



# LUNDS UNIVERSITET

## Ekonomihögskolan

*Institutionen för informatik*

## Scrums negativa konsekvenser

Kandidatuppsats 15 hp, SYSK02 Examensarbete i informatik.

Framlagd maj, 2016

Författare: Petter Billgren  
Leon Wipp Ekman

Handledare: Magnus Wärja

Examinatorer: Benjamin Weaver  
Paul Pierce

# Scrums negativa konsekvenser

Författare: Petter Billgren och Leon Wipp Ekman

Utgivare: Inst. för informatik, Ekonomihögskolan, Lunds universitet

Handledare: Magnus Wärja

Examinatorer: Benjamin Weaver, Paul Pierce

Slutseminarium: Maj, 2016

Dokumenttyp: Kandidatuppsats

Antal sidor: 117

Nyckelord: Scrum, Agil, Negativ konsekvens, Stora projekt, Kommunikation, Tekniska problem

## Sammanfattning:

Scrum växer fram som ett svar på traditionell projektledning och har fått mycket positiv kritik eftersom det anses vara bättre anpassat för mjukvaruutveckling. Denna studie behandlar Scrums negativa konsekvenser, ett område som delvis förbisetts av tidigare litteratur. Vi bygger upp ett teoretiskt ramverk med aktuell litteratur som sedan används som grund till djupintervjuer. Vi finner stöd i litteraturen för problem under tre huvudområden; tekniska problem, stora projekt och kommunikation. Dessa områden består i sin tur av flera negativa konsekvenser. Intervjuerna har gjorts med fem personer med olika befattningar inom Scrum för att innefatta så stor del av arbetsmetodiken som möjligt. Av studien framgår det att de tidigare identifierade problemen både avvisas och bekräftas samt att vi får stöd för andra negativa konsekvenser. Av totalt nio identifierade negativa konsekvenser avvisas två helt, ytterligare två kan varken avvisas eller bekräftas och fem får stöd i empirin. Vi finner även stöd hos de intervjuade för ytterligare två problem som vi inte funnit nämnda i tidigare litteratur.

---

# Innehållsförteckning

|  |           |
|--|-----------|
| <b>Figurer .....</b>                       | <b>6</b>  |
| <b>Tabeller .....</b>                      | <b>7</b>  |
| <b>1. Inledning .....</b>                  | <b>8</b>  |
| <b>1.1 Bakgrund .....</b>                  | <b>8</b>  |
| <b>1.2 Problemområde .....</b>             | <b>9</b>  |
| <b>1.3 Forskningsfråga .....</b>           | <b>10</b> |
| <b>1.4 Syfte .....</b>                     | <b>10</b> |
| <b>1.5 Avgränsningar .....</b>             | <b>10</b> |
| <b>2. Litteraturgenomgång .....</b>        | <b>11</b> |
| <b>2.1 Scrum .....</b>                     | <b>11</b> |
| <b>2.2 Scrum i Scrum .....</b>             | <b>14</b> |
| <b>2.3 Kritik mot Scrum .....</b>          | <b>15</b> |
| <b>2.3.1 Tekniska problem.....</b>         | <b>15</b> |
| Generella lösningar .....                  | 15        |
| Teknisk skuld.....                         | 15        |
| Stress .....                               | 16        |
| <b>2.3.2 Stora projekt.....</b>            | <b>17</b> |
| Olika geografiska platser .....            | 17        |
| Förmedling av strategi.....                | 18        |
| Svårt att involvera underleverantörer..... | 18        |
| <b>2.3.3 Kommunikation.....</b>            | <b>19</b> |
| Intern kommunikation .....                 | 19        |
| Extern kommunikation.....                  | 19        |
| Ingen formell dokumentation.....           | 20        |
| <b>2.5 Identifierade problem.....</b>      | <b>21</b> |
| <b>3. Metod .....</b>                      | <b>22</b> |
| <b>3.1 Metodval .....</b>                  | <b>22</b> |
| <b>3.2 Urval.....</b>                      | <b>22</b> |
| <b>3.3 Genomförande av intervju.....</b>   | <b>23</b> |

---

|   |           |
|---|-----------|
| <b>3.4 Intervjuguide .....</b>                          | <b>23</b> |
| <b>3.5 Datainsamling och analys av intervjuer .....</b> | <b>24</b> |
| <b>3.6 Undersökningskvalitet .....</b>                  | <b>25</b> |
| 3.6.1 Validitet .....                                   | 25        |
| 3.6.2 Reliabilitet .....                                | 26        |
| <b>3.7 Etik .....</b>                                   | <b>26</b> |
| <b>4. Resultat .....</b>                                | <b>28</b> |
| <b>4.1 Tekniska problem .....</b>                       | <b>28</b> |
| 4.1.1 Generella lösningar .....                         | 28        |
| 4.1.2 Teknisk skuld .....                               | 29        |
| 4.1.3 Stress .....                                      | 29        |
| <b>4.2 Stora projekt .....</b>                          | <b>30</b> |
| 4.2.1 Olika geografiska platser .....                   | 30        |
| 4.2.2 Förmedling av strategi .....                      | 31        |
| 4.2.3 Involvera underleverantörer .....                 | 32        |
| <b>4.3 Kommunikation .....</b>                          | <b>32</b> |
| 4.3.1 Intern kommunikation .....                        | 32        |
| 4.3.2 Extern kommunikation .....                        | 33        |
| 4.3.3 Formell dokumentation .....                       | 34        |
| <b>4.4 Andra problem .....</b>                          | <b>34</b> |
| 4.4.1 Otydlig karriärstege .....                        | 34        |
| 4.4.2 Brist expertis .....                              | 35        |
| <b>4.5 Sammanfattning .....</b>                         | <b>35</b> |
| 4.5.1 Tekniska problem .....                            | 35        |
| 4.5.2 Stora Projekt .....                               | 36        |
| 4.5.3 Kommunikation .....                               | 36        |
| 4.5.4 Andra problem .....                               | 36        |
| <b>5. Analys .....</b>                                  | <b>37</b> |
| <b>5.1 Tekniska problem .....</b>                       | <b>37</b> |
| 5.1.1 Generella lösningar .....                         | 37        |
| 5.1.2 Teknisk skuld .....                               | 37        |
| 5.1.3 Stress .....                                      | 38        |
| <b>5.2 Stora projekt .....</b>                          | <b>38</b> |
| 5.2.1 Olika geografiska platser .....                   | 38        |
| 5.2.2 Förmedling av strategi .....                      | 39        |

---

|  |            |
|--|------------|
| 5.2.3 Underleverantörer .....                      | 39         |
| <b>5.3 Kommunikation .....</b>                     | <b>40</b>  |
| 5.3.1 Intern kommunikation .....                   | 40         |
| 5.3.2 Extern kommunikation .....                   | 40         |
| 5.3.3 Formell dokumentation .....                  | 41         |
| <b>5.4 Andra problem .....</b>                     | <b>41</b>  |
| 5.4.2 Avsaknad expertis .....                      | 41         |
| <b>6. Slutsats .....</b>                           | <b>43</b>  |
| 6.1 Avvisade problem .....                         | 43         |
| 6.2 Varken bekräftade eller avvisade problem ..... | 43         |
| 6.3 Bekräftade problem .....                       | 43         |
| <b>7. Bilaga .....</b>                             | <b>45</b>  |
| <b>7.1 Ordlista.....</b>                           | <b>45</b>  |
| 7.1.1 Roller .....                                 | 45         |
| 7.1.2 Ceremonier .....                             | 45         |
| 7.1.3 Artefakter .....                             | 46         |
| <b>7.2 Intervjumall.....</b>                       | <b>46</b>  |
| <b>7.3 Intervjuguide .....</b>                     | <b>50</b>  |
| <b>7.4 Intervju 1 .....</b>                        | <b>50</b>  |
| 7.4.1 Kompletterande frågor till intervju 1.....   | 65         |
| <b>7.5 Intervju 2 .....</b>                        | <b>65</b>  |
| <b>7.6 Intervju 3 .....</b>                        | <b>78</b>  |
| <b>7.7 Intervju 4 .....</b>                        | <b>91</b>  |
| <b>7.8 Intervju 5 .....</b>                        | <b>102</b> |
| 7.8.1 Kompletterande frågor till intervju 5.....   | 114        |
| <b>7.9 Referenser .....</b>                        | <b>115</b> |

## Figurer

|               |    |
|---------------|----|
| Figur 1 ..... | 13 |
| Figur 2 ..... | 14 |

---

## Tabeller

|   |    |
|---|----|
| Tabell 1 - Identifierade problem .....                              | 21 |
| Tabell 2 - Identifierade problem kopplade till intervjufrågor ..... | 24 |
| Tabell 3 - Intervjupersoner.....                                    | 28 |
| Tabell 4 - Alla identifierade negativa konsekvenser.....            | 44 |
| Tabell 5 - Intervjuguide .....                                      | 50 |

# 1. Inledning

## 1.1 Bakgrund

Det pågår en debatt kring agil mjukvaruutveckling och dess konsekvenser. Debatten avser projektledning som är iterativ, levererar inkrementellt och har nära kontakt med kunden. Denna typ av metodik har under de senaste tio åren blivit väldigt populära och företag antar agila metoder för att bli mer konkurrenskraftiga och effektiva. Andra företag förhåller sig mer skeptiskt till de påstådda fördelarna och ser även nackdelar med denna typ av metodik. (Barlow et al. 2011, s. 26)

Agil mjukvaruutveckling kommer under 90-talet som ett svar på traditionell projektledning som visat sig vara bristfällig i många avseenden. Den då dominerande vattenfallsmetoden ansågs vara för tung och svårmanövrerad i en bransch som utvecklades snabbt. The Standish Group visade 1994 i "*The Chaos Report*" att endast 16 % av projekten avslutades i tid och höll sig inom budget. På grund av detta ansågs inte vattenfallsmetoden vara en tillförlitlig arbetsmetodik och agila metoder började växa fram. (Joey Cho 2010, s. 9-11)

Efterhand som agil projektledning växer sig populärare får metoden mycket positiv kritik. Den anses av många företag vara lösningen på problemen som projekt kantades av i sekventiell projektledning. Den forskning som har gjorts visar också att förändring hanteras bättre, kommunikationen ökar och projekten avslutas snabbare. (Pikkarainen et al. 2008, s. 303-304) Scrum har blivit den populäraste formen av agil projektledning. Scrum Alliance påstår att 60 % av alla agila projekt antingen är Scrum eller en variant av Scrum (Scrum Alliance 2016).

Även om Scrum får stöd i aktuell forskning är inte alla övertygade att projektledningsmetoden endast har positiva effekter. Relativt lite forskning behandlar Scrum, en ännu mindre del belyser nackdelarna med metoden. Det finns således en kunskapsbrist kring baksidorna med Scrum. Framförallt saknas empiriska studier om effekterna av Scrum, detta trots att kritik länge framförts mot Scrum och agila metoder. (Lei et al. 2015, s. 1-3) Det finns exempel på studier som försökt identifiera Scrum's negativa konsekvenser, men det finns inget konsensus kring vilka negativa aspekter som Scrum ligger till grund för. Den kritik som förs mot Scrum har inte heller följts upp empirisk på ett adekvat vis. Detta är den grundläggande orsaken att behandla detta ämne i denna studie.



## 1.2 Problemområde

Scrum har under 2000-talet vuxit till en populär arbetsmetod för mjukvaruutveckling och har kommit att ersätta traditionell projektledning både i små och stora organisationer (Hummel 2014, s. 4712). Trots att Scrum har fått erkännande av en mängd olika företag i olika branscher har relativt lite forskning gjorts om arbetsmetoden (Elbanna 2014, s.1). Flera studier har visat på de fördelar och positiva effekter Scrum och agila metoder har på mjukvaruutveckling.

Kommunikationen ökar, arbetet blir mer transparent, det är flexiblere och lättare att anpassa arbetet till nya krav och kunden är involverad i arbetsprocessen på ett bättre sätt. Framförallt har agila metoder setts som lösningen på de många problem som traditionell projektledning har som konsekvens. (Joey Cho 2010, s. 20-21)

Forskningen om agila metoder har delvis präglats av en positiv snedvridning där många av de negativa aspekter som kommer med metoden har förbisetts. De agila metoderna ställs i perspektiv mot till exempel vattenfallsmetoden och påvisas vara ett bättre alternativ. På så vis begränsas förståelsen om hur Scrum påverkar projektledningen som helhet. (Dybå och Dingsøy 2008, s.847)(Elbanna 2014, s.2) Det finns också en brist på studier som behandlar Scrum i stora organisationer. Fokus har istället legat på små och medelstora företag där Scrum har fått mest positiv kritik. Det finns även de kritiker som menar att de negativa aspekterna av agila metoder i slutändan blir större än fördelarna. Framförallt i stora organisationer som genomför komplexa projekt (Barlow et al. 2011, s. 26). Trots detta har få empiriska studier gjorts om Scrum och dess konsekvenser, endast 3 % av de vetenskapliga rönen om agila metoder behandlar Scrum (Lei, et al. 2015, s. 1-3).

Traditionell projektledning för mjukvaruutveckling har studerats mer och över längre tid. För- och nackdelar är definierade och erkända vilket gör det lättare att anpassa metodiken efter sina behov. För att få samma kunskap om agila metoder krävs att området studeras mer. Om Scrum ska kunna vara till så stor nytta som möjligt behöver de negativa konsekvenserna av arbetsmetoden förebyggas. Detta är endast möjligt om problemen har identifierats. Agila metoder lyckas ta itu med många av de problemen som anses som de främsta orsakerna till att mjukvaruprojekt har misslyckats och fortfarande misslyckas. Men med nya projektledningsmetoder kommer nya risker och ett företag behöver känna till dessa för att kunna hantera dem. Genom att undersöka riskerna med en arbetsmetod kan företag lära sig att undvika de problem som den kan vara upphov till, och kan på så vis ha större framgång med sina projekt.(Elbanna 2014, s.13)(Joey Cho 2010, s. 20-21)

### 1.3 Forskningsfråga

Det finns en kunskapsbrist kring Scrum och dess negativa konsekvenser. Därför vill vi belysa detta område för att identifiera och definiera vilka aspekter av Scrum som har en negativ inverkan på projekt. Vår forskningsfråga formuleras:

Vilka negativa konsekvenser kommer med användandet av Scrum?

### 1.4 Syfte

Uppsatsens syfte är att undersöka den agila arbetsmetodiken Scrum's negativa konsekvenser på projekt för att belysa ett område som tidigare förbisetts. Genom analys av litteratur och intervjuer ska vi identifiera och definiera de problem som uppstår med Scrum som arbetsmetodik.

### 1.5 Avgränsningar

Vi kommer att begränsa oss till att definiera problem kring Scrum, således är uppsatsen inte en komparativ studie där Scrum jämförs med andra agila metoder eller traditionell projektledning. För att ge kontext åt vår analys kommer dock andra arbetsmetodiker behandlas. Vi ämnar också avgränsa oss till de nackdelar eller svagheter som Scrum kan ge upphov till vid en korrekt användning. Därför kommer implementationssvårigheter och vanliga fel som organisationer gör när de använder sig av arbetsmetodiken inte att utredas.

## 2. Litteraturgenomgång

För att kunna identifiera de negativa konsekvenserna av Scrum måste vi bygga upp en teoretisk grund kring ämnet. Vi kommer att gå igenom Scrum och förklara dess olika delar och roller. Detta för att ge läsaren en utgångspunkt att se problematiken från. Efter genomgången av Scrum går vi in på de olika negativa aspekter som vi identifierar i tillgänglig litteratur. Beroende på vilket språk och vem som har skrivit om Scrum används olika termer för att beskriva arbetsmetodiken. Vi har därför gjort en ordlista där vi förklarar de mest grundläggande Scrumtermerna som vi håller oss konsekvent till i uppsatsen. Ordlistan finns i bilagan. Litteraturgenomgången avslutas med en tabell där alla problem finns representerade och hur de är kopplade till litteraturen. Det är problem som blir grunden för dessa vår intervjumall.

### 2.1 Scrum

Scrum är en iterativ arbetsmetod som bygger på att kontinuerlig anpassa, kontrollera risker och förbättra träffsäkerheten för ett projekt. Projektledningsmetoden baseras på tre grundstenar: transparens, inspektion och adaptation. Genom transparens ska alla involverade i projektet kunna se hur processen ser ut. Inspektion innebär att Scrumanvändare ständigt ska granska produktbacklogen för att hitta eventuella fel. Processen ska kunna justeras genom adaptation till de förändringar som görs om något anses vara fel. (Lei et al 2015, s. 1-3) Liksom andra agila processer har Scrum också ett tidigt stadiet av tidsestimeringar och kostnader. Men inom Scrum används features som delleveranser för produkten. Att skapa en feature är att på något sätt tillföra ett mervärde för kunden till den slutgiltiga produkten. Under en sprint tillförs en eller flera features till produkten. På så vis kan processen följas och i slutet av varje sprint, när en sprintreview hålls, granskas vad som åstadkommit. (Larson och Gray 2011, s. 596) Eftersom mervärde tillförs inkrementellt istället för att leverera hela produkten på en gång ökar kommunikationen mellan Scrumlaget och kunderna. (Schwaber och Sutherland 2013, s.7-8)

Produktägaren representerar alla olika stakeholders och ansvarar för att kundernas krav fullföljs. En produktägares uppgift består av att skriva om kraven till användarberättelser, prioritera dessa berättelser efter behov och kostnad samt lägga till dem till produktbacklogen. Kommunikation är en viktig del av rollen som produktägare. Produktägaren kommunicerar med alla stakeholders och utvecklarna och är en slags medlare eller mäklare för dessa grupper. De övergripande målen och strategierna ska också förmedlas till utvecklarna som annars endast är insatta i sina specifika uppgifter. (Larson och Gray 2011, s. 598) (Schwaber och Sutherland 2013, s.5)

---

Utvecklarna ansvarar för att realisera produkten. De jobbar i tvärfunktionella grupper där alla de kunskaper som behövs för att utveckla produkten ska finnas med. Laget ansvarar själva för vem som jobbar med vad och hur de arbetar med uppgiften. Deras uppdrag är alltså endast att genomföra alla användarberättelser de tar på sig inför och under sprinten. Antalet utvecklare i varje Scrumlag varierar men det rekommenderas att vara cirka sju stycken. (Larson och Gray 2011, s. 598) (Lei et al 2015, s. 3)

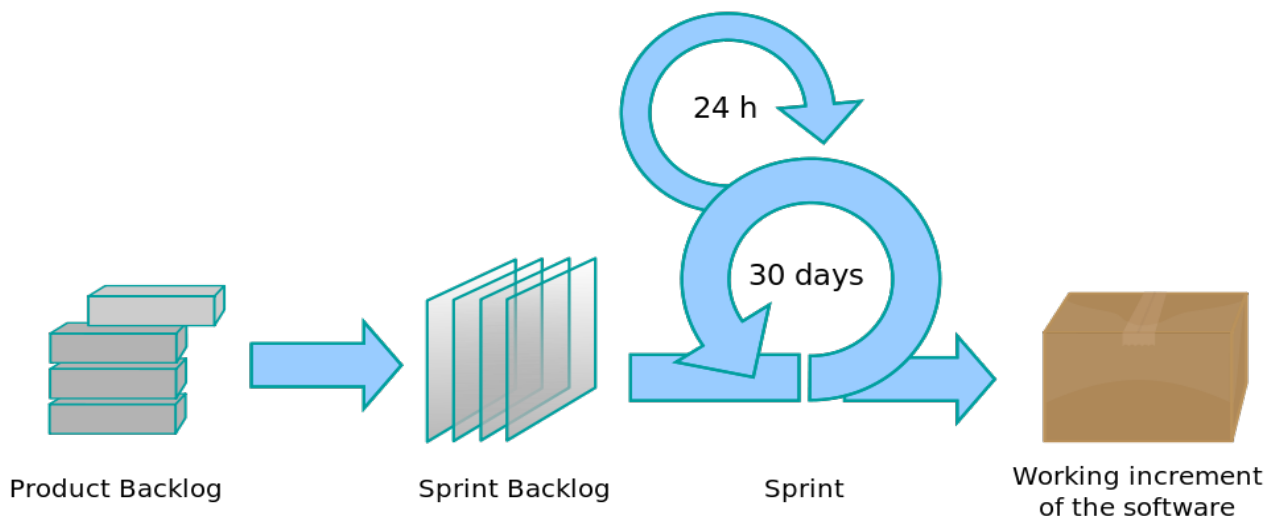
Scrummasterns roll är att underlätta Scrumprocessen och att lösa problem och störande moment för laget. Det är viktigt att poängtera att Scrummastern inte är ledaren för Scrumlaget. Uppgiften är snarare att se till att fokus bibehålls på Scrumprocessen. Det är Scrummasterns ansvar att se till att det dagliga Scrummötet och andra ceremonier genomförs under sprinten. Scrummastern fredar utvecklarna genom att facilitera Scrumns olika moment samt vara den personen som får störas i Scrumlaget. På så vis ska utvecklarna endast fokusera på värdeskapande processer. (Erik W. Larson och Gray 2011, s. 598)

Produktbacklogen kontrolleras av produktägaren. I produktbacklogen finns alla kundkrav, funktioner och förbättringar representerade som features i en prioriteringslista. Kundkraven ska vara avklarade för att produkten ska anses vara färdig. I produktbacklogen finns även estimeringar på hur lång tid varje krav kommer ta, hur länge laget har jobbat på den och hur långt det är kvar. (Larson och Gray 2011, s. 600-601)

Sprintbacklogen motsvarar det arbete som laget har tagit på sig under sprinten. Det är en lista över de uppgifter, tagna från produktbacklogen, som ska genomföras för att ny funktionalitet ska kunna implementeras till slutet av sprinten. Laget ansvarar själva över sprintbacklogen, vilket innebär att de skapar och kontrollerar den. De ger sig själva uppgifter, ser till att de blir gjorda, räknar på hur mycket tid det är kvar på användarberättelser samt bestämma om användarberättelser antingen är "inte påbörjad", "påbörjad" eller "avslutad". Innan en uppgift kan sättas som "avslutad" måste uppgiften ha blivit granskad av en annan lagmedlem. (Larson och Gray 2011, s. 600-601)

Inom Scrum håller laget dagliga Scrummöten. Detta är möten som görs stående och håller på i upp till 15 min. Här turas alla Scrummedlemmar om att berätta vad de åstadkommit sedan det senaste Scrummötet och vad det ska göra fram till nästa Scrummöte. Medlemmarna berättar även om det finns några blockeringar som gör att de inte kan jobba vidare på en uppgift. Efter Scrummötet kan enskilda medlemmar träffas och diskutera diverse problem som kan ha uppstått. (Larson och Gray 2011, s. 599)

Scrumarbetet fortsätter på detta iterativa vis fram till projektets planerade slutdatum. Efter att produkten blivit levererad börjar realeaseplaneringen för nästa projekt. Idén med releaseplaneringen är att sätta upp långsiktiga mål och planer för projektet. Här inkluderas alla medlemmar av ett Scrumlag, produktägaren, Scrummastern och utvecklarna, plus andra intressenter. (Larson och Gray 2011, s. 597-600)

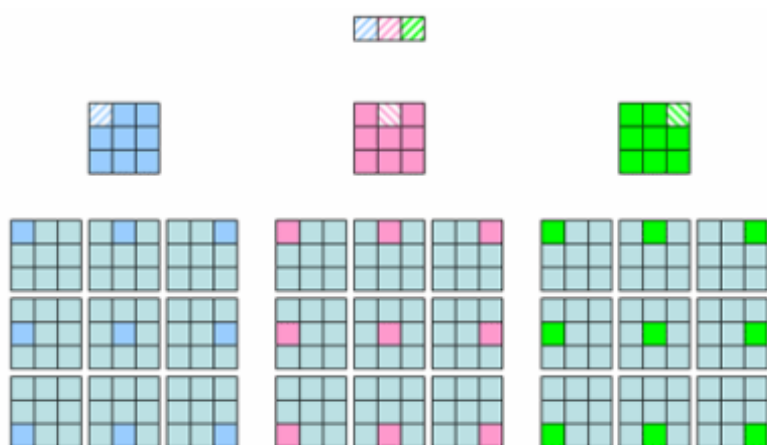


*Figur 1*

*Scrumprocessen. En del av produktbacklogen görs om till sprintbacklog som Scrumlaget sedan jobbar med under sprinten som i detta fall är på 30 dagar. Därefter släpps en prototyp. (Wikipedia 2009)*

## 2.2 Scrum i Scrum

Scrum är från början tänkt som en arbetsmetod för små upp till medelstora projekt, men efterhand som den växt i popularitet utvecklades Scrum att kunna tillämpas i större sammanhang. Scrum i Scrum är ett tillvägagångssätt för att skala upp Scrum för större projekt. Metoden bygger på att varje Scrumlag har en representant som tillhör ytterligare ett lag som har en mer övervakande och koordinerande befattning. På så vis ska produktägaren inte behöva hantera för många externa stakeholders på egen hand samtidigt som den informella kommunikationen ska fortsätta obehindrat.



*Figur 2*

*Scrum i Scrum. Varje Scrumlag har en representant i ett lag högre upp i organisationen. (Ktata och Lévesque, 2009, s.62)*

## 2.3 Kritik mot Scrum

Företag som utvecklar mjukvara måste snabbt kunna reagera på förändringar på marknaden. Dels måste mjukvaruprojekt anpassas till ny teknik, dels till ständigt uppdaterade krav från kunder. Sekventiell projektledning anses vara förändringströg och i dess ställe har agila metoder och Scrum vuxit sig populära (Pikkarainen et al. 2008, s. 304). Agil mjukvaruutveckling har dock redan från början fått kritik för de nackdelar som metoden kommer med. Året efter att *Manifesto for Agile Software Development* sammanställs, den gemensamma grunden för all agil projektledning, riktas också kritik mot vad som anses vara svaga punkter. Även om författarna ser positivt på agil arbetsmetodik finner de några scenarion där de inte anser det lämpligt att tillämpa dessa metoder. Efter detta har flera artiklar och böcker tagit upp svårigheter, främst med agila metoder, men också med Scrum. Eftersom Scrum tydligt kan härledas från *Manifesto for Agile Software Development* kommer vi se kritik mot agila metoder som kritik mot Scrum. De olika agila metoderna skiljer sig på detaljnivå och i det dagliga arbetet, men delar korta iterativa cykler, löpande kommunikation med kunder samt ständig anpassning till förändring. (Elbanna 2014, s.2-4) Elbanna argumenterar hur agila projektledningsmetoder, jämfört med traditionell projektledning, innebär nya fördelar men samtidigt kommer med nya risker. Vi sammanställer denna kritik för att kunna kategorisera tidigare identifierade risker med Scrum. Kritiken faller under vad vi har identifierat som tre huvudgrupper; tekniska problem, stora projekt och kommunikation.

### 2.3.1 Tekniska problem

#### Generella lösningar

En del i den agila idén är att försöka lösa problem på nya sätt. Enligt författarna Turk et al. ligger fokus på att lösa specifika problem snarare än generella lösningar. Därför undviks generisk kod för att främja innovativitet. Detta anser författarna vara problematisk eftersom många projekt kan spara tid och pengar av att använda design patterns eller delvis färdiga lösningar. (Turk et al. 2002, s. 45)

#### Teknisk skuld

Teknisk skuld (från engelskans technical debt) är ett samlingsuttryck som beskriver dålig mjukvaruutveckling, vilket inkluderar olika typer av icke-fullständiga lösningar. Koden fungerar men kommer i ett längre perspektiv att vara svår att hantera. Uttrycket introducerades 1992 men det är fortfarande oklart exakt vad teknisk skuld innefattar. (Tom et al. 2013, s.1499) Tom et al. försöker identifiera vad teknisk skuld omfattar i fem övergripande punkter. Kodskuld, det vill säga kod som kommer behöva refaktoreras eftersom den är svår att läsa, saknar logik eller inte följer god praxis. Arkitektonisk- och designskuld definieras som program som på grund av sin

design inte kan anpassas eller främjar vidareutveckling av funktionaliteten. Men teknisk skuld kan även komma i formen av manuellt utfört arbete som potentiellt skulle kunna automatiseras. Detta kan t.ex. ske när icke-uppdaterad mjukvara används. Om endast ett fåtal personer förstår programvaran blir detta en tekniskskuld. Saknas förståelse kring arkitektur och övergripande idéer är underhållet av koden svår. Testskuld är ytterligare en aspekt av teknisk skuld, som kännetecknas av icke-automatiserade tester av koden.(Tom et al. 2013, s.1502) Teknisk skuld ska dock inte alltid ses som något negativt, detta kan vara en utmärkt genväg för att snabbt kunna producera fungerande mjukvara. Den måste däremot hanteras för att inte bli ett kostsamt problem på längre sikt. (Tom et al. 2013, s.1498)

Då agila metoder har ett starkt fokus på funktionalitet riskerar utvecklingsmiljön att drabbas av teknisk skuld, det vill säga att icke-funktionella krav på en mjukvara som att städa i kod eller andra underhållsåtgärder förbises. (Elbanna 2014, s.10-11) Oliviera et al. menar att teknisk skuld inte går att eliminera utan endast kan minimeras till en hanterbar nivå. Samtidigt anser författarna att många agila arbetslag verkar anse sig immuna mot teknisk skuld. Scrumlag hanterar sällan detta moment trots att metoden på pappret, med sina iterativa cykler, hade kunnat vara en grund till att motverka problemet. I den snabba utvecklingen och leveransen av funktionalitet prioriteras mjukvaruarkitektur och underhållbarhet bort. Detta har visat sig vara problematisk med Scrum då ansvaret för teknisk skuld inte är självklart. Produktägaren tar sällan med icke-funktionella leverabler i sin produktbacklog samt att Scrummastern eller utvecklarna sällan prioriterar teknisk skuld i den agila miljön. (Oliveira et al. 2015, s. 50-51) Codabux och Williams pekar på liknande problematik. Teknisk skuld är något som ska prioriteras precis som andra delar av produktbacklogen. Är fokus på att leverera en fungerande produkt ska också teknisk skuld tillfälligt väljas bort till fördel för funktionalitet. Men att aldrig handskas med detta kommer leda till högre kostnader, svårighet att implementera ny funktionalitet utan att påverka andra delar samt att mjukvaran får en kortare livslängd. (Codabux och Williams 2013, s.8-9)

### Stress

Det ligger stor press på utvecklarna att uppnå de krav som de tagit på sig under en sprint. Detta gör det svårare att utveckla produkten korrekt och lämna så lite teknisk skuld bakom sig som möjligt. Men på grund av tidspressen är det många utvecklare som väljer att ta genvägar för att slutföra sina krav under sprinten. Detta medför en större teknisk skuld som behöver tas itu med i framtiden, annars kan det leda till försämrad kvalitet, brister i systemet och förseningar i projektet. Det ligger ett problem i att personer högre upp i organisationen inte vill höra tekniska aspekter och därför inte förstår att tid behöver läggas på att säkerställa en ordentlig produkt. (Elbanna 2014, s.10-11)



### 2.3.2 Stora projekt

Kritik mot Scrums och agila metoders lämplighet för större projekt har ifrågasattes från det att projektledningsmetoden kodifierades. Turk et al. argumenterar för att agila metoder blir svåra att hantera i stora projekt som involverar många stakeholders och utvecklare. De menar att traditionell projektledning med tydliga riktlinjer för dokumentation kan vara ett bättre alternativ då informell kommunikation blir svårare i takt med att projekt växer. Samtidigt poängterar de möjligheten till agilitet hos mindre grupper i projekten.(Turk et al. 2002, s. 45) Samma kritik återkommer senare och mer specifikt riktat mot Scrum. I större projekt påpekar författarna Ktata och Lévesque att kommunikationen äventyras. Eftersom den informella kommunikationen är det främsta sättet att förmedla information inom Scrum måste detta hanteras. (Ktata och Lévesque, 2009, s.65)

#### Olika geografiska platser

Problematik uppstår kring stora projekt som är utspridda i globala organisationer. Den informella kommunikationen är viktig inom Scrum, vilket är svårare att understödja i projekt där Scrummedlemmarna har olika geografiska arbetsplatser. När möten måste planeras kommuniceras det mindre inom Scrumlaget. (Turk et al. 2002, s. 43-44)

Sureshchandra och Shrinivasavadhani bekräftar vikten av att Scrumlaget är på samma plats för att det agila arbetet ska fungera smidigt. I jämförelse med vattenfallsmetoden är det svårt att distribuera Scrumlaget på olika geografiska platser på grund av vikten av den interna och informella kommunikationen. Men stora företags kompetens finns sällan på en plats och har gjort distribuerad mjukvaruutveckling svårt att undvika. (Sureshchandra och Shrinivasavadhani 2008, s. 217-218) Det är bland annat svårt att genomföra långa möten om det görs via telefon- eller videosamtal. Sprintplaneringen eller Sprint Retrospective kan ta flera timmar och faciliteras därför inte lätt med dessa kommunikationsmedel. Även språket utgör en begränsning. Alla personer behärskar sällan det valda språket till fullo. Detta kan leda till svårigheter att uttrycka sig och missförstånd mellan Scrummedlemmar. Tidszoner och kulturella skillnader utgör också hinder när distribuerade Scrumlag ska samarbeta.(Hossain et al. 2009, s. 179)

Det transaktiva minnet spelar också en viktig roll i ett Scrumlag. Detta innebär att de olika medlemmarna vet vilka kompetenser som finns hos de andra i Scrumlaget. På så vis finns det en kollektiv kunskap om vem som sitter på expertis inom olika områden. Ett väl fungerande transaktivt minne ökar produktiviteten hos ett Scrumlag men är svårt att upprätthålla i distribuerade projekt eftersom Scrumlaget sällan lär känna varandra.(Ralph och Shportun 2013, s. 11-12) Men Hossain et al. argumenterar för olika strategier för att handskas med problemen. För att Scrum ska kunna fungera i geografiskt utspridda lag kräver det förändringar eller tillägg i

---

Scrummetodiken. Det finns dock ingen allmängiltig lösning till alla dessa problem. (Hossain et al. 2009, s. 182-183) Sureshchandra och Shrinivasavadhani menar att ett Scrumlag kan spridas ut om en modell utformas för att bevara flexibiliteten och ett fritt informationsflöde (Sureshchandra och Shrinivasavadhani 2008, s. 217-218).

