är det någon nyckelresurs/kompetens från tredjepart som man tar in, då ingår dom som vanlig projektdeltagare.

Det är en så liten pöl vi verkar inom så ofta har dom varit anställda hos oss redan, så det är inga problem, vi har ju iofs har stött problem då var det för att man kontrakterat från andra länder. Eller köper upp någon tredjepartsprogramvara eller liknande och behöver tjänster man associerar med dom här utomlands kan dom vara. Men inom koncernen länderna är det inga problem

## Bilaga A2. Cybercom Intervjutrascript

### 1. Personliga frågor

#### 1.1 Vad heter du?

Magnus Månsson

Cybercom och Vi jobbar ju med consulting, inom telekom och mobila lösningar inriktningar i Malmö bland annat och jag jobbar som konsult mestadels som projektledare, produktägare, utredare och kravinsamling.

#### 1.2 Vad är din yrkestitel?

#### 1.3 Hur länge har du jobbat med Scrum (yrkestitel/projektledare)

Första scrumprojekten för 7-8 år sedan ungefär.

### 2. Generella frågor om Scrum:

#### 2.1 Använder ni Scrum idag?

Det har vi som grundläggande projektmetodik i våra projektupplägg, så vi använder det även ifall det är vattenfalls/fastprisprojekt så jobbar vi ändå med sprintar och dom best practices man har då men då kanske man planerar upp all de sprintarna från början mer eller mindre. Vilket inte är så agilt men man har glädje av att använda metodiken även då eftersom det ofta kommer förändringar som blir lättare att hantera. Då blir det ju så att sprintar som inte är invanda i en sprint blir en del i backlogen. Det är bara de redan prioriterade så det är inte så stor skillnad även om du har en prioriterad backlog där du sätter upp vissa avdelningar och säger att det här är sprint 1 och det här är sprint 2, sprint 3 sprint 4.. så därför tycker jag att det är liksom en gråskala mellan renodlat agilt arbete och mera traditionellt projektledning. Redan förr när man jobbade med säg vattenfallsmetoden så hade man ju ändå ändringshanteringen och man försökte ju skapa delresultat på vägen. Scrum är mera att man har renodlat det och skapat en tydligare metodik kring det. Det är lite granna som objektorienterad programmering innan det fanns officiellt så jobbade många utvecklare på det viset bara det att det kallades inte för objekt orienterat.

#### 2.2 Om svaret är ja:

##### 2.2.1 Varför använder ni Scrum?

##### 2.2.2 När använder ni Scrum?

Jag vet inte om jag skulle kategorisera det, men du känner ofta i ett projekt om alla i projektet jobbar i samma fas att du får en naturlig avdelning där en sprint kan avslutas. Men behöver du

göra mera kontinuerliga releaser då någon jobbar med något som behöver ut nu, och någon med något lite senare så är det kanske svårt att tajma ihop det hela så att det blir en sprint, utan det är bättre att släppa en "continues delivery" kedja, och har du den uppsatt effektivt så att det inte blir för mycket overhead med att göra en release då blir det inga problem egentligen att göra en release. Då blir det inga problem med att releasa mer eller mindre varje feature för sig sedan, men kanske du ändå samlar ihop allt för att göra en acceptanstest så att du har några saker samtidigt då. Det kan bli stelbent när du ska ha det här sprint, speciellt om du ska ha 2 eller 3 veckors sprintar och få ett jämt flöde och tajma alla inblandade i det. Men det är en utmaning, det är också en utmaning att inte ha det för det bli bökigare att köra rent Kanban med "continues delivery" så att jag tror nog att i majoriteten av alla projekten försöker vi följa scrum med sprintar.

### **2.2.3 Hur använder ni Scrum?**

### **2.2.4 Komplimenterar och/eller kombinerar ni Scrum med andra metoder?**

Ja i de fallen som då man inte har några man inte vill ha tydliga sprintavdelningar då kanske man arbetar mera kanban orienterat med backlogen och se till så att man arbetar med lagom mycket aktiviteter samtidigt.

## **3. SWOT frågor om Scrum**

### **3.1 Vilka styrkor ser du med att använda Scrum?**

Det är ju att du får en tydlighet du tidigt visar vad det är du gör för någonting för att undvika missförstånd ofta är projekt. Förr var ju kanske kunder och mottagare inte inblandade förrän ganska sent, och upptäckte du då någonting "nej det var ju inte så jag menade" då var det försent att ändra så det tycker jag är den största styrkan att du får väldigt fokus på att leverera rätt saker. Det andra är att du får ett naturligt avbrott där du kan ändra dig då delvis om du upptäcker "nej det var inte så jag menade" men också för att förtydliga och förbättras kraven efterhand. Så att försöka skriva kraven från början är ett av de största problemen med vattenfallet. Det går inte för att du vet för lite. Du lär dig så mycket genom att arbeta med en sak så att kraven bör utvecklas efterhand och det är kanske den egentliga att "embrace change" är i grunden för Scrum men och agilt men jag tror nästan att den här tydligheten att man visar vad man gör är ännu viktigare i praktiken. De två sakerna ser jag som styrkorna.

### **3.2 Vilka svagheter ser du med att använda Scrum?**

Det är ju om man tror att det är ett kokboksrecept som bara följer jag de gör jag de här dojorna så blir det bra men så är det ju inte. Det blir det inte du måste ha ett skickligt team ändå. Du kör daily standups och retrospectives och gör du inte det på ett bra sätt och skapar rätt förutsättningar för att folk ska interagera på rätt sätt, så hjälper det inte att du har dom bitarna så att du måste förstå vad det verkligen är och ha ett skickligt team. Det finns en övertro på att bara om vi kör Scrum så ordnar sig allting men så blir det stora besvikelser och det kan ofta finnas en stor skillnad mellan ingenjörerna som jobbar med Scrum i implementationsteam och mottagarna som kanske har uppfattningen att jag har förstått vad

Scrum är men sedan så har man ganska olika bilder om det. Så att det är nog svagheten i sig att det finns missförstånd kring vad Scrum är. Man tror att det är en ”Silver bullet”. Det är viktigt att man utbildas och kommer i samma nivå kring vad det är för någonting, och komma överens om vad definitionen och sedan inte tro att det i sig inte hjälper så jättemycket. Scrum och agilt är en samling best practices det gör ju inte ett inkompetent team kompetent helt plötsligt utan gör bara det att de kan arbeta effektivare, och verkligen förstå att de jobbar rätt.

### **3.3 Vilka möjligheter ser du med att använda Scrum?**

Det engagerar ju mottagaren kunden i ett tidigare skede så det tycker jag är den stora möjligheten att man får naturligare punkter som man sätter upp. Det kan vara väldigt svårt att boka in tidigare, men när vi säger att nu ska vi göra scrum, och det betyder att ni måste ha en produktägare, och ni måste åtminstone vara med på demo, och retrospectives och blir då ger det ju fler naturliga punkter med kunden. Det nära samarbetet är sammanfattningsvis, och det förbättrade samarbetet är möjligheterna. Det engagerar eftersom de känner den pressen att nu måste jag säga någonting och utvärderar någonting så kan det vara så att de engagerar sig under sprinten också med Scrum mastern det är ju största styrkan tycker jag men det blir sen på samma sätt internt inom teamet att det blir naturligt att man har till exempel retrospectives som kan vara lite jobbigt att komma igång med men när det väl är igång så blir några grejer som kommer upp till ytan och sedan det blir ett bättre team till nästa sprint. Kunderna blir mer nöjda när de arbetar med Scrum.

### **3.4 Vilka hot ser du med att använda Scrum?**

Det är det jag sa att blir det missförstånd kring vad någonting är för något så kan det bli stora besvikelser. Att det hela tiden handlar om att hantera förväntningarna. Kontakten med kunden är aldrig fel men det kan ibland vara en utmaning med det. Det är inte självklart, och löser sig inte av sig själv bara för att man kör Scrum.

## **4. Följdfrågor**

### **4.1 Vad tycker du om Sprintar**

När det gäller den regeln då så tycker jag den finns ju av en anledning. Det är best practice för att den drar med så mycket som man inte tänker på ”ja men vi skjuter en dag” eller ”vi gör si eller så”, men det får inverkan på det andra då om man är i ett team då kan man inte fatta beslut på att en person behöver en dag till för då blir det problem. För de andra så man bör hålla på sprint perioderna, men märker man då, som jag sa i början, att det här projektet fungerar inte riktigt, så då behöver man kanske överge sprintar då. Men sen får man ju inte vara så stelbent så att man inte i ett specifikt fall kan säga ”ja ok” här tar vi ett medvetet beslut och förlänger sprinten en dag.

### **4.2 Vad tycker du om rollerna i Scrum (ScrumMaster, Product Owner, Team)**

Jag tycker att de är bra! De ger en tydlighet! Problemet och utmaningen kan vara att det kan vara ett problem att hitta en produktägare på mottagarsidan ofta har de ett antal stakeholders, och det kan vara svårt att hitta en som är beredd att ta på sig rollen att samla ihop alla de

kraven och då kanske man som ScrumMaster får man ibland ta ett steg upp, och ta rollen som produktägare på kundsidan, vilket inte är optimalt och man får inte den interaktionen mellan produktägaren och ett team. Kunden måste ha tiden att förbereda inför demon, och förbereda inför sprint planeringen, så har man en två veckors sprint så måste de kunna göra det annars har man en produktägare. Det är en möjlighet för en produktägare att vara med dagligen, men det behöver inte absolut något de behöver vara . Det kan finnas en risk med en produktägare som engagerar sig för mycket som kommer med ändringar till teamet löpande utan då får de göra det via ScrumMastern så att teamet inte blir distraherat. Det är en stor huvudsak med sprinten att teamet ska få arbeta ostört då. Nu har vi bestämt att vi gör det här och nu gör vi det. För att du får ner effektiviteten så mycket, så fort. Det blir distraktioner kring ”kanske skulle vi göra så här istället?”, utan då får man ta det som en change i nästa Sprintplanering (tyvärr). Det blir lite disciplin det handlar ju om ScrumMastern till skillnad från en vanlig traditionell projektledare. Hur kan man hantera konflikter genom det delade ansvaret i en självorganiserande grupp?

Det handlar mycket om att komma ifrån Micro management, där man står och pekar på folk och säger vad de ska göra, då tar man bort deras egna engagemang. Det är väldigt viktig princip, men man får inte övertolka det, utan finns en konflikt så kan ScrumMastern gå in och lösa konflikten. Men det ska vara efter att teamet själv försökt lösa det och kanske snabbt samlar dem, och har man en daily scrum så kan man snabbt ta upp det. Det betyder ett annat sätt, man jobbar nära varandra och man slipper de missförstånden som sker när man sitter på olika ställen och kastar paj. Det handlar om att man också håller väldigt operationellt på teamet. Går något fel så är det hela teamets fel, alltid. Du slipper mycket av konflikterna för det är inte lönt att försöka peka på någon annan utan det är bara att lösa det.

### **Personlighetstyper och Scrum**

Det är alltid en fördel om du får ihop ett bra team. Det är nyckeln till framgång att sätta ihop ett bra team men du kan även forma det och har du någon sorts nivå på medlemmarna och där kommer retrospektiven in och du lyfter att jag tycker inte om att du jobbar så det är inget personligt men det fungerar sämre och oftast tar folk till sig och hittar en lösning så det är där man måste hitta en öppenhet i retrospectives där det är okej att lyfta vad som helst, både positiva och negativa grejer. Det handlar lite om att klappa varandra i ryggen också ”det där gick jävligt bra” så att ”det känns lite kul”. Jag tar oftast projektledarroll med hela den ekonomiska uppföljningen, och då tar jag ett ansvar för som gäller inom befogenhet och man får ju eskalera om där är något som sker, men det ser jag inte som ett problem. Det kan ju vara svårare om man är en av utvecklarna, och inte är så senior man har ett intresse av Scrum masteriet och att någon senior utvecklare inte respekterar vem som är ScrumMaster då möjligen. Det kan vara tufft att ha den auktoriteten när det verkligen är en konflikt som teamet inte själv löser, men jag har inte själv råkat ut för det, eftersom vi och teamet har hanterat det som sker.

### 4.3 Hur arbetar ni med att utveckla kravspecifikation?

### 4.4 Hur ser du på Scrum möten ?

Vi har använder alla (Scrum) mötena och jag tycker att dessa best practicen är bra.

### 4.5 Hur fungerar dokumentationen i Scrum?

Jag vet inte i vilken mån hur Scrum uttalar sig om dokument, men Lean handlar ju om att ha så lite dokumentation som möjligt men ändå tillräckligt. Så det är ju ingen ursäkt för att inte dokumentera men du ska göra det på rätt sätt. Man jobbar ofta med Wiki som vi har som standardverktyg så att man lägger upp saker där och sedan kan man pdf:a det så att man har det som dokument om man behöver det, och att man har det som en samlad vy av informationen . Vad du sedan tycker är dokumentation, men du ska dokumentera din kod, dina förfaranden och release notes. Sedan är det ju en fråga om att ska det bli en produkt så behöver du någon typ av produktinformation. Det är ju en förhandlingssak om det är produktägaren som hanterar det eller om det är teamet som skriver det men det kan ju bli lite bökigt om all informationen ligger i storys någonstans, och om det sedan kommer in någon från ett företag och säger att nu skulle jag vilja ha en bild över vad ni har gjort för någonting; ”ja titta i backloggen” eller i ”sprint loggarna”. Det kan ju vara lite svårt utan man behöver någon säljande sammanfattning om vad det är för någonting, och sedan kan man du ju behöva manualer, maintenance. Ofta får slutanvändaren ett så intuitivt gränssnitt så att de inte behöver dokumentation, men vissa andra saker andra saker kan det behövas punktlister för som hur man installerar systemet, och hur konfigurerar man det osv. Den dokumentationen måste ju fram precis som alltid annars.

### 4.6 Hur ser ni på, och hanterar risken att förlora gruppmedlemmar under projektets gång?

Det får man hanterat med en överlämning, men det är värre om de går och blir överkörda eller någonting. Ofta är dokumentationen out-of-date några dagar efter att den skrivits så jag tror inte att det hjälper så mycket då om man ska lägga halva sin tid på att dokumentera, om det ska vara så att man bara ska kunna försvinna från en dag till en annan, och gå in och ta över så att det är rätt utopiskt i alla fall. Jag tror problemet är istället att man förr lade för mycket tid på snygg dokumentation, för vissa projekt gjorde det förr då man kanske med Lean då inte lade lika stort krav och lika stort värde på tjocka tjusiga dokument. Men det är så med stakeholders att det står i dokument, och de sätter värde på dokument, men de har ingen aning om kod utan det ger istället en falsk trygghet. Det gäller att har rätt typ dokumentation men att säga att Scrum inte använder sig av dokumentation det är missbrukande av Scrum, utan Scrum genom det agila manifestet säger att vi värderar mera att vi kommunicerar men hade vi kört denna intervjun via mail så hade den blivit ganska omständlig så visst är det mycket effektivare med en kommunikation face to face då vi ser hur vi reagerar och så vidare .

### 4.7 Hur ser du på fördelning av resurser i Scrum?

Nej det tycker jag är bra, vi har en sprint log och det är bara att ta en grej. Tycker man att den borde jag ta då får man motivera det direkt, eller på en daily scrum eller inom teamen men

principen är den att var och en tar det som den tycker är bra. Det är ju estimerat du ska ju uppfylla estimaten går du på något som du inte behärskar så får du ju motivera det för då kanske du inte når dina estimat, men det är en styrka också att man försöker få tvärteam där alla kan göra allt så att det inte bara är du som kan leverera systemet och är du då sjuk vid leveransdagen ”ja då står vi där” och det bara är du som kan gå in och ändra något i databasen. Det är bra att få en bredd och en viss kostnad är jag beredd att ta på att man lär flera att kunna flera saker i teamet för då får man ett kraftfullare team framöver om du ska jobba framöver med sprintar och releaser över en längre tid.

#### **4.8 Hur upplever ni kundens centrala roll i projekten?**

Det är kanske inte något som Scrum kanske inte direkt berör hur man allokerar resurser. De ser ju poängen i estimeringen, men ibland kan det vara att kunden vill ha mera kontroll på att man rapporterar på allokering, beläggning och debitering, hur mycket som debiteras. Det går lite utanför Scrum men det är något som jag som projektledare får hantera.

#### **Är det ofta problem med agilt?**

Jag tycker att det är en mognadsgrad som ökar ofta från att man hör något nytt buzzword så tar det en fem år innan det verkligen börjar implementeras. Sedan tar det ett antal år till innan alla är uppe på den nivån. Vi kom igenom objekt orienterat, och vi är rätts så bra genom agilt tycker jag att det finns rätt så bra mognad kring det nu är det här med Devops som är hett att man samarbetar med operations så att man är ett team som både utvecklar, och driftar saker och ting. Det är bättre, och bättre att arbeta agilt. Alla har sin bild om vad scrum är för någonting och vad agilt är för någonting mer eller mindre skillnader.

#### **Framtiden och utvecklingen av scrum ?**

Scrum är väl scrum men hela det agila konceptet utvidgas med devops det är ju är en viktig grej för att få effektivitet hela vägen ut det för annars är det att man utvecklar någonting och det fungerar inte i driften då har man ju tappat en vinst så ser man det som en del av den agila utvecklingen så visst det kommer att komma mer. Devops har verkligen börjat användas men många har olika bild om vad det är för någonting.

#### **Det kommer lite trender.**

Så är det ju men det är ju grejer som folk känner att det skulle vara bra om vi gjorde så här och sedan är det någon som är en vid någon konferens någon auktoritet som sätter sig ner och sätter namn på saker och lägger ut det och säger att det här är någonting nytt. Många känner ju igen det ”så jobbar jag ju i princip” men det är ju bra att det vi satt namn på det så kan vi nämna det och prata om det ha en kultur för det och så blir det mera vedertaget också tidigare var det ju väldigt flummigt att jobba lean ”ja vadå ska vi inte dokumentera – era fuskare alltså” men idag vet man att det är en best practice, och det är vedertaget och det diskuteras på konferenser så därför är det okej att jobba lean.

#### **4.9 Har ni erfarenhet av att teamet är placerad i olika städer (distributed teams)**

Det har jag jobbat med också, och det är ju jobbigare det är lite som om vi gjort en telefonintervju här det blir jobbigare och lite lättare med missförstånd uppstår och ett



distribuerat team kan ju innebära att någon sitter i Indien eller Kina och då har de andra kommunikationssätt, kulturella skillnader och tidszoner och sådär vidare så där är utmaningar men med dagens verktyg med Skype inte minst så fungerar det. Vi kör med Gira i Scrum utan inga lappar på väggar. Vi har ett elektroniskt system för att skapa Stories, tasks och burn down chart och det. Det ligger online, och det gör ju också, att du kan jobba hemma om du ska vabba eller vobba så fungerar det bra att arbeta hemifrån. Men det är alltid kraftfullare att inte bara sitta inte bara i samma hus och samma rum utan verkligen nära varandra. Så om jag säger någonting så hör du och ser det direkt och du hör allting som ses i rummet även ifall det inte rör dig. Den osmotiska kommunikationen, ju närmare desto bättre och gärna att de som arbetar med samma sak sitter precis bredvid varandra så om du har två android utvecklare så ska de bara behöver vrida på huvudet så ser de varandras skärm och vad den andra gör då blir det oftare att de kommunicerar än om de satt rygg mot rygg eller mittemot varandra. Teamet ska vara tätt samlat det ger effektivitetsvinster.

### **Påverkar ljudnivån produktiviteten?**

Ljudmiljön, antingen lär du dig att jobba i ett sådant landskap annars passar du inte in i ett sådant team. Men du kan ju ha hörlurar på dig om du behöver absolut fokusera på någonting, och det har ju många men de ser fortfarande, och är det någonting så tar de av sig hörlurarna så att jag inte behöver gå fram till dom varje gång jag snabbt ska växla en mening.

## **Bilaga A3. Eniro Intervjutrascript**

### **Vad heter du?**

Jannika Björkholm

### **Vad är din yrkestitel?**

Teamchef för Core på Local Search. Det är mitt team som ansvarar för det som visas utåt på vår site förutom kartorna, så det är alltså företagssök och personsök samt index och sökmotorn dvs. söklogiken som ligger bakom vilka svar du får när du söker. Tex. att om du söker Clas Ohlson så borde du hamna på personer men just Clas Ohlson överridas för det är förmodligen företaget du vill söka efter men annars, om du söker på Jannika Björkholm, vill du hamna på personer också listar vi förhoppningsvis ut vart du vill.

### **Hur länge har du jobbat som (yrkestitel/projektledare)**

Teamchef sedan 2009

### **Hur länge har du jobbat med Scrum?**

Sedan 2008 då jag var produktägare

### **Använder ni Scrum idag?**

Vi försöker jobba agilt hela tiden men det är inte alltid vi lyckas. Vi jobbar med scrum med det största teamet, core teamet som jag ansvarar för. När det gäller index-teamet så är det inte lika uppstyrt som i scrum, snarare lite kanban inspirerat får jag väl lov att säga. Det är inte så att teamet genomsyras av kanban heller, utan snarare kanban-influerat i flödet som de har på sin tavla ,eller rättare sagt på ett av våra fönster.