### Förmedling av strategi

Det finns också en svårighet att förmedla den övergripande strategin för företaget om projekten är stora. I små projekt är det relativt lätt för en produktägare att förmedla kundens behov och hur saker prioriteras. I takt med att projekt blir större tillkommer fler beroenden och den övergripande bilden blir mer komplex. Detta gör det svårare för Scrumlagen att se vart projektet ska. En gemensam idé och strategi om det slutgiltiga målet för produkten ökar Scrumlagets förmåga till lagarbete, vilket i längden leder till bättre produktutveckling. Utan strategin saknas en gemensam grund att utgå ifrån om problem uppstår eller om prioriteringar måste göras. Detta gäller framförallt i miljöer för utveckling av UX. Saknas förståelse för helheten har Scrumlaget svårt att designa ett bra gränssnitt för slutanvändare. (Ktata och Lévesque, 2009, s.62-63)(Kollmann et al. 2009, s. 17)

### Svårt att involvera underleverantörer

Ett annat problem som identifierats är de begränsade möjligheter att involvera underleverantörer. Eftersom produktbacklogen ständigt uppdateras är det svårt att ge en underleverantör precisa specifikationer över vad som ska göras. Således blir det svårt att inkorporera ett externt företag i den iterativa processen. (Turk et al. 2002, s. 43-44) Efterhand som Scrum införts i större organisationer har problematiken kring underleverantörer vuxit. Stora organisationer utvecklar sällan, tillskillnad från små, mjukvara helt på egenhand. Istället måste de ta hänsyn till många olika underleverantörer när mjukvara utvecklas. Vattenfallsmodellen har i dessa lägen varit praxis vilket gör det problematiskt om en av parterna arbetar agilt, dels för att underleverantören inte känner till metoden men framförallt för att det är svårt att matcha traditionell och agil projektutveckling. Ytterligare en aspekt av detta är att förhandlingar kan bli svåra utan tydliga tidsramar och specifikationer. (Penttinen och Mikkonen 2012, s. 195-196)

### 2.3.3 Kommunikation

#### Intern kommunikation

Även om kommunikationen ökar inom Scrumlaget kan den försämrans mellan utvecklarna och andra avdelningarna på företaget. Agil utveckling av system kan missgynna samverkan mellan olika avdelningar om de jobbar sekventiellt. Detta kan till exempel gälla en operativ IT-avdelning som inte har möjlighet att vara en del Scrumprocessen då de hanterar problem efterhand som de kommer in. Detta innebär att avdelningar som arbetar sekventiellt måste utföra arbetet under väldigt kort varsel vilket problematiserar processen. (Elbanna 2014, s.9)

Elbanna pekar på att Scrumlaget får för lite feedback på användarberättelser av andra avdelningar, t.ex. marknads- eller finansavdelningar, om arbetsbördan är hög. Då tenderar dialogen och deltagandet i utvecklarnas arbete att prioriteras bort och minska, vilket leder till att Scrumlaget får gissa sig till vilken funktionalitet som ska implementeras. (Elbanna 2014, s.12) Mjukvaruutvecklingen blir då inte samordnat med övriga avdelningar på företaget. Eftersom dialogen mot stakeholders, som kan representeras av andra avdelningar, är vital för Scrum riskerar den höga arbetsbördan att hindra arbetet genom försämrad kommunikation mellan avdelningar. (Elbanna 2014, s.13) Då Scrum bygger på att information förmedlas verbalt och informellt, vilket gör det problematiskt i stora projekt. I små eller medelstora projekt är det möjligt att hålla alla parter informerade samt föra en kontinuerlig dialog för att understödja arbetet. Men i större projekt där beroendeförhållanden mellan avdelningar är många fler är det svårt att föra all kommunikation informellt. (Ktata och Lévesque, 2009, s.65)

#### Extern kommunikation

Scrum beskrivs av Scrum Alliance (2016) som ett hjälpmedel för att förbättra och öka den informella kommunikationen. Pikkarainen et al. visar också att Scrum bidrar till bättre kommunikation både inom företaget och med externa stakeholders (Pikkarainen et al. 2008, s.332-333) Men vid större projekt tenderar produktägaren att få större roll än vad denna klarar av. Detta leder till en dålig dialog mellan produktägaren och stakeholders.(Barlow et al. 2011, s. 28) Ett stort problem med detta, som i slutändan leder till minimalt mervärde, är att produktägaren missförstår stakeholdern. (Ktata och Lévesque 2009, s.65) Produktägaren har även svårt att hantera alla olika stakeholders förväntningar och krav som kommer med större projekt. Svårigheten är att som ensam person medla mellan flera olika stakeholders som har olika och ibland motstridiga krav. Detta kan göra prioriteringen komplicerad och tidskrävande. En följd av fel prioriteringar och missförstånd är att produktbacklogen står tom i slutet av många projekt eller att onödiga features skapats. Ibland kan kraven från olika stakeholders vara så svårhanterade vara

så illa att dessa features inte alls kommer användas och då har inget mervärde skapats till kunden. (Ktata och Lévesque 2009, s.65)

### Ingen formell dokumentation

Ytterligare ett led i den iterativa utvecklingsprocessen är att inte fokusera på tunga modeller och dokumentation för att snabbt komma igång med skapandet av en lösning. Detta kan vara problematiskt vid utvecklingen av stora och komplexa system anser Turk et al. I dessa fall menar författarna att det är olämpligt att arbeta agilt, eftersom det finns en hög risk för misstag som i längden kan bli dyra. (Turk et al. 2002, s. 45) Samma kritik kring formell dokumentation i stora projekt återkommer när Barlow et al. analyserar agil projektlednings svagheter. De pekar på svårigheten med underhåll över tid om dokumentationen kring ett stort system är knapp. (Barlow et al. 2011, s. 28) Det finns också en risk för att nya konstellationer av Scrumlag skapas mellan projekt, något som ytterligare försvårar förståelsen för mjukvaran som skapats tidigare. Med lite och icke omfattande dokumentation blir det svårt att sätta sig in i logiken bakom koden. Eftersom inlärningsperioden är lång minskar ett Scrumlags produktivitet när en medlem byts ut. (Elbanna 2014, s.11) Även om en organisation inser vikten av formell dokumentation och vill öka den är det ett svårt moment att involvera det i Scrum. Att avsätta tid till att ständigt uppdatera all dokumentation i takt med hur projektet ändras från sprint till sprint skulle vara väldigt tidskrävande och kontraproduktivt. (Ktata och Lévesque 2009, s.62-63)

## 2.5 Identifierade problem

I tabellen nedan finns de negativa konsekvenser som vi lyckats identifiera i tillgänglig litteratur. Problemen är indelade i tre huvudgrupper eftersom de i vissa fall delar egenskaper. I ett fall har vi identifierat ytterligare en aspekt av huvudproblemet. Under problem två redogörs för den stress som Scrumlaget utsätts för. Detta är en delorsak till varför teknisk skuld uppstår, men stress kan också ses som ett problem i sig. Problemen extern och intern kommunikation uppstår främst vid stora projekt, men är ytterst ett kommunikationsproblem.

| Huvudgrupp              | Nr. | Problem                     | Ytterligare aspekt | Litteratur     |
|-------------------------|-----|-----------------------------|--------------------|----------------|
| <b>Tekniska problem</b> |     |                             |                    |                |
|                         | 1   | Generella lösningar         |                    | 24             |
|                         | 2   | Teknisk skuld               |                    | 2, 4, 13, 23   |
|                         | 2.1 |                             | Stress             | 4              |
| <b>Stora projekt</b>    |     |                             |                    |                |
|                         | 3   | Olika geografiska platser   |                    | 5, 16, 20, 24, |
|                         | 4   | Förmedling av strategi      |                    | 9, 10          |
|                         | 5   | Involvera underleverantörer |                    | 14, 24         |
| <b>Kommunikation</b>    |     |                             |                    |                |
|                         | 6   | Intern kommunikation        |                    | 4, 10          |
|                         | 7   | Extern kommunikation        |                    | 1, 10, 15, 18  |
|                         | 8   | Formell dokumentation       |                    | 1, 4, 10, 24   |

Tabell 1 - Identifierade problem

---

## 3. Metod

### 3.1 Metodval

Det finns två typer av empiriska undersökningar som kan utföras, kvalitativa och kvantitativa. Den kvalitativa undersökningen samlar in djupgående data, ofta med hjälp av intervjuer på ett fåtal personer. Den kvantitativa undersökningen går på bredden, ofta genom enkäter med förutbestämda frågor till många personer. Datan blir alltså inte lika djup med en kvantitativ undersökning men du får ett bredare resultat då fler personer har svarat. (Jacobsen 2002, s.92)

Vår empiriska undersökning har gjorts med hjälp av intervjuer för att skapa en kvalitativ undersökning. Detta passar en explorativ uppsats vilket vår problemställning är. Genom att gå in på djupet på några få personer får vi en klarare bild av specifika problem. (Jacobsen 2002, s.93) Vi har kunnat fördjupa oss i hur dessa problem artar sig samt hur de har uppkommit med hjälp av bland annat motfrågor och diskussioner. Målet har varit att identifiera negativa konsekvenser av Scrum och kunna visa upp mer av en helhetsbild av problemen.

Intervjuerna har skett ansikte mot ansikte. En personlig intervju blir mer givande när den sker vid ett fysiskt besök. Den intervjuade har lättare att besvara känsliga ämnen eftersom det lättare skapas ett förtroende under en besöksintervju. Detta beror på att människor får mer kontakt när de närvarar på samma plats i jämförelse med t.ex. en telefonintervju. Det finns dessutom en större risk för att den intervjuade ljuger under en telefonintervju. Telefonintervjuer lämpar sig inte för en intervju med många öppna frågor. Vi arbetar med öppna frågor och valde därför att träffa de som intervjuades fysiskt. Vi har dock skickat mail med kompletterande frågor efter att vi genomfört intervjun. (Jacobsen 2002, s.161)

### 3.2 Urval

Vi har tagit kontakt, via e-mail, med diverse stora företag som är aktiva i Lund. Det har varit viktigt att företagen och personerna som blivit intervjuade arbetar med Scrum. En annan preferens har varit att få tag på personer med olika roller inom Scrum. Anledningen till detta är att få olika perspektiv. Olika roller kan uppfatta problem på olika sätt eller ha helt olika ståndpunkter kring vad som är ett problem. För att kunna belysa de negativa konsekvenserna med Scrum måste vi behandla så stor del av arbetsmetoden som möjligt.

De personerna som intervjuats jobbar på företagen Ericsson, Axis och Qlik. De har roller som produktägare, Scrummaster, utvecklare eller linjechef. Linjechefen har egentligen inte en bekräftad roll inom Scrum utan är en roll som finns kvar sen den gamla vattenfallsmetoden användes på dessa företag. De har ett nära arbete med Scrumlagen och även högre upp i organisationen. På grund av sitt nära arbete med Scrumlagen samt att de är delaktiga i processen får de en övergripande blick över arbetet med Scrum. Detta gav oss ännu ett perspektiv på hur arbetet med Scrum funkar och de negativa aspekter som kan uppstå med Scrum.

### 3.3 Genomförande av intervju

Vid en kvalitativ intervju finns det ett behov av en förberedd struktur. Risken vid en intervju utan förberedelse är att vital information inte uttrycks av den intervjuade. Genom att ha en viss förberedelse blir det enklare att belysa de intressen som ska undersökas. Vid motsatsen riskerar det kvalitativa idealet att försvinna. (Jacobsen 2002, s.272)

Intervjuerna genomfördes i ett enskilt rum på respektive företag. Vi ville vara i ett enskilt rum för där finns få störande moment. Vi ville även vara på företaget så att den intervjuade hade nära till och från intervjun. Det är viktigt att den intervjuade inte känner sig stressad över att göra intervjun eller att för mycket tid försvinner från dess dagliga aktiviteter. (Jacobsen 2002, s.272) Genom att komma till den intervjuades kontor fick vi också en bättre förklaring av deras arbete med Scrum. De kunde visa upp deras Scrumboards och placering av laget. Detta hjälpte oss att förstå likheter och skillnader i tillvägagångssättet av Scrum på de olika företagen även om detta inte framgår i uppsatsen.

### 3.4 Intervjuguide

Med hjälp av en tabell visar vi upp de problem vi har funnit i litteraturen, problemen är kopplade till de frågor vi ställt vid våra intervjuer. Litteraturen och intervjufrågorna finner ni numrerade i bilagan. Frågorna var inte strikt begränsade till Scrumns negativa konsekvenser utan behandlade även mer allmänna områden. Detta var till exempel vara hur arbetet går till och hur stor del av företaget som arbetar med Scrum. Vi försökte ringa in företagets arbetsmetodik så väl som möjligt, dels för att täcka problemen som tas upp i litteraturgenomgången men också för att undersöka ytterligare problem. Vi skickade även mail med kompletterande frågor till deltagarna efter att intervjuerna genomförts.

| Huvudgrupp              | Nr. | Problem                     | Ytterligare aspekt | Litteratur     | Intervjufråga                      |
|-------------------------|-----|-----------------------------|--------------------|----------------|------------------------------------|
| <b>Tekniska problem</b> |     |                             |                    |                | 6.1-6.3, 7.1-7.9, 7.13             |
|                         | 1   | Generella lösningar         |                    | 24             | 6.1-6.3                            |
|                         | 2   | Teknisk skuld               |                    | 2, 4, 13, 23   | 7.4-7.5, 7.13                      |
|                         | 2.1 |                             | Stress             | 4              | 7.6-7.9                            |
| <b>Stora projekt</b>    |     |                             |                    |                | 3.1-3.2, 4.4-4.4.1, 4.9, 5.11-5.12 |
|                         | 3   | Olika geografiska platser   |                    | 5, 16, 20, 24, | 3.1-3.2, 4.9                       |
|                         | 4   | Förmedling av strategi      |                    | 9, 10          | 4.4-4.4.1, 4.7-4.9                 |
|                         | 5   | Involvera underleverantörer |                    | 14, 24         | 5.11-5.12                          |
| <b>Kommunikation</b>    |     |                             |                    |                | 4.1-5.12                           |
|                         | 6   | Intern kommunikation        |                    | 4, 10          | 4.1-4.8                            |
|                         | 7   | Extern kommunikation        |                    | 1, 10, 15, 18  | 5.1-5.10                           |
|                         | 8   | Formell dokumentation       |                    | 1, 4, 10, 24   | 4.2, 4.6                           |

Tabell 2 - Identifierade problem kopplade till intervjufrågor.

### 3.5 Datainsamling och analys av intervjuer

Datan har samlats in med hjälp av semistrukturerade intervjuer samt ett kompletterande mail för förtydligande av specifika frågor. Dessa bifogas i bilagan. De som intervjuats har haft olika roller inom Scrum för att omfatta olika uppfattningar av negativa aspekter. Intervjuerna har spelats in med hjälp av en mobiltelefonsapplikation som heter Röstmemo. Dessa inspelningar har sedan transkriberats för vidare analys och efter det kategoriserats. Alla intervjuer skedde på den intervjuades arbetsplats.

Transkriberingen gjordes på de inspelade intervjuerna för att sedan kunna bryta ner och kategorisera materialet. Det transkriberade materialet är analyserat av båda författare för att se till



att misstag inte har gjorts.

Efter transkribering gick vi över till kategoriseringsfasen. I kategoriseringsfasen delas frågorna in i kategorier som grundar sig i de åtta negativa konsekvenserna samt en ytterligare aspekt vi funnit i litteraturen. Vi skapade även en kategori till, där problem som inte påträffats i litteraturen grupperades. Vi valde att ta med ett problem som inte påträffats i litteraturen om tre eller fler av de personer vi intervjuat tog upp samma problematik. Kategoriseringen hjälper oss att visa att en särskild data behandlar samma ämnen som en annan data. (Jacobsen 2002 s.230) Att gruppera datan på detta sätt förenklar analysen för oss. Några få kategorier ger en bättre överblick. Vi kan även på ett mer systematiskt sätt jämföra de olika intervjuerna. Nedbrytningen till dessa kategorier har vi gjort med hjälp av applikationsverktyget Annotations.

### 3.6 Undersökningskvalitet

Det är viktigt att kritiskt granska en undersöknings kvalitet. Här finns tre aspekter: Har vi fått den datan vi ville få tag på, kan vi koppla datan till andra sammanhang och är datan vi samlat in trovärdig. Detta anses viktigast i en kvantitativ undersökning men även i en kvalitativ undersökning bör giltigheten (validitet) och tillförlitlighet (reliabilitet) beaktas. (Jacobsen 2002, s.256)

#### 3.6.1 Validitet

Inom den kvalitativa undersökningsmetoden finns två typer av validitet den interna och den externa. Den interna giltigheten syftar till hur giltiga resultaten är. För att mäta denna giltighet krävs en prövning av resultaten gentemot andra människor. Vi kommer närmre sanningen ju fler som håller med om resultatet. Detta kan göras på två sätt. Antingen genom att visa upp resultaten för de undersökta och diskutera ifall de håller med om resultatet. Det andra är att validera mot andra teorier, empiri och andra kompetenta inom samma område. (Jacobsen 2002 s.256-259) Vi har valt att skicka ut våra uppsatser till alla vi har intervjuat. De har fått chansen att ifrågasätta våra resultat. Vi mailade uppsatsen och resultaten med en kort introduktion där vi uppmanar de intervjuade att läsa uppsatsen, och som Jacobsen föreslår, bad vi de intervjuade att berätta om de känner igen sig i den beskrivning som rapporten ger. (Jacobsen 2002 s.257) Detta är dock bara en del av valideringen. De intervjuade är begränsade i med att de endast ser det ur sitt perspektiv och därför kan saker som den intervjuade inte känner igen sig i fortfarande räknas som giltiga. (Jacobsen 2002 s.257) Därför krävs det också att en validering görs mot den befintliga litteraturen. Vi har jämfört de resultat vi har fått med den litteratur som finns kring ämnet och funnit att det finns stöd för vår empiri i den tidigare litteraturen.

### 3.6.2 Reliabilitet

En intervju kan påverkas av flera olika typer av distraktioner. Den intervjuade blir påverkad av olika typer av stimuli och signaler. Detta innebär att den intervjuade blir påverkad av hur vi som intervjuare klär oss, hur vi ser ut och beter oss. (Jacobsen 2002, s.270) Vi har valt att klä oss i kläder som inte är allt för färgglada eller som sticker ut för mycket. Vi har också sett till att vi under intervjuerna varit lugna och försökt att hela tiden låta den intervjuade tala till punkt. Vi har även valt att vänta lite efter ett avslutat svar för att möjliggöra ett tillägg från den intervjuade. En av de viktigaste aspekterna för att få goda resultat under en intervju är att den intervjuade är komfortabel med intervjun. Det är viktigt att välja en plats som inte känns onaturlig för den som intervjuas. (Jacobsen 2002, s.271-272) Vi valde att ha alla intervjuer på de intervjuades kontor. Vi valde att utföra intervjun i ett enskilt rum för att få så få störande moment som möjligt. Vi har på förhand informerat om innehållet i intervjun och en ungefärlig längd på den. Detta har gjort att de intervjuade har haft möjligheten att avsätta tid för att bli intervjuade. Eftersom de intervjuade var informerade om intervjufrågorna och syftet har de varit förberedda inför intervjun och på så sätt givit oss genomtänkta synpunkter och tankar kring ämnet.

## 3.7 Etik

Vid en undersökning kommer vi som undersökare att ta sig oss i den privata sfären hos en annan individ och detta bör behandlas med respekt. Enligt Jacobsen (2002) finns det tre grundkrav som ska uppfyllas under en undersökning för att följa den etiska normen/regler och säkerställa den intervjuades rätt. De tre grundkraven är informerat samtycke, rätt till privatliv och krav på riktig presentation av data. (Jacobsen, 2002, s.480)

När det gäller det första kravet, informerat samtycke, innebär det att den undersökte frivilligt går med på att bli undersökt. Det inkluderar risker och vinster som deltagandet kan leda till. (Jacobsen 2002, s. 482-483) För att den undersökte ska kunna frivilligt välja att delta behöver denne också bli informerad om intervjun och dess syfte, för- och nackdelar och hur uppgifterna ska nyttjas. (Jacobsen 2002, s. 484) Vi har skickat ut ett e-mail till våra intervjudeltagare med information om uppsatsens syfte, hur vi kommer behandla informationen och möjligheten att läsa uppsatsen innan den har blivit opponerad.

Det andra kravet, rätt till privatliv, innebär att intervjun endast berör ämnet och att det finns en separation mellan intervjun och resten av den intervjuades liv. (Jacobsen 2002, s. 485) En kvalitativ studie gör det oftast enkelt att med några få fakta lista ut vem en person är även om denna inte nämns vid namn. (Jacobsen 2002, s. 485) Eftersom personen som intervjuas ska känna

sig trygg med att dela med sig av informationen är det viktigt att försäkra denna om dess anonymitet. Vi kommer därför inte att nämna namn, ålder eller boende som personen jobbar på men däremot kommer vi att nämna vilken roll som personen jobbar med inom Scrum. Detta leder till en viss brist i anonymitet men den intervjuade har blivit informerad om detta innan intervjun sker.

Det tredje kravet, krav på riktig presentation av datan, handlar om att förmedla resultaten på ett korrekt sätt. Resultaten ska förmedlas i sin rätta kontext samt så komplett som möjligt. (Jacobsen 2002, s. 488) Vi har givit de intervjuade möjligheten att gå igenom uppsatsen för att se hur deras intervju har behandlats. Det gav dem möjligheten att bekräfta att det som sades under intervjun stämmer överens med uppsatsen.

## 4. Resultat

Intervjuerna genomfördes med fem olika personer på tre olika företag. För att förstå kontexten har intervjuerna till stor del behandlat allmänna frågor. Dessa tas dock inte upp i resultatet som begränsas till Scrums negativa aspekter. Utöver de problem som presenteras i litteraturgenomgången har vi även identifierat ytterligare problem som presenteras. Nedanför finns en tabell med vilken roll och vilket företag som varje intervjuperson arbetar på.

| Referens           | Roll                       | Företag  |
|--------------------|----------------------------|----------|
| Intervjuperson IP1 | Utvecklare                 | Ericsson |
| Intervjuperson IP2 | Linjechef                  | Axis     |
| Intervjuperson IP3 | Linjechef                  | Qlik     |
| Intervjuperson IP4 | Scrummaster och utvecklare | Ericsson |
| Intervjuperson IP5 | Produktägare               | Ericsson |

Tabell 3 - Intervjupersoner

### 4.1 Tekniska problem

#### 4.1.1 Generella lösningar

Problematiken kring generella lösningar och generisk kod får lite eller inget stöd hos de intervjuade. Det finns ingen uppfattning om att Scrum begränsar möjligheten till generiska lösningar, snarare hurvida det är relevant eller ej. IP4 upplever att design-patterns och generisk kod används i den mån det går. Det finns överlag en vilja att hålla sig till beprövad arkitektur och att bygga på fungerande lösningar snarare än att göra något helt nytt, eftersom det kommer med många risker. Även IP5 berättar om krav på standardisering för att göra funktionalitet bakåt kompatibel. Det finns alltså en tydlig struktur för hur nya delar av program ska utformas och implementeras. IP2 och IP3 anser att återanvändning av kod eller design-patterns är svårt att dra nytta av i deras dagliga arbete. Detta beror främst på att funktionaliteten som läggs till är helt ny och utvecklas inte på samma vis som tidigare delar. På så vis upplevs inte generisk kod vara något som går att tillämpa i det dagliga arbetet. *“Klart vi fyller på parametrar men det är inte så att vi kan ta ut stora kodblock som används som grunden till nått som är helt nytt.”*(IP2, paragraf 89). IP1 pekar på att det finns initiativ att hantera generella lösningar för återkommande kod för att slippa lösa samma sak i olika projekt. Det är än så länge inte väl implementerat i

verksamheten och vid vissa tillfällen löser olika Scrumlag samma problem istället för att se hur andra har gjort det. IP1 påpekar dock att detta inte är ett problem som han upplever att Scrum är orsaken till.

#### 4.1.2 Teknisk skuld

Alla intervjupersoner är överens om att teknisk skuld är svårt och måste hanteras. Det är problematiskt att få med användarberättelser som handlar om mjukvaruarkitektur eller andra långsiktiga förbättringar eftersom kunden inte får någon direkt nytta av detta. Den kortsiktiga lösningen prioriteras oftast före långsiktig kvalitet. *“Har du en marknadsmässig aspekt där det handlar om pengar som ska in i slutändan och du har några som jobbar mycket mer tekniskt och är intresserade av att hålla en snygg mjukvaruarkitektur så är dem två intressena tämligen svårförenliga. Det ena kräver tid och det andra kräver snabbhet och där i uppstår en skärningspunkt som man försöker flytta till ett lämpligt ställe.”* (IP5, paragraf 80)

IP1 menar att det saknas grundläggande incitament i Scrum för att hantera teknisk skuld i förhållande till budget, tid och vad som ska göras. De tre sistnämnda restriktionerna kommer alltid prioriteras om kunden får bestämma. Således är det en utmaning att hantera teknisk skuld. IP3 pekar på liknande problematik där det är en svår prioritering mellan en ny feature och långsiktig kvalitet. IP4 ser teknisk skuld som en konsekvens av stress före en deadline. När det inte finns tid för att lösa något på bästa sätt hamnar fokus på att leverera något som fungerar. Både IP1 och IP2 anser att utvecklingsarbetet drabbas på längre sikt om teknisk skuld inte hanteras. Det blir svårare och svårare att implementera in ny funktionalitet och mjukvaruutvecklingen går långsammare.

Alla intervjupersoner är väl medvetna om problematiken kring teknisk skuld. De ser det som en balansgång där kvalitén på mjukvaran stundom måste prioriteras, trots att mervärdet för kunden är diffust. IP1 och IP5 nämner att åtgärder tagits för att hantera teknisk skuld. Detta görs av mer erfarna personer i Scrumlagen som har ett ansvar för kodens arkitektur. Enligt IP5 kan inte teknisk skuld testas bort utan måste lösas av någon som förstår koden. *“Man kan inte testa dig ur alla problem, man måste se de här logiska mönstren i mjukvaran.”*(IP5, paragraf 160)

#### 4.1.3 Stress

IP5 menar att det finns en intressekonflikt mellan produktägaren och laget. Laget har tagit på sig den mängd uppgifter de anser att de klarar. Samtidigt ser produktägaren gärna att laget gör mer användarberättelser per sprint. Men IP5 påpekar dock att det är laget som har sista ordet i frågan. IP1 diskuterar samma ämne och säger att det kommer tryck utifrån. Ingen vill välja bort någon feature. Det gör att allt försöks klämmas in i samma release. Utvecklarna hinner inte med detta

och något måste prioriteras. Då menar IP1 att saker och ting kommer att falla ur eller bli ajournerade till nästa release. Lösningen är inte att jobba övertid, det har man redan försökt i vattenfall. Problemen ligger ofta i tidsestimaten. Något tog längre tid än väntat menar IP1. Den tid de tar för ett projekt att slutföras prutas ofta ner. Produktägarna uppfattar estimaten som alldeles för höga och ber utvecklarna att korta ner sina projekttestimat. Det är en deadline som passar produktägaren bättre än utvecklarna enligt IP1.

Ett problem som IP4 uppfattar som ett stressmoment är pressen att producera features. Det finns ingen tid att lära sig djupgående kring ett ämne. IP4 menar att man endast lär sig det som behövs för att klara av implementationen men inget mer. Han hävdar att där skiljer sig teorin från verkligheten. *“De säger i teorin att man ska ha tid för att lära också men det blir låg prioriterat när det väl gäller.” (Paragraf 46, IP4)*

IP4 fortsätter att berätta att hans Scrumlag har haft problem att estimerat sprintarna. Enligt IP4 beror detta på att de inte längre är experter på ett område. Eftersom de ofta börjar på nya områden, som de inte är särskilt bevandrade i, får de svårt att estimerat hur mycket de hinner med i en sprint. IP4 tillägger att ett ytterligare problem är att de tvingas lägga för mycket fokus på här och nu, vilket gör att de inte hinner undersöka nästkommande användarberättelse. Detta hade givit dem en bättre bild av vad som skulle komma närmast och i med det hade de blivit bättre på att estimerat.