### **Varför använder ni Scrum?**

Det är extremt mycket mera snabbriktigt och transparent så alla kan hänga med i vad som görs. Vi kör 3 veckors sprintar på Eniro så vi lanserar var tredje vecka vilket gör att vi

prioriterar om backloggen var tredje vecka. På det sättet kan vi prioritera in nya saker ofta. Vi kan alltså bestämma oss att vi gör något visst efter den här sprinten som vi jobbar i nu, vilket betyder att vi kan ha det ute i produktion 3-6 veckor senare. Alla är mer delaktiga också. Innan vi arbetade med scrum använde vi vattenfall och då var lanseringarna ofta ett halvår när vi ville ändra på något, även om det var något mindre. Så jag tycker att en väldigt stor fördel är att det är mycket mera snabbt och föränderligt nu. När man kör Scrum är det teamet som tar på sig att de ska leverera en viss feature och för att kunna göra den så behövs det lite av designen, lite av html-utvecklaren och lite av java-utvecklaren och man måste jobba på sakerna tillsammans. Efter tre veckors utveckling och testande så släpper man sen tillsammans ut någonting till produktion. I den ultimata världen, vi är inte riktigt där ännu, så har man en prioriterad backlog på tavlan med tasks som ska göras och där kan då vem som helst ta vilken lapp som helst. Vi är fortfarande, till stor del i alla fall, kvar i att "jag är javautvecklare och måste ta den här lappen som det står java på även om den är nummer fyra på listan", men i det ultimata teamet kan naturligtvis att vem som helst ta vilken lapp som helst. På Eniro har vi en produktägare som prioriterar backloggen.

### **När använder ni Scrum?**

Många team använder Scrum i större eller mindre skala. Just nu är vi 6 olika team på online fronten, sedan har vi team på backend också, där flera jobbar med Scrum. Vi försöker få in de flesta projekten också så att vissa projekt delar upp sig, och just kör det projektet, som utvecklar en speciell sak, som kanske är ett halvårs projekt då dom sitter ibland tillsammans, men att de kör scrum inom det projektet. Vi använder typ Scrum jämt mer eller mindre.

### **Hur använder ni Scrum?**

#### **Komplimenterar och/eller kombinerar ni Scrum med andra metoder?**

Kompletterar med kanban, är det något team som är tre personer, då fungerar det inte lika bra kanske att köra Scrum på samma sätt, utan du behöver vara ännu snabbare än de andra tre veckors sprintarna som vi annars kör. Då har det funkat bättre att bara ha ett jämt inflöde helt enkelt och få till utvecklingen på det viset. Då planerar de inte så mycket innan. Tex.index teamet som är tre personer och jobbar väldigt tight försöker mer ha ett jämt inflöde istället för att commita sig till en sprint. Det gör att de är mer lättroliga så att om nån kommer på att vi tex. måste tweeka på kategorierna på något sätt eftersom de inte fungerar optimalt så kan dom väldigt snart hugga tag i det. Hur vill vi göra om det? Brainstorma kring det! Också kör dom. Dom är inte så planerande alls, men sen om det är så jättekban eller inte det är en annan femma men de har inte planerade sprintar och släpper saker var tredje vecka. De hoppar över sprintarna och har istället sina saker på en tavla (eller rättare sagt ett fönster ut mot gatan där de sitter).

## **SWOT frågor om Scrum**

### **Vilka styrkor ser du med att använda Scrum?**

Snabbrörligheten, man får ihop ett team lätt eller lättare, som gör att man gör någonting tillsammans man har ett gemensamt mål, var tredje vecka får man fira segrarna i sprinten och ha en lansering

### **Vilka svagheter ser du med att använda Scrum?**

Det kan vara för trögrörligt att bestämma sig i förväg för tre veckors jobb. Några kan tycka att det är för lång tid och då har man kanske valt att köra kanban-influerat istället då men annars tycker jag inte att det är så mycket svagheter. En svaghet som kan vara, det är ju om man ska jobba flera team tillsammans på ett och samma projekt till exempel. Då är det ju att samordna allting, men å andra sidan så får man ju inte det i vilken metod man än väljer. Vi har försökt göra det ibland med hjälp av tex. scrum av scrum där det kan vara ett möte varje vecka som kanske alla scrum masters går på för att informera varandra. Vad är på gång i ditt team? Vad är på gång i mitt team? Så gäller det förstås att få ihop kommunikationen däremellan. Det blir lite som distribuerade team, även fast teamen sitter på samma ställe så träffas en från varje team och pratar ihop sig om vad de gör i de olika teamen, annars så kör vi Dailys. I de fall då teamen har medarbetare i distribuerade team, så att man har någon som jobbar i Polen och resten sitter i Sverige och någon i Danmark men resten sitter i Sverige, då har man dailys med Hang out, Skype eller video och får till dailyn på det viset.

### **Hur är det att ha en teammedlem i Polen?**

Det är inte lika bra så som om det suttit här självklart! Just i kartteamet där de har en från Polen, de har jobbat tillsammans ganska länge, så de känner ändå varandra ganska bra plus att dom har ganska isolerade uppgifter, det som de gör ändå liksom, även om det ingår i en sprint och det ingår olika delar i det men det är mycket svårare naturligtvis men vi har en ganska bra skypekultur. Vi har ganska mycket teamchatten där dom tex. skriver "jag har det här problemet hur ska vi lösa det" men det är klart att det är ett hinder att inte sitta bredvid varandra. Det är väldigt mycket lättare när man bara kan prata med grannen. Men vi klarar det ganska bra med Scrum och det beror nog på att man har de här daily och sprint planeringen och att man verkligen bestämmer tillsammans, även ifall man inte är med fysiskt så är man med på video då. Och man bestämmer tillsammans att man har ett gemensamt mål. Och det är inga stora tidsskillnader, det hade varit jobbigt om man kanske suttit i Indien

### **Vilka möjligheter ser du med att använda Scrum?**

Samarbetet och flexibiliteten, det blir väldigt öppet med en tydlig backlog som alla och hela företaget kan komma åt om man vill och läsa backlogen och se vad är det som vi tänker

släppa om tre veckor. Det blir ganska tydligt så. Och i och med att man har som team mål att teamet gör saker, så då tycker jag att det går mycket bättre än att man gör som individer, jag refererar till tidigare då vi körde mera vattenfall. Först satt där kravmänniskor och då gjorde dom sitt, och sedan varsågod och så lämnade dom det till någon designer som gjorde sin magi och sedan lämnade de över det också var det vattentäta skott emellan. Och man får naturligtvis ett mycket bättre flöde, samarbete med teamen, öppenhet och tydlighet när man istället kör Scrum. Det är mycket tydligare om var man ska vända sig någonstans om man har kommentarer eller frågor

### **Vilka hot ser du med att använda Scrum?**

Det är lite det här med att man inte vet vad det andra Scrum teamet gör. Dom som jobbar med kartorna kanske ändrar någonting i våran header, och det vet inte det andra teamet om, också påverkar det oss också så det är ju kommunikationen mellan teamen ska jag säga.

### **Konflikthantering**

Rent teoretiskt om det skulle komma en sådan situation så är det upp till teamet att lösa det men naturligtvis så är det också mitt ansvar som teamchef att gå in och då får man väl trampa in och säga att vi måste lösa det. Men som jag ser det så är det inte upp till Scrum mastern att skulle lösa det utan det är hela teamet, och då får teamchefen steppa in på något sätt. Tror jag rent teoretiskt .

### **Vad tycker du om Sprintar (Intern)**

-Regeln att ej ändra målet under en Sprint

Det är ganska viktigt. Det som kan vara vårt problem, om man tänker det som visas utåt på siden så är det ganska viktigt om något hittas till exempel, men vi har problem med en banner som inte visas riktigt på rätt ställe till exempel. Vi har kanske haft en sprint planering och bestämt att om tre veckor så ska vi släppa det här och det här och alla comittar sig och säger ”japp” vi tror det här hinner vi på dom här tre veckorna och sedan så kommer någon från banner teamet och ropar ”oj oj” bannern ligger ju alldeles för långt till höger. Hur ska vi göra nu? Och då måste vi ägna några timmar en dag åt att fixa det. Kanske det går att fixa då, men sen så dröjer det till två dagar senare så kommer någon och ropar att det är ett telefonnummer som ligger fel. Vi har löst det hos oss att istället för att säga att nu måste vi bryta sprinten och planera om så försöker vi att inte göra det så, vi har fegat lite där, vi planerar upp oss till varje person kanske ungefär sex timmar om dan, istället för 8:a timmar om dagen och då så tar vi inte in mera case utan vad vi tänker att vi hinner med på X timmar, eller hur långt vi har en sprint och sedan så vet vi, att vi kommer att bli störda. Varje person kommer i snitt att bli störd två timmar om dagen, antingen med det eller så kan det vara teammöten också naturligtvis men vi räknar bort tid för att vi vet att det här kommer att

hända, att människor kommer att bli störda hela tiden för vi vet om, att om det är något akut på siten, då måste det här rättas till inom de här tre veckorna och då har det varit en sånär pragmatiskt avstickare från scrum sprint egentligen då liksom. Sedan så har vi också provat någon gång att ta in ett case som är ett timeboxat case , tre dagars produktionsstörning, också har vi tagit in det, så bara för att hitta någon lagom pragmatisk sak, för naturligtvis när man sitter i teamet då man faktiskt påverkar och löser saker som syns på front så är det svårt att säga ”Nej det får vi ta i nästa sprint” om det är någonting viktigt, så gör det att sex veckor bort i tiden är för långt liksom så det är så vi försökt lösa det.

### **Vad tycker du om rollerna i Scrum (ScrumMaster, Product Owner, Team) (Intern)**

#### **-Vad tycker om samarbete mellan rollerna inom Scrum?**

Jag tycker att rollerna fungerar bra absolut, jag har en scrum master i teamet så den rollen behöver jag inte ta, jag har en produktägare också men sedan så som teamchef så handlar det ofta om att hitta i alla fall någon slags roll mellan produktägare och scrum master, dels hjälpa produktägaren att få till en backlog och få det prioriterat. Produktägaren har väldigt mycket andra ansvarsområden också, vilket gör att produktägaren kan bli svår att få tag i . Det handlar om att vara någon slags acting produktägare, för man behöver sitta emellan och säga att ja då är vi på erat spår A och är produktägaren inte där och svarar, så gör jag det på något sätt. Det behövs liksom någon mera där, och samtidigt så scrum master rollen, när det handlar om att sopa banan för teamet då känner jag att den tar jag också en del av, eftersom det tar en hel del tid av oss för det, så då tycker jag att en team chef också behöver kunna göra det, rycka in och göra dom sakerna, så det är väl ganska splittrat så men jag tror att svårigheten är ju att den största. Största svårigheten för produktägaren på Eniro i alla fall, där allting som rör sig på förstasidan är så och på våra undersidor på Eniro så är det ju så mycket, vilket gör att han kanske har en sex sju åtta stakeholders att hålla reda på , där någon vill göra någonting litet ett litet säljverktyg och den andra tycker att det är jätteviktigt att få in banners format, något nytt format och en tredje tycker att dom här Facebook grejerna dom är jätteviktiga och någon fjärde tycker ratings då kan jag sitta emellan och säga “japp nu kör vi dom här sakerna” nu är det de som är viktigast. Det tror jag är mycket av prioriteringsproblem så där känner jag att jag måste hjälpa till så mycket det går liksom.

#### **Så prioriteringen kan också vara ett litet problem,**

ja ett litet hot, visst det är inte så att det hade funnits en lösning på det om man suttit med vattenfall och ju mindre och färre team man har desto större bli problemet . Vi har haft lite om man går tillbaka några år på Eniro så hade vi lite flera team, och då var det naturligtvis lättare eftersom man kunde gå genom mera utveckling men ju färre utvecklare desto svårare blir det. Det blir hårdare att prioritera liksom, så det är ju den sitsen vi har just nu. Det är ju så men annars så tycker jag att rollerna är bra.

## Hur arbetar ni med att utveckla kravspecifikation? (Extern)

Vi använder ett system som heter Gira som är ett ärendehanteringssystem egentligen och där ligger alla userstories och till det så ligger det kanske bifogat någon kravspec eller någon bild som berättar någonting när någon ändå gjort någon design. Och beroende på vad det är för något, det blir så väldigt varierat hos oss. Ena gången är det så, att du ska flytta den här knappen därifrån till dit ”ahh då behöver vi inte så mycket specifikation för det”. Nästa gång som det vi sitter och ska börja jobba med nu och nästa vecka så handlar det om att vi har en site som heter rejta.se för rating och Reviews då, den ska vi nu lyfta in i vår core tjänst istället, så att när man går in och tittar på ett företag så kan man redan idag då trycka på att man vill gå in och rejta företaget, men då kommer man till rejta.se som är vår site då och det är där omdömet visas, men nu så kommer man inte göra det utan man kommer till Eniro och det är ju ganska stort istället och då naturligtvis så kräver det mycket mera krav, och där så brukar det vara så beroende på vilka människor som kommer att vara inblandade, så är det kanske designen i teamet som kommer med på ett tidigt stadium och kommer att vara med och rita då, men vi har ingen inte i något team faktiskt så har vi inte någon speciell kravspecialist utan det är väl en liten brist på Eniro att man inte har någon bra sådan men därför ligger mycket av det jobbet på produktägarsidan att dom får ta fram hyfsade krav som vi sedan jobbar tillsammans med på i teamet också. Vi brukar ha ett möte som vi kallar backlog refinement eller backlog grooming kallas det också, kanske i mitten på sprinten eller i slutet på sprinten där vi tittar vad ligger närmast i backlogen och då kan vi se att om det ligger något sån här större grej som vi känner att det här måste vi veta mera om så kan vi prata med produktägaren om det och säga att det här case måste du utveckla. Vad är det du vill att vi ska göra då? Och då blir så ibland samarbete med någon i teamet, ok jag måste låna en javautvecklare till exempel och sitta tillsammans och komma överens om att det här, det här som är scopet för just den här uppgiften då. Eller på produktsidan själva då att dom hittar på någon spec för det så det är lite olika från case till case skulle jag säga men vi har user stories, de ligger i Gira ibland estimerar vi dom. Oftast så estimerar vi inte dom inte så långt innan på backlog för det här med att sitta och estimerar massor med case på backlogen det gör att sen, så när man väl, när man lägger ner för mycket tid på det också kommer vi kanske aldrig utveckla det så är det bara slöseri med tid utan vi försöker att i själva sprint planeringen säga att, det här tror vi tar fem dagar, det här tar två dagar och någonstans där lista ut att ungefär såhär många case är det rimligt att vi hinner med. Sen förra sprinten så testade vi köra lite storypoints för att se om det skulle tyckas att det kändes bättre, så vi får se vad teamet kommer att tycka om det i retrospectivet, om de tycker att det känns bättre.

## Själv idén till projektet...

Själv idén till projektet kommer oftast inte från utvecklarsidan utan det kommer från produktsidan, det är den vanligaste. Sedan så har vi faktiskt något på Eniro som vi kallar för Eniro labbs, det är två dagar per kvartal där utvecklarna bestämmer helt själva vad de vill göra i två dagar. Ibland är det om någon vill läsa på något om något nytt bibliotek på i java script och ibland så blir det, att men jag har gjort en ny suggest funktion man börjar skriva

någonting och så kommer det förslag i en liten meny som kommer in när de börjar skriva. Någon har gjort om den, det vore mycket bättre om den såg ut såhär istället ” Ahh det var en jättebra ide då kanske vi implementerar den ” och nästa var att man skulle flytta senast att man skulle flytta koordinatören på kartan man tycker att den här sprinkler bilden som står med mitt företags namn på där står det en lastbil i vägen, jag skulle vilja att man istället för att man tittar rakt fram från mitt företag så kan man stå lite åt höger för då så får man en snygg bild på min entré att man då skulle jag kunna flytta den punkten här internt eller till och med externt, så att många roliga idéer som då kommer från utvecklarna själva för annars är det typiska att idéerna kommer från produkt sidan också ska utvecklarna utveckla och leverera. Vi försöker få bort den här beställare-leverans organisationen, utan få utvecklarna att vara lite mera delaktiga åt alla håll och kanter i den Agila världen så vill man faktiskt att man ska jobba tillsammans. Gain all the man brainpower heter det på något så här fint som någon har sagt. Det är lite intressant att man får flera delar i det.

### **Hur ser du på Scrum möten ? (Intern)**

**-Planeringsprocessen (planning meeting)**

**-Daily Scrum**

**-Sprint Review**

**-Sprint retrospective**

Planeringsprocessen, det vi försöker ha emellan är backlog grooming eller backlog refinement då, och det är egentligen anledningen till det är att få just de här planeringsmötena bättre så att det inte är så att vi sitter på de här planeringsmötena, hela teamet sitter tillsammans mot produktägaren och så säger vi ”aa vi ska göra” story A men då är det ingen som förstår riktigt vad vi ska göra utan för att vi ska göra det så måste vi först få koder från teamet till exempel för att kunna visa upp det här så måste vi få den här koden, annars så kan vi inte påbörja caset och då vore det så skönt att man har allting sådant på plats och börja med varje case när sprinten börjar, så försöker vi istället ha det här, att vi försöker ha det här backlog mötet en vecka innan för att kunna tala om för produktägaren att du måste skaffa svar på de här åtta frågorna som vi faktiskt saknar svar på, för att vi ska kunna starta med dom här casen som du vill att vi ska starta med och sen handlar det lite om att vi ska få en bra start på sprinten. Vi hjälper produktägaren för att han ska kunna få svar på sakerna som gör att vi kan börja på en gång. Det man inte vill är, att ta in saker i sprinten som e för otydliga helt enkelt, där inte produktägaren inte riktigt vet vad vi ska göra och vi vet inte vad vi ska göra då blir det såhär hur lång tid tar det att göra.. ”Ahh vi vet inte för vi vet inte vad vi ska göra”. Då blir det lite svårt att estimeras och komma igång på.

Produktägaren är en extremt viktig och central roll för scrum är det de liksom, att den måste vara med och ta besluten. Sedan på sprint planeringen då brukar vi försöka ta första timmen



tillsammans, där jag är med och produktägaren är med. Där man går genom och säger vad handlar varje case om, och man säger att det här är tydligt, ungefär att det är så långt ungefär, så långt ner på listan som man tror att man kan hinna på den här tre veckors sprinten plus lite till då. Sedan brukar vi lämna och så går teamet kvar, i verkligheten i detalj så att det här första caset, vad innebär det jo det innebär att öppna upp den här instansen, den här brandväggen, utveckla den här featuren då. Och då ser man att det kommer att behövas en java resurs, en front end resurs, ja det kommer att ta frontend resurs för det kommer att ta frontend resursen tre dagar och Java en dag. För att där på något sätt beta ner för att säga hur många timmar vi är uppe i kan vi ta med något mera case eller inte så det är väl där det blir tydligt att ”Japp det är det här” så då committar vi oss till att det är det här som kommer att ligga i sprinten. Och sedan brukar det naturligtvis vara så att backloggen är prioriterad en ganska lång bit ner, så att även om man nu tror att vi behöver de här 15 översta casen så är dom 15-20 också prioriterade, så att är vi färdiga innan så kan vi lyfta in något case i sprinten eller från backloggen som inte låg i sprinten från början. Eller som vi pratade om innan det här sakerna som kommer till ”oj oj oj det brinner här på siten” nu måste vi ta in dom också ett bra sätt som också är viktigt naturligtvis.

Daily Scrum kör vi också. Ja det kör vi kvart i 10 till 10. Där produktägaren är med varannan gång och jag är med varje gång och teamet med som är bara det vanliga vi kör väldigt standard så, Vad gjorde jag igår? Vad ska jag göra idag? Har jag några hinder?

Och flyttar lappar så kör vi små avatrar och det är lite olika på olika team och kanske också på olika företag hur tavlan ser ut. Tidigare så hade vi, när jag satt i ett annat team så valde dom att ha en rad för varje person, också tog man saker och satte på sin egen rad men istället så har vi valt sedan 2 eller 3 år, och faktiskt något som alla på Eniro valt, att köra att backloggen uppifrån och ner i prioriterad ordning och sedan så har man en kolumn för tasks, userstories först och vilka tasks som man ska ha, och sedan så har vi in progress, blocked test, vilket gör att man går uppifrån och ner i backloggen, tar ett case och sätter det i progress och sedan så har man sin egna lilla avatar, så man kanske har Bart Simson eller någon annan så sätter man den gubben på att nu har jag, det är just jag som jobbar med det här caset och det är i progress. Det gör ju det hela lite mindre personligt, på det sättet att det här är Jannikes rad och det här borde Jannike jobba med utan vem som helst kan faktiskt ta vilket case som helst, i alla fall i utopi världen.

Sprint review- vi är lite dåliga på detta, bättre innan men som sagt 19e mars så lanserade vi UX14 som är ett jättejobb där vi har skrivit om all HTML Java script kod på hela siten. Java ligger kvar ungefär likadan i botten, men allt annat är omskrivet från blankt papper och det har ju tagit oss sedan stenhårt jobb sedan augusti till mars då. Och det gjorde att det inte fanns så mycket att titta på efter var tredje vecka och då bestämde man att det inte är något kul med sprint Reviews då det fanns liksom inte något att visa men nu hoppas jag att det är något som vi kan komma in i igen. Vi är faktiskt inne i vanliga sprintar igen med det här kommer att komma ut och då är det intressant att visa upp det.

Sprint retrospective – den kör vi stenhårt, den tycker jag är den viktigaste mötet i Scrum eller Agilt liksom. Vad gjorde vi? Vad kan vi faktiskt göra bättre? Och verkligen lyfta upp de här förbättringspunkterna? Dels att det här gjorde vi bra det här gjorde vi riktigt bra och dels att det här tar vi med oss det här måste vi få ordning på till nästa gång så att man hela tiden jobbar för att bli lite bättre på det Agila tänket där liksom.

### **Hur länge vara en sprint som mest?**

3 veckor men sedan kan det vara så att vi väljer att lansera först efter 6, eller 9 veckor i värsta fall, som det var i det här tidigare projektet UX14, men att vi i alla fall säger att de här grejerna är vi klara med om tre veckor. Annars blir det lite ohållbart om vi säger att det här, ska vi lansera om jättelänge så därför väljer vi att var tredje vecka så planerar vi om helt enkelt. Och dels för att man ska få möjligheten att känna att man ska kunna säga att det här som jag trodde var viktigast för tre veckor, sedan det tror jag inte längre, det kan vi putta ner på backloggen. Det här har istället kommit upp som en bubblare, det är viktigt att det är färdigt om tre veckor. Det är också därför vi lanserar.