Alla intervjupersoner uppfattar att mängden arbetad övertid är mindre än tidigare. IP5 menar att man i ett tidigt skede i Scrum märker om det behövs jobba lite mer under en sprint. Detta gör att lagmedlemmarna snabbare är medvetna om de måste anpassa sitt schema för att kunna ta sitt ansvar. I Scrum är det enklare att tillsätta en till grupp att arbeta med en uppgift om det finns behov för extra hjälp. Något som IP5 menar att hans chefer är duktiga på att göra. IP3 anser att arbetsbelastningen är hanterbar i alla avseenden. IP2 menar att man känner sig säkrare i början av en sprint och att det blir stressigare i slutet men det är aldrig så att arbetsbelastningen ökar särskilt mycket.

## 4.2 Stora projekt

### 4.2.1 Olika geografiska platser

Det kommuniceras mycket mellan lag på olika geografiska platser, både hos Qlik och Ericsson. För att möjliggöra detta finns det olika verktyg så som videokonferenser eller digitala Scrumboards. Det råder olika uppfattning om hur väl detta fungerar. Det är dock ingen av intervjupersonerna som längre jobbar i distribuerade Scrumlag, utan kommunikationen sker

---

mellan olika Scrumlag eller mellan olika avdelningar. IP1 har tidigare jobbat i ett distribuerat Scrumlag och anser att det inte fungerar. *“Ja, ehm, absolut den informella kommunikationen dör, helt. Man sitter ju inte ens med varandra. Jag har jobbat med team i Indien, Polen, England, Norge och liksom försökt köra det här med olika sett. Kanske med digitala Scrumboards, telefonkonferenser, videokonferenser på standupen, Skype eller så. Om man hårdrar det så är det så jävla svårt att säga vad det är som egentligen inte fungerar. Man kan inte riktigt svara på det, men det rinner alltid ut i sanden.”*(IP1, paragraf 136)

De andra intervjupersonerna har inte jobbat i distribuerade Scrumlag eftersom deras avdelningar har helt samlokaliserade lag. De har däremot mycket kontakt med andra Scrumlag på andra platser över hela världen, bortsett från IP2. Jämfört med att ha alla Scrumlag på samma plats anser IP1, IP4 och IP5 att det är problematiskt med olika tidzoner och att den interna kommunikationen försämras. Det bidrar också till en viss ledtid när ett projekt utvecklas på olika platser. IP4 anser att det största problemet är när det skapas beroenden mellan Scrumlag på olika platser. Ett Scrumlag kan vara beroende av att ett annat lag lyckas med sin implementering, eller att man behöver vänta på att ett annat lag ska utföra ett test. IP3 har en liknande situation där andra lag har ett beroende av att dennes funktionalitet ska implementeras i tid.

IP5 och IP3 säger att kommunikationen mellan olika geografiska platser trots allt fungerar bra. IP3 menar att den dialog som sker mot andra orter någon gång i veckan fungerar smärtfritt. De olika verktygen som finns tillgängliga främjar en god kommunikation. IP5 poängterar dock att samlokaliserat alltid är bättre för dynamiken på arbetsplatsen, men att detta är nästan omöjligt att realisera i stora företag.

IP1 ser också en risk med att jobba med distribuerade Scrumlag som är att medlemmarna inte lär känna varandra. Det finns en brist på kunskap om de andra personerna i laget, synergieffekterna försvinner när man inte jobbar bredvid varandra. Det är svårare att veta vad lagmedlemmarna har utfört för arbete, vad de är duktiga på och vad de kan mindre bra. IP1 liknar det med ett lagspel *“Det är lite grand som att spela fotboll. Det går inte att spela ihop utan att titta på varandra.”*(IP1, paragraf 142)

#### 4.2.2 Förmedling av strategi

Enligt både IP2 och IP5 finns det flera verktyg för att förmedla visioner, mål och strategier. De har diverse möten inom varje avdelning där kortsiktigare mål presenteras. IP2 nämner visiontalks som förmedlas till hela organisationen. Liknande beskriver IP5 deras stormöten och Tec-days. Tec-days visas via videolänk för att ge alla tillgång och ibland körs det även lokalt. Både IP2 och IP5 menar att den här informationen är frivillig och alla kan komma åt den. Det viktiga är att fylla

---

den med tillräckligt viktig information för att folk ska vilja ta del av den menar IP2.

Produktägaren ska alltid vara delaktig i förmedlingen av strategi enligt Scrumprincipen. Desto närmre produktägaren sitter Scrumlaget desto mindre kommer han att representera kunden menar IP1. Samtidigt blir produktägaren bättre på att jobba med Scrumlaget och att representera Scrumlaget uppåt.

Både IP3 och IP4 menar att deras primära fokus ligger på den närmsta tiden som de menar är ungefär 4-5 sprintar. De får information om större visioner, mål och strategier men IP3 menar att det kan vara svårt att se den röda tråden i det vardagliga arbetet när man kombinerar de kortsiktiga målen med de långsiktiga.

#### 4.2.3 Involvera underleverantörer

Ingen av företagen jobbar med underleverantörer i den bemärkelsen att andra företag anlitas för att göra delar av mjukvaran. Däremot har Ericsson i Lund tidigare anställt konsulter för att hjälpa till i processen. Då har de integreras med ett Scrumlag. Nu hyr de istället in konsultgrupper som jobbar som enskilda Scrumlag. Enligt IP4 har det varit svårt att integrera konsulterna i Scrumlaget framförallt om de är mitt uppe i en feature. Det tar tid att lära konsulten om processen vilket gör att hela laget blir långsammare då de tar tid från arbetet. Axis jobbar med interna leverantörer och det arbetet sköts mest av produktägaren och den person som representerar produktägaren. När det krävs, vid väldigt tekniska fall, ligger kommunikationsansvaret långt nere i lagen. Detta fungerar smidigt enligt IP2. IP3 har inga erfarenheter av att jobba med underleverantörer i Scrum. IP1 och IP5 jobbade tidigare med underleverantörer, IP5 i form av konsulter som integrerades i ett Scrumlag och IP1 arbetade mot underleverantörer på flera olika vis.

### 4.3 Kommunikation

#### 4.3.1 Intern kommunikation

Alla intervjupersoner är överens om att den interna kommunikationen ökar med Scrum. Den agila miljön ger mer utrymme för dialog och deltagarna i de olika Scrumlagen är mer engagerade i samtalen jämfört med vattenfall. Den informella kommunikationen inom Scrumlagen anses vara utbredd och fungera tillfredställande. Men även mellan Scrumlag, framförallt på samma avdelning, sker det utbredd kommunikation. IP1 menar att han inte går på andra Scrumslags dagliga standup men att det utan problem går att vända sig till medlemmar av andra lag om det finns frågor. IP3 anser att folk flyttar på sig mycket för att prata face-to-face med andra lag som



det finns beroenden mot. IP2 menar att ingenjörer från olika Scrumlag uppmuntras att gå på andra lags standup när det börjar närma sig en release för att bevaka olika beroende mellan olika delar av mjukvaran.

Men det verkar finnas en tendens att kommunikationen går lite långsammare ju längre ifrån Scrumlagen jobbar från varandra. Det finns en viss ledtid mellan Scrumlag som ska ta över användarberättelser upplever IP2. Användarberättelserna kan komma sent vilket ger nästa lag dåligt med tid inför ett släpp av en produkt. Fokus ligger på sin egen sprint och då är det svårt att inkludera de som ska ta över. IP5 Anser att steget är längre om någon måste lämna sin avdelning. Här är kommunikationen inte lika utbredd, men han påpekar att det snarare handlar om att lämna sin bekvämlighets zon. Det finns också en svårighet att ha en kontinuerlig dialog från Scrumlagen upp i organisationen. IP1 menar att det finns en benägenhet att kommunikationen blir enkelriktad, det krävs neråt och rapporteras uppåt. IP1 fortsätter och menar att det finns en kunskapsbrist mellan ledning och utvecklare, att det saknas detaljkunskap högre upp i organisationen vilket försämrar möjligheterna till en bra dialog.

IP4 och IP3 pekar på en annan aspekt av kommunikationen mellan avdelningar. Andra delar av organisationen jobbar inte nödvändigtvis likadant som avdelningarna som utvecklar mjukvara. IP3 pekar på att säljare ofta måste kunna säga att det ska finnas en viss funktionalitet vid en viss tidpunkt. Det finns en långsiktighet hos företagets försäljare som skiljer sig från utvecklingsdelen. IP4 anser också det problematiskt att jobba mot avdelningar med andra arbetsmetodiker, till exempel vattenfall. Han pekar främst på svårigheten att bryta ner stora delar till användarberättelser för närliggande sprintar. *“Vi ville ha det sprintmässigt och inte hela grejen samtidigt. Utan bara den delen som behövdes för vår user-story. Men deras sätt att jobba var att göra hela featuren och kanske några grejer som vi kommer behöva nästa två-tre månader. Så det kan vara ett problem om man har olika mål.”* (IP5, paragraf 65)

#### 4.3.2 Extern kommunikation

Det kan finnas många krav från diverse stakeholders och IP2 menar att det inte alltid är den lättaste uppgiften att prioritera, men det tvingar man helt enkelt produktägaren att göra. Enligt både IP2, IP3 och IP5 är det nästan aldrig att flera krav är motstridiga, utan när det är möjligt att de kan bli gjorda vilket blir en prioriteringsfråga. IP3 menar att det kan bli roliga samtal i den bemärkelsen att de delaktiga ofta tycker olika. IP5 anser att det finns risker med att öppna upp kanaler vad gäller kommunikation från Scrumlaget och uppåt i organisationen. Då är det lätt att det kommer ner tre nya saker uppifrån uppmärksammar han.

Både IP1 och IP4 menar att det är för många led mellan utvecklarna och kunden. De menar att det finns en tröghet i kommunikationen. IP5 berättar också att det finns en ledtid från kund till

utvecklare. Men tror att det är en kvarleva från vattenfall. Trögheten gör att man inte kan få in nya krav förrän nästa delprojekt mål menar IP3. IP1 menar att produktägarna inte alltid förstår sig på vad utvecklarna gör. Ett problem med detta är att utvecklarna inte själva kan vända sig till kunden, utan många gånger behöver gå andra vägar för att få svar på sina frågor om kraven. IP4 hade velat ha en närmre dialog med den riktiga kunden för att få reda på mer om vad kunden tyckte om de features som de levererat. *“Ja jag tror det hade varit bättre, man vet vad de vill egentligen. Och sen hur den grejen vi levererat funkar. Det är bra att veta vad de var intresserade av. En feature kan ha så mycket grejer så om man pratar med dom så vet man vilken huvuddel de var efter.”* (IP4, paragraf 86)

### 4.3.3 Formell dokumentation

Ingen av de intervjuade anser att det råder en brist på intern formell dokumentation. Vid brister i kommunikationen menar IP2 att det aldrig är bristen på formell dokumentation som är orsaken. Han pekar snare på att så lite formell dokumentation som möjligt eftersträvas. *“Man vill aldrig gå mot mer formell kommunikation och dokumentation om det inte är slutkundsdokumentation.”* (IP2, paragraf 57) IP5 menar att det egentligen bara finns fokus på kundsdokumentation som går genom många led innan den godkänns. IP4 fyller i med att han tycker det finns för mycket intern formell dokumentation för tillfället, kommunikationen sker bättre informellt. Å andra sidan menar IP3 att man aldrig kan ha för mycket formell dokumentation när det kommer till kod, när någon ny kommer kan det vara bra att förstå sig på funktionaliteten. Men fortsätter att berätta att det inte heller är ett problem eftersom man sitter nära varandra. Då kan man få hjälp av utvecklare som är mer bevandrade i koden och funktionaliteten. Den informella kommunikationen överlappar den formella.

## 4.4 Andra problem

### 4.4.1 Otydlig karriärstege

IP2, IP4 och IP5 ser alla ett problem i att Scrum gör karriärvägarna otydliga. Jämfört med vattenfallsprojektledning är organisationen mycket plattare och antalet titlar är färre. Nästa steg i karriären är därför inte lika utpräglat. IP4 påpekar att alla fortsätter vara utvecklare trots att kompetens inom andra områden, till exempel ledarskap, ökar med tiden. IP2 menar att det måste finnas en företagskultur där alla ser det som något fint att vara med i ett utvecklingslag. *“Man måste se till att alla inom organisationen tycker att 'jobba inom ett Scrumteam det är det finaste som finns, vi ställer upp för varandra’”*. (IP2, paragraf 37)

IP4 och IP5 anser också att en utvecklare måste sätta stolthet i att ha lång erfarenhet, eftersom det är svårt att ta steget vidare som projektledare eller en annan position med mer övergripande

ansvar. IP4 ser gärna att det hade införts fler chefer på lägre nivå, för att en anställd ska veta var i karriären den befinner sig. Han menar också att olika ledarskapsposter går till samma personer eftersom en utvecklare inte på pappret har erfarenhet inom detta område. Anställda förblir således fast på liknande positioner även om de vill avancera eller ändra sina uppgifter. IP3 håller med om att organisationen blir plattare men ser inte det som ett problem för den individuella karriären.

#### 4.4.2 Brist expertis

Enligt IP4 skapar Scrum en miljö där alla fostras till generalister. Han menar att utvecklarna sällan kan fördjupa sin kunskap i specifika ämnen eftersom Scrumlagen ofta byter arbetsuppgifter. Han påpekar också att det enligt Scrumlitteratur ska ges tid åt fördjupning men att detta ofta prioriteras bort till fördel för ny funktionalitet och nya features. IP5 ser också en tendens till mer generell kunskap. “[...] du ska ta ett ansvar både för systemeringen och för verifieringen du kan inte bara komma in mitt i och göra en implementation och sen släppa allting.” (IP5, paragraf 37) En Scrummedlem kan inte endast sitta och programmera utan måste ta del i andra delar av arbetet. På så vis får alla en bredare kunskaper, men som går mindre på djupet.

En annan sida av det här är att fler utvecklare utan expertis ändrar i koden. IP1 menar att det kan vara farligt att fler människor redigerar kod i många olika områden. “*Din kompetens i alla de här områdena sänks ju, du får ju mer idioter som springer runt och pillar i kod som de inte riktigt fattar.*” (IP1, paragraf 56) IP1 förtydligar att även om spetskompetens försvinner finns många fördelar med bredd inom Scrumlaget. Han menar att folk tenderar att lära sig snabbare eftersom kunskap om andra domäner finns. IP3 menar också att det finns en risk i att Scrumlag tar över eller hjälper till med features från andra Scrumlag. Saknar utvecklarna kunskap inom detta område riskerar de att göra med skada än nytta. Scrum som metod tillåter inte att utvecklare kan koncentrera sig på endast en sak utan kräver ett större helhetsansvar från varje person. Det ger bredare kunskaper men expertisen inom specifika områden riskerar att försvinna.

### 4.5 Sammanfattning

#### 4.5.1 Tekniska problem

Problematiken kring generella lösningar får lite stöd hos intervjupersonerna. Det verkar inte heller finnas en uppfattning att det är Scrum som arbetsmetod som hindrar företagen från att använda sig av generisk kod. Det används i den mån det går, men ofta är projekten av olika art och lämpar sig därför inte av återanvändning. Teknisk skuld är ett problem som alla intervjupersoner är medvetna om och har olika åtgärder för att hantera. Det finns en problematik kring prioriteringen av långsiktig kvalitet kontra ny funktionalitet. På kort sikt, i förhållande till

---

budget, tid och omfattning finns det lite incitament att jobba med teknisk skuld. Men det finns samtidigt en vetskap att blicken stundtals måste lyftas för att hantera den långsiktiga kvalitén.

Stress upplevs främst kring antalet användarberättelser som ska tas med i sprint. Det finns en vilja uppifrån i organisationen att mer ska göras som ska vägas mot vad utvecklarna tror att de hinner. Sammantaget finns en viss stress kring estimering men det går att hantera.

#### 4.5.2 Stora Projekt

Samlokaliserat anses vara det optimala sättet att jobba med Scrum. De av intervjupersonerna som arbetat med distribuerade Scrumlag har slutat med det eftersom det inte fungerat. Främst på grund av att den informella kommunikationen blivit sämre. Kommunikationen mellan Scrumlag på olika platser verkar dock fungera relativt bra även om det finns svårigheter.

Förmedlingen av strategi hanteras lite olika men anses överlag vara ett problem. Svårigheter kan uppstå i produktägarens ansvar som förmedlare av framtida strategier. Om produktägaren är väldigt nära laget har den god insikt i det dagliga arbetet men har mindre koll på vad som händer högre upp i organisationen. Produktägaren kan på så vis tappa den röda tråden i det långsiktiga arbetet. Företagen har dock kommit runt detta med andra kanaler för kommunikation.

Om underleverantörer anlitas arbetar de som ett eget Scrumlag eller som en utvecklare i ett Scrumlag. Det kan ta tid att integrera och lära upp en konsult som kommer in mitt i en feature.

#### 4.5.3 Kommunikation

Den interna kommunikationen är problematisk om andra avdelningar inte jobbar efter Scrum. Långsiktighet hos andra avdelningar kan skära sig med Scrumarbetet.

Den externa kommunikationen försvåras av många led mellan Scrumlag och kunden. Företagen använder interna kundrepresentanter vilket resulterar i en viss ledtid.

Avsaknad av formell dokumentation för inget eller lite stöd.

#### 4.5.4 Andra problem

Det ses som problematiskt av tre av de intervjuade att Scrum skapar otydliga karriärvägar. Trots ökad kompetens är det svårt att avancera. Expertisen minskar med Scrum. Scrum tillåter inte att utvecklare kan koncentrera sig på endast en sak utan kräver ett större helhetsansvar från varje person. Det ger bredare kunskaper men expertisen inom specifika områden riskerar att försvinna.

## 5. Analys

Vi har i vår empiri funnit stöd för några av Scrum's negativa konsekvenser och i några fall har kritiken helt avfärdats. Många av de negativa konsekvenser som vi förhört oss om hos de olika intervjupersonerna har delvis känts igen men kan ha haft andra orsaker. Eftersom vi har genomfört semistrukturerade intervjuer har inte de olika områdena inte berörts på samma sätt med de olika personerna. Alla intervjupersoner har därför inte behandlat samma ämnen i lika stor utsträckning. Det finns också skillnader mellan rollerna, framförallt tenderar de två utvecklarna att belysa samma områden. Men även linjecheferna och produktägaren som har jämförbara roller uppfattar Scrum liknande. Ett annat moment som påverkar de intervjuades uppfattning om Scrum är organisationen i övrigt. Likt rollerna, har de personer som arbetar på samma företag pekat på liknande problem. Detta bör tas i beaktning då det kan röra sig om hur just detta företag arbetar och inte en allmän negativ konsekvens med Scrum.

### 5.1 Tekniska problem

#### 5.1.1 Generella lösningar

Empirin har visat oss att ingen av intervjupersonerna uppfattar användandet av generella lösningar i kombination med Scrum som ett problem. Det finns också svagt stöd för detta i litteraturen, den enda källa vi hittat är från 2002. Eftersom vi inte lyckats hitta mer litteratur kring problemet efter 2002 samt att ingen av de intervjuade uppfattar det som ett problem, anser vi att generella lösningar inte är ett problem att ta i beaktning.

#### 5.1.2 Teknisk skuld

Teknisk skuld är något som alla intervjupersoner är medvetna om och beskriver som en svårighet. Det stora problemet ligger i att Scrum inte har en tydlig roll som har ansvar för att ta hand om detta område. Det krävs därför en medvetenhet hos de som jobbar med Scrum att ibland flytta fokus från här och nu till framtiden för att förhindra att den tekniska skulden inte ska byggas på för mycket. Codabux (2013) menar att teknisk skuld ska prioriteras som allt annat, om det inte görs tenderar det att bli svårare att implementera ny funktionalitet samt att livslängden på produkten blir kortare. Intervjupersonerna uppfattar det svårt att prioritera kvalitén eftersom det är de kortsiktiga lösningarna som ger värde till kunden.

Elbanna (2014) skriver om att det finns stor press på utvecklarna att utveckla features och att kvalitén blir bortprioriterad på grund utav detta. Intervjupersonerna uppfattar också detta som ett problem. Som konsekvens har de olika företagen inrättat egna ansvarsområden för att täcka den tekniska skulden. På så vis hanteras problemet men endast om Scrum modifieras. Vi anser därför

---

att brist på kunskap om teknisk skuld kan skada ett företag på lång sikt. Det går att hantera men det kräver åtgärder, likt de som företagen vi intervjuat infört.

### 5.1.3 Stress

Elbanna (2014) pekar på en press på utvecklarna att producera den önskade funktionaliteten under en sprint. Scrum har fokus på att leverera mervärde till produkten, allt annat blir sekundärt. Stressen resulterar i hastiga lösningar med sämre kvalitet eller förseningar på grund av dåliga estimeringar. Detta ligger i linje med vad de personer vi har intervjuat tycker, även om de uttrycker sig mer nyanserat och ofta upplever att stressen är mer sporadisk. De menar på samma vis som Elbanna att stressen går ut över kvalitén. IP1 som är utvecklare och IP5 som är produktägare beskriver en intressekonflikt mellan vad som produktägaren vill få gjort och vad Scrumlaget tycker att de hinner göra under en sprint. De båda representerar var sin sida i förhandlingarna men ser båda en viss svårighet att hitta en balans. IP4 anser det problematiskt att estimerar användarberättelser inför en sprint. Pricksäkerheten är för dålig vilket gör att arbetsbördan kan bli högre än väntat. Det är dock noterbart att IP2 och IP3 som jobbar på andra företag inte upplever stressmoment i samma utsträckning. Vi anser att stress bör ses som negativ konsekvens, inte bara som en aspekt av teknisk skuld.

## 5.2 Stora projekt

### 5.2.1 Olika geografiska platser

Ett av de stora problemen med att ha ett Scrumlag på olika platser är att det transaktiva minnet inte fungerar särskilt väl. Enligt Ralph et al. (2013) gör detta att Scrumlagen inte lär känna varandra. IP1 menar något liknande då han beskriver de tillfällen han har jobbat i distribuerade Scrumlag. Han menar att det inte går att hålla koll på sina lagmedlemmar, om det var på jobbet, vad de kunde bra och vad de kunde mindre bra. Att sitta på olika platser förhindrar den interna kommunikationen och gör det svårare att ha en informell kommunikation menar även Sureschandra (2008).

Ingen av intervjupersonerna jobbar längre inom distribuerade Scrumlag men de har mycket kontakt med Scrumlag på andra platser. De största problemen med kontakt mellan Scrumlag på olika platser enligt intervjupersonerna är beroenden och tidsskillnader. Beroenden gör att arbetet inte blir lika effektivt, de måste vänta på andra Scrumlag. Även tidsskillnaden har en negativ effekt på beroenden då ledtiden blir förlängd. När något skickas till ett lag i en annan tidzon kan processen förskjutas enligt IP4.

Om Scrumlagen är samlokaliserade anser vi att olika geografiska platser mellan olika lag är fullt hanterbart. Splittras lagen får det dock fördömande konsekvenser för samarbete och de huvudsakliga fördelarna med Scrumns krossfunktionella lag försvinner. Vi anser därför att Sureschandra och Shrinivasavadhani (2008) analys bekräftas av de intervjuade kring distribuerade Scrumlag.

### 5.2.2 Förmedling av strategi

Ktata och Lévesque (2009) beskriver att produktägaren har svårt att förmedla övergripande mål och strategier. Vi ser en tendens hos intervjupersonerna att inte koppla ihop närliggande mål och långsiktiga mål. IP3 och IP4 anser båda att informationen finns även om det kan vara svårt att se den röda tråden. Men förmedlingen i sig, det vill säga distributionen av övergripande mål anses fungera. IP2 och IP5 pekar på olika åtgärder som gjorts för att hålla Scrumlagen uppdaterade på strategin och andra övergripande mål. Alla på företaget är välkomna på stora presentationer där företagets mer långsiktiga planer förmedlas. På så vis har produktägaren har inte ensamt ansvar för distributionen av information. Problematiken som Ktata och Lévesque (2009) beskriver upplevs således inte av intervjupersonerna eftersom det finns etablerade kanaler för att förmedla övergripande idéer. Trots detta anser vi att problematiken kring strategiförmedling inte går att avvisa helt eftersom företagen åtgärdat problemet på olika vis. Scrum kan fortfarande vara en delorsak till varför det är svårt att förmedla strategier.

### 5.2.3 Underleverantörer

Alla de intervjuade har ingen eller liten erfarenhet av underleverantörer. Detta gör bedömning om detta problem svårt. Turk et al. (2002) och Penttinen och Mikkonen (2012) ser både att avsaknaden av kravspecifikationer och preciserade tidsramar gör arbetet mot Scrumlag svårare. De anser också att det svårt att matcha en underleverantör som arbetar enligt vattenfall med en Scrumorganisation. De intervjuade beskriver dock ett annat sätt att förhålla sig till underleverantörer. Istället för att låta en specifik del av koden göras av någon annan, togs konsulter in vid behov. På detta vis går företagen runt problematiken kring kravspecifikationer och svårigheten att förhålla sig till andra typer av projektledning. Däremot dyker andra problem upp, IP4 beskriver att det är svårt att integrera en konsult i arbetet om Scrumlaget är mitt i utvecklandet av en ny feature. Eftersom förhållningssättet till underleverantörer är annorlunda samt att inte alla intervjupersoner har erfarenhet av området är det besvärligt att bedöma. Av intervjupersonerna som har erfarenhet av underleverantörer beskriver enskilda konsulter eller hela konsult-Scrumlag som välfungerande. Det finns svårigheter kring involverandet av konsulter om en avdelning är mitt i en feature men i övrigt anses det fungera.

## 5.3 Kommunikation

### 5.3.1 Intern kommunikation

Intern kommunikation inom Scrum är beroende av att informell kommunikation är möjliggjord. Det är en märkbar försämring av kommunikation desto längre bort Scrummedlemmarna och Scrumlagen sitter. Något som IP2 uppmärksammar är att processen blir långsammare när lag som sitter längre bort ska ta över en användarberättelse. Eftersom den informella kommunikationen är vital när man arbetar med Scrum är det problematiskt när denna försämras. Det leder till längre ledtider och det riskerar att bli en sämre produkt. Elbanna (2013) anser att Scrumlagen får för lite feedback av de andra avdelningarna när deras arbetsbelastning är hög. Detta försämrar kvalitén på både produkt och arbete. Risken finns att utvecklarna inte gör rätt och i med det inte skapat något mervärde till kunden.

Elbanna (2013) menar även att kommunikationen mellan avdelningar blir försämrad när andra avdelningar inte jobbar med Scrum. Det blir svårare att anpassa arbetet efter olika arbetsmetoder. Detta är något som IP4 menar är väldigt svårt. När andra avdelningar jobbar med metoder som t.ex. vattenfall, där de bygger hela featuren på en gång, måste man vänta på att de ska bli klara innan man kan fortsätta med sin del om det finns ett beroende. Det blir svårare att jobba med den informella kommunikationen som Scrum bygger på när det ska samarbetas med andra arbetsmetoder.

### 5.3.2 Extern kommunikation

Ktata och Lévesque menar i sin artikel från 2009 att om det finns flera stakeholders i ett projekt, kan produktägaren få svårt att prioritera och representera alla dessa viljor. Med många stakeholders har produktägaren mer att ta hänsyn till vilket gör det svårare att göra en korrekt bedömning. Särskilt problematiskt är det om kraven är motstridiga och eftersom besluten påverkar produkten slutgiltigt. De intervjuade ser viss problematik kring prioriteringen av olika stakeholders krav, men ingen upplever detta som ohanterligt. IP2, IP3 och IP5 menar att det mycket sällan handlar om motstridiga krav i bemärkelsen att om en görs kan inte den andra göras på grund av att tekniken är oförenlig. Däremot kan tid vara en faktor som gör det omöjligt att hinna med båda två. De intervjuade beskriver också svårigheter som Ktata och Lévesque inte tar upp. IP1, IP3, IP4 och IP5 anser alla att det finns en viss tröghet i kommunikationen mellan lagen och kunden. IP1 och IP4 beskriver en problematik som kring produktägarens förståelse för vad utvecklarna gör. Därför vill de ha en snabbare och närmre kontakt med kunden för att få rakare besked kring krav på produkten. Problemen kring den externa kommunikationen kretsar alltså inte kring prioriteringar och motstridiga krav utan snarare kring ledtider och för många lager mellan utvecklare och kunden.



### 5.3.3 Formell dokumentation

Ingen av de intervjuade känner igen problemen med formell dokumentation. Barlow et al. (2011) och Elbannas (2014) kritik om att den formella dokumentationen understödjer lärande och avsaknad av denna typ av information begränsar nya utvecklare att sätta sig in i koden. Istället pekar IP2, IP3, IP4 och IP5 hur den informella kommunikationen överlappar formell dokumentation. Även om det önskas mer kommentarer i kod verkar det inte finnas någon vilja att lägga mer tid på dokumentering av arbetet. De intervjuade håller således snarare med Ktata och Lévesque (2009) som pekar just på svårigheten att förena Scrum och mycket dokumentering. Det går inte heller att bekräfta Turk et al. (2002) påstående att agil utveckling inte lämpar sig för komplexa projekt. Ericsson utvecklar tämligen avancerad mjukvara med många beroenden mellan Scrumlag och får det att fungera.

## 5.4 Andra problem

### 5.4.1 Otydlig karriärstege

Tre av intervjupersonerna uppfattade det som ett problem att karriärstegen inte är lika tydlig med Scrum. Detta problem ses av IP2, IP4 och IP5 som en direkt konsekvens av Scrum, som än så länge ingen har en bra lösning på. Jämfört med t.ex. teknisk skuld som flera författare pekar på som bekymmersamt har företagen sätt att hantera problemen som uppstår. Det är anmärkningsvärt att flera personer på olika företag uppfattar detta som ett problem samtidigt som det inte diskuteras i litteraturen. Kompetens som ledarskap översätts inte lika bra på papper i Scrum. Ledarskap förbättras över tid menar IP4 men i Scrum kan du inte klättra på ledarskapsstegen och har därför inga bevis för att du är en duktig ledare när du väl söker dig mot andra tjänster. Vi anser därför att den otydliga karriärstegen är mer av en systematiskt art som bottnar i Scrums organisering. Många titlar och poster försvinner med Scrum och Scrumlagen måste se på att vara utvecklare som något fint. Många av de andra problemen som kommer med användningen av Scrum finns också i andra typer av arbetsmetodiker och hanteras ofta sämre av dessa. Men en otydlig karriärstege fanns inte tidigare i organisationen enligt IP4. Det är således ett problem som är direkt kopplat till Scrum som metod.