### **Hur fungerar dokumentationen i Scrum?**

Där är ju regeln att dokumentera så lite som möjligt men ändå tillräckligt. Den är jättesvår där man förut i vattenfall satt och dokumenterade ihjäl sig med bilagor hit och dit, till att nu gör vi det väldigt enkelt för oss själva just nu och förhoppningsvis så kommer det även att hålla sedan. Vi har ett system som heter conference som är vår interna Wiki, där lägger vi upp mycket skärmdumpar kan vi säga så att där man har en bild också har vi ritat in på den och säger att här söker man, när man gör en sökning så händer det här och det här. Relevans nivå och annat finns på wikin att hitta då. Och hur systemet funkar och hur olika funktioner funkar helt enkelt, så det är ju där vi försöker dokumentera plus att det dokumenteras i koden också naturligtvis men framförallt så är det ju, den här visuella dokumentationen som är den centrala och fungerar som kunskapsspridare.

### **Hur ser ni på, och hanterar risken att förlora grupplemmar under projektets gång?**

Jo Wikin fungerar som stöd för hur man får upp system som vi har på wikin då. Det gör det men sedan är det ju något man naturligtvis inte vill göra att förlora en grupplemm under någon gång, men där är vi lite sådär att vi försöker med UX14 projektet bakom axeln. Vi försöker ha så korta projekt som möjligt och försöker att ha delleranser istället. För det här när man sitter med ett projekt som läser upp större delar av teamet eller kanske hela delar av teamet i ett halvår, så är det helt värdelöst när man inte får ut någonting annat på siten, då är det så mycket som försöker stoppar upp och blir liggande, liggandes när vi ”ooo vi ska levererar det här jättestora projektet, det stora big bang här om ett halvår”. Utan vi vill istället försöka dela upp det. Jag vet inte om du har sett det men Henrik nyberg har ritat en ganska

skön bild, du kan googla den jag tror att den heter "Whats agile" det är en pdf som han har lagt ut. I denna är det, det här att vi ska leverera en bil också sitter vi och så levererar vi ingenting, för att vi håller på och bygger ratten men sedan så bygger vi motorn och sedan har vi ett däck där och ett däck här, om ett halvår så levererar vi den här medan som vi ska tänka i den agila världen så vill man att man ska tänka att vi vill ha något som rullar, det är en skateboard också kan du åka på den och sedan kommer du på "Aaa men vänta jag skulle behöva att den gick att styra också", vi bygger ett styre åt dig. Och så kan du styra din skateboard och sedan så tredje men jag behöver ett tak över huvudet det är så kallt när det regnar. Och då bygger vi det. Och sedan så helt plötsligt så har vi en bil men det är hela tiden en leverans på vägen som är användbart istället för att vänta sex månader på att få sin bil så kan man istället få delar.

### **Hur ser du på fördelning av resurser i Scrum?**

Naturligtvis så den här ultimata världen där alla kan göra allt, den finns ju inte på riktigt. Många gånger i alla fall så är det ju inte så att alla kan allt även ifall man skulle vilja det, men på riktigt så är det många gånger så att man är javautvecklare eller man är frontendutvecklare sedan så finns vissa gråzoner där man kan ta av varandras men vissa grejer e sådana att det här måste vi vänta med att en frontendutvecklaren blir ledig tills vi kan göra de här grejerna och då får du javautvecklare, kanske göra något som har mindre prioritet men som ändå behöver bli gjort. Det är väl den liksom och det är väl det tråkiga att det vore naturligtvis mycket enklare om alla kunde allt för då vore det bara att fortsätta, men där är vi ju inte tyvärr.

Jag och produktägaren tillsammans kanske, där vi tar in det här är väldigt mycket som bara frontend utvecklare kan göra. Då inser man det någonstans, sent i sprintplaneringen att vi är väldigt sent med mycket jobb åt de här personerna, men inte så mycket jobb åt de här personerna. Då blir det okej att då kan vi göra något av det här, som dom ska göra men som vi kan göra. Ja men dom här casen kan vi göra. Och då behöver vi hela tiden fylla varje dag i sprinten och så tar vi med de här casen också som är Java case då till exempel, så man försöker fördela det lite så.

### **Hur upplever ni kundens centrala roll i projekten?**

Ja det gör vi det även om det är internt men det kräver ju ett tätt samarbete. Det är inget snack om det hela tiden och den som är närmast oss, som är produktägaren får liksom ta det vidare eller någon annan bakom den, bakom produktägaren som faktiskt är beställare på riktigt, så då handlar det om att, Tittar du nu, vad tycker du kom bort till oss, vi är så många ändå så att vi sitter ju ändå ganska tätt tillsammans. Och när det gäller annonser då finns ju väldigt specifikt vad de får och inte så där behöver vi inte göra så mycket. Det blir inga sådana här specialare för företag, det de kan göra är isåfall inne i sin annons lägga in något men det är mera en flash

så det är inget som vi behöver anpassa oss mot, så det är mycket standard som finns, annars skulle det inte gå.

### **Har ni erfarenhet av att teamet är placerad i olika städer?**

#### **-Hur ser du på Scrum och samarbete i utspridda grupper, ex scrum möten?**

Vi har ett team som sitter i Polen och sedan har vi ett team som jobbar med vår mobil utveckling som sitter i Norge och sedan så har vi resten av teamen i Sverige. Där vi har till exempel kart teamet, så har vi en resurs som sitter i Danmark som tillhör teamet och en resurs i Polen som tillhör kartteamet då. Annars sitter vi här i Stockholm då.

Teamkänslan fungerar bra men är naturligtvis bättre i vissa team där de sitter tillsammans naturligtvis, men så försöker de naturligtvis att få hit de alla någon gång per år för visst har det en fördel om de någon gång får sitta tillsammans men det fungerar ändå ganska bra eftersom dom kör med sprint planering. Det hade varit sämre att göra det med vattenfall till exempel där man inte har samma sprint comitment som Scrum där teamet ändå gör och committar sig tillsammans. Det tror jag underlättar det blir också en viss kunskapspridning.

## **Bilaga A4. Stretch Intervjutrascript**

### **Vad heter du?**

Daniel Terborn

### **Yrkestitel?**

Jag är ju VD för Stretch öresund AB, men jag har också arbetat som arkitekt under min karriär, utvecklararkitekt.

### **Hur länge har du jobbat som arkitekt och VD?**

Sedan 2000 på heltid, så är det 14 år.

### **Hur länge har du jobbat med Scrum?**

Jag tror att, ska säga så här; alltid anpassad Scrum, enligt Scrum om man anpassar Scrum så blir det inte Scrum längre. Men om man jobbar som konsult och levererar, eller som leverantör och beställare så måste man göra anpassningar runt om, lite. Men jag tror att, 2006 började jag med en agil metodik som inte riktigt var Scrum, och sedan 2007 då, då nånstans, Scrum mera formellt men med anpassningar ska tilläggas.

### **En kombination?**

Ja, precis så.

### **Använder Stretch Scrum idag?**

Ja.

## Varför använder ni Scrum?

För att leverera våra mjukvaruprojekt, som vi antingen leverar till kunderna eller som vi leverar "in-house", det är alltid till kund såklart men. Så men det gör vi för att produktionsstyra, kan man väl säga, våran utveckling.

## När använder ni Scrum?

Vi levererar mjukvara till kunderna, så vi använder det i flera av våra leveranser till olika kunder. Det typiska är väl när det är nån slags affärsnära web-lösning, då använder vi Scrum för det. Det är vårt sätt att styra internt och att också ha interaktion med kunden. Det kan vara SharePoint men det kan vara lika mycket modern web, alltså "responsive" webblösningar, senaste klient Angular tekniker, så det är fram för allt inom de området, än så länge. Andra områden vi levererar inom, SAP etc., vi är inte vana vid det än, men det är en sån sak som kommer, mer och mer. agila, Scrum-liknande metoder. Så vi gör det även hos exempelvis ute hos våra kunder, så fram för allt det som vi levererar härifrån till våra kunder, men vi är också ute hos våra kunder. Så vi är ju ScrumMasters ute på EON exempelvis, styr olika processer. Men det är ju samma typ av upplägg, men där kanske det används inte för ett specifikt mjukvaruprojekt utan för lite större sammanhang, där flera olika produkter är inblandade.

## Hur använder ni Scrum?

Scrum tycker vi är jättebra, men det är ju inte nån helhetslösning, jag skriver på tavlan också men i grunden så Scrum är ju, stöttar oss i de dagliga arbetet; att planera och styra vad vi ska utveckla per iteration. Men det hjälper oss inte med många av dom stora frågorna i form av att hållbara, alla intressenter och stakeholders hos våra kunder, det kan vara många som har ett intresse, många som har makt och vill påverka det här finns ute i organisationen och det här hjälpa ju inte Scrum oss att hantera dom. Alltså att Scrum hjälper oss inte fullt ut att förstå vad kunden vill ha, det hjälper oss inte heller att förankra med dom, vad dom kommer få. Scrum hjälper oss inte med någon ekonomisk styrning eller hela den biten. Runt om, styrgrupp när vi har stora leveranser för många miljoner så måste man ha en hel del saker runt om. Så där använder vi mer en klassisk projektledningsstruktur det kan vara Prince2, PMI, PMP eller liknande som vi oftast använder med riskstyrning och allt. Så Scrum är ju ingen helhetslösning, förr presenterades Scrum så, men det är det ju inte. Det skulle kanske funka helt internt, men även internt så har man ju ofta en budget som man sätter upp, en viss kostnad då, så jag tror man måste få kombinera. Och sedan så är det ju inte heller nåt, styrgrupp och övergripande styrningen, och stakeholderhantering, och riskhantering etc, det är ju problempjektledning. Sedan så hjälper ju inte Scrum oss heller hur vi ska bygga våra tekniska lösningar, så bygger webblösning säger vi så kanske vi använder test-driven development eller liknande metodik, under Scrum, som säger hur vi ska faktiskt utveckla och det är en arkitektur där också. Så Scrum är ju jätteduktig på sån här koncern

produktionsstyrningen, men man ska inte tror att det är en helhetslösning att man kan köra Scrum i projekt och sen är allting bra. Då kan man åka på problem och det gör man väldigt ofta också med Scrum. Jag har sett så många fallerade projekt där man, det finns ju kunder som tycker att agilt inte är bra för att dom har testat Scrum och det funkade inte.

### **Passar bra en viss typ av miljö.**

Ja, det är ju en del vi också jobbar med att inse att agilt är större än Scrum. Scrum är en metodik som är jättebra men man kan inte bara börja med det i en organisation som är van vid en vattenfalls-metodik eller liknande, där man sätter en budget som man följer upp och sen styr man efter tid, pengar och kvalitet så där klassisk projektledning. Men de som man är van att jobba krävs det också en inställning, en förändrad inställning till det agila på högre nivå kanske. Man måste inse att man styr på ett annat sätt. Så en del i att vi få det här att funka bra, att förankra med verksamheten, förankra med styrgrupp och beställa den ena eller andra saken. Om man inte gör det, förankra det så tror jag att man misslyckas med ett scrumprojekt då. För man styr ju inte längre efter en klassisk projektledare. Orädd för förändring för att man vill, man har ju sina, mitt Excel-ark säger att jag måste vara klar v17 och jag ska bränt så här mycket pengar, varje förändring blir ett hot mot det, det är det som jag ska styra mot. En förändring är att man måste se annorlunda på det, man måste tycka om förändring, se att det är en fördel. Man lär sig saker och ting under resans gång. Det är en inställningsförändring som inte Scrum i sig kommer ge, det måste man ta på en lite högre nivå.

## **SWOT**

### **Vilket är styrkan som du fram för allt ser med Scrum?**

Det är ju att man lär sig längs vägen. Det innebär om man vill skapa en lösning som är väldigt nära ambitionen som kunden har, så på resans gång om man hanterar kunden bra, så kommer dom förstå läsa mycket och förstå mycket bättre. Och där också komma till insikt att många av dom hårda kraven som dom tänkte sig från början inte stämmer. En hel del saker kan man ta bort, längs vägens gång och andra saker vill man lägga till. Så att man göra förändringar längs hela vägen, och det behöver ju inte nödvändigtvis bli dyrare utan jag har varit med om att man ta bort stora funktioner som har kostat så mycket pengar att man har insett att de behöver man inte vara så viktigt när man kommit halvvägs. Så ja, den stora, stora styrkan är att man lär sig längs vägen och just då att om då ser till att bjuda in verksamheten, kunden på rätt sätt så att de blir en automatisk förankring då. Men det är verkligen inte alla som jobbar med Scrum på de sättet. Men Scrum, använder man det så, så är det superstarkt. Det är den absolut stora biten

## **Svagheten?**

Svagheten i det stora, det är egentligen ingen svaghet i sig, men det finns en svaghet i att folk kanske tror att det är en komplett lösning och att tror man på det, och går efter det så kommer man ju åka på väldigt mycket nitar i väldigt många fall. Så att när det kom så var det som att det ersatte mycket annat. Men jag ser det som att de fyller ett hål för tar man dom klassiska Rupo om du känner till dom här rational unified process, dom där klassiska vattenfalls-metoder, dom har ju skapat agila varianter av sig själva nu också men dom beskrev egentligen inte så mycket tyckte jag, hur man skulle jobba i dagliga, dom beskrev mest om faserna. Så jag tycker Scrum fyllde ett hål, alltså det fanns lite annat också innan dess då. Så jag tycker inte Scrum i sig har någon svagheter det handlar mer om att komplettera med, mer runt om då. Och så en detalj som jag alltid lägger till i varenda Scrum-projekt som jag någonsin gjort, det är ändå tidsuppföljning, det är ju inte inbyggd i Scrum-modellen, i dom här verktygen som standard. Men jag har väldigt svårt att se vem som då inte vill se hur mycket tid man har lagt på nånting. Man ska alltid hålla koll på hur mycket som är kvar av ändå ren praktikalitet, men att ändå veta hur mycket man har lagt (på en viss aktivitet). Jag har svårt att se varför man inte har det som standard i alla såna Scrum-mallar och liknande och det lägger man alltid till. Jag behöver ju veta det för att kunna veta om jag kan fakturera och annat också. Men jag vill ändå lära mig, man gör ju ändå estimat även i Scrum. Och det är ju intressant att veta hur mycket tid man, det är nånting som jag tycker är konstigt. Men jag vet faktiskt inte ifall det är, det är mest mallarna vi använder, om det faktiskt inte ingår i Scrum. Man kan se velocity och liknande men det är ju övergripande, men det är ju intressant att veta, "oj, där estimerade jag väldigt fel". De tar ju ut varandra, jag kan ju fortfarande leverera blocket på en viss tid. Men som sagt det gäller ju bara att komplettera, alltså det finns ju inga konstigheter och jag tror inte Scrum säger att man inte får följa upp det kanske. Så att sen säger jag ju inte nödvändigtvis att det är Scrum med däremot alla implementationer av Scrum som jag sett på. Så far man, som tur så tillåter man nästan alltid konfigurering så kan man konfigurera till det.

## **Möjligheter?**

Det kan vara enabler att förbättra samarbetet till verksamheten. Jag kommer mycket från att man jobbar med IT-lösningar och då är kundens verksamhet som man jobbar med, kärnverksamheten dom kanske inte kan IT. Det är ett klassiskt problem, eller största problemet som alla brottas med inom IT är att man ofta är dålig på att ta den kommunikationen. Vi som Stretch, vi jobbar väldigt mycket med den här biten, tycker att det är väldigt viktigt. Men Scrum kan vara en enabler för att på ett bättre sätt ha en dialog med dom här och intressenterna, och få verksamheten att vara engagerad under projektets gång. Så man inte ser det som man startar nånting, väntar ett halvår och sen kommer det tillbaka nånting från andra sidan. Så Scrum metodiken hjälper oss, tillför lite mer struktur till det, och det tycker jag är en väldig "möjlighet" med Scrum i sig. Så är Scrum bredare än de vi använder idag, så man kan ju använda mer av det. Mer och mer att det kommer medföra i mjuka stationer också att det inte bara är IT-leverans. Det kan ju vara större projekt som ponerar flera olika typer av införande, exempelvis ett program av något slag eller liknande.



Kan använda Scrum lite bredare också så, det har inte jag sagt så mycket, vi gör lite så också så exempelvis hos en kund det är väldigt spännande att få se hur det faller ut.

### **Lärandet?**

Lärandet är kopplat där ju. Att få verksamheter att förstå lösningen eller inte och att fånga upp det synpunkten att det kommer förändras väldigt mycket under gången. Jag brukar säga att om hade vart superduktiga på det där, tar fram en komplett kravlista på dag 1 i projekt tillsammans med kunden och man gör det jättebra då kanske man träffar man kanske 70% av dom som man egentligen skulle behöva, men då tror jag man ska vara väldigt nöjd. Sen kan ju behoven förändras under resans gång på ett projekt som håller på ett år, men i grunden så har du ingen chans att leverera nånting som är riktigt, riktigt bra om du bara fångar in behov eller så, krav från start. Om du där emot uppdatera det och synka det under hela resans gång och verksamheten kan se hur det växer fram, så kommer man väldigt mycket mer få samma bild. Och det kan ju hjälpa att få 90% eller 100% av behovet levererat. När man har 100% av behovet levererat, det är det som är fantastiskt. Det kommer vara mer diskussioner under gång men det blir mindre diskussioner. Det är också en annan del i det, det går ju, men alla gör fel och kommer alltid göra fel men med Scrum finns det ju en god chans att man upptäcker det tidigare. Kanske bara lite? sprids om man gjorde fel, åt fel håll, men ibland kortare tid än så. Men jag har ju märkt ofta att o man får, våra kunder vill ju hänga med oss väldigt mycket när vi jobbar på det här sättet. Så nästan alla leveranser så har vi kund här en dag i veckan. Det har vi haft tidigare, har till och med haft kunder som har vart på plats nästan heltid. Just för att kunna vara nära och vara med under resan, men det är roligt att se det här växa fram, så det är en möjlighet och kraft, om man använder rätt.

### **Snabbheten? Att det går fortare att leverera med Scrum?**

Gör det det? Jag tror väl kanske det för om man, men det är bara kopplat om man, det upplevs fortare för man ser resultat oftare, två, tre eller fyra veckor. Vi brukar köra två veckors, så man får ju nytt demo varannan vecka, då märker man ju kunden upplever fart på ett annat sätt. Och det kanske går fortare om man upptäcker felen fortare. Så du lägger inte två månader åt fel håll, då tappar du väldigt mycket tid och kraft utan det rättas till snabbare, så ja. Jag vet inte om det är fortare eller snabbare metodik i sig än klassik, det vet jag inte ifall det är så. Men effekterna bli så, man upplever att det blir så.

### **Ser du några hot med att använda Scrum?**

Hot är inte kopplat till Scrum i sig utan vad som är förväntad av Scrum. Så kunder, många kunder som har mycket bekymmer, klassiska organisationer och fungerar inte så väl och vet inte riktigt vad som är felet och sen hört talas om den här agila metodiken, och sen säger man att vi ska bli agila och sen kanske man utbildar sig på Scrum och sen börjar man jobba på det,

och sen tror man att man är agil. För man ser, man förknippar ofta agil och Scrum, likhetstecken mellan det. Jag menar det är ju större konceptuellt och tänka lite annorlunda också. så det som händer när man kör ett projekt med Scrum men man har inte anpassat sig i övrigt. Man förstår inte vad det innebär, hur ska man tänka med budget och det kan vara lite övergripande. Det gör att man ofta fallerar, man tror att allt ska lösa sig med Scrum, men det är andra saker runt om som är, inte är tillräckliga som inte kommit tillräckligt långt på den här resan. Så fallerar det, det projektet och sen får Scrum och de agila, finns ju organisationer som där det är ett skällsord. Men det är ju inte Scrums fel eller så utan det är vad effekten blir för man tror. Det är ju inte ett hot med Scrum. Det finns en mogenhet som behövs, det ställer lite andra krav på, jag menar jag har varit med Scrum när man inte fått någon attention från verksamheten, där man inte haft tid att vara med. Då blir det bara ett internt redskap för att styra produktionen, det använder ju jag ändå, förvissat att jobba så. Men då tappar man väldigt mycket av det, då tror man jobbar med Scrum kanske men egentligen så är det bara ett sätt att planera sin tid och sen har man inte fått någon dialog med verksamheten och då fallerar ju allting. Det är ju inte Scrum som är problemet i sig, det är det som är runt om. Inför man Scrum och inte är mogen att förstå vad det innebär och man måste ändå investera samtidigt för att få ut effekterna. Då kan det vara bättre att ha ett klassiskt vattenfall som alla förstår sig på, då kan man i alla fall göra det någorlunda bra. Med Scrum kommer man kanske falla totalt, om man inte har det här.

Det kräver ju faktiskt, för att det ska funka, för att hela konceptet ska fungera att du måste ha en kontinuerlig dialog. Om du är med verksamheten, någon som är med, som ser behoven och som kan vara med och styra. Om bara utvecklare sitter där, som inte förstår verksamheten, då kan man ju springa åt vilket håll som helst. Och det kan ju vara en stor risk då många utvecklare förstår inte den här delen, hur viktig det är att man har den förankringen. Så den organisationen, dom väljer ju då Scrum och så ger dom inte någon tid, med sina personer att jobba med teamet, då blir det nästan värre än vattenfall. I vattenfall då har dom förstått att dom i alla fall behöver ta fram en jäklig detaljerad kravspec, och jobbat med leverantören och med oss. Jobbat med workshoppar och annat, så då har jag haft det i alla fall då har vi fått dedikerad tid under någon månaders uppstart från verksamheten, det förstår dom ju, eller hur är det så. Så får man chans att komma upp i 70% träffbild. Men Scrum då har man inte den stora grejen och om dom inte förstår allvaret att dom måste vara med varje vecka, då får man faktiskt mindre attention totalt sett, uppmärksamheten och då kommer man inte leverera nånting då, är väl så. Jag ser inga hot i sig annars utan bara man måste vara noggrann med vad det innebär

## Följdfrågor

### Vad tycker du om sprintar och dess regler

Konceptet med sprinter är jättebra, det skapar väldigt mycket struktur. Ändrar vi i sprintar ibland? Ja det gör vi absolut. Saker och ting händer, det kan hända att vi bara inte hinner med och så plockar man ut nånting, och så lägger man till en funktion. Vi försöker att inte röra så mycket under sprintarna men det kommer ju hända ibland. Vi har alltid en pragmatisk inställning, vi kommer inte följa någon regel slavisk, det är det jag menar också utan vi gör lite anpassningar ibland. Man ska ha det som ett mål att man inte ska behöva ändra för då funkar ju hela mekaniken mycket bättre. Det är klar vi behöver ha ändringar in ibland.