### 5.4.2 Avsaknad expertis

Scrum som metod möjliggör en bredare kunskap kring utveckling, men i med ett bredare fokus försvinner expertisen kring ett specifikt ämne. Eftersom utvecklarna inte längre ges tid och fokus till endast ett område försvinner den spetskompetensen de tidigare haft. Detta är något som uppfattas som negativ av framförallt IP4. Han menar att litteraturen påstår att det ska finnas tid för att lära sig om specifika ämnen men att verkligheten inte ser ut så. Det finns ingen tid att lära

sig om nya ämnen eller specifika saker i ett ämne, fokus ligger på att göra features och att lära sig precis så mycket som behövs för att skapa featuren. Denna negativa konsekvens är därför ett tekniskt problem. Vi anser inte att generell kunskap i sig kan ses som ett problem eftersom det ger en mer övergripande förståelse. Men en organisation som inte främjar teknisk expertis kan bli uddlös. Vi tror, likt en otydlig karriärstege, att problemet är begränsat till agil projektledning, eftersom problemet beskrivs av intervjupersonerna som en förändring från tidigare. Det finns inte heller samma helhetsansvar i sekventiell projektledning som främjar generell kunskap. Åter igen är det fokus på mervärde för kunden, som är utmärkande för Scrum, som också gör att långsiktiga fördelar bortprioriteras. I detta fall väljs fördjupning bort mot ny funktionalitet.

## 6. Slutsats

*Vilka negativa konsekvenser kommer med användandet av Scrum?*

I vår studie har vi undersökt Scrum's negativa konsekvenser. Grunden har varit att försöka styrka redan identifierade problemområden som nämns i aktuell litteratur. Genom semi-strukturerade intervjuer har vi lyckats bekräfta och avvisa tidigare funna problem, men också fått stöd för att andra problem finns.

### 6.1 Avvisade problem

Vi har inte hittat underlag för generella lösningar varken hos intervjupersonerna eller i litteraturen efter 2002. Vi har inte heller funnit stöd bland intervjupersonerna för att det uppstår problem när den formella dokumentationen saknas. Snarare tvärtom då fyra av intervjupersonerna ansåg att det antingen fanns för mycket formell dokumentation eller att man inte behövde mer formell dokumentation.

### 6.2 Varken bekräftade eller avvisade problem

Eftersom ingen av intervjupersonerna jobbar med underleverantörer har vi ingen möjlighet att bekräfta eller dementera problemen. Ingen har problem med det eftersom företagen inte jobbar med underleverantörer i den bemärkelsen. De av intervjupersonerna som anlitar konsulter anser att det fungerar. Att det skulle vara svårare att förmedla strategier kan vi inte heller bekräfta. Företagen har på olika vis kommit runt problematiken genom andra kanaler för att förmedla övergripande idéer. De ser således inget problem i detta. Men Scrum kan möjligtvis orsaka ett problem utan dessa åtgärder.

### 6.3 Bekräftade problem

Teknisk skuld anses av alla intervjupersoner vara svårt eftersom Scrum per automatik inte kommer tillrätta med detta problem. Alla de intervjuade har vidtagit någon åtgärd för att kunna hantera den tekniska skulden, men anser fortfarande att det är problematiskt. Stress upplevs delvis, även om det inte ständigt är ett problem. Vi anser att stress inte bara är en aspekt av teknisk skuld utan en negativ konsekvens på egenhand. Den flyttas från tekniska problem till andra problem i tabellen.

De intervjuade arbetar idag inte med distribuerade Scrumlag där medlemmarna har olika geografiska platser, men den erfarenhet som finns pekar på att det inte fungerar. Däremot anses det vara hanterbart att kommunicera mellan Scrumlag med olika geografiska platser. Den interna kommunikationen fungerar i stora drag bra med Scrum. Det finns dock orosmoment. Framförallt

kan kommunikationen mot andra avdelningar som inte jobbar med Scrum vara problematisk. Det finns även problem kring delar av den externa kommunikationen. Delvis är prioritering av backloggen problematiskt men främst anses de många leden mellan Scrumlag och kund som negativt.

Två problem som inte vi funnit nämnda i tidigare litteratur är en otydlig karriärstege samt brist på expertis. Tre av intervjupersonerna ser den otydliga karriärstegen som problematisk och svår att hantera i Scrum eftersom arbetsmetodiken gör organisationen plattare. Bristen på expertis är också ett direkt led av hur Scrumlagen organiseras. Generell kunskap främjas före teknisk expertis vilket kan bli ett problem i längden. Vi väljer att kategorisera detta som ett tekniskt problem.

Nedan finns en tabell med alla negativa konsekvenser som vi lyckats identifiera vid användandet av Scrum.

| Huvudgrupp              | Nr. | Problem                   |
|-------------------------|-----|---------------------------|
| <b>Tekniska problem</b> |     |                           |
|                         | 1   | Teknisk skuld             |
|                         | 2   | Brist på expertis         |
| <b>Stora projekt</b>    |     |                           |
|                         | 3   | Olika geografiska platser |
| <b>Kommunikation</b>    |     |                           |
|                         | 4   | Intern kommunikation      |
|                         | 5   | Extern kommunikation      |
| <b>Andra problem</b>    |     |                           |
|                         | 6   | Otydlig karriärstege      |
|                         | 7   | Stress                    |

Tabell 4 - Alla identifierade negativa konsekvenser

## 7. Bilaga

### 7.1 Ordlista

#### 7.1.1 Roller

**Produktägare** - Ska förmedla vad olika stakeholders vill med projektet och prioritera olika krav genom en produktbacklog. Produktägaren ska maximera värdeskapandet genom att tydligt förklara övergripande mål till utvecklarna.

**Utvecklare** - Laget som ansvarar för utvecklingen av features till produkten.

**Scrummaster** - Faciliterar arbetet för utvecklarna och ser till att alla moment inom en sprint genomförs.

**Scrumlag** - Alla de personer som är med i ett projekt och utvecklar en produkt, d.v.s. Produktägaren, Scrummastern och utvecklarna.

**Stakeholders** - De som på något sätt påverkas av projektets utgång, t.ex. kunder eller ledning.

#### 7.1.2 Ceremonier

**Sprint** - Den cykel på 2-4 veckor som Scrumlaget har på sig att utveckla en del av slutprodukten.

**Dagligt Scrummöte** - Stående möte på max 15 min där gårdagens och dagens arbete går igenom. Hela Scrumlaget deltar.

**Sprintplanering** - Detta är en planering inför nästa sprint. Här ingår hela Scrumlaget där produktägaren presenterar prioriteringslistan på användarberättelserna. Efter sprintplaneringen ska det finnas ett sprint mål och en sprintbacklog.

**Sprintreview** - Under en sprintreview presenteras vad som åstadkommit under den senaste sprinten för produktägaren och ibland även kunder.

**Sprintretrospektive** - Ett möte med utvecklare och Scrummastern för att återkoppla till den senaste sprinten. Tanken är att laget ska lära sig av de problem och svårigheter som uppkommit under sprinten.

**Releaseplanering** - Detta är planering för släppandet av produkten. Här diskuteras långsiktiga planer och risker. Alla medlemmar inom Scrumlaget och andra intressenter inkluderas i denna process.

### 7.1.3 Artefakter

**Feature** - Ett mervärde för kunden som kan tillföras till produkten. Varje feature består av flera användarberättelser.

**Användarberättelse** - En nedbrytning av en feature för att göra arbetet mer överskådligt. En användarberättelse är mindre omfattande och har mer specifika krav på funktionalitet än en feature. En utvecklare jobbar med en eller två användarberättelser åt gången.

**Produktbacklog** - Innehåller alla krav och features som finns på produkten. Den förändras dynamiskt med utvecklingen av projektet och kritik från stakeholders efter varje avslutad sprint.

**Sprintbacklog** - De användarberättelser som ska bli klara under den rådande sprinten.

(Lei et al 2015, s. 1-3)(Tanner och Mackinnon 2015, s. 5-7)(Schwaber och Sutherland 2013, s.4-15)

## 7.2 Intervjumall

1. Allmänt:

- 1.1. Nuvarande titel?
  - 1.2. Hur länge har du varit på företaget
    - 1.2.1. Hur länge har du varit på nuvarande position?
    - 1.2.2. Om du varit på några andra positioner, vilka och hur länge?
    - 1.2.3. Har du jobbat med Scrum sedan tidigare?
  - 1.3. Har du tidigare jobbat med andra arbetsmetoder?
  - 1.4. Uppenbara skillnader för dig?
  - 1.5. Uppenbara skillnader i organisationen/avdelningen?
  - 1.6. Trivs du med Scrum som arbetsmetodik?
2. Hur arbetet går till:  
Berätta mer ingående om din erfarenhet av Scrum.
- 2.1. Hur jobbar ni med Scrum? Hur går arbetet till?
    - 2.1.1. Jobbar ni med Scrum i Scrum?
    - 2.1.2. Hur går det arbetet till?
    - 2.1.3. Följer ni Scrum principfast eller har ni kombinerat med andra arbetsmetodiker?
    - 2.1.4. Varför har ni valt att gör ändring? Varför har ni inte valt att göra ändringar?
    - 2.1.5. Finns det skillnader inom företaget? Jobbar olika avdelningar på olika sätt?
- Olika team på olika vis?
- 2.2. Vad är din roll inom Scrum?
  - 2.3. Har du haft olika roller?
    - 2.3.1. Skillnader mellan dessa?
    - 2.3.2. Vilken roll är svårast att hantera?
    - 2.3.3. Vad upplever du som de största fördelarna med Scrum?
  - 2.4. Finns det negativa aspekter? Vilka är dessa?
  - 2.5. Finns det någon form av förändring du skulle vilja göra?
3. Scrumlaget och kommunikation
- 3.1. Hur många ingår i ert Scrumteam?
  - 3.2. Hur ser fördelningen ut?
  - 3.3. Hur många brukar ansvara för backlogen?
4. Intern kommunikation
- 4.1. Hur brukar kommunikation gå till inom ert Scrumteam?

- 
- 4.2. Hur brukar den formella kommunikationen fungera? Finns den?
  - 4.3. Hur brukar den informella kommunikationen fungera?
  - 4.4. Hur förmedlas övergripande mål och strategier?
    - 4.4.1. Förstår man som utvecklare helheten?
  - 4.5. Hur definieras användarberättelser?
    - 4.5.1. Vem brukar involveras i processen?
    - 4.5.2. Svårigheter?
  - 4.6. Pratar ni med andra Scrumteam?
  - 4.7. Finns det någonstans där kommunikationen brister?
  - 4.8. Kommunikation mellan avdelningar?
  - 4.9. Kommunikation mellan olika geografiska platser?
5. Extern kommunikation
    - 5.1. Hur kommunicerar ni med externa intressenter?
    - 5.2. Hur många stakeholders brukar vara involverade i ett projekt?
    - 5.3. Hur prioriterar ni era intressenters krav?
      - 5.3.1. Hur hanteras motstridiga krav?
      - 5.3.2. Finns det svårigheter i att göra alla stakeholders nöjda?
    - 5.4. Förs det en kontinuerlig dialog med alla stakeholders?
    - 5.5. Hur brukar den formella kommunikationen med stakeholders fungera?
    - 5.6. Hur brukar den informella kommunikationen fungera?
    - 5.7. Finns det någonstans där kommunikationen brister?
  
    - 5.8. Hur delaktiga är era kunder i processen?
    - 5.9. Vill de vara mer eller mindre delaktiga?
    - 5.10. Förstår de varför de bör vara delaktiga?
  
    - 5.11. Har ni underleverantörer som skriver kod?
    - 5.12. Hur fungerar involveringen av dessa i processen?
  6. Återanvändning
    - 6.1. Använder ni er av generella lösningar?
    - 6.2. En del litteratur pratar om svårigheten att använda sig av Scrum vid väldigt komplexa problem, där det ofta finns en specifik arkitektur som måste användas, har ni upplevt detta?
      - 6.3. Tenderar varje nytt projekt att "uppfinna hjulet"?
      - 6.4. Anser ni att ni behöver mer struktur till planering av projekt?



- 6.5. Frestas ni av vattenfall eller någon annan form av sekventiell arbetsmetodik?
- 6.6. Dokumenterar ni ert arbete?
- 7. Resultat och estimeringar:
  - 7.1. Hur många projekt blir klara i tid sen ni började använda Scrum?
  - 7.2. Hur pricksäker brukar tidsestimeringen vara?
  - 7.3. Hur pricksäker brukar uppskattningen av antalet användarhistorier(userstories) per sprint vara?
  - 7.4. Hur påverkar Scrum kvalitén på era produkter?
  - 7.5. Hur hanterar ni technical-debt(Hastiga lösningar, snarare än "korrekt" kodning)?
  - 7.6. Hur är arbetsbelastningen?
  - 7.7. Finns det stressmoment?
  - 7.8. Är arbetsbördan jämt fördelad över en sprint?
  - 7.9. Blir det övertimmar? Hur ofta?
  - 7.10. Scrum bidrar till en platt organisation. Ser du ett problem i att det saknas en hierarkisk struktur?
    - 7.11. När man läser om Scrum får man ofta höra om vad man ska göra, men hur vet ni att ni gör rätt?
      - 7.12. Hur mäter ni resultaten?
      - 7.13. Hur mäter ni kvalitén på era produkter?
      - 7.14. Finns det resultat att jämföra med från innan ni började använda Scrum?
        - 7.14.1. Ser ni någon förändring i er arbetsmetodik?

### 7.3 Intervjuguide

Denna intervjuguide visar på vilka rader det går att läsa om de olika problemområdena för varje intervju. IP står för intervjuperson och numret vilken rad ämnet behandlas på.

|            | <b>Tekniska problem</b>      | <b>Stora Projekt</b>                | <b>Kommunikation</b>  | <b>Andra problem</b>            |
|------------|------------------------------|-------------------------------------|---|---------------------------------|
| <b>IP1</b> | 59-63, 67, 75-77             | 41, 49, 53-55, 80-84, 131, 135      | 82, 92-94, 104, 109, 123  | 19, 23, 33, 37, 55, 59, 148-150 |
| <b>IP2</b> | 76, 82, 86, 88, 94, 103      | 59, 72                              | 15-17, 45, 53-58, 60-70, 91-97  | 36, 41, 98                      |
| <b>IP3</b> | 52-59, 125, 146-158          | 32-36, 52, 59-61, 74-76, 119, 134   | 32-34, 42, 67-76, 82-87, 93, 99-101, 107-117, 134, 138, 140-144, 152, 170-180 | 28, 36, 148, 182                |
| <b>IP4</b> | 46, 93, 103-111              | 25, 33, 60, 69, 87                  | 56, 64, 75, 81-83, 99   | 35, 43, 47, 92, 115, 122-123    |
| <b>IP5</b> | 46, 52-62, 81, 103, 109, 113 | 8-18, 34-38, 44, 71, 117-119, 31-38 | 7, 11, 17, 30-42, 44-52, 64-69, 71-80, 103-105                                | 12, 20-24, 89-93, 101, 115-117  |

*Tabell 5 - Intervjuguide*

### 7.4 Intervju 1

|     |  |
|-----|--|
| 1.  | Då kan vi börja med din nuvarande titel?   |
| 2.  | Ja asså jag är en Scrummedlem, det är väl titeln.  |
| 3.  | Utvecklare?  |
| 4.  | Utvecklare.  |
| 5.  | Inte Scrummaster, inte produktägare?   |
| 6.  | Nä, precis.  |
| 7.  | Alright, hur längre har du varit på företaget?   |
| 8.  | 11 år  |
| 9.  | Hur länge har varit där du jobbar nu?  |
| 10. | Svårt att säga, det beror på vad du menar, jag har inte varit produktledare eller chef, men man har haft en hel del olika typer av roller, som Module key responsible technical lead, lite mera tekniska avancemang men inte ledarskaps. |
| 11. | Det är alltså utveckling det har varit fokus på, men på olika?   |
| 12. | Design, systemdesign, Technical Lead, rita många bollar och ringar på papper. Såna bitar också, ur ett tekniskt perspektiv.  |
| 13. | Men har du jobbat med Scrum sedan tidigare?  |
| 14. | Sedan 2008.  |
| 15. | Och det var då Scrum infördes här?   |
| 16. | Nej, jag har nog varit med i fem olika uppstarter av Scrum från helt nya konverteringar, från vattenfallsmodell till...  |
| 17. | Okej, för att du varit på olika ställen på Ericsson, man har börjat på en avdelning, och sen så....?   |
| 18. | Och sen så dör den avdelningen, och så fortsätter det så. Där vi är nu är egentligen den sista och mest motståndskraftiga mot agila verksamheter.  |

|     |   |
|-----|---|
| 19. | Okej.   |
| 20. | Så, där vi är nu är ju sista anhalten av hardcore tekniker som tycker det ska fungera på det gamla sättet.  |
| 21. | Varför tror du det är så mycket svårare på denna avdelningen, är det just bara för att folk är vana vid ett visst sätt?   |
| 22. | Just nu är det den enda avdelningen som finns kvar, haha, det finns inte de gamla NS-delarna av när vi gjorde de här Sony-Ericsson, då gjorde vi hela plattformar.  |
| 23. | Det som var ST-Ericsson ett tag?  |
| 24. | Det var det också, det var vi alla här. Då var det enbart ST-Ericsson i hela Lund. Förutom Ericsson Research. Och då, desto mer tekniskt komplext det blev desto mer vill man ha kvar ett modulägarskap, att man rör sig inom färre rader kod, i princip. Medans, och istället köra planerade överlämning mellan olika delar. Planerad integration med samordning från olika håll, samleveranser och sen försöka köra testningen efter att man har levererat. Så då gjorde man nånting och så fick nån annan fixa skiten. För det är i princip det som det blir.  |
| 25. | Vad det just det här, man vill ha det planerade, om två månader på dagen då kommer det här levereras, med exakt den här specifikationen?  |
| 26. | Det var ju fantasibilden. Tidigare har vi jobbat enligt PROPS, det är en projektmodell som Ericsson själva tagit fram. De har sålt de till andra och varit rätt framgångsrika med projektplaneringsstrukturen. Den modellen är den klassiska vattenfallsmodellen, först ska man ta reda på om det finns ett behov, man gör en prestudy, ska vi ens göra vår produktidé, sen ska man ta reda på om det är genomförbart, sen ska börja planera, sen ska man börja exekvera, i mitten av exekveringen ska man revidera den, och sen ska man göra en handover till stakeholder. Sen där efter går man in i maintenance av den här. Och alla de här faserna var, i teorin, det egentligen mest effektiva sättet man kan göra det på.   |
| 27. | Då var det fast och man visste att allt skulle vara klart, men sen i verkligheten blev en annan?  |
| 28. | Exakt, det var ju aldrig klart. Det var ju alltid förändrade krav och det var ju alltid det här går ju inte längre eller det här tänkte vi inte på och kundernas krav har förändrats. De flesta här har ju jobbat med extremt mycket saker som inte kommit ut. MBMS, som är mobile broadcasting, alltså tv i mobilen telefonen. DVBH som är en adaptation av DVBT, digital marksänd TV, för handheld istället för terrestrial, DVBS för satellit för DVBH tog de fram för att vara deterministisk för, för att din telefon skulle kunna sova och sen bara precis gå upp och för att avkoda det den var intresserad. Istället för att avkoda alla kanaler hela tiden, så bara precis i de tidsluckorna din kanal sändes för att vara en strömsnål TV-mottagare för streaming, för broadcast streaming. Det har ju aldrig hänt. |
| 29. | Tror du det har varit en konsekvens av att man har jobbat med vattenfall?   |

|     |  |
|-----|--|
| 30. | Nä, nä men däremot så blir flexibilitet att hoppa av. Till och börja med måste du planera in det, i så god tid att du inte vet om du verkligen vill ha det eller inte. Vi måste vara på tåget, så vi har lite DVBH, vi har lite jättekonstiga grejer från Vodafone, som ville ha en jättehög nivå av olika tjänster och liknande som de ville ha. När man har projekt som löper i tre fyra år så måste man vara på banan redan innan med alla kraven.  |
| 31. | Just det, ja.  |
| 32. | Och sen så när du väl går in i en planeringsfas för att säkra upp alla de resurserna som behövs för att genomföra det här, en detaljerad plan, när är vad klart, då måste du ju ha gissat marknaden två år framåt i tiden för att kunna göra den. Och i teorin är det ju jättebra, om det inte ändra sig, hehe.  |
| 33. | Trivs du med Scrum som arbetsmetodik?  |
| 34. | Ja, absolut, Scrum i sig, om man hårdrar det så är Scrum den vedertagna, accepterade implementationsmodellen av en agil eller lean utveckling. Den negativa med Scrum tillskillnad från andra agila metodiker är att du försöker timeboxa det. Det är egentligen den största skillnaden, i mellan, att du verkligen kör en timeboxing och en demo oavsett vad du är klar med eller inte. Det svåra är ju att hitta features som du verkligen kan klämma ihop, utveckla en sprint och sen levera och vara klar med något vettigt. Oftast är det ju 0.7 sprint eller 1.2 sprint.                           |
| 35. | Och den går inte riktigt att, det går inte ha en halv leverabel? Hel funkar den att leverera?  |
| 36. | Precis.  |
| 37. | Hur långa sprintar är det ni har?  |
| 38. | Oj, definiera (missförstånd).  |
| 39. | Ja veckovis eller?   |
| 40. | Ah, hur långa sprintar? Tre veckor. Tre veckor är en bra avvägning, vissa försöker köra två, det är för kort, fyra funkar men, men är lite långt. Vissa kör fem-sex. Ofta brukar det kanske bli att man kör fake-sprintar, så man säger att man är halvvägs klara efter ett sprint. Jaha, vad innebär det? Då är det svårt att dema och se att vi blev klara med allt det vi sa eftersom det är en subjektiv bedömning av vad är halvvägs. Vi har gjort hälften av alla modultester, jaha, okej? Vi har skrivit hälften av koden. Ja, okej? Det kan man inte veta förens du är klar, om det var hälften. |
| 41. | Jobbar ni med Scrum i Scrum, det här konceptet med att man har ett Scrumteam finns det en person som jobbar i ytterligare ett Scrumteam som har en mer övergripande funktion?  |
| 42. | Ja, absolut.   |

|     |   |
|-----|---|
| 43. | Det är så tanken är?  |
| 44. | När man skalar upp det så absolut. Men det är ännu värre höll jag på att säga. Det är inte riktigt så enkelt som Scrum of Scrums. Men ja. Vi har, vad ska man säga multipla loopar. vi har våra loopar och sen har vi en rapportering i ytterligare en loop och den loopen har också en planeringsloop. Kaydens kallas det, eller Relayplannig eller nånting, och då brukar man köra x antalsprints per en relay och så planerar man på en högre en nivå.   |
| 45. | Det blir liksom fler snurror på varandra? Man har en daglig Scrum och sprinten och nästa där?   |
| 46. | Precis.   |
| 47. | Men det är för att tackla den här uppskalningen av Scrum?   |
| 48. | Ja. Och för att kunna synka, var ligger vi egentligen. Framförallt är det för vi som är utvecklare. Timeboxing är jättebra, ur ett perspektiv och det är att: timeboxing är egentligen gjord för att skydda oss utvecklare från att hela tiden bli avbrutna, att hela tiden komma in och "Du vems fel är det här? Kan du kolla på detta?". Och så hamnade alltid på utvecklaren tidigare att försöka göra det också för ni skriker om det. Det som är bra med timeboxingen är att du egentligen inte får ändra planen.  |
| 49. | Man fredar utvecklaren?   |
| 50. | Man fredar utvecklaren, man skyddar dem. Det är det som Scrummatterns största arbetsuppgifter, det är egentligen att skydda teamet. Vara nån sorts gateway in i utvecklingen. Där han egentligen då ska gå in i den andra loopen i Scrum of Scrum och redovisa hur det går, "Vad är det för progress?". Men det där blir lite rörigt nu, ur ett Scrumperspektiv ska det ju alltid finnas en productowner närvarande. Och beroende på hur man delegerar productownerskapet, hur långt ner product ownern sitter, hur nära productownern sitter teamen, ju närmre teamen han sitter ju mindre han egentligen vad som ska göras, ur ett marketingperspektiv, ju sämre är han på att representera kunden. Men ju bättre är han på att representera oss uppåt. |
| 51. | Men hur ser det ut, har varje Scrumteam en productowner? Eller?   |
| 52. | ----- är ju en productowner och det brukar vara så att de har två teams och varje team driver en feature, ibland ett team på en feature ibland två teams på en feature för att öka progressen.  |
| 53. | Men det här fortsätter sen i den här Scrum i Scrum? Sen finns det en produktägare för varje sån här cykel?  |
| 54. | Om vi tar, på nått klurigt vis kommer ju programmet in i det här, och ett program är egentligen en samling av projekt. Och projekt och agilt är ju svårkombinerade men man försöker ändå göra det på en högre nivå och då är frågan vad är det egentligen för ansvar som ligger i programmen och vad är det som programmets uppgifter egentligen är i en agil arbetsmetod. Tidigare har det varit att de har sagt dem är systemkraven eller produktkraven, först bröt vi ju ner, nu pratar vi innan, vi fick ett produktkrav, det produktkravet kunde vi bryta ner i antingen subproduktkrav eller systemkrav. One-   |

|     |  |
|-----|--|
|     | liners i princip, vi ska kunna göra det här, vi ska kunna göra det här. Och sen så kunde man skicka ut de här kraven till de olika modulerna och de olika modulerna implementerade de här kraven och man försökte man integrera det och få det att funka i ett system som inte fungerade. Och där kunde projekten gå in och säga, nu har vi brutit ner allting, vi har en analysfas, och man ska liksom systemera det här och säga vad är det som behöver göras i vilka moduler och sen så kunde man tracka då, vad är progressen på dem här bitarna som vi har distribuerat ut. Och försöka få en kolla och en avrapportering på hur långt har vi kommit med det och det och det. Och nu, när man vänder på det här, det behöver ju inte nödvändigtvis vara Scrum, men att man tar ett lite mer ett ax till limpa-ansvar man jobbar med så många bitar som möjligt för att få en feature, eller vad man ska kalla det, att funka så man tar helhetsansvaret, man tar testningen, man tar högt, man tar lågt och man tar alla de bitarna som behöver göras. Man tar den slutliga designen, man gör intefacen, man gör alla de här bitarna som behövs göras. Då tar man ett helhetsansvar från start till slut.   |
| 55. | Tror du det är bättre?   |
| 56. | Det är mycket bättre. Ur perspektivet att produktkvalitet ökar, något fruktansvärt. Din kompetens i alla de här områdena sänks ju, du får ju mer idioter som springer runt och pillar i kod som de inte riktigt fattar. Och det är en farlig, det är en downside. Om du har folk som har suttit med den här koden i tio år och han vet det ska läggas till och det här. Han kommer ha världens finaste kod och kommer se ut som preussisklaghylla med alla moduler snyggt undanstoppa med strukturer i datalagringar kommer vara ordnat för den modulen. Men det betyder inte att det fungerar ihop. När man tar det ansvaret så kan du testa av det direkt, för du får själv testa alla de olika bitarna. Köra modultesterna i den modulen och köra modultesterna i den modulen och köra ett fulltest, fullnod som vi kallar det då, men att du kör ett verkligt test på något sätt. Vi har ju massor av olika nivåer av tester, ibland stubbar vi, ibland stubbar man API:er för att göra ett modultestliknande och där man bara dummieskickar svar eller pumpar en funktion med gränsvärden från högt till lågt och ser så den inte funkar. Men sen kan man köra en massa olika processer i en avgränsad miljö som man kör nästan ett halvt system, vi har något som kallas multikomponenttest, man har massor av komponenter som samverkar. Så petar du in något här och, man stubbar allt som det eventuellt det kan studsas i, och så får du ut något här som du verifierar och skickar tillbaka ett svar också excererar du en större mängd av koden tillsammans, av det här subsystemet eller vad man ska kalla det. |
| 57. | Och sen kopplar man på det på ytterligare, och så går det vidare?  |
| 58. | Och sen tillslut kör man det på en riktig nod med, med instrument som t.ex. fejkar att man är en telefon. Man kan köra 2000 telefoner och massor av olika celler och så har vi system som via dämpning och förstärkning telefonernas olika signaler fejkar en förflyttning, i princip. Den här cellen blir svar och den här blir starkare och då är det i princip att du kör en bil mellan två celler. Allt det här blir bara större och större.   |
| 59. | Men intentionen är att köra mer eller mindre samma projektledningsmetod genom hela organisationen?   |
| 60. | Nä, det är egentligen inte intentionen, utan det är motståndet i den redan befintliga organisationen som inte vill köra på något annat sätt. De dem tror att de tappar är determinism. Förutsägbarheten i när blir det här klart. Eftersom deras, om man växlar ifrån vattenfall till nån typ av agil utveckling, nästan alla av de agila utvecklingsmetoderna utgår från en backlogg, och en backlogg är en unikt prioriterad lista av arbetsuppgifter. Det ska aldrig finnas något tvivel om den arbetsuppgiften är  |

|     |   |
|-----|---|
|     | <p>viktigare än den arbetsuppgiften, ur ett backloggersperspektiv. Den måste vara unikt prioriterad. Då blir det väldigt enkelt, för deras enda styrmekanism är att ändrar prioritet i backloggen, om de vill att något annat då får de ändra prioritet i backloggen. Jag har en polare som har varit chef på lite olika ställen som har ett favorituttryck: "Att prioritera är att välja, och att välja är att välja bort, vad väljer du bort" - och det vill aldrig nån göra. För du kan bara få en, eller du kommer få x antal saker, men vilka av dessa kan du undvara. Ingen vill ju undvara något. Alla vill att allt ska finnas med. Och då försöker man skohorna in allting. vi kommer bara göra det vi hinner med och det vi hinner göra ordentligt, resten kommer att falla ur. Eller komma till nästa drop. Då måste man skjuta på det, man kan inte bara säga "Kan vi inte bara jobba lite hårdare", för det är ju vad traditionell projektledning har försökt göra. Kan vi inte bara jobba lite övertid? Och så, kan ju problemen vara by design eller att något tog längre tid än väntat. För det är alltid problemet. Om du kommer utifrån och frågar hur lång tid tar en arbetsuppgift och får svaret 12 veckor så svarar man kom igen, jag skulle kunna göra det på 4. Säger du verkligen 12? Och börjar man gå igenom vad som verkligen behöver göras, det är kanske inte så mycket, och så prutar alltid projektet ner tidsestimaten för att få en bättre deadline. Som passar deras behov bättre.</p> |
| 61. | Men är inte detta en kvalitetsrisk? Med technical-debt? Man pressar fram en lösning?  |
| 62. | Du har helt rätt, det finns inget incitament för ett projekt att skapa kvalitet, och där kommer projekt alltid säga "nä, så kan du inte säga, kvalitet är jätteviktigt". Ett projekt har en fixed budget, den har en deadline och ett scope. Han bryr sig inte ett piss om kvalitet, dom skohorna alltid in till en viss kvalitet. Men kvaliteten var under budget, scope och tid.  |
| 63. | Hur är det med övertid?   |
| 64. | Det har egentligen ingenting med arbetsmetodiken att göra. När vi jobbade här i slutet av mobiltelefon, när vi gjorde modem, ja framförallt Samsung försökte vi komma ut till. Med LTE-modemet. Det var så mycket övertid, det var bara hela tiden hela tiden. Oavsett om det var sprintar eller inte. Övertid är övertid.  |
| 65. | Men hur ser det ut idag? Är arbetsfördelningen jämt fördelad över tre veckor? Eller är det, är det tuffare i slutet?  |
| 66. | Det där beror lite på vilken direkt chef du har, skulle jag vilja påstå. Linje chefernas roll i en Scrumorganisation är också lite tvetydig. Vad gör min chef? Han som är min lönesättande chef, vad är hans roll i en sån här organisation. Han är aldrig med och är arbetsledare, han är ju själv inte utvecklare, han äger egentligen processen och Scrumteamen och ska egentligen ansvara för den långsiktiga utvecklingen av Scrumteamen. Har vi rätt kompetens osv? Är vi rustade för framtiden? Men linjen i sig, det tidigare som hände, när vi pratar vattenfallsmodell pratar vi ofta en matrisorganisation, linjen åt ena hållet och projekten åt det andra hållet. I det här så sker en förhandling, där det förhandlas, du kan få så och så många resurser inom det här kompetensområdet, så många inom det här. I Scrum försvinner det här, då får vi en prioriterad lista och har x antal Scrumteam. Varje Scrumteam drar arbetsuppgifter från den prioriteradelistan. Och sen blir vi klara när vi blir klara.  |
| 67. | Använder ni er av generella lösningar? Från sprint till sprint, om koden är till 50 % samma, kan det vara att ni använder den igen? Om det finns en grund i varje feature som är snarlik, som skulle kunna  |