### Scrum roller?

Jag tycker det är jättebra, jag tycker bara att problemet, ska jag vara ärlig, att man alltid pratar om ScrumMaster i såna fall då. Att de är sån, att man alltid fokuserar på, Product Ownern var väldigt, alltså väldigt viktig men jag tror det är väldigt så ofta man bara pratar om ScrumMaster men man har inte så mycket koll på vem som är Product Owner om man säger så. Nånstans så är ju Product Owner det viktigaste. Balansen, bilden av de dom lägger fokus på det är inte rätt för det mesta tror jag. (Product Owner) Äger backloggen tänker jag och styr behovet och värdet och det är ju personen som är nyckeln till verksamheten. Så för deras talan, och ser till att vi har sagt saker. Alla går ju i princip och utbildar sig till ScrumMaster, vi är som 20 stycken utbildade ScrumMasters här på kontoret, men Product Owner hur många har gått den utbildningen i sig?

### Hur arbetar ni med kravspecifikationen

Förr i tiden när jag jobbade mycket med vattenfall, hade man ofta en 3-månaders fas eller liknande, vid lite större projekt, där man satte kraven. Jag skrev tiotals sidor med text, små punkter med specifikationer. Vid scrumprojekt gör vi det fortfarande, en kravuppsamlingsfas innan, men den är mycket mindre. Det handlar mer om att förstå flödena, processerna, affärsvärden, det är dom bitarna. Man tar ändå fram en kravbild så man kan göra en grundarkitektur från start och man vet ungefär var man är på väg åt i grova drag. Så det gör man fortfarande, jag fortfarande en sån fas och det tror jag egentligen inte ingår i Scrum, det nånting som man behöver ha.

Vi kanske kan ta det såhär; jag tänker på projekt på 5 miljoner då är det kanske bara en månad som man gör en uppstart, man gör lite bemanning och alla dom här bitarna och sedan så bygger man ju upp. Inför varje ny Sprint har vi ett möte med Product Ownern och kanske några till, där vi diskuterar just, detaljerar dom sakerna som ligger högst upp i backloggen.

Och sedan har man, när vi jobbar så har vi normalt sett ett möte varannan vecka med slutanvändare.

När vi levererade en lösning åt Region Skåne, målplanen då hade jag möte med 10-20 sjuksköterskor de måste vi ge lösningen varje sprint då, och ta in synpunkter och anpassningar och så. Så ofta tar 80% de av man visar, diskuterar med slutanvändarna, riktiga användare som inte kan nån teknik, se hur dom reagerar, och sen så tar man med sig det i nästa sprint.

Vi gjorde en leverans som var klar nu i januari. Där vi släppte ny vårdplan, som används. Vi levererar lösningar som man använder när, om någon kommer till sjukhuset för vård, och sen behöver man hjälp efter man lämnar sjukhuset. De kan vara allting från att man ska ha hemtjänst, eller behöver en rullator, eller arbetsterapeut som du ska ha möte med eller vad som helst. I alla dom fallen blir det samarbete mellan sjukhus, primärvård och kommun. Då använder man vårt system för att planera det, samarbetet dom tre emellan. Då måste dom träffas i praktiken, diskutera innan vem som gör vad för patienten. Så det är en plan som vi haft länge. Nu kommer behovet att hantera patienten i sitt hem. Finns ju många, fram för allt äldre men även personer med mentala sjukdomar. Men om man nu kan få vård hemma. Det är onödigt, det kostar mycket att komma till sjukhuset och dessutom låter patienten vara hemma. Så då behöver man en ny plan här för kommun och primärvård, dom åker ut till många patienter då, eller kanske inte patient då men brukare eller så, så dom behöver hemtjänst där man kanske behöver byta bandage eller liknande. Primärvård gör vissa saker och kommunen vissa. Så där är ju dumt för patienten när det kommer hem en massa olika människor, så det gör att det kommer hem en person som kanske gör alltihopa. Så där kan man koordinera. Då har tagit fram en ny plan, det har varit massa med jobb med Region Skåne. Har jobbat i mer än 5 år att ta fram en gemensam plan hur man ska jobba med detta. När vi kom in och skulle leverera IT-lösning för detta då trodde vi att det fanns en gemensam plan, med det visade sig snabbt att det fanns minst fem olika. Malmö stad hade en, Örkelljunga hade en, östra göringe hade en, psykiatrin hade en fjärde och så, och dom var helt olika. Så då fick vi ju börja med att hjälpa då att komma överens om en gemensam plan. Då gjorde vi den resan, tog ungefär ett år och sen levererade vi januari och i januari åkte först sjuksköterskan ut till en patient, satt vid middagsbordet hemma, och hade dator med sig då använde våran plan. Och kom tillbaka och sa att det var helt fantastisk sa att det passade hennes samtal med brukaren så bra. Det var precis som att de var att hon fick ett mentalschema sa hon. Så det blev en jätteframgång första gången dom använde det. Och så tänker jag på den första specifikationen vi fick när vi började, den fanns ju inte ens nästan. Så det är väldigt bra exempel på hur Scrum, man jobbar och är genom det här så kunde man ju ta fram en tydlig lösning. Men då fick vi alla dom här olika intressenterna att gå med, att bli förankrad, dom fick komma med på gemensam röst också, de var mycket diskussioner där; hur ska det funka? en massa regler, när ska det godkännas? när är den aktiv? och så vidare; vilka krav måste kommunen göra och hur gör primärvården och så vidare. Det är en enorm resa dom har hölt på med det i så många år och diskuterat det. Men det är först nu vi tar upp det diskuterar, och visualiserar det konkret och då kan man komma överens, det är så himla viktigt. Det är ju inte Scrum, Scrum hjälper oss där men det är ju väldigt mycket annat som utöver det som man måste tänka på och använda

det på det sättet. Men då är det en fantastisk resa, då träffade vi nästan 100%, utgick från de vi hade, hade vi träffat 10%, så är det ju.

### **Har ni haft kunder som inte kunnat hantera Scrum?**

Historisk så har jag många olika baksidor, alltså delvis att kunden inte kunnat lägga tillräcklig tid, så då har man inte fått ut kraften som det innebär att jobba på det sättet. Det är ju kopplat till budget, oftast får man en fast budget och sen ska man ändå jobba såhär och då handlar det om att ha rätt inställning, vad det innebär. Så det kan vara olika utmaningar. Vi är ju lite transparenta, blir man ju per automatik mer med Scrum också. Hade man jobbat i vanligt vattenfall hade man inte sett allting, alla fel som görs och så där. Tvärtom jag tycker inte det är något problem alls egentligen, men jag kan ändå se är man inte van vid det så som kund så få man bli lite överraskad ibland. När har utvecklat något som inte passar, det är något som bör var styrkan i det, men man måste vänja sig vid det. Men däremot i slutändan alla som har jobbat med oss med det här älskar ju det, vill inte jobba på något annat sätt sen. Men det är en förändring i inställning.

### **Hur ser du på Scrum möten?**

Ingen speciell synpunkt så. Vi har alltid ett möte inför Scrum eller inför en ny Sprint och vi har alltid demo efter varje sprint. Retrospective brukar vi ha, så har vi alltid varje vecka ett klassiskt projektledningsmöte där vi brukar på vartannat ha retrospektive där. Daily Scrum får jag säga är vi rätt dåliga på här, men vi varierar ju, i vårt fall så sitter vi hela teamet bredvid varandra och det handlar också om hur många man är. Jag har jobbat historisk mycket i distribuerade team både i Indien och Uppsala, då blir det viktigare att ha daily Scrum. Och sen i större team, när det kanske är team som är flera till och med teams sådär. Jag tror det blir mer kvalitet med ett mindre team som jobbar en längre tid än väldigt många som jobbar kortare klandertid för att leverera nånting. Så när vi kanske är 3-4 stycken i ett projekt och gör på ett år istället vara 8 st för ett halvår. Daily Scrums är viktiga i lilla teamet men om vi däremot hade varit 8 st hade det varit ännu viktigare. Det märker jag att vi glider på en hel del.

### **Dokumentationen?**

En typisk grejer som inte Scrum hanterar alls ju kring dokumentation. De ser vi hela tiden, projekt som slarvar med det, men dokumentation måste vara en leverabel som allt annat i Scrum. Som jag ser det, i varje utvecklingsuppgift som ingår måste man också uppdatera sin dokumentation. Men det tycker jag är faktiskt ett problem till viss del, att i Scrum inte har en höjd fördel för det borde nästan vara inbyggd för att det är så viktigt. Vi tar precis över ett nytt web-projekt här från en stor firma, alltså välkänd firma som har duktiga utvecklare, men som inte kan mycket av de jag pratar om, om man ska vara helt rå. Dom har ju levererat en stor lösning till en kund under flera år och det är bara skit allting kan man säga. Men det är ju inte

för att dom inte är duktiga tekniker, det är för att dom inte har haft kunnat att kommunicera med verksamheten, verksamheten har inte förstått vad dom säger, och så har dom inte dokumenterat nånting under möten. Så tar vi över de här efter två år och ska göra det här rätt, så inser vi att det finns ingenting. Vi ska ta över detta som ska vara nästan färdigt trodde vi när vi tog över det men nu inser vi mer och mer att det inte alls är färdigt, det är mycket som inte funkar. Vi ska försöka leverera-upp detta. Men vad vi gör vi våra kundmöten, vi skriver alltid protokoll, som vanlig historik, vilka beslut som togs, och vi ska ju såklart dokumentera lösningen. Dom har inte gjort det alls, det är svårt för oss att ta över såklart, det är otroligt svårt att få fram vem sa vad i dom diskussionerna som dom har. Men dom har slarvat så mycket, dom har inte ens tagit fram som en installationsanvisning som funkar.

Dokumentationen är jätteviktig, men den ska hållas på rätt nivå. Förr i tiden, ni ska se specarna som jag skrev för planlösningen jag berättade om. Där skrev jag ett samlat arkitektokument som jag tror var på 120 sidor och sen en design dokumentation på nästan 200 sidor. Jag tror 300 sidors dokumentation som jag skrev för den specen, helt galet. Idag, jag skulle aldrig få för mig att skriva nåt sånt. När jag ger den till någon utvecklare, kan ju inte läsa det, det tar ju veckor att läsa. Så man gör det på ett annat sätt kanske, vi visualiserar mycket mer än skrift, men dokumentationen är ändå superviktig. Vi plussar på det där, det har inte med Scrum i sig att göra, men det är en leverabel i Scrum. Och sen ser vi till att vi har protokoll på våra möten och så där.

### **Hur är risken med att förlora gruppmedlemmar?**

Det kommer man aldrig ifrån. Om det är en team medlem i Scrumgruppen så kanske det är lite enklare för man är mer medveten om vad den andra har gjort, men det är fortfarande samma. Scrum pratar ju om att man tar vilken task alltså uppgift man vill men i praktiken fungerar det inte riktigt så, man kommer vara olika bra på olika saker kunna olika delar av en lösning olika bra. Så det är klart att det fortfarande kommer vara jobbigt om en viss person med viss specialkompetens slutar. Jag tycker inte Scrum gör så himla mycket att förbättra det. Det handlar inte bara om kunskap det handlar om intresse också. Någon person tycket det är jävligt mycket coolare att bygga coola webbgränssnitt med javascript och någon annan älskar att bygga integrationerna, älskar att bygga tunga algoritmer i back-enden. Och så är det ju, om man är ett bra team kommer man ha dom här nischningarna, man kommer tycka om olika saker. Man kommer inte ha fem stycken som är likadana. Det är inte bästa varianten.

### **Fördelning av resurser i Scrum**

pratad om

## Kundens centrala roll

pratad om. Men dom ska inse så mycket som möjligt, det finns ju upplärning för kunden, om man gör det rätt kommer dom tycka det är underbart sen.

## Scrum och samarbete i utspridda grupper?

Har gjort det väldigt mycket. Då tycker jag Scrum är en jättestyrka. Jag jobbade ju tidigare på ett av dom stora internationella konsultfirmorna, Capgemini i fallet då. På den tiden hade vi 20 000 indier. Jag var ansvarig för utvecklingsgrupperingen. Jag drev dom största projekten vi hade i Sverige kring utveckling och där var det med indier, jag var dessutom högst rankad inom Capgemini, man hade ett sånt rankingssystem och då var jag top-50 i världen. Så då ingick i en Squad som åkte runt och räddade projekt som havererade i Europa. Så jag har sett en del utmaningar just i distribuerade team. Varit ner i Österrike exempelvis och då var det en SharePoint lösning där som skulle levereras om två veckor och ingenting funkar och då var det nästan bara byggt från Indien. Så alla projekt 2006 och framåt, 2006-2011 var det alltid distribuerade team. Jag var alltid arkitekt, jag var projektledare nån gång men min preferens var att vara arkitekt och utvecklare och nån annan som var projektledare som är bättre på det. Min preferens har alltid varit att äga den tekniska preferensen. Så det är ju typiskt jag som tar hand om dom här utvecklarresurserna, jag som är den som hanterar det distribuerade teamet, medans projektledaren hanterar det övergripande; stakeholders, förväntningar, budget och så vidare. Så Scrum, i Indien jobbade man överhuvudtaget inte med agilt utan det är klar vattenfall det man är van vid. Så man jobbar i distribuerade. Jag såg ju det agila som ett jätte starkt verktyg för att jobba distribuerat. Nu är inte dom vana det i Indien och, men jag fick ju utbilda dom då. Man ska komma ihåg det är olika kulturer, i Indien är man väldigt van vid att en överordnad som bestämmer vad man gör, så gör man exakt de man blir tillsagd, och så är det bara så. Men i ett agilt projekt så måste man ju ta eget initiativ. Det där går inte utbilda på distans, så jag lärde mig så snabbt, så tog jag hit indierna, dom som jobbade i teamet så jobbade vi ett par månader i Sverige med oss och fick den förståelse för vilka förväntningar som vi hädelsen kunde dom åka till Indien och jobba därifrån, då tog dom mer egna initiativ sen, och då började det fungera bättre. Det handlar om att förstå varandra, förstå kulturer, det är ju specifikt för Indien och varje sån har ju sina utmaningar. Men man kan jobba med Sverige, jag jobbar ju men folk i Uppsala exempelvis, återigen då är det också jättevärdefullt med just detta. Jag tycker det är en superenabler för att jobba distribuerat. Uppsala var ju ett annat företag, men man måste bestämma sig när man ska jobba tillsammans och hur man ska göra. I mitt fall handlar total transparens, då funkar det ju bra med agila. Men om man tänker sig man ska ha några slag olika affärsmodeller, man vill ta betalt, dölja sina timmar då blir det väldigt väldigt svårt. Jag är för ärligheten där, man måste vara helt öppna med varandra. Hur tar jag betalt? Och man är inte intresserat av att göra nånting, fulspel. Så det är viktiga komponenter i det, för annars kan man nog kolla efter ett annat arbetssätt för det blir väldigt tydligt. I distribuerat finns ju andra utmaningar, Scrum i sig är jättebra

## Bilaga B – Frågor för intervju

### 1. Personliga frågor

#### 1. Personliga frågor

1.1 Vad heter du?

1.2 Vad är din yrkestitel?

1.3 Hur länge har du jobbat som (yrkestitel/projektledare)

1.4 Hur länge har du jobbat med Scrum?

### 2. Generella frågor om Scrum:

2.1 Använder ni Scrum idag?

2.2 Om svaret är ja:

2.2.1 Varför använder ni Scrum?

2.2.2 När använder ni Scrum?

2.2.3 Hur använder ni Scrum?

2.2.4 Komplimenterar och/eller kombinerar ni Scrum med andra metoder?  
-Varför?

2.3 Om svaret är nej:

2.3.1 Varför använder ni inte Scrum?

### 3. SWOT frågor om Scrum

3.1 Vilka styrkor ser du med att använda Scrum?

3.2 Vilka svagheter ser du med att använda Scrum?

3.3 Vilka möjligheter ser du med att använda Scrum?

3.4 Vilka hot ser du med att använda Scrum?

### 4. Följdfrågor

4.1 Vad tycker du om Sprintar (Intern)

-Regeln att ej ändra målet under en Sprint

(Koppling till teori: 2.1.1 Sprintar)

4.2 Vad tycker du om rollerna i Scrum (ScrumMaster, Product Owner, Team) (Intern)

-Vad tycker om samarbete mellan rollerna inom Scrum?

(Koppling till teori: 2.1.3 Scrum Roller)

4.3 Hur arbetar ni med att utveckla kravspecifikation? (Extern)

-Hur samarbetar ni med kunden efter att projektet har påbörjat?

-Har ni haft kunder som ej har kunnat anpassa sig till Scrum metoden?

(Koppling till teori: 2.2.2 Kravhantering)

4.4 Hur ser du på Scrum möten ? (Intern)

-Planeringsprocessen (planning meeting)

-Daily Scrum

-Sprint Review

-Sprint retrospective

(Koppling till teori: 2.1.4 Scrum Evenemang)

4.5 Hur fungerar dokumentationen i Scrum? (Intern)

(Koppling i teori: 2.2.3.8 Dokumentation)



**4.6** Hur ser ni på, och hanterar risken att förlora gruppmedlemmar under projektets gång? (Extern)

(Koppling till teori: 2.2.3.8 Dokumentation, 2.2.4.4 Processer)

**4.7** Hur ser du på fördelning av resurser i Scrum? (Intern/Extern)

Koppling till teori: 2.2.2.2 Resursallokering)

**4.8** Hur upplever ni kundens centrala roll i projekten? (Extern)

-I hänsyn till planering av resurser

-uppbyggnaden av projektet

-plötsliga ändringar i projektet av kunden

(Koppling till teori: 2.2.4.1 Kunder, 2.2.4.6 Kundinblandning)

**4.9** Har ni erfarenhet av att teamet är placerad i olika städer (distributed teams)(Intern)

-Hur ser du på Scrum och samarbete i utspridda grupper, ex scrum möten?

(Koppling till teori: 2.2.3.6 Kommunikation och distribuerade teams)

## Referenser

Alvesson Mats och Sköldbberg Kaj (1994): Tolkning och reflektion vetenskapsfilosofi och kvalitativ metod. Studentlitteratur, Lund.

Eriksson Lars Torsten (2001): Att utreda forska och rapportera. Liber Ekonomi, Malmö.

Nils Brede Moe, Aybuke Aurum och Tore Dybå (2011): Challenges of shared decision-making: A multiple case study of agile software development. Norway School of Information System, Technology and Management,

Nilse Brede Moe, Dingsøy Torgeir , Tore Dybå (2010) : A teamwork model for understanding an agile team: A case study of a Scrum project. . Information & Software Technology. May2010, Vol. 52 Issue 5,

Cho Juyon (2008) : Issues and challenges of agile software development with Scrum.. Colorado State University-Pueblo

Holme Idar Magne och Solvang Krohn Bernt (1997): Forskningsmetodik Om kvalitativa och kvantitativa metoder. Studentlitteratur, Lund.

Holmström, Helena och Brian Fitzgerald, Pär Ångfalk, Eoin Conchuir (2006): Agile practices reduce distance in global software development.Source: Information Systems Management. Summer2006, Vol. 23 Issue 3

Pries-Heje, L.Pries-Heje (2011) : A case study from an agile distributed project in Denmark and India Source: AGILE Conference, 2011; 2011, p20-28, 9p

Holzmann Vered, Panizel, Ilanit (2013) : Communications management in Scrum Projects. Proceedings of the European Conference on Information Management & Evaluation. 2013, p67-74. 8p

Jacobsen Dag Ingvar (2002) : Vad, hur och varför? Om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen. Studentlitteratur, Lund

Lee Chih Rich (2012) :The Success Factors of Running Scrum: A Qualitative Perspective System Technology Group,I BM, Chinese Taipei. Department of Computer Science, National Taipei University of Technology.

Repstad Pål (2007) : Närhet och distans Kvalitativa metoder i samhällsvetenskap. Studentlitteratur, Poland 2007

Rubin Kenneth S (2013): Essential Scrum A Practical Guide to the Most Popular Agile Process. Edwards Brother Malloy, Fourth printing.

Schwaber Ken (2004) : Agile Project Management with Scrum. Microsoft Press

Strauss Anselm, Corbin Juliet (1998) : Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Sage Publications: London.

Tanner, Maureen, Mackinnon Angela (2013): Sources of disturbances Experienced during a Scrum sprint. Proceedings of the European Conference on Information Management & Evaluation. 2013.

Team FMW (2013) , SWOT Analysis Strategy Skills. [www.free-management-ebooks.com](http://www.free-management-ebooks.com). ISBN 978-1-62620-951-0.

Vlaanderen Kevin, Slinger Jansen, Sjaak Brinkkemper, Erik Jaspers: The agile requirements refinery: Applying SCRUM principles to software product management. Department of Information and Computer Sciences, Utrecht University, Utrecht, The Netherlands

Williams Laurie (2011) : Scrum + Engineering Practices: Experiences of Three Microsoft Teams  
North Carolina State University.