|     |  |
|-----|--|
|     | använda sig av. Tittar man på Scrum överlag förespråkas inte att man använder sig av generella lösningar, utan man ska ju gärna på något sätt spåna fram en ny idé och inte hämmas av att det finns ett givet sätt att lösa den på.  |
| 68. | Du får nog definiera på vilken nivå, är det typ tasknedbrytning eller är mönster vi jobbar efter, eller är det produktkod, produktlösning?   |
| 69. | Ett givet sätt att lösa en viss typ av problem och att man återanvänder det?   |
| 70. | Nä det vill jag inte påstå att vi har, det man börjar lyfta upp lite mer i de här bitarna är något som kallas softwarecraftmanship, alltså hantverksmässiga kunskaperna om utvecklingen. Och där i ligger det mycket designpatterns och lite sånna saker. Det finns ett visst antal sätt att göra en sorteringsalgorithm, det finns ett visst antal sätt att göra länkade listor. Självklart ska man inte göra om länkadelistor varje gång.  |
| 71. | Uppfinner ni hjulet igen?  |
| 72. | Såklart försöker vi inte göra det. Och när det är så många teams så är det svårt att överbrygga vad alla gör hela tiden. Vi har precis gjort en rätt generell lösning som kallas tokenbucket som är en flödeskontrollmekanism där man lägger ner några tokens, varje gång man ska göra något tar man ut några tokens. Finns det inga tokens kvar i bucketen får du inte göra det. En flödeskontrollmekanism som används rätt ofta i telekom. Även om vi leverar detta i generisk kod så att alla kan börja använda den, hur många andra tokenbucket-lösningar har vi i systemet? Rätt många gånger uppfinner vi hjulet igen, ibland.   |
| 73. | Är Scrum i vägen för att använda sig av den här generiskakoden eller design pattern?   |
| 74. | Nä, inte Scrum per se, men börjar vi prata att man jobbar låst i en modul. I ett visst antal filer eller om du jobbar krossfunktionellt, från ax till limpa. Vilket man skulle kunna göra utan att jobba med Scrum. Men man brukar bundla det, att du vill jobba krossfunktionellt för att minimera antalet handovers. Nån ska inte göra design, lämna över det till någon som ska tolka det, göra implementationen och lämna över till någon som ska testa det. Men det är bara, det finns ju massor av köteoridelar i det här, att du vill inte ha buffrar av halvfärdigt arbete. I en fabrik vill du inte ha nån som gör vänster framskärmar, han är sjukt duktig på det, vi har sjukt många vänster framskärmar. There is no point. Det är en buffert av halvfärdigt arbete. Olackade framskärmar till en Toyota Corolla, till slut blir de obsolete och man måste slänga eller göra om. Så man ska försöka minimera de bitar som ligger och blir gamla, för att minimera waste. Men de tär ju inte Scrum per se, det ju mer Leantänket, just in time kommer in från Toyota. |
| 75. | Nånstans i slutet av en sprint lämnas det ju över en feature, hur går det till? Den här mer formella dokumentationen, dokumenterar ni? Kommenterar ni kod?   |
| 76. | Ja asså, man försöker ju gå ifrån, timeboxingen är bra, ur ett kortsiktigt perspektiv, men så fort vi är klara med någonting, ska vi ju leverera det, oavsett om det är sprintslut eller inte. När vi är klara med någonting. Säg att vi har gjort en lib-funktion som vi kommer behöva senare, så gör vi funktionen, testar den och levererar den. Sen påbörjar vi nästa del där vi använder lib-funktionen då ju tanken att  |

|     |   |
|-----|---|
|     | den ska in i Continuous integration-kedjan så ska den ligga på baseline, den rebaseas och alla ska se att den finns där. Då blir våra leveranser mindre och mindre. Och då är det inte vid sprintets slut att vi ska leverera nånting och se om det funkar utan vi siktar på att varannan dag ha nånslags leverans. men det är ju jättesvårt att göra det, det är som att serva ett flygplan i luften. Den ska aldrig landa den ska bara flyga och flyga och vi ska göra den bättre och bättre och då kommer man in på andra problem. Vad finns det för metodiker för att leverera kod på ett bra sätt? Som gör att det inte ligger en massa död kod i produkten som aldrig används.  |
| 77. | Men är det inte produktägarens ansvar att rätt saker ska göras?   |
| 78. | Productowners ska ur ett featureperspektiv representera kunden i vad kunden vill ha. Kunden har ingen känsla för kvalitet. Projektarbete, det är lite som barnarbete de har ingen känsla för kvalitet. I det här har vi en annan struktur som heter productguardians som långsiktigt är där för att driva fram en arkitektur som håller, moduler som funkar, kvalitet, mönster, tester. Kodgranskning och de bitarna för att se att folk jobbar på rätt sätt i koden. Det är ju för att lyfta upp kodkvaliteten. Produktägaren är ju intresserad av vad han kan sälja, och har ingen intresse för när ska refactorera kod. Det kan han inte sälja. Få honom att förstå att det är jätteviktigt med för att vi ska kunna fortsätta bygga vidare och lägga till mer features. Refaktoring är, som jag ser det som utvecklare, det är att göra plats varje gång du ska göra en ny feature. Rensa ut en hyllplats för att lägga in en låda, men det får man inte göra, du ska bara försöka plåstra in den funktionaliteten. Men du måste röra det annars... |
| 79. | ca 33 min   |
| 80. | Hur förmedlar man den informationen?  |
| 81. | Det är jättesvårt, det är lite där projektet och produkten kolliderar. För vi har ju nånslags projekt/program, när har lovat... här har vi cykler, en och samma produkt som lever under väldigt lång tid. Det här är multistandard basstationer, som vi ska kunna köra 2G, 3G, 4G på en och samma station men vi bryr oss bara om LTE, det är ju då vad vi jobbar med. Då släppte vi 15a, typ vid nyår 2015, så ska vi ha en drop, dvs runt jul eller så. Då ska vi ha något som är GA färdigt, General Availability, då ska det vara testat och ha ett scope som är testat. Sen kommer 16a, 16b osv.   |
| 82. | Hur förmedlas detta då (övergripande mål)? Jag gissar ni har dagligt Scrummöte där man går igenom taskar man håller på med just nu?   |
| 83. | Framförallt är det ju burndownen, som blir tydligt visualiserad, sen är det operational product owners som går ut och Scrummasters går in i Scrum of Scrums och där finns det flera stakeholders representerade. Och progressen är betydligt enklare att se, ni såg den feature-anatomin vi har av userstories (refererar till innan intervjun satte igång) de gröna rutorna som var kopplade till varandra där vi försöker visualisera så fort vi hittar "det här måste vi göra eller det här måste vi göra", popularer vi en anatomi som är beroende av, ur det tekniska beroendet, identifiera olika värdesteg. Om vi gör den här kedjan kan vi göra det här, om vi gör den här kan vi enable det här i produkten eller få andra att testa. Men den förmedlingen går egentligen via Operational product owner, OPO:n.  |
| 84. | Som då är ännu högre upp än produktägaren?  |

|     |   |
|-----|---|
| 85. | Nä, nä, nä.... ska jag rita?  |
| 86. | Ja.   |
| 87. | Vi har ju en sjuk massa backlogs, vi har inte bara en backlog, utan det finns många olika stakeholders. Vi har en backlog för capacity, som handlar om....  |
| 88. | Ert team?   |
| 89. | Nä, nä organisationen. Vi har capacity, som är hur celler kan vi driva på en basstation, vilket är en jätte viktigt salespoint. Vi kan säga det finns olika requirement areas, varje requirement area har en area product owner.  |
| 90. | Och detta är de väldigt övergripande featuresarna?  |
| 91. | Ja, det är högsta nivån av features. Sen har vi, här finns massor med olika såna (requirement areas). Det är Capacity, MVB och det är mobile broadband. Och då har den features för mobile broadband, inte maintenance osv. En area product owner äger sin backlog och här kommer det lite kluriga. Här har vi ett program (organisatorisk program) som egentligen inte passar in i den här strukturen. Han trackar en backlog kan man säga, som dras ner till ett Scrumteam. Vi kallar det inte för Scrumteams utan för Crossfunctionalteams, och här finns det en operational product owner som egentligen ska gå som en förlängd arm från APO:n och sen så har vi en Scrummaster som är entrypointen i Scrumlaget. En OPO har två Scrumlag(XFT), och varje XFT har ett backlog. Det är ju inte säkert att vi drar ur samma backlog. Och det är här den agila biten kommer in. Program manager är mer eller mindre en projektledare men de trackar progressen på en backlog och rapporterar upp. Det ska ju egentligen den här killen göra (pekar på tavlan, OPO).  |
| 92. | Den interna kommunikationen är alltså lite delad?   |
| 93. | Den är lite difus och frågan är var ansvarsrollerna ligger. Programmet är egentligen en kvarleva av old-days of working. Då har man försökt har ett program (manager?) som asvarigt för x antal program, så drar man ihop dem. Då har vi tex 16b som vi försöker få ihop till midsommar. Då har vi massor av olika features i de här olika backloggarna som vi försöker tracka till 16b. 16b är ju en viktig grej för oss, då har vi lovat våra kunder att ni ska få det här, och det här kommer att funkar. Så här kommer en matris in, vad har vi lovat de olika kunderna till de olika tidpunkterna och hur långt har vi kommit nu. Skillnaden mellan då och agilt arbetssätt är ju att vi måste jobba mer intimt med de riktiga kunderna. Så våra kunder ska ju egentligen vara lovade det här och det här, men däremot ska vi försöka sammarbeta med kunden så vi kan få fram det kunden verkligen vill ha. Det försöker vi göra med tidiga leveranser, early deployment, continous deployment, vi har ju vissa nyckelkunder där vi går ut och, inte bara vid 16a eller 16b. Det kan ta 3 veckor från det att vi levererar till det är deployat på en riktig basstation i texas. |
| 94. | Har ni kontinuerlig kontakt med kunder?   |

|      |  |
|------|--|
| 95.  | Definiera kontakt, haha. Vi strävar dit mer och mer. Tanken är att de här APO, de ska ha super bra kontakt med kunden, de ska verkligen veta vad kunden vill ha.   |
| 96.  | Är det en slags indirekt kontakt med kunden (via APO:on)?  |
| 97.  | Nån slags representativ kontakt.   |
| 98.  | Det är ingen operatör som kommer hit och tittar?   |
| 99.  | Nä, nä!  |
| 100. | Så det finns en tröghet i kontakten?   |
| 101. | Ja, absolut, jättetrög. Samma problem här som tidigare. Vi gjorde en feature (Förklarar en feature på tavlan). Vad KAN vi göra? Vad är det kunden har bett om här? Då måste man gå till en solutions architect inne hos operatören som kan förklara vad som önskas. Då kan vi säga, men du, ni kan konfigurera på det här sättet, då kan ni uppnå detta på samma sätt, så att man gemensamt bollar fram det som verkligen är den kortaste vägen till det som kunden faktiskt vill ha.  |
| 102. | Är detta så tekniskt svårt att det är omöjligt för kunden att komma med vettiga förslag?   |
| 103. | Ja, samt att de här,(pekar på tavlan på APO) inte fattar det här(pekar på utvecklar team).   |
| 104. | Med storleken kommer olika kunskapsnivåer som gör kommunikationen trög?  |
| 105. | Även komplexiteten i produkten. Alla kunder har olika sätt, när vi deployar en basstation. Skjuta upp den på väggen, fota en QR-kod på basstationen via en androidapp, trycka iväg det till ORM-nätet och ORM-nätet ska då konfigurera upp via konfigurationsfiler väcka noden via IPsec, per automatik, SFTP:ar hem alla initiala konfigurationerna som man behöver och sen vara uppe och snurra. Tanken är att vi slippa många olika handovers. Det ska bara vara att smälla upp stationen med nån förkonfiguration. Detta blir helt absurt med LTE i de större städerna. Jag var i Seoul i tunnebannan. Kollade upp i taket, det är ett kluster av antenner överallt. Här går en operatör(ritar), som går kors och tvärs. Ska man då smälla upp en ny station för att täcka upp dålig täckning. Då måste man ju stuva runt gamla stationer. Då vill man forma om hela dispositionen av antenner. Då har vi massa olika protokoll för att hantera detta. |
| 106. | Men det var ju inte detta, det var ju detta med kundkontakt. Vem är kunden? kunden är ju inte en singularitet. Om du går till Sony så har du ju 3000 personer där inne som är intresserade av vad vi gör. Men vem ska man lyssna på?   |
| 107. | De har olika idéer om olika saker?   |

|      |   |
|------|---|
| 108. | Absolut, den ena är ansvarig av att se till att dom får in alla våra leveranser. Sen har du någon som är ansvarig för att de ska kunna sälja någonting och vara klara i tid. Han är ju intresserad av ha ett bra samarbete med oss. Men han som bara är intresserad av att följa en tidtabell och se till att underleverantören levererar, han är ju inte lika intresserad av att ha ett samarbete där vi börjar förhandla om scope och vad är det egentligen vi ska göra? Och sen nästa som är kvalitets ansvarig, han är ju bara intresserad av att det som kommer in fungerar.   |
| 109. | Ni har ju ett Scrumteam och jag gissar att det fungerar som de flesta Scrumteam, ganska informellt, man kan ställa frågor, ni hade det här med två knappar per task (refererar till Scrumboarden)   |
| 110. | Exakt täcka upp lite för varandra.  |
| 111. | Precis, men hur funkar, kan man gå till andra teams? Gör man det?   |
| 112. | Ja, eller nä, jag skulle inte säga att man går till andra team, man går till andra individer. Jag går ju inte till ett annat Scrummöte och presenterar mig och mina problem. Men de informella vägarna.   |
| 113. | Och det är accepterat?  |
| 114. | Absolut, absolut men många andra delar i... En tanke som de har jobbat med är att ju tidigare du får ut en ny feature ju mer värd är den. Om ett halvår har alla det. Desto tidigare du kommer ut med saker och ting, desto bättre. Så ur det perspektivet kan det vara lönt att istället för att ha två teams som jobbar med olika features, som blir klara. Då kanske det är lönt sätta två teams på samma feature, även om vi inte blir dubbelt så snabba, så får vi ett litet försprång. Även om vi får ut mer produkt på det andra sättet är värdet högre att vara snabb. Då kan det vara bättre att vi trampar varandra på tårna, ur ett värdekjedjersperspektiv. När man gör det får det andra positiva effekter, man ökar de informella kontakt vägarna inom teamet. Man är tvungen att samarbeta mellan de olika teamen. |
| 115. | Hur välj era team?  |
| 116. | Vår uppdelning är helt strikt så att en linje har två teams och en OPO idag. Min linjeförman har två team och typ sju personer i varje team, och en produkt ägare.  |
| 117. | Men hur sammansätts teamen?   |
| 118. | Vårt estimat var att vi ska kunna vara klara med shared-ram feature vecka 40, två olika operatörer ska kunna dela på alla resurserna i en cell. Dvs varje tidslucka ska vi kunna dela 70/30, så allt ska kunna dela. Så ska vi med fairness kunna styra över trafiken. Och då, äh vecka 40 går inte, vecka 25 är lagom. Sen kommer en massa 5G features. Vi måste banka ihop alla dem här features. Då måste man stuva runt rejält (i organisationen), och det är här flexibiliteten i det agila kommer in. Man kan börja klippa och ändra eller om ----- kommer med strategiska beslut som säger att vi ska gå mer åt detta hållet. Då måste allt det här räknas om. Vi behöver egentligen inte göra nya backloggs, men vikterna i dem behöver göras om. Och helt plötsligt får vi en helt annan prioritet.                      |

|      |   |
|------|---|
| 119. | Men byts teamen då också?   |
| 120. | Nä, nä vi försöker hålla en viss kontinuitet i teamen. Pga vi måste byta till 5g features, pga förändrade prioriteter vill vi bli klara med det här, vi vill inte bara kasta det här. Vad kan vi göra för att blir klara i tid? Och vad kan vi göra för att gå från vecka 40 till 25. Vi saknar de här och de här bitarna. Så fick vi låna in tre killar, låna in de här och den här kompetensen. De kunde en jätteavancerad simulationsprogram som kunde göra simuleringar på riktig hårdvara, för att se om invärden om ger samma utvärden. Dem gör det i en simulator och vi ska se till att verkligheten också generera samma värden. Men eftersom vi inte kunde detta fick vi låna in lite kompetens för det, lite andra förstärkningar och även en product guradian som var väldigt duktigt på detta.   |
| 121. | Är det Scrummastern som fattar detta beslutet då?   |
| 122. | När, nä det är OPO:n. egentligen är det ju så att Scrummastern inte har en ledande roll utan. En Scrummaster ska inte vara en teamleader. En Scrummaster ska vara kunnig i det agila, ska vara kommunikativ utåt mot stakeholders, när de kommer och frågar så ska det bara vara att han man ska störa då, det är en del av jobbet, att bli störd. Samt att han får gärna vara en facilitator av alla ceremonier ett Scrumteam ska ha. Så att vi håller retrospektive, så vi håller sprintplanering, vi måste förbereda nästa sprint via backlogrefinement, så att vi verkligen groomar backlogen ordentligt. Det är alltid det svåra. Att grooma nästa sprint features eller userstories kommer alltid slå på den här sprintens commitment. Ska vi göra det före vi påbörjar på nästa sprint? Avvägninen är svår. Speciellt om man bara trycker ännu mer, "Kan ni bli klara med ännu mer snabbare? Ja men då blir det ostrukturerat nästa sprint". Så där sker alltid en avvägning. Beroende på vilken managemet stil, skulle jag vilja påstå, så har jag märkt enorma skillnader mellan... det finns ju ett sunt förnuft i den agila utvecklingen, som det andra, gammal projektledning är lite mer som marxistisk planekonomi, si och så mycket resurser behöver vi för det här och då funkar det i teorin. Men i verkligheten ser inte ut så, då börjar det flyttas runt, det blir missvisande och man får, vad ska man kalla det, orättvisor, man är överbemannade på vissa delar och saknar andra delar. Det är där den adaptiva biten måste komma in och där man känner av om man har för lite eller för mycket folk, om saker tar längre eller kortare tid. |
| 123. | Finns de en dialog som går på båda hållen? Det kommer inte bara uppifrån?   |
| 124. | Det där är svårt. Den där (pekar på tavlan mellan OPO och linjechef?) är för enkelriktat. Det krävs neråt och rapporteras uppåt.  |
| 125. | Det är inte en dialog?  |
| 126. | Nä, det är inte ett samarbete där man försöker nå bästa resultat. men lite gran tror jag han där uppe inte förstår vad vi gör här nere, vi har ett lite för stort avstånd. Han sitter med slips och käkar middagar, hehe, lite åt det hållet. Och har lite olika tankar. Medans OPO:n är mycket närmre oss.   |
| 127. | Dialogen fungerar alltså längre ner, närmre utvecklarna medan men efter det blir det ett hopp?  |

|      |  |
|------|--|
| 128. | Yes, när man börjar prata marknad eller så. Men det finns samtidigt sjukt mycket duktigt folk som sitter där uppe. Men i alla de här detaljerna har de ju inte koll.   |
| 129. | För att det är omöjligt?   |
| 130. | Precis, i alla dessa (produktbackloggar för utvecklingsteam), finns det ju kärleksbarn som de verkligen är jättemåna om och har en tät dialog om. Sen vissa features som förblir lite sladdisar.   |
| 131. | Hur funkar det om man försöker involvera nån underleverantör?  |
| 132. | Nu har vi ingenting, inte vad jag vet eller jobbar med. Tidigare har vi ju jobbat med outsourcing, offshoring, Multi site team, Cross site team.   |
| 133. | Det var fortfarande Scrum då?  |
| 134. | Det försökte vi.   |
| 135. | Var det mer problematiskt med informell information?   |
| 136. | Ja, ehm, absolut den informella kommunikationen dör, helt. Man sitter ju inte ens med varandra. Jag har jobbat med team i Indien, Polen, England, Norge och liksom försökt köra det här med olika sett. Kanske med digitala Scrumboards, telefonkonferenser, videokonferenser på standupen, Skype eller så. Om man hårdrar det så är det så jävla svårt att säga vad det är som egentligen inte fungerar. Man kan inte riktigt svara på det, men det rinner alltid ut i sanden. Utan, det som funkar det är att man sitter verkligen så här, co-located, vi sitter, alla våra team sitter bara från den korridoren och bortåt. Sen har vi linjeförfaren och OPO:n bredvid oss.   |
| 137. | Varför vill man köra crosssite då?   |
| 138. | Det vill man inte. Man tror i perioder att man behöver det. Förut när man har haft modul ägarskap, och så försöker man göra krossfunktionella team, så har vi haft SIM-moduler. Då har man ett API här upp och sen en Android-applikation, hela vägen ner till hårdvaran. Det här var några killar i Lund (pekar på tavlan), detta var några i Beysinstock(?), det här var vi i Lund. Sen ska man försöka göra nått i det här, så blir det jättekross site. Det här har hänt överallt, om man tar alla de här lagerna för att genomdriva en feature i produkten, behöver du alla de här lagerna, det ska uppdateringar, för att det ska hanteras i alla lagerna. Hela vägen från vad vi kallar verkligheten till hårdvaran, då blir det lätt att man såg kompetensen på olika ställen och försökte samla ihop till ett Scrumteam. Utan kontinuitet, utan fysisk, vad ska man säga face-to-face kommunikation. Så sprintplaneringar blev helt.... |
| 139. | Det håller helt enkelt inte?   |
| 140. | Nä, det gör det inte. Sen tappar du de verkliga synergieffekterna av att jobba nära. Det är ju inte så att man har koll på om någon är borta. Ännu mindre vad han gjorde igår. Och när han frågar mig om   |

|      |  |
|------|--|
|      | hjälp (om vi är på samma plats), så fattar jag var är han, vad kan han, vilken, vad är han duktig på, vad är han svag på och hur kan vi hjälpa varandra.   |
| 141. | Det blir mer lagspel?  |
| 142. | Ja, absolut, det är lite grand som att spela fotboll. Det går inte att spela ihop utan att titta på varandra.  |
| 143. | Kan man mäta resultat? Mäter ni resultat?  |
| 144. | Ouch, svår fråga.  |
| 145. | Kan man mäta att det går bättre nu än utan Scrum?  |
| 146. | På toppnivå kan vi mäta det, och framförallt, kan vi mäta felhantering. En kund hittar ett fel, rapporterar in ett fel och så får dem en fix i den riktiga versionen produktkoden. Där har du en ledtid, det är inte ovanligt att vi pratar ibland... tremånader är kort, sex månader är rätt bra och två år rätt dåligt. Det är ju inge roligt och sitta och vara kund då, och anmäla fel och få svar två år senare och få "Vi har nästan löst det, det borde inte uppkomma så mycket nu". Där är vi ju jätte, där kan man se de absolut största skillnaderna i framförallt den här krossfunktionaliteten. Om du ska försöka planera en fix alla de här olika lagerna måste du koordinera och planera om.(Vattenfall) |
| 147. | Avbrott pga telefonsamtal.   |
| 148. | Mätningen av velocityn är sjukt svår. Riktigt svårt. Enkelt i teorin, man gör nån övergripande point estimering som man ska kunna återanvända i den yttreloopen, Scrum of Scrums eller ett relay planering, Kaydens planerings perspektiv och se om ett team estimerar på det här sättet så borde vi klara åtta månader. Teorin okej, men rörigt i verkligheten. Hur vet du hur snabbt du gör något du inte vet hur man gör. Det säger sig självt, det kommer bli ett jädra estimat nänstans.  |
| 149. | Då tackar vi för intervjun, eller hade vi något mer? Om du har något att säga?   |
| 150. | Jag kan hålla på länge, hehe. Nä, nä jag har inget mer att tillägga. Det är rörigt och stort det här. Man kan ju gå in på verkliga detaljer och, eller upp till jättestora övergripande problem. Men generellt är det i små team, där man har en produkt som är väldigt begränsad, då är det säkert sjukt enkelt. Jag gillar inte den här timeboxingen, jag vill hellre ha ett löpande, där man försätter beta av, oavsett om man kallar det sprint eller inte. Vi fortsätter beta av och gör så mycket vi kan så effektivt som möjligt, för att komma upp i hastighet. Sen om det är tre eller fyra månader. Sen om vi kallar det sprint eller inte...  |
| 151. | Slut.  |



#### 7.4.1 Kompletterande frågor till intervju 1

Hur ser du på att Scrum gör organisationen platt? Att det saknas hierarkiskstruktur? Är det svårt att se sin egen karriärstege?

Den gör egentligen inte organisationen plattare i sig, den gör organisationen mer korrekt. Rätt beslut ska tas på rätt ställe och onödiga lager skalas bort och får göra saker som dem egentligen ska göra. Kontakten med kunden är av vikt, kundens behov och önskemål in kombination av strategiskt fokus. Allt kunden vill ha ska inte tillhandahållas eftersom företaget måste värna om att sig själv i första hand. Men företagen ska välja det som är viktigast, men kundens behov i fokus.

Tenderar Serumlagens kompetens att bli väldigt generell? D.v.s. försvinner spetskompetensen eftersom alla måste göra lite av varje?

Känner inte till Serumlagen men den generella kompetensen som du säger blir mycket högre. Dvs, generellt sett kan många fler leverera värde i flera områden. Ja, spetskompetensen blir högre i en mindre rörligt organisation, det är rätt på kort sikt. Det jag har märkt när team tar bredare ansvar är att tiden för att nå kompetens är kortare. Dvs, det tar inte lika lång tid för en generalist att skaffa sig kompetens inom ett område när han känner till de samverkande domänerna mycket bättre. Downside med djup specialist kompetens är att den blir protektionistisk i sin natur. En specialist vill inte att ens kompetensområde ska bli inaktuellt. Så motståndet från dessa specialister kan bli väldigt hög och arbeta emot företagets och produktens bästa. Med detta sagt så måste det också finnas kontinuitet i området man jobbar i. Alltså, det kostar väldigt mycket att i alla områden alltid kunna allt. Det är kontraproduktivt och man måste få en chans att lära sig det man gör. Fördelen med generalism är att (mer eller mindre) vem som helst kan gå och fixa eller förbättra. Generalisma medför också att verktyg, testmiljöer etc ska fungera som professionella verktyg, dvs du ska inte vara expert på något litet script nånstans för att köra tex ett test. Dessa verktyg ska hålla hög generell nivå i form av stabilitet och trovärdighet.

#### 7.5 Intervju 2

|     |  |
|-----|--|
| 1.  | Din nuvarande titel?   |
| 2.  | Jag är teamchef som är Axis motsvarande till sektionschef inom många företag, Jag har ett testteam på 12 personer som jag är linjechef för sen har jag lite processansvar gränslandet mellan linje och projekt i Scrum,  |
| 3.  | Så du håller inte i ett Scrumteam?   |
| 4.  | Nä jag håller mer i en organisation och resurssätter Scrumteam och sådär, man kan kalla det ett funktionellt testansvar, jag har ändå ansvar övergripande för testningen av vår produkt  |
| 5.  | Men du har inte så mycket med att organisera gruppen?  |
| 6.  | Jo det skulle jag säga som teamchef så tillsätter jag gruppen. Så det är rätt personer i grupper sen hur de arbetar, det är efter Scrum. Då har vi en extern Scrummaster som ligger utanför teamet vi har en projektledare som driver projektet och i ett av mina Scrumteam som innehåller bara testare så har vi också en person som är testlead för hela produkten så han tar liksom mera ett projekt ansvar testmässigt för produkten.  |
| 7.  | Ni har liksom en version av Scrum?   |
| 8.  | Det kan man nog säga.  |
| 9.  | Vi jobbar med en produkt som heter aohs som är en hostad lösning för kameraövervakning så det ända kunden behöver är en kamera sen behöver man en PC och sen gör vi resten i molnen så kunden kopplar kamerorna till internet och pc till internet sen sköter vi alla databaser och servers här(i molnet) så kunden behöver bara öppna en sin webbläsare ställer in vad han vill övervaka stänger pcn igen nästa dag tar han upp den och kollar på sina videos och ser vad som händer, det är produkten  |
| 10. | Sen är vi två stycken linjechefer ,detta är lite mer old school, jag säger inte att det är bästa jag skulle gärna se att man slår ihop det, vi har en teamchef/linjechef för qa det är jag och en teamchef/linjechef för dev en kille som heter -----, detta är linjeorganisation här finns ungefär 15 resurser(qa) och här finns ungefär 25 resurser (dev). Och sen här ur ett mer scrumperspektiv då så har vi ett antal teams som jobbar med produkten, alla är Scrumteam men de har olika ansvarsområden. Det kan inte dra vilt ur samma backlog utan det finns lite olika tankegångar där. Då har vi två stycken backend team som jobbar med back enden och dom är ungefär fyra stycket eller fem är det nog, fyra utvecklar och en qa, quality assurance testare, då ska man säga att alla är egentligen utvecklare men qa kommer från min organisation och de fyra kommer från ----- organisation och qa resursen försöker vi tillsätta på ett sådant sätt så att han eller hon så att han eller hon kan jobba med utveckling och skriva kod och så men har ändå ett funktionellt testansvar inom teamet teststrategier och förnya features och projektledare leder resten av teamet vid testning ser till att man tar ansvar vid testning. Sen har vi ett fornt end team så jobbar med webbiten dom jobbar likadant. Vi har ett customer support team ser likadant ut men jobbar med kundcase mot kunderna. Sen kommer det här vi har ett vi kallar det för releaseteam och detta teamet har ungefär fem personer där alla kommer från qa och dom tar ansvar för all testning som sker innan en realease ut mot slutkund. För vi har inte kommit till continguous integretation eller vi har continous integretation men vi har inte kommit till continous delivery vi har |

|     |  |
|-----|--|
|     | fortfarande fyra släpp om året mot våra kunder.  |
| 11. | Det är stående det specifika datumet?  |
| 12. | Ja det är målet men sen är det ju så här ofta när vår produktägare jobbar med oss på den här produkten sen är det inte alltid så tydligt att tid går före scope, ibland vill man ha rätt scope och då skjuter man på relasen istället  |
| 13. | Så det är lite flexibelt?  |
| 14. | Det är lite flexibelt vi har ofta en buffert ett datum inåt och ett datum utåt så kan vi ta in buffertar ta lite ökad risk kvalitetsmässigt  |
| 15. | Så inom de här små teamen är det Scrum men i övrigt är det en slags linjeorganisation?   |
| 16. | Ja kan man säga. Utan för detta finns det då en produktowner och en som vi kallar för produktowner proxy det är egentligen bara ett sätt att få in en riktig projektledare i projekt. Det är lite fult för att det är mycket projektledare som håller ihop alla de här Scrumteamen men man kallar dom produktowner proxy vi har en produktägare som ligger lite utanför som verkligen träffar kunder som ställer krav inom organisationen och sen har ve en proxy som kommunicerar med honom som bryter ner till teamen skapar backlogen mer i detalj. |
| 17. | Dom har ett gemensamt ansvar för backlogen?  |
| 18. | Ja jag tror man kan säga att PO ställer krav på hög nivå och petar inte i backlogen han skriver inte i version one och de mjukvaror som håller ordning i backlogen utan det gör proxyn sen har de tät kommunikation så PO ligger ganska nära sales   |
| 19. | Så det är mest utåt?   |
| 20. | Precis.  |
| 21. | Sen har vi i varje team så har vi både de rollerna jag säger plus en av utvecklarna eller testarna är team produkt owner som egentligen har lite mer ansvar för team backlogen det är bara för.. att det blir lite av en dialekt av Scrum på det sättet eftersom våra team inte är eller har olika roller vi har inte lyckats hitta ett sätt där alla team är lika och kan ta från samma backlog så det har lite olika inriktning .  |
| 22. | Så ni har inte ett Scrum i Scrum där det är en från varje team som går upp ett steg utan det finns en produktägare för varje team och sen finns det någon högre PO som har en övergripande ?   |
| 23. | Det kan man säga, sen vet jag inte vad skillnaden är med att jobba Scrum i Scrum liksom. Ja i  |

|     |  |
|-----|--|
|     | praktiken kanske det blir ganska likt. Jag har inte koll på det riktigt?   |
| 24. | Petter förklarar Scrum i Scrum (07:30)   |
| 25. | Jag tror vi är rätt nära det. Vår product owner proxy som är duktig och intresserad av Scrum driver ju lite ett Scrum team med sina team produkt owner dom flyttas upp med lite kommunikationsansvar. De har möten tittar på vilken nedbrytning som behövs.  |
| 26. | Men då har ni inte någon Scrummaster?  |
| 27. | Jo det finns en Scrum master. Som jobbar som Scrum master i alla team. Han är Scrum master på riktigt liksom vill säga att han håller i dialys och retros. Han kallar till backlog refinements och han jobbar med Scrumprocesser i hela organisation så han är teknisk kunnig men han lyckas bättre med att blanda sig in i teamen.  |
| 28. | Så han tar en daily med alla?  |
| 29. | Ja han sätter ett schema.  |
| 30. | Har du varit med i något team innan du blev linjefe?   |
| 31. | Jag har aldrig jobbat i ett team i Scrum, vi har faktiskt Scrum team, vi också men vi försöker jobba Scrum i linjen. Det är min andra organisation där jag jobbar Scrum på det här viset. Sen har jag varit på Ericsson och jobbat i priceis den organisationen som ————— jobbar i. Så på exakt samma sätt som dom jobbar idag. Jag var en av de team eller sektionschefer där. Och innan dess har jag jobbat som sektionchef och tidigare som testare objektledare på Ericsson medans dom jobbade vattenfall så jag har sett båda sidor.  |
| 32. | Har ändringarna ni gjort varit för att så har det funkat helt enkelt? Har ni provat er fram?   |
| 33. | Jag har inte hela historien men jag vet att man har provat många olika vägar på axis vi gjorde en del förändringar efterhand även på Ericsson. Där drog vi igång tillsammans med Softhouse dom körde ju en implementering av Scrum hos oss dom är en konsultfirma som jobbar med Scrum och efterhand har vi gjort förändringar.  |
| 34. | Vad upplever du som den största fördelen med Scrum?  |
| 35. | Om man ska isolera något så tycker jag den största fördelen med Scrum är att man planerar inte längre fram än, man hävdar inte att man kan planera längre fram än man kan. Man säger inte vi ska vara här om ett år utan man tar hänsyn till vad som händer i på vägen och man planerar på i detalj det som kommer nära i tiden det är det ena, det andra är att man blir väldigt snabb på att byta fokus man blir väldigt snabb på att byta arbetsuppgifter i teamen man kan dra igång på en uppgift och ett par veckor in på en uppgift när folk kommit igång kan man säga att vi har en ny prioritering nästa sprint ska vi jobba med något annat. Jag har en historia från Ericsson där man hade team där varje team det |

|     |  |
|-----|--|
|     | <p>var inte Scrum team men ett team jobbade på ett lager av den här extremt komplicerade stacken. det fanns ett fysisktteam lager två lager tre och test på en annan nivå och att dra igång det maskineriet för att jobba med en feature tog flera veckor innan alla hade synkat och då hade folk börjat ställa in sig på vad de skulle arbetet med. Så att bara stanna arbetet tog flera veckor asså man kunde säga att nu ska vi börja med det här i stället och då kunde man hitta en ingenjör någonstans som jobbade på det gamla för han hade hittat ett intresse, ”vi ska göra det förr eller senare så jag gör det färdigt” Det blir inte så bra</p> <p>Sen finns det alla det vanliga sakerna som transparens och sånt men det vet ni. Men de här är det två stora sakerna som gör skillnad för mig.</p> |
| 36. | Och negativa aspekter, upplever du att det finns några sådana?   |
| 37. | Ja det gör det. I min roll är det mest negativa jag ser är att karriärvägarna är ganska otydliga innan hade ingenjörerna i en vattenfallsmetod finns det ju fler roller, projektledare objektledare, man kunde göra folk till seniors och olika ansvar inom arkitektur och sånt. Det här med att alla är lika blir liksom det gör att det kan bli problem inom organisationen. Man måste se till att alla inom organisationen tycker att ”jobba inom ett scrumteam det är det finaste som finns, vi ställer upp för varandra”  |
| 38. | Sen finns det säkert mer projektmässiga, i grunden är det bra att det är som det är. Teamet gör sina estimat och sina commitments men det kan finnas frustration ibland, att man vill komma in i det gamla linjesaken att sätta en deadline och sen jobba mot det och som chef att ta ett ansvar att det sak hända. Det leder inte till något bättre men det kan finnas en frustration i att man inte kan styra så mycket.   |
| 39. | Tror du att det är en brist i Scrum eller är det en inkörd på det traditonella projektledningen?   |
| 40. | Jag tror det är att man är inkörd på det traditionella jag tror inte det är en brist, snarare en tillgång men det kan ändå skapa frustration. Men det är nog ingen nackdel i Scrum, resultatet blir bättre.  |
| 41. | Ni är ungefär 5 personer i varje scrumteam?  |
| 42. | Ja vi har varit så mycket som 8 personer. Vi har växt som organisation och man märker direkt att det funkar inte team som team. Dialysen och standupsen blir långa det blir subgrupper i teamen som jobbar med olika saker där har liksom varit gränsen 7 har funkat 8 funkar aldrig så då har vi splitat upp så nu är dom nu mellan, minsta teamet är 4 största teamet är 6st.  |
| 43. | Den externa scrummastern också?  |
| 44. | Japp   |
| 45. | Hur brukar kommunikation gå till? t.ex. har ni någon formell kommunikation?  |

|     |  |
|-----|--|
| 46. | Vi har veckoprojekt tänk, det finns rätt mycket vi kan börja med att det finns mycket informell kontakt såklart vi sitter väldigt nära varandra alla utom kundteamet dom får inte plats på vår våning men vi håller på att lösa det, vi sitter i samma korridor och kommunicerar så mycket vi kan mellan rummen i med att man sitter nära varandra och sen finns det en del, vi uppmuntrar folk att gå på varandras stand ups framförallt nu vid releasetider där det finns beroende, vi ska släppa ny mjukvara nu på tisdag, då märker man att ingenjörerna går på varandras stand ups och pratar och ställer frågor kring det man själv jobbar med ”när kommer er leverans, när ska vi leverera med det här beroendet?” och sånt. När man tittar mer formellt har vi ett projektsynk i veckan och det hålls av vår produkt owner proxy/projectledare och det är han och det är tpo:na och sen är det linjecheferna sen finns det ett ticketprio möte per dag det vill säga då sitter vår kund-TPO han leder den gruppen tillsammans med de andra TPO:erna där de går igenom alla de nya ticketsen och prioriterar dom i vår ticketbacklog, striktprioritering, tanken är att det ska göras varje dag och det ska ta tio minuter bara för att vi ska ha koll för vilka tickets som finns och hur mycket kapacitet som ska avväjas för det sen har vi en weekly project stand up. Och då samlar man hela organisationen och då står product owner proxy och pratar om hur det ser ut i vårt projekt sen har vi en vi kallar det för line stand up. Då samlar vi linjechefer hela organisationen och så pratar vi mer åt ett linjeperspektiv typ visioner långsiktiga planer tidsrapportering, allt sånt mer traditionella linjemöte. |
| 47. | Men det är ändå nå vad ska man säga så här halvformell halvinformell?  |
| 48. | Ja det är formell i det att alla är inbjudna men den är informell då det iär inget som dokumenteras, ingen agenda som skickas ut i förväg, ingen sådär alla ska va på den här det är obligatorisk utan vi måste fylla dom med så pass bra innehåll att folk är intresserade  |
| 49. | Det här är bara en del och sen fortsätter det här i resten av organisationen?  |
| 50. | Faktiskt är det så att Axis inte är så scrumifierat. Vi är 1200 personer i Lund kanske 800 som jobbar med utveckling och RnD, traditionellt sätt jobbar alla med kameror och kameraprodukter lösningen i en låda. Vi är några i organisationen som jobbar med applikationer och mjukvarulösningar i det här och det är dom i organisationen som jobbar med Scrum. Det är egentligen bara 2 st. Det är de som gör windowsapplikationer och mobilappar. Vi är kanske 100 av de 800. Resten jobbar vattenfall.  |
| 51. | Är ni påväg över mot Scrum?  |
| 52. | Ja intresset är väldigt stort kring molnlösningar, leverans av tjänster och servicar istället för applikationer och hårdvara. Där med Scrum för det passar mycket bättre i den typen av produkter. Det är nästan omöjligt att jobba Scrum och vattenfall för kamera är kanske Scrum bättre men inte så tor skilland men att jobba vattenfall för en molntjänst är nästan dumt.   |
| 53. | Brister kommunikationen någonstans? i scrumteamet eller mellan de olika?   |
| 54. |  |
| 55. | Det varierar ju såklart över tid dom två problem som är vanligast att ett team som blir lite för stort om det är mer sex personer delar upp sig fullständigt och gör två helt olika uppgifter under en sprint och där med inte kommunicerar spå mycket med varandra. De går på samma dialys men det är som   |

|     |  |
|-----|--|
|     | <p>två olika möten i mötet. Vilket kan vara en nackdel. Det andra är nog mellan teamen. När vi inte har lyckats skapa storiesen oberoende. Att man tittar på sitt eget, man levererar sent. Och så får man inte riktigt ihop det där det drar ut på tiden leveransprocessen blir längre. Sen är det utmanande en tredje sak det är det här mellan releaseteamet som har ansvar för branchen och ökad kvalitén och släppa mot kund, vilket vi inte kan idag. Då ska releasefarten öka kvalitén och resten av tiden ska man hålla uppe kvalitén. Vi har ett sätt att jobba mot dom där varje team har gjort en story eller har en story på gång så flyttar man sin story till en speciell kolumn som heter risky kolmn. Det betyder att här har vi gjort en funktion här ser vi en testrisk det måste vi kolla, då kan release teamet fastän den är testad lokalt kan releaseteamet göra ett djupare test i en mer verklig miljö. Så det går urnt och plockar från denna kolumnen i olika team. Och här kan det vara så att kommunikationen är dålig med va som är på väg dit. Det kommer dit försent t.ex. vi ska ha ett släpp i slutet av april så säger vår testlead han jobbar mycket med de här teamet men det är en egen roll. Så säger han vi har två veckors testtid bara för att säkra det vi skeppar externt. Även om vi hostar den här lösningen så har vi även andra företag som hostar den och kallar den sin och vi leverar till dom. När man leverera till dom har de inte tid att sitta och patcha och så utan måste vara något med väldigt hög kvalitét. Då säger han allt behöver göras på tre veckor för test. Kommunikerar man inte här då så kommer de trycka den testtiden så det bara en vecka kvar och de händer och de kan man kalla dålig kommunikation.</p> |
| 56. | <p>Ligger det då en birst i att det inte finns, den informella kommunikationen brister och därför borde man ha mer formell kommunikations? eller ser du det som att den informella brister och man borde jobba på den?</p>   |
| 57. | <p>Jag tycker det senare. Man vill aldrig gå mot mer formell kommunikation och dokumentation om det inte är slutkunds dokumentation. Så då får vi jobba på de informella processerna. Utvecklare tenderar att vara lite såhär titta på sitt. Och plötsligt tittar de upp och så finns det ingen tid kvar</p>   |
| 58. |  |
| 59. | <p>Övergripande mål och strategier. Man får en task som utvecklar och kör ner huvudet men hur förmedlas övergripande strategier? Och vem förmedlar det?</p>  |
| 60. | <p>Jag skulle vilja säga att det är eftersom vi är en så pass liten organisation och vår produktägare ligger väldigt nära vår business developer och sales. Vi får en väldigt ytlig input från kundsidan. Vi ligger väldigt nära våra kunder. Vi har folk i våran organisation som har kontaktuppgifter till våra kunder. Man jobbar rätt så ofta mellan tekniskkontaktuppgifter hos kunden och rakt in i våra Scrumteam. Produktägaren bygger en wishlist. Som egentligen är högnivå backlogen, den kallas så för att han inte ska tror att han kan få allt. Lite som tomtens önskelista. Den förmedlas sen till vår product owner proxy och blir en backlog. parallellt med det pratar produktägaren med oss teamchefer i ett litet längre perspektiv var är vi på väg, vart vill vi utveckla produkten. Och de kommunicerar vi sen i visiontalks till hela organisationen. Så dels blir det varje vecka i lines stand ups och weekly project stand ups man står och pratar om saker och ting. Just nu är vi i en fas där vi har gått från att levererat en mjukvara en server mjukvara till det vi kallar hostingproviders. Det är dom som erbjuder tjänsten mot marknaden. De får en mjukvara installerat på sina serverar drifrar den mot molnet och tar ut felrapporter och levererar. Vi känner att vi kommit lite långt från kunden på det sättet så nu har vi börjat hosta själva. Vi sätter upp systemet i Amazon och så säljer vi till en service provider som åker omkring och knackar dörr och säljer kameror. Och sen så är det vi har som 24/7 support och sköter driften. och den här omsvängningen håller på att hända nu så vi går från en mjukvaruleveransorganisation till en tjänsteleveransorganisation. Och detta är typiskt så här nu håller</p>             |

|     |  |
|-----|--|
|     | vi på att svänga om produkten och då har vi hållt en visionstolk det är två veckor sen nu då bokar vi en timme med alla och så pratade Jonas som vår chef heter om var vi är på väg med organisationen och jag pratade om vilka testutmaningar vi har när vi blev en sån organisation, sen fanns det några talare till. Så de det formella sättet. Sen finns det såklart massa informella samtal hela tiden. Ibland måste man tänka efter, vi linjechefer och sen förmedla det.  |
| 61. | Jonas pratar om hela organisationen och visionen jag talar testutmaningar och sen det tredje benet var att vår arkitekt Hannes som pratade om hur vi förändraprodukten. Vi vill jobba mycket mer med isolerade microservices. Teknisk, marknadsmässigt och organsiatoriskvision som informeras. Vi är duktiga på att informera hela organisationen.  |
| 62. | Det var helt omöjligt på ericsson där vara alla en kugge och så fick man veta att som individ jobbade man med något jättelitet som team något lite större som avdelning något lite större men även om man omfamnade hela lund så på min tid iaf så nådde man inte särskilt långt för den hela förståelsen. Här vet ju alla hela tiden eftersom vi är en så "liten" organisation.   |
| 63. | På ericsson var man många många tusen som jobbade inom samma kod och här är man kanske 40.   |
| 64. | Den externa kommunikationen, som då är produkt ägaren som gör en wish list som han ger till pop men då säger du att det är ganska tätt kommunikation?  |
| 65. | Dom två om man ska spetsa till det lite där så är det lite hat/kärlek. Det är ju inte riktigt så men produkt ownern kan vara väldigt tydlig med att det här är min whislist och det är det här jag väntar mig och han kan ju liksom vara lite övertydlig för att vi ska lyckas på marknaden bör vi ha det här färdigt, sätta förväntningarna, verkligen det är inte bara en backlog och så plockar man uppifrån och ser hur långt man kommer. Utan han ställer krav på organisationen och då är ju proxyns roll att säga det här är alldeles för mycket att gå till teamet och sätta i en roadmap och bryta ner i ett detaljerat halvår. Vi kommer aldrig komma så långt. vi ska inte jobba med så mycket nedbrytning. Du prioriterar och vi säger att sätter vi linjen här och jobbar med nedbrytning av det har vi fyllt backlogen med mer arbete än vad vi hinner den här tidsperioden. Sen så är det så att i början av en roadmap diskussion så är det ganska mycket friktion och man tar ställning för olika sidor och sen så länge de kommer desto mer samarbetar det mer och mer och när man satt den övergripande planen och teamen har kommit till en plan över tid. En hel releasecykel. Då börjar man samarbeta mycket mer istället. |
| 66. | Hur många stakeholders har ni ungefär? hur många är involverade? Hur mycket är det att ta hänsyn till?   |
| 67. | Tittar man på vilken grupp produktägaren här har runt sig så har han i vårt styrgruppsmöte så har man head of new business, han som leder den enheten så han är chef för 150 personer, sen har vi produktägaren, sen har vi business developer, och sen har vi enavdelningschef, och så sitter vi två linjechefer och produkt ownern och proxyn och det är gänget som sitter i styrgruppen för proxyn och de flesta är interna förutom de första fyra nämnda. Så de är stakeholder tittar man sen vilka kunder som finns bakom dessa så har vi kanske mellan 5 och 10 kunder som är väldigt viktiga för oss. Närmre 5 skulle jag säga. Sen har vi ler vi leverar till men de är dom 5 som vi tar hänsyn till deras krav.   |



|     |   |
|-----|---|
| 68. | Är det ofta det är motstridiga krav från kunder eller internt? och hur hanterar man det? finns det ett till sätt att prioritera det?  |
| 69. | Tydligt är det nog inte men det är något vi tvingar produktägaren till. Han måste prioritera strikt och det är inte allt lätt. Sen är det inte så att kraven är motstridiga det är nog inte så att kund A vill ha någonting och kund B vill ha någonting och att båda lösningarna passar inte in i produkten. Så komplex är inte produkten sätt ute ifrån asså feature mässigt. Det handlar me om att kund A vill ha denna Featuren och Kund B vill ha denna featuern. Båda vill nog ha båda men den ena vill ha den ena mer och den andra den andra och vi måste välja vilken som är viktigast.De flesta är nog överens om vart vi ska me ndet handlar mer om leverans datumet.  |
| 70. | Formell kommunikation med stakeholders, du nämner innan att det finns en slutdokumentation?   |
| 71. | Precis. Men det är nog mer en slutdokumentation är mer en teknisk dokumentation till kund hur man sätter upp produkten och hur man använder den. För stakeholders har vi en operationellt styrgruppsmöte varannan vecka. Där man diskuterar beslut som behöver tas investeringar som behöver göras vi rapporterar verksamheten och får input på den och den dokumenteas, och det är egentligen det ända mötet som dokumenteras. Vilka beslut som tagits och så. Och det är inte sådär paragraf ett och ordförande öppnar mötet och så. Det är inte alls så utan man bara skriver lite. Vi skriver presentation där vi presenterar hur det ser ut i produkten de ger feedback på de och vi skriver ner lite beslut och så. En powerpoint presentation som vi tar upp varje gång. Sen finns det också en formell styrgrupp på högsta nivå där det mer så här beslutas om vi ska fortsätta med produkten ska vi investera mer pengar i produkten men dom har bara möte varje kvartal eller halvår kanske . Och där sitter riktigt höga chefer i axis såna som flerdelar pengar mellan detta projektet eller detta projektet. Och det vet jag inte så mycket om egentligen men det finns ju dokumentation och så. Och det kommer aldrig ner till oss mer än på ett ganska filtrerat sätt. |
| 72. | Underleverantörer har ni någrasådan? Som skriver kod eller på något sätt ska involveras i processen?  |
| 73. | Vi har en del open source, en del komponenter på serversidan. Bl.a. en videoöversättare som översätter streamvideos i realtid. Vi har internleverantörer som kamerorna, vi levererar ett system så måste ju våra axiskamereror som är de enda kamerorna som stöds och vår servermjukvara, dom måste spela väldigt bra tillsammans. Så hela kamerautvecklingen på axis är en internleverantör, så vi har några stycken.  |
| 74. | Är det svårt att involvera dom i Scrum? er process få det att klaffa med deadlines och att de är involverade med vad ni har för ide?  |
| 75. | Jag tror inte vi gör det till en tydlig del av Scrum. Det är mycket bakgrundsarbete som vår produktägare och vår productowner proxy som dom jobbar med utanför Scrum. Dom driver saker som produkten kräver utanför Scrum och ibland när de innebär en dialog som är väldigt teknisk eller ett integrationsjobb med en tredjeparts leverantör då går man rakt ner i teamet. Ni som team har ansvar i den här sprinten att få det här färdigt tillsammans med den här tredepartsteamet.Det är ganska smidigt och vi trycker ner ansvaret väldigt väldigt långt.  |

|     |   |
|-----|---|
| 76. | Hur hanterar ni technical debt? Om ni har sådant.   |
| 77. | Det har vi verkligen. Det är en jättesvår fråga. Det är det i vattenfall också. Men inte minst när prioriteringen blir så tydliga. Då har vi väldigt svårt att få in refractoring, technical debt stories i backloggen och det är ett problem när man pratar visualisering. Vi har ett litet av berg av teknisk skuld kan man säga. Som gör att processen faktiskt blir långsammare att utveckla och vi vill ju minska på den bördan snabbare än vad vi lyckas med idag. Just därför att vår arkitekt, han skriver stories som blir väldigt svåra att få in i produktägarens backlog, hela hans prioritering eller hans wishlist. och då får vi ju välja mellan att antingen dölja våran och presentera en minera kapacitet till produktägaren vilket inte känns bra, eller förhandla med honom och försöka få in det. Och ingen av de sätten att arbeta har lyckats helt bra egentligen. Det arkitekten kan påverka det är den löpande arkitekturen, hur man gör refractoring samtidigt som man jobbar med en modul. Och vilket tankesätt man har vi nyutveckling. Däremot den tekniska skulden är vi inte bra att hantera och ibland lägger vi på den för att deadlines med kunder känns så pass viktiga. |
| 78. | Så det är svårt att kommunicera med produktägaren han vill prioritera kunderna och ni vill prioritera produkten?  |
| 79. | Precis så faktiskt.   |
| 80. | I och med att det inte är ett funktionellt leverabel så blir den väldigt svår att involvera i processen?  |
| 81. | Ja precis. Jag tror den skulle vara rent tekniskt är den inte svårt att integrerar eller testa. men den ger ju inget kundvärde här och nu. För den är ju rätt så långsiktig. Sen kan man säga att hade vi kunnat hantera den tekniska skulden kontinuerligt så hade vi fått ut mindre här och nu men det är lätt att se att om ett år hade vi haft mycket högre velocity i organisationen. För det är alltid problem med den tekniska skulden. Inte bara att de ser stökigt ut och funkar sämre. Det viktigaste är att nyutveckling görs så pass mycket långsammare   |
| 82. | Kanske framförallt om man ska gå mot mer Service? Det finns inga egentliga releaser utan kontinuerligt. Man behöver en stark grund.   |
| 83. | Precis.   |
| 84. | Annars blir det den här man tvingar in det nya liksom?  |
| 85. | Precis och det är inget man gör independent, alla stories och features vi trycker in det slår väldigt brett.  |
| 86. | Är det här ett större problem inom Scrum eller vattenfall?  |
| 87. | Jag tror inte det. Det är den här klassiska att du har samma problem i Scrum och Vattenfall men i Scrum ser du problemen. Och har chans att göra något åt det. I Vattenfall är det bara så att vi lyckas inte med våra projekt. Vi får inte riktigt ihop det i slutet. Det tar mycket längre tid och det är lite  |

|     |  |
|-----|--|
|     | svårt att säga varför.   |
| 88. | Använder ni er av generiska lösningar? Återanvänder kod från projekt till projekt och sådär?   |
| 89. | Nä det kan jag inte säga att vi gör. Vi bygger på i produkten hela tiden. Men det blir, varje gång är det något som är rätt annorlunda. Klart vi fyller på parametrar men det är inte så att vi kan ta ut stora kodblock som används som grunden till nått som är helt nytt. Men det beror också mycket på att vi inte bygger en produkt som vi släpper till markanden så sätter vi en support en viss tid det är inte så vi vill att kunderna följer oss hela tiden och uppgraderar. Det ligger lite i produktens karaktär. Den här produkten jobbar med microbetalningar. Vi tar inte betalt för mjukvaran utan vi tar betalt för den här kamerakanalen. Så den kanalen kostar x dollar per månad. För vi har koll, lösningen är så att kameran kan inte koppla upp sig mot vår produkt utan att vi är med på ett hörn. Så slutar man betala så tappar man kontakten. Har man kopplingen så får man en faktura. Hela iden är att växa detta till väldigt mycket kameror och få väldigt många små betalningar varje månad. Så vi tar inte betalt för uppdateringar utan vi vill att alla kunder ska ha det senaste. Det vill säga vi har liksom inte fork at vår produkt. Och sen kan vi återanvända det här eller så bygger vi en annan produkt och återanvänder ut den så allt är ett spår. Så allting är en återanvändning kan man säga. |
| 90. | Så det finns inget behov av generiska lösningar?   |
| 91. | Nä jag tycker inte.  |
| 92. | Det kanske blir samma spår men det här, om det någon gång skulle vara väldigt komplicerad arkitektur eller man vet från början att den här biten kommer vara väldigt komplicerad. Då kan det vara svårt med Scrum i med att man på något sätt ska man gå in och sen ska man kasta in utvecklarna på det så ska de på något sätt hitta på hur man ska lösa det medans det egentligen finns ganska tydliga man måste gå den här vägen för att det ska gå att lösa. Händer det eller är det ett problem?  |
| 93. | Jag skulle vilja säga att vi är ganska bra på att involvera teamet med utvecklarna tidigt. Redan när vi gör dom första estimaterna. Man tittar liksom på kvartalsplaneringen eftersom vi ligger så nära kunderna måste vi även titta hur tror vi att vi kan positionera oss på marknaden tror vi att vi kan levera under ett kvartal. SÅ har vi med ingenjörer och tpo:er så väldigt tidigt börjar vi diskutera tekniska lösningar arkitektur och så. Så vi kommer aldrig till en tydlig tidpunkt där vi lämnar över till någon som ska hitta en lösning utan det jobbas fram hela tiden. Och de bästa är ju arkitekterna och sen är det team medlemmarna och ingenjörerna och de har väldigt mycket informell kontakt. SÅ jag tycker inte att det är ett problem om jag uppfattade din fråga korrekt. Ja jag tror det.  |
| 94. | Så här projekten överlag? Man tittar på hur lång tid de tar, går de oftare över tid nu än innan? Är det ofta de blir försenade?  |
| 95. | Jag tror de blir försenade mindre idag. Alltså mindre tid. Som jag sa har vi byggt ut lite gummiband. Att vi har ett externa datum och ett internt datum. Kunden är medveten om att vi jobbar Scrum och kunden vet att dom behöver hög kvalitet. Och vi kan förhandla med kunden vilket innehåll vi levererar när. Jag skulle vilja säga att de nästan alltid drar ut på tiden men de är väldigt korta tid. Det handlar om fåtalet veckor och då har vi en release 3 gånger om året. Så på fyra månaders utveckling har vi en parveckors slip men det hanterar vi med det interna gummibandet. Och det handlar ofta om   |

|      |  |
|------|--|
|      | utmaningen att få teamen och ingenjörerna att verkligen veta hur viktigt det är. De vet hur riktigt det är men det är så svårt som utvecklare att sätta en deadline man vet att vi ska leverera till kund i slutet av april och 7e april ska vi vara färdig med koden och sen är det testfas resten. Asså människan funkar inte så när det är en vecka kvar och man känner att det är 10 dagar arbetekvar då är det inte inbyggt i processen och människan att ja men då blir vi lite sena denna gången också men så är det. Så länge tiden inte ökar så kan man ta hänsyn till det. Men jag tycker vi är där fortfarande vi kan inte få commitment och estimerar rollerna fullt ut. Det är inget problem. |
| 96.  | Är ni ganska pricksäkra när man estimerar task eller user story. Det funkar ganska bra?  |
| 97.  | Japp, det funkar bra skulle jag vilja säga. Jag tror vi har hit rate på avklarade sprint mål och avklarade sprint poäng på över 70%. Och då är det ju inte så att resten är en total katastrof. 70% klarar vi oss hela vägen eller kanske till och med gör något lite extra. Och på 30% kanske det saknas lite. Kanske på externa beroenden. Kanske för att något blev mer komplext kanske pga. sjukdom. Det finns många anledningar. Jag tycker det är väldigt bra faktiskt. Skulle det vara 100% skulle jag bli orolig för då tror jag man har under commitat.   |
| 98.  | När man planerar sprintar och ska göra user stories och sådär kan det vara svårt att t.ex. en user story är en halv sprint eller 1,5 eller det kanske inte går att dela den på mitten? Har ni tre veckors sprintar eller?  |
| 99.  | Två. Vi har korta sprintar och det funkar bättre här än vad det gjorde på Ericsson att ta ett steg i taget. Vi har nästan aldrig en sprint med en story. Vi har ofta 3 eller 4 stories per sprint. Vi kan prioritera dom vi kan lämna en story om vi inte hinner. Och det är så pass många poäng och story att det blir en dynamik. Något får lite ledtid då fokuserar gruppen på något annat. Har man en story bli man ju väldigt sårbar ifall det händer något i början av sprinten. För då tappar man ledtid direkt och har inget annat att jobba på.   |
| 100. | Jag tänker mer på om man ska planera sprinten så kan det ju vara så att man har en task som inte går att dela på mitten och dela upp till två olika sprintar. Det kan ju gå att lösa men om det ofta är så att man måste bryta ner den här och förlorar iden om vad den är?  |
| 101. | Nä det är inte ofta och vi som stakeholders som går på demo efter sprintar med alla team tycker nog att vi leverar kundvärde i varje sprint. Vi leverar saker som är intressanta att göra demo på och det har inte varit stora problem att splits så att vi hamnar där.  |
| 102. |  |
| 103. | Hur ser arbetsbelastningen ut över sprinten eller är det jämnt fördelat eller är det så att allt de sista dagarna?   |
| 104. | Axis har ju en historia över att ha en personalpolitik som innebär att man har en balans mellan arbetsliv och privatliv på ett helt annat sätt än vad Ericsson har. Ericsson i Lund har ju en historia av väldigt mycket overtid och kniven mot strupen. Nu måste vi leverera till kunden. Och det har gjort att man är bättre här på att tuffa på i jämntakt över hela sprinten. Det är klart att alla känner sig alltid tryggare i början av sprinten och det blir ju stressigt men det är inte så att arbetsbelastningen stiger   |

|      |  |
|------|--|
|      | mycket utan vi, folk jobbar här fulla veckor inte speciellt mycket mer ibland mindre och lyckas ändå med sina sprintmål. Så det tycker jag är bra.   |
| 105. | Så det är inga tydliga stressmoment?   |
| 106. | Nä det ända man kunde säga blir stressigt är väl ofta att i slutet av en sprint på en fredag och sen har man demo på måndag förmiddag och det ligger också i den mänskliga naturen att demo tittar man inte på förrän man verkligen försökt bli klar med allting och på fredagen är det bara helg och sen på måndagen kommer man tillbaka ska fixa demon liksom. Så demot kan bli lite stressigt att förbereda men jag tycker ändå att det är samma fokus i början av sprinten som i slutet.   |
| 107. | Hur vet ni att ni gör rätt? Hur mäter ni resultat? Hur vet ni att så som ni gör Scrum är rätt?   |
| 108. | Du tänker på själva processen och själva Scrum mer än produkten?   |
| 109. | Precis.  |
| 110. | Vi har några mätvärden som Scrummastern håller i. Självklart är det velocity alltså man kan ju arbeta med continuous improvement ständig förbättring. Det vill säga vi ska få velocity ska gå upp kontinuerligt. Det finns en lång historia här av att estimera på ungefär samma sätt. Med road map poäng och story poäng och så. Så man kan rätt lätt mäta velocity sen har vi ett annat index som vi tycker är väldigt viktigt. Som vi kallar happy index. Som är i retron samlar alltid Scrummastern in happy index från teamens medlemmar. Det är bara så här 1-5 hur happy är du för din arbetssituation och för jobbet ni gör och hur bra saker och ting är. Och det mäter rätt mycket frustration i teamet. Frustration och tvärtom till frustration typ arbetsglädje och harmoni. ett harmoniskt team högt happy index det är bra. Då har man hög velocity också. Det är mycket roligare att prata om det än att prata om hur fort har det gått liksom och det visar lite samma sak. Så vi försöker mäta happy index och bemötta den feedback vi får på retron prata om vilket riktning man ska gå bolla tillbaka till teamet. Coacha dom så att de tittar inåt och verkligen själv tar hand om sina problem. Så vi mäter happy index och vi hanterar det som gör att de inte blir happy och ser till att velocityn där med blir bättre. |
| 111. | Mäter ni detta innan ni började med Scrum?   |
| 112. | Det här har tillkommit efter Scrum. Tidigare var de, vad jag förstätt jag var ju inte med på den tiden här, men jag har förstätt att det var mycket det här klassiska att det står en produktägare och en projektledare och tycker det går för sakt piskar på, det är någon i organisationen som säger det här är orimligt.  |

## 7.6 Intervju 3

|     |  |
|-----|--|
| 1.  | Din nuvarande titel, vad gör du här på företaget?  |
| 2.  | På Qlik är jag TDM, som betyder triad development manager. Vilket innebär att jag har ett team på 16 personer, som, där dessa 16 jobbar i lite olika team. Men de rapporterar till mig allihopa. Och sen har jag en kollega som också har lika många och den biten med performance management och lön och målsättningar, hela vägen, som göra att vi har delat ner antalet gubbar i hanterbara mängder. Sen jobbar dem lite kors och tvärs projektmässigt. |
| 3.  | Nån slags linjechef-liknande?  |
| 4.  | Liknande.  |
| 5.  | Jobbar du efter Scrum?   |
| 6.  |  |
| 7.  | Njåe, en del av min dagliga tillvaro är jämförbart med att vara Scrummaster/projektledare. Även ett visst produktägandeskap är inblandat också.  |
| 8.  | Är det för ett team?   |
| 9.  | Nä, jag har flera, tre teams som jag är för.   |
| 10. | Hur stora är teamen?   |
| 11. | Det varierar, det största är nio personer i. Det minsta just nu är det bara två personer i. Beroende på pappaledighet och semester inblandat. Men i vanliga fall 5-6 personer, upp till nio.   |
| 12. | Har du haft någon liknande position innan?   |
| 13. | Nä, jag har varit här sen augusti förra året. Innan dess jobbade jag 14 år på Ericsson.  |
| 14. | Så då har du jobbat med Scrum förut?   |
| 15. | Absolut. Ja det var inget nytt för mig som chef att jobba med Scrumteam. Det har jag gjort länge.  |
| 16. | Har du haft någon annan roll inom Scrum?   |

|     |  |
|-----|--|
| 17. | Njae, jag hade OPM, kallar man det Operational Product Manager och det var ju liknande med den linjechefsmässiga biten utöver det där var jag varken Scrummaster eller produktägare. Jag var med som en agil coach för teamen.   |
| 18. | Är det några uppenbara skillnader hur ni arbetar med Scrum?  |
| 19. | Nja, alla bolag kör nog sin variant av vanilla Scrum. Det har man gjort här så väl som på Ericsson.  |
| 20. | Trivs du med Scrum agilt eller Scrum som arbetsmetodik?  |
| 21. | Absolut. Jag föredrar det många gånger framför vattenfall.   |
| 22. | Det känns som ett naturligt steg att ta?   |
| 23. | Absolut, naturligt att titta på vad som genererar värde och i vilken prioritetsordning vi ska göra saker. Det tycker jag är bra.   |
| 24. |  |
| 25. | Om vi går in mer ingående på hur ni arbetar med Scrum. Kör ni Scrum i Scrum?   |
| 26. | Ja det kan man väl säga att vi gör. Även om man in har kallat det så, egentligen exakt det, så är det som händer. Man tittar på vad det är för leverabler eller värden som teamen kommer tillföra produkten över den närmsta framtiden. Har haft flera gånger i veckan då som har man kunnat be om hjälp om man behöver det. |
| 27. | Men det är så att du har flera Scrumteam, finns det någonting över dig, en längre cykel som man går mot?   |
| 28. | Vi har ju övergripande, så kör vi milestones, som ligger som övergripande och dessa är 10 veckor långa. I varje milstone, har vi delat ner det i fem beats av två veckor.  |
| 29. | Det finns en form av nedbrytningsprocess där man går uppifrån och så blir det mindre och mindre taskar?  |
| 30. | Ja, men inte per automatik men i alla fall. Det finns ett övergripande där man omvandlar business scenarion till produkt scenarion som vi använder för att göra våra userstories.  |
| 31. | Har ni kombinerat med någon annan arbetsmetodik? Eller har ni någon egen twist på Scrum?   |
| 32. | Ja det är väl en egen twist på Scrum, skulle jag vilja påstå. Tex jobbar vi med flera produktscenarion parallellt istället för sekventiellt, som är en twist på det. Man använder Jira, som verktyg för att ha   |

|     |   |
|-----|---|
|     | koll på sina leverabler. Det är ju ett verktyg som är väl anpassat för en agil värld där man får upp boards istället då för fysiska tavlor.   |
| 33. | En digital program för att hantera tasks?   |
| 34. | Absolut, och jag använder en kombinationen av båda. Man har inte jättetydligt, kanske, en produktägare som man alltid kan fråga om prioritet eller vad kunder tycker i det fallet. I det teamet som vi har som är enduserteamet så gör vi saker till flera olika missions. Det vanliga hade kanske varit en tydligare bild ifall man hade ett team per mission, då hade man haft en backlog och en produktägare. Men nu är det indelat i ett antal sådana.  |
| 35. | Är det mer kunskapsspecifika teams som alla kan behöva hjälp ifrån? Finns det mycket kommunikation mellan de olika teamen?  |
| 36. | Ja det gör det ju, alla eller alla övergripande produkt scenarion som bara involverar engine teamen det kan involvera olika team. Då får man ha en koordinering mellan dessa och se vad som behöver göras, prioriteras och när eventuellt teamen kan vara klara.  |
| 37. | Händer det nångång att man måste vänta(på andra team)?  |
| 38. | Haha, absolut, man har ju det här som är bra att om ett Scrumteam blir stående och väntar så kan de erbjuda sig att hjälpa att göra det. Det gör vi ju bara under förutsättningen att vi vet vad vi ska göra. Många gånger är det så specifik kompetens att man skulle göra mer skada än nytta om man rullar runt där inne.   |
| 39. | Men det är delade backlogs mellan teamen och så blir Scrumteamen tilldelade uppgifter från olika backlogs?  |
| 40. | Alltså, missions har backloggar med saker som beställare. Sen sätter vi oss en gång varje milestone och tittar på vilka av de här sakerna kan vi göra. Så plockar vi ner då och tar ner dem i teamen och man har nån form av design från våra arkitekter först. Där är våra teammedlemmar ibland med så de är väl insatta i vad det är. Och sen gör vi våra planer. Och där är ju också något som skiljer vanilla Scrum. Vi gör, just nu gör vi planer som är sliding window på tre av våra beat i taget. I första beatet som kommer närmast har 80% konfidens i det vi ska göra, sen har vi 50 % konfidens i beatet efter sen har vi 40% i beatet som kommer där efter. Så gör man en planering med teamen varje beat. Och så tittar vi då, så här tror vi det kommer se ut, tre beat fram, men vi gör det med 40 % konfidens. |
| 41. | Dels att omvärlden förändras, dels vad man hinner med?  |
| 42. | Absolut.  |
| 43. | Jobbar ni på olika sätt i olika delar av företaget?   |



|     |  |
|-----|--|
| 44. | Jag tror nog man har lite olika sätt att jobba på olika ställen. Det tror jag. Sen att nån skulle jobba exakt som vi det tror jag inte. Vissa tittar på hela milestonen alla fem beaten redan på förhand, mer vattenfall ur den aspekten, samtidigt så får man en lättare överblicka ifall man är bra på att göra sin milestoneplanering.  |
| 45. | Så det finns variation men det grundas på samma tänk och terminologi?  |
| 46. | Det är samma.  |
| 47. | Vad är det största fördelarna med agilt?   |
| 48. | Det finns ju flera fördelar. En av fördelarna är ju diskussionen som man har löpande, vad det är som är viktigt. Det kan förändras, men när man har en diskussion och kommer fram till "Det här är viktigt" att man med teamen att de känner att de gör saker som tillför värde till produkten och att de syns på en gång. Det är en slags produkt utveckling. Slår man på sin telefon och så har man gjort en ny sida, så får man omedelbar feedback. Att man ser hur man lägger till features. Till exempel. Att omvärlden förändras, det gör den då ibland, att man inte tappar fokus. Ja, men nu bestämde vi detta förra året, ja men ett år är jättelång tid. Det är detta som gäller nu. Man fångar upp den föränderliga världen genom att vara lättroilig.  |
| 49. | Och detta är främst om man jämför med vattenfall?  |
| 50. | Absolut, det man bestämmer två år i förväg är inte samma sak när man väl sätter igång två år senare.   |
| 51. | Framförallt i en så föränderlig branch?  |
| 52. | Absolut.   |
| 53. | Negativa aspekter? 14:20   |
| 54. | Det finns ju alltid negativa aspekter, allt har ju en och anra sidan myntet. En sak som man får vara noga med på något sätt, det är ju den långsiktiga bilden i en kortsiktig vardag. Vart strategiskt vi är på väg. Göra ett förarbete som är tillräckligt bra gjort för att teamen senare kan tugga på. Det är ju en bit som, systemeringsbiten som man får passa på lite extra. Sen har man ju alltid när man har gjort en feature och man inser att man skulle kunna optimera den delen eller lägga till det för att bara göra den ännu bättre. Ja okej, låt oss lägga det i förbättrings delen eller den tekniska skulden. Att föra dem diskussionerna om tekniskskuld tillsammans med nya features är ju alltid tung att ta. Man pekar på att vi behöver att förbättra det här, just nu är det ingen bra upplevelse för kunden. Ja men vi måste ju ha den här feature. |
| 55. | Feature mot tekniskskuld?  |
| 56. | Ja, den är ju alltid en svår grej att få in tillräckligt många förbättringar. Det är svårt, lite tungrott.   |

|     |   |
|-----|---|
| 57. | Att det saknas incitament i strukturen för att få kvaliteten före funktionalitet?   |
| 58. | Det kanske inte alltid är kvalitet, men produkten kanske har tillräckligt bra kvalitet men det hade gått ännu fortare eller ännu snyggare eller ännu mer användarvänligt. De parallellerna kan man dra i alla fall. Då kan det vara svårt att få igenom att få med sig det helt enkelt.   |
| 59. |   |
| 60. | Fokus ligger snarare på att leverera ny funktionalitet, snarare än det långsiktiga, technicaldebt, att man hanterat den?  |
| 61. | Ja, och mappa. Om man tittar på förbättringarna så är det en sak, men om man ska kombinera det långsiktiga med det kortsiktiga så kan det ibland vara svårt att se den röda tråden i det du gör varje dag.  |
| 62. | Förmedla visionen eller långsiktiga visionerna?   |
| 63. | Ja.   |
| 64. | Hur gör ni det? Hur får man reda på det som utvecklare? Är det du?  |
| 65. | Närmsta milestone är väl där jag har mitt fokus. Sen kan man ju titta på, övergripande, har för gissningar för vad som kommer flera milestones fram i tiden. Den återkopplingen är kanske inte lika självklar. Närmsta milestone är vanligtvis där fokus ligger.  |
| 66. | Närmsta milestone är det som motsvarar sprint?  |
| 67. | Nä, 10 veckor. Så det är fem sprintar a 2 veckor.   |
| 68. | Det finns formell kommunikation och så finns det informell kommunikation. Hur förs den formella kommunikation internt?  |
| 69. | Ja, den förs i vårt dokument system, N, och Jira där den formella, där man har sina commitments, när saker är klara. Där man har sina beroenden och det återspeglas i den planering som man gör i de här sliding window, sprint eller beat. Så att vem som helst kan gå in och titta vad det är för värde vi ska tillsätta och teamen ser att det är möjligt att den eventuellt finns på plats. |
| 70. | Och det informella är som vanligt agilt, man rör sig mycket?  |
| 71. | Man rör sig väldigt mycket. Med alla sina interfaceytor om iaf om man inte har en feature som bara är engine-delarna. Vi har team klientdelarna ovanpå, där de jobbar i samma team som engine-delarna,  |

|     |   |
|-----|---|
|     | ett par team jobbar så.   |
| 72. | Så dels inom Scrum och med andra Scrumteam?   |
| 73. |   |
| 74. | Och sen är det lite mer koordinering, lite mer spring för att se så att grejerna vi är beroende av är på plats eller grejerna som dem är beroende av är på plats. Men jag tror det är omöjligt bedriva verksamhet med fler än flera team utan att behöva springa lite.  |
| 75. | Sker utvecklingen på många olika ställen, så att ni har kontakt med andra ställen i världen?  |
| 76. | Ja, både och. Det är mycket utveckling här i Lund. Men det finns på andra siter också. Det är inte så att vi implementerar samma funktionalitet i samma ställe på själva produkten däremot kan ett annat ställen i världen vara beroende av att vi får vår funktionalitet på plats för att dem ska kunna få ut någonting.   |
| 77. | Men då får man kommunicera på andra sätt än att flytta sig i rummet?  |
| 78. | Ja då är det ju andra, Skype eller Slack eller Webx   |
| 79. | Och det funkar?   |
| 80. | Absolut.  |
| 81. | Är det dagligen?  |
| 82. | Nä, det är en gång i veckan kanske. Annars är det ju öga mot öga. Man använder otroligt lite telefon på det här stället, vilket jag gillar.   |
| 83. | Det uppmuntras?   |
| 84. | Nä nä, det uppmuntras inte men jag tror jag har ringt kanske åtta gånger sen jag började. Resten är att man letar upp den man behöver at prata med. Det är inte större än så länge.   |
| 85. | Svårigheter med den interna kommunikationen?  |
| 86. | Jag kan inte sätta på fingret på att det är svårigheter med den interna kommunikationen. I dem fallen där man sätter sig ner och pratar output från teamen, tex att de är klara om två veckor, och när det har gått två veckor och de teamen inte är klara då kan du ju välja som scrummaster att springa till din motpart som du har sagt till att grejerna ser ut att vara färdiga. Men dem är ju inte det. Det kan du ju |

|      |   |
|------|---|
|      | göra. Men det kan ju vara att den informaitonen inte ripplar och inte kommer förrän två dagar senare.   |
| 87.  |   |
| 88.  | Det finns för lite en buffert?  |
| 89.  | Det kan vara nån dags fördröjning.  |
| 90.  | Lite ledtid?  |
| 91.  | Ja. Men inget som är svårt att hantera.   |
| 92.  | Du skulle inte se det som ett organisatoriskt problem?  |
| 93.  | Nä, nä.   |
| 94.  | Hur kommunicerar ni med externa intressenter eller stakeholders? Ha du kontakt med dem?   |
| 95.  | Nä, jag har inte direktkontakt med kunderna, de gånger jag har haft det är det för att vi har haft fel som kunden som supportavdelningen har kunnat ge dem tillräckligt med kött på benet. Då har vi blivit inkallad från R&D. Oftast telefonmöte.  |
| 96.  | Så kommunikationen kan gå så långt ner?   |
| 97.  | Den kan gå rätt ner till oss.   |
| 98.  | Vem hanterar Stakeholders?  |
| 99.  | Man har ju Product Management som är ute hos kunderna och man har försäljningsavdelningar ute hos kunderna som samlar sig önskemål, kunderna berättar at detta hade ju varit toppen ifall ni hade kunnat göra såhär. Tas tillbaka och tuggas och blir ofta businessscenarion av det. Där finns det förmodligen en tröghet så att man har inte en önskad funktionalitet förrän milestonen efter.             |
| 100. | Hur många stakeholders som är involverade i ett projekt?  |
| 101. | Ja det kan ett ju vara, just för del så har vi ju, eftersom levererar värde till flera missions, så kan det ibland vara flera iblandade. Och man kan ju hamna i den situationen där man måste samla flera stakeholders när man ser att vi hinner inte med alla de här tre grejerna. Kanske två av dem men inte alla tre. Vilken är då viktigast. Vilken är näst viktigast. Det kan bli roliga diskussioner. |

|      |  |
|------|--|
| 102. | Är det ofta motstridiga krav eller är det krav som är olika saker?   |
| 103. | Det är mer att de vill få olika saker gjorda.  |
| 104. | Och då får man helt enkelt... Dialog är lösningen?   |
| 105. | Ja.  |
| 106. | Är finns dessa stakeholders med även under själva utvecklingen? Tittar man efterhand om?   |
| 107. | Ja, det kan man väl säga de gör. De är ofta med när vi gör våra beatsammanfattningar av vad vi har gjort under sprinten, vad har vi tillfört för värde och vad har vi inte hunnit tillföra. Så har man den dialogen då.  |
| 108. | Är de representerade av någon på företaget eller är dem där i fysisk person?   |
| 109. | Nä dem är ofta med via telefon och video.  |
| 110. | "Det är ser bra ut, fortsatt på detta hållet"? (Som ett exempel på kommentarer)  |
| 111. | Ja. "Oh det där måste vi ha"   |
| 112. | Kan inte det blir svårt att hantera, Kunden kanske vet vad dem vill ha men inte hur det fungerar, att det finns ett kunskaps gap?  |
| 113. | Ja, asså nu är det inte slutkunden, utan product management som är repsentater för kunden kan man säga. Jag vet inte om jag upplevt att de önskemål som kunden har uttryckt inte har kunna omsättas i nånting som vi skulle kunna göra eller förstår. Nä, jag har nog inte upplevt det. Kunden har ganska så bra koll på produkten och har många gånger bra ideer på var vi behöver lägga tid. Det tycker jag. |
| 114. |  |
| 115. | Finns det någon formell kommunikation med Stakeholders?  |
| 116. | Det finns ju möten ovanför mitt huvud, som riktar sig mot product management än vad vi känner på i teamen. De är ju också samtidigt input till vår milestoneplanering.   |
| 117. | Den kommer ner för eller senare?   |

|      |  |
|------|--|
| 118. | Ja.  |
| 119. | Åter igen, tycker du det finns några brister här i den externa kommunikationen?  |
| 120. | Nä, om ett sånt fall så får man försöka jaga ner den personen som kan svara på frågorna. Okej, vi kan göra detta på fyra sätt, vilket vi ska vi göra? Spelar ingen roll för oss om vi gör vilken av dem. Då involverar jag i alla fall product management. Så får de ta reda på det om de inte vet.  |
| 121. | Underleverantörer, har ni det?   |
| 122. | Nej.   |
| 123. | Inte över huvudtaget?  |
| 124. | Nej.   |
| 125. | Körde ni något innan Scrum här?  |
| 126. | Ja det har väl dem gjort, men det är nog en tre år sedan som det var mer åt vattenfall, klassisk vattenfall. Men det inget jag vet tillräckligt mycket om.   |
| 127. | Använder ni er av någon generiska lösningar? Design patterns som återanvänds?  |
| 128. | Nä jag tror, det är möjligt att det finns men inget som jag kan komma att tänka på. All feature utveckling jag ha varit inblandad i har inte varit för våra delar något man lägger in i ett pattern sätt att lösa. Det har varit, ofta är det ju nya saker som inte finns på plats redan eller så finns det inte på plats men så utökar man funktionaliteten. Man har inga såna jättebra patterns som hela världen använder för ett visst problem. Sen tror jag inte det är helt tillämbart. |
| 129. |  |
| 130. | Scrum bygger på att an ska se saker med nya ögon men ibland kan ju en generisklösning vara den bästa.  |
| 131. | Ja man får alltid ha det i åtanke.   |
| 132. | Men ni känner inte att det är så att ni tenderar att uppfinna hjulet igen i nya projekt?   |
| 133. | Nä, nä.  |

|      |   |
|------|---|
| 134. | Det är så pass nya saker i varje projekt?   |
| 135. | Ja, uppfinner hjulet igen gör vi inte. Definitivt inte, det har vi inte tid med.  |
| 136. | Frestas ni nån gång att gå mot mer av en vattenfallsstruktur? Eller är folk överns om att agil är ett bra sätt att arbeta på.   |
| 137. | Det beror på vem du frågar. Kopplingen mellan att R&D i en agil tillvaro, säljarna säljer saker långsiktigt, R&D och top management i vilka olika milestones de olika funktionaliteten kommer in. Det är ju väldigt vattenfall. Än så länge har vi inte tagit det som, "Ja idag har vi tagit det som finns masterbrachen så får dem de, och de och de och det, men dem fick inte de och de och de." |
| 138. | Det finns strukturer kvar av vattenfall?  |
| 139. | Det finns det nog på de flesta ställen som bedriver agil utveckling, längre upp i kedjan, i försäljningsledet.  |
| 140. | Det blir mer matrisorganisation?  |
| 141. | Där måste man ha kvar överblicken längre fram i tiden för att kunna sälja en produkt som innehåller saker. "Dem här två sakerna får du, och går det riktigt bra får du det här också"- det tror jag inte riktigt är.... Men de tär möjligt.   |
| 142. | Dokumenterar ni ert arbete? Du sa innan att ni har något program?   |
| 143. | Ja, vi har asså Jira där du har siffror, id-nummer på taskar och userstories och även buggar som är inrapporterade. Sen använder vi ett dokumentssystem. Där kan du dokumentera vad du vill, de har inga kopplingar till många andra ställen. Om man inte vill. Jag vet inte riktigt vad du menar med när du säger dokumentera vårt arbete. Det är ju inte en dagbok som förs löpande.              |
| 144. | Jag tänker kod, om den är jättedokumenterad om tex någon annan vill sätta sig in i den?   |
| 145. | Där är det ofta mer att önska, oavsett, att man som ny utvecklare kommer och kan läsa i text hur allt hänger ihop och alla funktioner, likasåväl som i koden. Kod kan alltid kommenteras bättre. Tror aldrig jag har läst kod som har varit tillräckligt bra kommenterad. Haha.   |
| 146. | Det kan ju överlappas av informell information, ser du det som ett problem att det saknas dokumentation?  |
| 147. | Jag ser inte det som ett problem i sig, därför det förhindrar inte våra nya som vi tar in att få progress. Men det gör man ju ofta för man sitter brevid någon som kan peka i koden där det finns tillräckligt med information för att sätta sig in i det. Men det är inte en sån "Välkommen, här är ett 800 sidigt   |

|      |   |
|------|---|
|      | kompendium som du ska ta dig igenom”, det finns ju inte.  |
| 148. |   |
| 149. | Om vi tittar på tidsestimeringar, blir de pricksäkra?   |
| 150. | Jag tycker ju de blir bättre och bättre på att tidsestimera, i våra sliding windows vill vi ju gärna få till det och 80% konfidens i det kommande beatet. Och där är det ganska bra träffsäkerhet. Sen är det ju inte alla saker man tar i som blir mindre, det är ju snarare så att om man börjar gräva i nånting brukar det ju svälla. Jag tror inte jag har en så länge, under dessa 18-19-20 år som jag jobbat träffat en mjukvaruingenjör som är klar tre veckor innan. Ofta är det att man är optimistisk.  |
| 151. | Upplever du time-boxingen som ett problem?  |
| 152. | Det är inga problem.  |
| 153. | Hur tror du Scrum påverkar kvaliteten på produkterna?   |
| 154. | Om du med kvalitén menar att det inte finns några förbättringar ytterligare att göra så kan kanske lägga skulden eller förklaringen på att det är Scrum vi kör som är varför vi inte gör tillräckligt många av förbättrings sakerna man skulle vilja göra. Jag tror det är ganska klassiskt egentligen. Oavsett, även i ett vattenfall, där du har definierat vad du i huvudsak ska göra, sen när du i klar med det utveckla det, “För tusan, där är ju grejer...” och då börjar nästa projekt. Jag tror inte att man kommer irkigt undan att teamen lägger sin själ i att få fram en feature som dem tycker är så bra som möjligt, det kanske är det bästa med att om teamen är kommunikativa, det här är tillräckligt bra men vi bör få in detta också. Om det då gör så att driver långt in i nästa milestone, så får man ta upp det för diskussion om vi är nöjda där eller om det är mindre saker som är relativt snabba saker så fixar vi det bara. |
| 155. | Men kommunikationen går på båda hållen? Ni får dels era missions uppfrån, och sen om ni upptäcker något så går det på andra hållet?   |
| 156. | Absolut.  |
| 157. | Hur ser det ut med arbetsbelastningen? Finns det stressmoment?  |
| 158. | Det beror nog helt på vem du frågar. I Teamen så bedömer jag det som en hanterbar arbetsbelastning ur alla aspekt. Det är inte mycket beordrad övertid. Det är i princip inget jobb på helger. Och när man är här är det full rulle som gäller, det kan vissa tycka attt man har en lite, vad ska man säga, en dynamisk tillvaro när man är här. Som testutvecklare eller som utvecklare tror jag ändå att vi försöker ha dem så ostörda som möjligt, så de kan göra så mycket som möjligt utan contact visuals. Sen blir det nog mera mer, dynamik ju mera upp du kommer. Men för teamens del tycker jag absolut det är en bra arbetsmiljö.  |



|      |  |
|------|--|
| 159. | Och sprintarna eller beatsen är jämt med arbetsbelastningen, det är inte så att det går upp jättemycket inför releasen?  |
| 160. | Ja men det som med teams som är lite positiva eller lite överoptimistiska, då blir det ju lite mer dynamiskt mot slutet när man ska knyta ihop säcken. Men det är inget jättekonstigt.   |
| 161. | Hur vet ni att ni gör rätt? Om ni håller rätt hastighet?   |
| 162. | Asså, vi har egentligen här och nu har vi inga mått på hur många pungråttor eller kameler som teamen tuggar per beat. Det har vi inte. Däremot gör vi t-shirt-sized på allt vi bryter ner. Och vi påbörjar inget som har en större t-shirtsiz än large. Vilket vi motsvarar ungefär 20 mandagar, 15-20 mandagar. De flesta taskarna är i storleksordningen small, som är mindre än en manvecka. vi har inte börja sätta KPI:er på hur många av de taskarna vi gör i snitt varje beats. Det kommer vi göra. |
| 163. | Så tanken är att man ska ha en måttstock?  |
| 164. | Man ska ha en måttstock på vad varje kan tugga.  |
| 165. | Och samma sak med kavlitén? Har ni en måttstock på kvalitén?   |
| 166. | Ja asså måttstocken är ju snarare där du får, systemteståterkoppling att något inte fungerar i värsta fall. Eller att kunden som kunden sen upplever problem. Och då återkommer det ju rakt in till oss.   |
| 167. | Blir det då högsta prioritet?  |
| 168. | Absolut.   |
| 169. | Det går före?  |
| 170. |  |
| 171. | Vi gjorde ju så fram till slutet på förra året att vi hade flera olika team som hade både feature utveckling och hantering av buggar och förbättringar. Men det blir väldigt task-switching när det brann buggar som kom in med 20 meter höga lågor så vi gjorde så att vi har tre renodlade feature teams och ett kvalitetsteam som tar hand om de brinnande buggarna, har det som fokus..  |
| 172. | Så då är det det bara det som de jobbar med?   |
| 173. | Precis. De taskswitchar mellan de olika buggarna, men de har inga leveranser till mer värde genom featuretillväxt.   |

|      |   |
|------|---|
| 174. | Men då är det så att det behövs ganska mycket kommunikation mellan det teamet och de andra teamet? Som har utvecklat funktionen från början?  |
| 175. | Ja kommunikationen brukar vara så att har man som featureteam färdigställt en feature som man levererar i produkten som har man i sin tur får buggar på, så är det featureteamen som gjorde den featuren vid det tillfället. Dem rättar också buggar. Så man kan inte koda dåligt och sen är det ens polare som rättar upp grejerna.  |
| 176. | Det finns fortfarande en ax till limpa...   |
| 177. | Den kommer tillbaka till dig.   |
| 178. | Att ha det här helhetsansvaret, tror du det är nånting som påverkar kvaliteten positivt?  |
| 179. | Ja det tror jag. Men snarare tänkt när man själv sitter med produkten och använder den innan att det ens är tänkt att det ska komma ut till kunden. Är man då nöjd med den nya featuren som egen användare så vill man ju gärna också att nån ska tycka det är bra, och man jobbar det här lilla extra. Nu lägger vi på nått testfall här för att testa de här helt crazy grejerna som nån kan göra. Man kanske lägger på det utöver vad man hade tänkt från början.  |
| 180. | Och det kanske kopplar tillbaka till det här att man snabbt får feedback på vad man har gjort?  |
| 181. | Ja, det är väldigt tydligt.   |
| 182. | Då tror jag vi är rätt så klara här. Om inte du själv har något att fylla i?  |
| 183. | Nä, det vet jag inte vad det skulle vara.   |
| 184. | Finns det något du skulle vilja ändra?  |
| 185. | Ja du, det beror nog lite på vilket ställe man är på. När man har, om hela världen hade jobbat efter vanilla Scrum tror jag inte det hade varit bra i alla lägen. Men det är inte bra heller att jobba strikt vattenfall. Där man har möjlighet att jobba agilt så är det också en stor fördel för dem som är inblandade. Sen inser man också att det kan vara svårt att sälja saker längre fram i tiden. "Ja men till 3.0 så är det det här som kommer" och sen ska man rent försäljningsmässigt lova bort något som man inte riktigt vet om ett R&D team orkar med att leverera förään det börjar närma sig. Det kan ibland vara svårt. |
| 186. | Organisationer blir ju väldigt platt, att det är svårt att se sin egna karriärstege?  |
| 187. | Ja, det kan det väl vara. Men jag ser inte det något beskymmer egentligen. Inte den aspekten.   |

|      |  |
|------|--|
| 188. | Det kommer ju med många andra fördelar att det är platt. |
| 189. | Ja, fördelarna överväger tycker jag.                     |
| 190. | Alright, då tackar vi så mycket.                         |
| 191. | Ja, bra!   |

## 7.7 Intervju 4

|     |   |
|-----|---|
| 1.  | Din nuvarande titel?  |
| 2.  | Jag jobbar som Scrummaster men också som utvecklare   |
| 3.  | Så du är både och i Scrumteamet? Så du har ett team som du är scrummaster för?  |
| 4.  | Ja. Innan hade vi så att man var scrummaster heltid men nu är det så att man får utveckla också.  |
| 5.  | Men var du bara scrummaster då?   |
| 6.  | Innan? Nej då var jag bara utvecklare.  |
| 7.  | Har du varit på någon annan position än utvecklare och scrummaster?   |
| 8.  | Nä bara dom två.  |
| 9.  | Så du har gått från bara utvecklare till att nu vara båda två?  |
| 10. | Mm.   |
| 11. | Har du jobbat med Scrum sen tidigare?   |
| 12. | Japp, vi började sedan bmod-dagar, när vi jobbade med mobiltelefoner nu har vi flyttat till basstationsidan. 2013 tror jag att det var.   |
| 13. | Men det har varit på Ericsson alltihopa? Eller det har ju bytt.   |
| 14. | Ja det var inte faktiskt, då var vi en annan business unit. Så vi fick inte agile som resten av organisationen, vi ville prova något själv. Så när vi började var vi ett slags pilotprojekt för dom i lund. För att se hur det funkade. Vi hade inte bestämt oss här att börja med den (metoden). Efter 5-6 |

|     |   |
|-----|---|
|     | månader tog de beslutet att vi skulle vara agilt.   |
| 15. | Har du jobbat med någon annan arbetsmetod?  |
| 16. | Japp innan 2013 hade vi ett traditionellt sätt att jobba.   |
| 17. | Är det några uppenbara skillnader?  |
| 18. | Ja där hade man komponentteams och man var bara ansvarig för sin egna grej inte hela kedjan. Då va det allt som typ man läser i böcker och teori vi upplevde det också. Så det var typ om man är först i kedjan och när man är klar pushar man till nästa och sen finns det ingen kommunikation mellan de två. Men när det här teamet började jobba så hittade det något helt nytt så då behöver man ändra design helt och hållet så man var inte inblandat i designfasen. SÅ det var många sånna olika problem.  |
| 19. | Så du trivs bättre med Scrum som arbetsmetodik?   |
| 20. | Ja det är bättre  |
| 21. | Hur jobbar ni med Scrum? hur ser arbetet ut? hur ser organisationen ut?   |
| 22. | Jag vet inte hur de jobbar i kista eller i Linköping men här är det så att vi är en del av en stor RA heter det . Och sen har man samma requirement area, organisation jag vet inte riktigt vad man ska kalla den, typ lund Ottawa eller någon annanstans och där har man en stor backlog med features och sånt och sen typ om vi har 5-8 teams här så bollar man någonting till lund. Det är dom här features som vi kommer jobba med och deadline brukar vara slutet av året eller nästa sex månader. Och sen har vi vår egen backlog som skapas för oss i lund. Alla teams här dom säger att det här är prioriterat i en lista ni kan sätta önskemål och ta den här och den här.   |
| 23. | Är det per avdelning då?  |
| 24. | Inom RA har man redan tagit beslut över vilka features som kommer till lund. Då har vi olika avdelningar.   |
| 25. | Det delas bara in efter kunskapsområde?   |
| 26. | Ja. Så då har man en feature och sen ja här är dom inte inblandade. eller dom är inblandade på en nivå som inte är synlig för teams i alla fall. Sen kan det vara så ibland att särskilt nu händer det ganska ofta för innan i början var det så att vi fick features som var typ 2-3 år och då jobbade dom litegrann som förut. Det skulle vara systemerat och sen pushar man RD eller utvecklings teams och testning. Och det är fortfarande så att system inte är som en del av vår agilteam. Så det är inte typ att i teori är det så att man skulle ha alltid en crossmixed team. Men så är det inte. Kanske är det inte möjligt heller. Att man har allt där- Så vi har bara utvecklare, testare och sen alltid vår software development område allt som vi kan ha. Men inte den systemisation del. Jag har inte riktigt koll där |

|     |  |
|-----|--|
|     | men jag tror inte de jobbar agilt heller. Så de försöker vara klar med hela features som de har. Sen skickar dom vidare den till oss. Men nu händer det så att det inte finns så mycket tid så man får requirements snabbt och man behöver den här och man behöver den här så då får man en chans att jobba tillsammans med system.  |
| 27. | Men tills ni får era tasks eller user stories då kommer dom från att man har bara valt R&D och sen delas det upp man tar vad man vill från den här gemensamma backlogen?   |
| 28. | Dom är blandad för varje feature system. Och dom jobbar på en helt annan.  |
| 29. | Jag tänker på hur ni jobbar?   |
| 30. | När den kommer till oss är den redan klart och estimerat. Då försöker vi förstå vad som står i dokumenten, ställer frågor. När du börjar med den börjar du läsa den och förstå för hur man ska implementera för sen bryter den ner i OPOs i user stories. Dom user stories som vi kanske kommer att jobba med i nästa sprint. Eller stora delar bryter vi ner de i. Sen har vi, bryter vi dom nästkommande typ. Som gäller för nästa sprint så detaljerat som möjligt.   |
| 31. | Följer ni Scrum principfast eller använder ni er av andra arbetsmetodiker?   |
| 32. | I början hade vi som Scrum by the book men typ 2013 den jag nämnde. Men nu så leker vi litegrann med det också. Om det följer Scrum så följer vi några huvudregler i Scrum. Varför behövs den här och varför behövs inte den här. Vi använder litegrann Kanban ibland. Så vi försöker se och sen olika ceremonier eller artefakter som vi tror. Vi försöker hålla den så enkelt som möjligt  |
| 33. | Vaför har ni blandat in Kanban?  |
| 34. | Det beror på vad man jobbar med. Vi har jobbat med Kanban också. Och när vi började jobba med Kanban blandade vi in lite från Scrum och sen följer man bara Kanban by the book så finns det inga dailymeetings och jobbar man bara med Scrum så kan det bli så att man bara jobbar med troubleshooting eller det blir typ slutet av eller om det är några dokumentationer eller sånt. Så kan man flytta för det är stora delar trouble reports och man vet inte om det är sådana taskar så man tror inte det går att planera då kan man byta till Kanban. Det har vi gjort för enkla skäl. |
| 35. | Din roll som Scrum master vad är svårast att hantera?  |
| 36. | Nu har vi ändrat, by the book är det så att man måste försöka låta inte någon annan störa teamet och sånt och ta bort impediments osv. Men i vårt team har vi valt så att, teamet har bestämt själva att vi kan ta upp impediments själva så vi behöver inte ha, vi vil inte att du gör den så det är inte riktigt. Vi har blandat den lite grann eller ändrat den. Svåraste... i början var det med att timeboxa våra möten.  |
| 37. | Asså estimeringen av tasksen?  |

|     |  |
|-----|--|
| 38. | Nä asså vi har retrospektivmöten, sprintplaneringen och daily stand ups och man måste ha dom och har vi sagt att vi ska ha sprintplanering i 3 timmar och retrospektiv inom en och halv timme men vi hade problem för vi hade diskussioner och det var lite svårt. Då bestämde vi eller vi tittade på det i retrospektiv och hur ska vi göra? Och jag tror det är bra nu. Sen nästa sak är att det svåraste är typ att man är beroende av, det kanske är en fördel men det kan vara en nackdel också, att man är beroende av teamet. För en Scrummaster har ingen makt. Så det beror på vilka människor man har i teamet, vilken filosofi har dom, är de lätta att jobba med. Så det blir, ja, lite frustrerande. Så det är två grejer även om man tror att man måste övertyga. Om man tror att något inte är bra och man läst om det i böcker eller hört från andra organisationer så måste man försöka övertyga teamet. Även om det är något man har bevisat kan man inte få in det i teamet så man måste försöka övertyga.  |
| 39. | Det tar lång tid från att man har en ide till man kan implementera den?  |
| 40. | Mm. Och man har folk som ibland har svårt att följa scrums grundregler. Det är några som bara kan se att det är dumt eller att de inte vill. Det har inte hänt i mitt team men det har hänt här och där.   |
| 41. | Vad upplever du som de största fördelarna med Scrum?   |
| 42. | Att alla är inblandade, testare, crossfunctional teams, det är bra i jämfört med det traditionella sättet att jobba. Man kan hitta problem snabbt. Man gör ingen detaljerad planering, man är flexibel, man är agil. Sen tror jag det är bra att man, det är lätt en mer synlig än det var innan och estimeringar är bättre eftersom det görs av teamet. Och sen tar man ansvar för det man säger att man ska göra. Retrospektivt är ganska bra tror jag. Så man sitter innan i vattenfall har man aldrig sånt. Här är det vi själva som jobbar med processen och ser hur vi kan förbättra osv.  |
| 43. | Finns det några negativa aspekter? Något som du upplever som en nackdel med arbetsmetodiken?   |
| 44. | Nää jag vet inte faktiskt. Jag kan inte komma på något   |
| 45. | Men om man jämför t.ex. med vattenfall, många av de problem som finns där finns de kvar? eller försvinner de med användandet av Scrum?   |
| 46. | Förresten en annan sak till det är lite annorlunda nu men i vattenfall eftersom du nämner den en fördel vi hade där var att man var expert i sitt område. Man visste inte så mycket om andra grejer men man visste om sin del. Här, by the book, behöver man ha T—kompetens, man måste vara expert i ett område och sen man måste kunna vara generalist i resten. Det funkar inte. Det funkar inte så bra faktiskt. Jag har tänkt i mitten också om det är ett nytt team och man sen gör nya grejer var sjätte månad och sen även där man hoppar olika roller, typ nu försöker jag göra lite testning och något annat. Där tror jag man bara blir generalist inom Scrum. Jag ser det som en nackdel, dom i boken säger att man ska göra så men jag har inte sett en sådan implementation här. VI har redan sådan t-kompetens eftersom vi jobbade i sådana kompetensteam men om du går tillbaka 3 år när vi började med Scrum ser jag ingen utveckling. Man kan göra saker snabbare för det är snabbt men man blir ingen expert. Eller det finns ett beroende av några experter fortfarande när det gäller detaljerad kunskap. SÅ det är en nackdel. Kanske finns det redan i teorin hur man löser det? men det kan kanske vara att man läser eller att man sitter själv och fördjupar sig i det här. Men det funkar inte faktiskt. Man har ett stort krav man ska vara snabb och leverera. En annan sak är att man inte får lika mycket tid att lära sig inom Scrum. Om man har högprioriterade features, en efter en annan, har man |

|     |  |
|-----|--|
|     | inte så mycket tid att sitta och gräva och försöka veta mer. För att göra den snabbt så lär man sig bara vad som är viktigt och vad som behövs för den implementation eller delen man ska göra. Det är en annan som kanske kan lösa den men det gör, de säger i teorin att man ska ha tid för att lära också men det blir låg prioriterat när det väl gäller. Båda grejer samtidigt.   |
| 47. | Finns det någon förändring du skulle vilja göra?   |
| 48. | Ja det jag ville, fast jag inte kunde övertyga mitt team, jag tolkar det annorlunda. Jag tror man behöver sån här T-kompetens men de andra i mitt team tror inte på det. Det kanske inte ser det som ett problem. Det ser det som bra att vara generalist för saker förändras snabbt så man kanske inte kan ägna tid åt att lära sig något. För nästa år kanske man jobbar med en helt annan grej som kanske är högprioriterad av kund. Och de har rätt också så då får man hitta något nytt kanske hur man löser det problemet. Sen problem som vi har haft själva, jag kunde inte dela och man blir stressad av att jobba med högprioriterade grejer. Jag kunde inte fokusera på mitt jobb som Scrummaster för jag ville bara fokusera på min sak som utvecklare. Så det blir lite svårt att ha en balansgång där. Vi försöker ju. Vi har tagit upp det i retrospektiv. Så vi kommer göra något åt det, särskilt om Scrummatterns grej.  |
| 49. | Hur många ingår i ert Scrumteam?   |
| 50. | I början var vi ganska stor, jag tror de flesta är 6-7 men i början ar vi 8. Men nu är vi 5.   |
| 51. | Hur tycker du skillnaden är mellan att jobba 8 och 5?  |
| 52. | Man får bara veta hur man ska hantera eller vad är potentiella problem som kan uppstå med ett stort team och ett litet team. OM vi hade 8 så var daily standups större och då anpassade vi oss istället. Det finns två sätt att göra daily standups, personvis och taskvis, så vi försökte gå taskvis. Sen kunde man som 8 parprogrammera mycket. Även om vi jobbade två och två så hade vi 4 tasks, men om man är 5 så är man för få. man kan inte bara jobba med två task. Annars var det inga problem   |
| 53. | Hur många ansvarar för backlogen?  |
| 54. | Vi har en backlog på högre nivå en teamnivå för alla på vår avdelning där har vi en backlog. Den heter produktbacklog och det finns features som vi pullar där ifrån. När vi har pullat en feature så äger vi den då bryter vi ner den till mindre delar. Så vi har inte som det finns i boken att man måste ha alla user stories prioriterat och man pullar de högsta därifrån utan vi har backlog refinement meetings så vi försöker se till att vi har något att göra i nästa sprint. I första backlog refinement vi försöker bryta ner. Vi kanske behöver det jhär den här user storken osv. Det finns ingen detalj där ingen estimering på den men sen senare säger vi att vi behöver bryta ner dom fyra fem så att vi har något att pulla i nästa. Och sen för denna sprint så säger vi att vi kommer ta dom här. och sen andra user stories de ligger, det blir inte backlog, de ligger i anomin, hur de kommer se ut bestämmer vi. Det kanske skulle bli annorlunda om man är mer teams. En fördel för oss är att vi har synkroniserat våra sprinter så alla blir klara samtidigt. Om vi jobbar mer än ett teams på en feature så behöver vi bara ha en output innan vi börjar nästa sprint. De här user storiesen kommer vi jobba med nästa sprint. |

|     |  |
|-----|--|
| 55. |  |
| 56. | Om sprintarna går parallellt hur kommunicerar man mellan teamsen? finns den en formell dokumentation eller är allt informellt?   |
| 57. | Det finns dokumentation men det är inte för kommunikation. Vi får ett dokument från system management med requirements, det är en typ av diskussion så att vi förstår vad som finns i features och vad som behövs. Men om vi jobbar tillsammans om vi är t.ex. två teams då sitter iv bara i backlog refinements tillsammans och försöka bryta ner och kommunicera där. Sen kan det vara om man jobbar i samma område eller om man har brytt ner user stories så de är oberoende av varandra kanske man inte behöver kommunicera. Men om det finns beroende då går vi och pratar med varandra, så det finns ingen kommunikation där utan den är informell.   |
| 58. | Tycker du det saknas dokumentation?  |
| 59. | Nej, jag tycker vi har lite mer än vi behöver.   |
| 60. | Hur förmedlas övergripande mål och strategier? inte för en sprint utan snarare långsiktiga mål.  |
| 61. | Vi har olika DUs och sen på en större nivå så har man en strategi, vad vi ska sikta på från kundens perspektiv och roadmapps osv. Och sen när det kommer ner på avdelningsnivå försöker dom ta in de grejerna. Och sen kommer det ner till teams hur de vill ha sina mål och sen ett mål för nästa år och sånt. Och sen har vi någon typ av releaseplanering också men vi har inte så mycket att göra med strategi utan snarare när vi kommer vara klara med våra features och hur det kommer se ut de närmsta 4-5 sprintarna.   |
| 62. | Har du något ansvar som scrummaster att förmedla de här övergripande strategierna till teamet?   |
| 63. | Nä det kan vara helt annorlunda om ni snackar med någon annan, det finns ingen defined roll att man ska göra så här eller så här. Jag har inte gjort det utan det har varit linjechefen som har gjort det. Vi har haft presentationer på varje nivå om det varit DUs strategi eller avdelningsstrategi.  |
| 64. | Tycker du att det någonstans som kommunikationen brister, att det är svårt att kommunicera med andra teams eller andra avdelningar?  |
| 65. | Vi har inte haft så mycket jobb utanför, jag har varit med i en feature men två releaser. Men det finns problem eftersom man vill uppnå olika grejer. Om system har gjort sin grej och de får inte bara systemering dokumentation utan simulation också. Då vet vi hur det betar sig så vi hade lite problem där. För det var inte färdigt. Och när vi började jobba och behövde lite saker från dom. Vi ville ha det sprintmässigt och inte hela grejen samtidigt. Utan bara den delen som behövdes för vår user story. Men deras sätt att jobba var att göra hela featuren och kanske några grejer som vi kommer behöva nästa två-tre månader. Så det kan vara ett problem om man har olika mål. |



|     |  |
|-----|--|
| 66. |  |
| 67. | Men är det också kanske så att eftersom det är olika arbetsmetodiker som det skär sig?   |
| 68. | Ja men sen man kanske behöver simulationer innan färdiga. Vi hade några grejer som man ska jämföra med kod och det var simulation. Senare har vi några testavdelningar so mkör test oc hed har sin egna backlog så när det går utanför vår avdelning eller team är vi mer beroende av deras backlog och prioritet där.   |
| 69. | Har ni distruberade lag? Har du jobbat med det? alltså personer på olika platser geografiskt.  |
| 70. | Nej.   |
| 71. | Ingen erfarenhet av det?   |
| 72. | Vi hade några som reste för att lära sig om teams. Några av oss var tvungna att resa. Så inte hela tiden men över kortare perioder.  |
| 73. | Har det funkat eller finns det svårigheter med det?  |
| 74. | Det minns jag faktiskt inte. Men om de är i Kina eller USA tappar man ju tid. T.e.x om man är beroende av något test eller så. Det är ett problem om tester ska göras av en avdelning på annan plats. Så då ibland när man vil ha något testat, skickar man det och man vill ha resultatet snabbt men det finns ett gap på ungefär en dag.   |
| 75. | Om man tittar på extern kommunikation hur kommunicerar ni med de olika intressenterna, de olika stakeholdersen? Hur många stakeholders är involverade i ert projekt?   |
| 76. | Vi har ingen kund hos oss. Så då har vi någon som projektchef som säger vad som är viktigt och har kontakt med kunden. Sen har vi operational product owner. Eftersom vi är en stor organisation har vi en OPO här och sen på huvudnivå också som snackar med kunder   |
| 77. | Så kunden har representanter ner i organsiationen?   |
| 78. | Ja eller dom är en slags proxy som vet vad kunder behöver. Så i början är det de features som vi har i vår backlog så det syns där vad knduer vil ha. Efter det blir det för det mesta vår OPO och avdelnings OPOn som är i kontakt med kunder. Och han är involverad hela tiden. Men det varierar från person till person. I värsta fall kan det vara såå att man ser OPO i backlog refinement och sen i sprint reviewn men i vårt fall är han där hela tiden, han sitter också bredvid och han är där vid varje daily standup. |
| 79. | Hur hanterar ni olika prioriteringar och motstridiga krav? På team nivå  |

|     |  |
|-----|--|
| 80. | Det gör inte vi på teamnivå. jag har inte haft det problemet som scrummaster. Allt har kommit från OPO och då är det färdig prioriterat.   |
| 81. | Hur brukar den informella kommunikationen funka? Talar man fritt inom teamet?  |
| 82. | Ja man sitter bredvid varandra sen tar vi upp grejer i daily standup sen parprogrammerar vi också. Och om det är något kan man alltid ta upp det. Så det är inget problem.   |
| 83. | Ni har ingen direktkontakt med kunderna så ni vet inte hur involverade?  |
| 84. | Nä , men vi har tagit upp det så kanske kommer det att ändras  |
| 85. | Hade du velat att det fanns en närmre dialog med slutkunden eller den riktiga stakeholdern?  |
| 86. | Ja jag tror det hade varit bättre, man vet vad de vill egentligen. Och sen hur den grejen vi levererat funkar. Det är bra att veta vad de var intresserade av. En feature kan ha så mycket grejer så om man pratar med dom så vet man vilken huvuddel de var efter.  |
| 87. | Har ni underleverantörer? Som skriver kod? Beställningsjobb kan man kalla det  |
| 88. | Inte nu längre men innan hade vi   |
| 89. | även under Scrum? funkade det?   |
| 90. | Nä egentligen inte. Det kanske har funnits, men vi har använt oss av konsulter och då var de en del av teamet.   |
| 91. | Är det svårt när man får in en ny i teamet?  |
| 92. | Ja det är alltid svårt. Man blir långsammare, man får se till att den nya personen kommer in och tar sin tid och lära sig processen. Det beror på feature och när personen kommer. Om de kommer i början av en feature kanske det inte är så svårt. Alla behöver lära sig nya grejer för nya features men om det är i mitten av en feature så blir det lite långsammare. Man kommer rakt in i projektet och någon måste lära upp personen. |
| 93. | Använder ni er av färdiga designpatterns eller generisk kod på något sätt? Ni har gjort något tidigare så återanvänder ni det igen?  |
| 94. | Ja om man kan så gör vi det. Men man kan inte göra det hela tiden.   |
| 95. | Så det finns en viss likhet i produkterna?   |

|      |  |
|------|--|
| 96.  | Som organisation vill man att de bygger på varandra, det blir lättast. Sen kan det komma några där man behöver bygga något helt nytt.  |
| 97.  | Men är det ofta att man uppfinner hjulen igen? Scrum vill ju att man ska titta på saker med nya ögon   |
| 98.  | Jag har inte varit inblandad i sådant men det händer när man vill ha en ny produkt som är lite annorlunda. Men även då, den som jag har sett vill man inte ändra all arkitektur om det inte behövs. Det finns risker med ny arkitektur. Kanske funkar den kanske inte. Jag vet inte dom som jag har sett dom försöker ta så mycket av den befintliga. De har testat det och de vet att det funkar                                |
| 99.  | Hur dokumenterar ni arbetet? Dokumenteras det?   |
| 100. | Ja det finns på olika nivåer. Det finns på feature nivå. På avdelningsnivå också. Också systemisation hur de tror att det kommer se ut. Sen när iv jobbar skriver vi hur det ser ut i implementation och hur man ska testa. Det finns dokumentation om olika arkitekтуella enheter och har man lagt till något där får man uppdatera den. Det är en ny grej som jag lagt till.   |
| 101. | Olika nivåer för olika?  |
| 102. | Ja   |
| 103. | Hur pricksäkra brukar ni vara i tidsestimeringen av era user stories?  |
| 104. | Vi har faktiskt haft problem med att estimerar våra user stories. Förhoppningsvis lyckas vi ändra på det men vi har inte varit bra på det , vårt team. Och det var samma grej, det var en ny feature och om man vill estimerar någonting bra så får man veta ungefär vilka metoder kommer den att röra osv.  |
| 105. | Tror du det främst är en fråga om erfarenhet eller systematiskt?   |
| 106. | Erfarenhet hjälper såklart, om man har jobbat och är expert på området så vet man allt då är det inte så svårt att estimerar. Men för nya teams eller i ett nytt område. Problemet som vi hade var att vi alltid fokuserade för mycket på det vi gör just nu och hade ingen tid att lära oss om nästkommande user stories. Så då blev det mycket gissning. Vi hade kunnat ha en bättre bild av det var som skulle komma närmast. |
| 107. | Hur hanterar ni technical debt? teknisk skuld  |
| 108. | Jag tror vi har mer fokus på kvaliteten i början tills vi börjar närma oss deadline. SÅ det finns bra fokus där men när vi närmar oss deadline försöker vi först och främst leverera något som funkar. Det kanske inte är bästa lösning men den funkar och sen kanske man kan förbättra den senare. I nästa release eller uppdatering.   |

|      |  |
|------|--|
| 109. | Finns det tid för det i nästa sprint?  |
| 110. | Kanske inte. Men man brukar ha clean ups också. Sen om man tror att det är något man behöver göra lägger man till den i clean up som görs i slutet av en release. Men det varierar nog från team till team.  |
| 111. | Hur ser det ut med arbetsbelastningen? Över en sprint t.ex. blir det oftare stressigare i slutet eller är det jämt fördelat?   |
| 112. | Cheferna har gjort ett bra jobb med det. VI hade ganska mycket övertid i vattenfall eftersom folk var experter på sina områden kunde inte någon annan slutföra ens uppgifter. men här har de försökt lägga till nya teams vid behov. Sen kan det också hända att det kanske inte är värt att lägga till ett nytt team det kommer inte påskynda processen. I vårt fall hände det så, vi tänkte på vad vi skulle göra och hade det hjälpt att lägga till ett nytt team? eller blir det för mycket tid att lära upp dom? VI hade jobbat 9-10 månader med uppgiften och ett nytt team hade fått det svårt att hjälpa till. Och då fick vi jobba lite övertid. Men cheferna försökte i alla fall men det var inget man kunde göra i det här fallet vanligtvis brukar det hjälpa.  |
| 113. | Mäter ni era resultat? Jämför ni hur det går från sprint till sprint?  |
| 114. | Inte vi som ett team men OPO gör. Han har en burndownchart så han ser hur det går beroende på estimering av user stories. Och efter varje sprint gör han det. VI gör inget sådant. Vi kollar bara sprintvis vad vi kunde gjort bättre eller vad vi gjort fel. Och försöker förbättra den, experimentera med olika grejer. Men ingen strikt mätning.  |
| 115. | Scrum är ganska platt. Hur ser du på Scrum och att det inte finns samma tydliga steg som i vattenfall?   |
| 116. | Det är olika folk som tagit upp det. Personligen tycker jag det var bättre för, det beror från folk till folk. Det har det helt annorlunda på t.ex. toyota. Personligen tycker jag det var bättre på inte Ericsson men ST Ericsson. Där hade vi olika stegar. Vi hade grad 11 till 19. Det var bra. En bra anledning var att vi hade ganska många så man kände att man gick vidare. Sen hade de olika stegar. Här är ledarskapsstegen. Om man varit i grad 15 kunde man välja vilket håll och typ av steg man skulle välja. T.ex. Technical ladder. Då kunde man bli research eller arkitekt. I Scrum finns ingen ledarskapsgrej alls, eller den är väldigt liten. Så det känns svårt. Man ser inte vad som kommer hända hur utvecklas jag. Här har vi inte någon bra plan. Man behöver mindre ledare och mindre chefer. Även om man inte har såna positioner inom Scrum tror jag att man skulle kunna ha såna stegar så man vet vilken nivå man är på. Annars sitter man bara som en utvecklare även om man har ökat sin kompetens som en ledare. Hur de ska göra får man föreslå men det är en förbättring jag kan se. På teknisk nivå, innan hade vi arkitekter och sånt nu har vi bara utvecklare. |
| 117. | Några ser det som ett problem. Man kan inte heller bli expert eftersom man är generalist. OM man jobbar med generella grejer hela tiden blir man ingen expert och det känns inte bra. Vi märkte det när vi flyttade från ST Ericsson eftersom vi hade många experter där som hade jobbat många år och var ganska bra på mobilsidan och kom nu över till basstationssidan. Hur ser man karrärsutvecklingen här  |

|      |   |
|------|---|
| 118. | Tror du det är både för en person som själv att man inte ser en karriär? Men även uppifrån hur man ser utveckling då det inte finns titlar? Vem ska ta nästa steg?  |
| 119. | Jag vet inte hur de bestämmer sig, det kommer bli svårt. Om vi tar ett exempel om någon vill gå till en ledarskapsroll kommer det bli svårt. Det kommer antagligen bli samma gamla folk som jobbat i en sådan roll sen tidigare. För när man sitter och kollar CV:n tittar man, har du någon tidigare erfarenhet som ledare? Så dom som jobbat med det kommer ha det medans de som jobbat med Scrum har inte det pga av Scrum.  |
| 120. | Den tydliga stegen är bra att ha uppifrån så man kan se var folk befinner sig?  |
| 121. | Jag tror det skulle hjälpa. Men det blir mer på en team nivå och teamet leverar en mjukvara och det är det de tänker på och inte en personlig utveckling. Om man läser böcker om toyota där de kom på lean. Där är det helt annorlunda som jag tolkat det. Där jobbar de inte som utvecklare utan vanligt tekniskt jobb. Men där försöker man förbereda dom som ledare. Så varje person som de uppfostrar kan ta rollen som ledare. De har en så stark och bra kultur som de vill behålla som den lär in de nya personerna i den typen av kultur som de har. De gör personen bra. Den gör den så bra att de kan plocka den personen och sätta denna på någon ledarplats. Det är annorlunda här på ericsson för jag vet inte om vi har någon organisationskultur. Om jag var CEO hade jag tänkt på det med möjligheten att klättra på karriärsstegen.  |
| 122. | Ett annat problem som jag nämnde med teams var ju att övertyga teams. Jag hade ett problem. När man har så många teams som jobbar med Scrum som ericsson kan man lära sig mycket av att gå runt och observera de andra. Det blir svårt som scrummastern att jobba med ett team när de inte förstår vad agilt är. Det är majoritetsbeslut i slutändan så det kan inte vara så att scrummastern vet allt och teamet vet inget. Då är det scrummasterns jobb att lära dom men han kan inte lära dom för att de inte vill veta för de tror att det vet allt om agilt. Det beror på över den definition av agilt som finns i deras huvud. Hur presenterar man något nytt för någon som redan tror att de är på den nivån. Och nästa sak vad jag var efter var att gå till andra teams daily stand ups men det var svårt att övertala teamet. För de tyckte inte de behövdes men jag tror det. Det hade varit bra för då kan man lära sig mycket om Scrum eller hur de tolkat Scrum, va de gör som är bra eller dåligt. Men det var svårt att övertyga och det är sådana grejer som är problem. |
| 123. | Problemet ligger alltså i att man måste kunna mycket?   |
| 124. | Team ska ju kunna också. Jag visste att en sak var rätt men det visste inte mitt team. Man försöker förklara att gör vi inte detta kommer detta att hända. Men i slutändan är det teamets beslut. Några av oss påpekade på några saker som vårt team inte mät. Man får inte ge upp. Det är inte lika lätt att ändra på saker som det hade varit om man varit chef. Nu måste man försöka övertala alla. Det är frustrerande när man vet att något är det bästa sättet och de inte tror det. Lösningen är att försöka ta upp teamet till den nivån så att de förstår.   |

## 7.8 Intervju 5

|    |  |
|----|--|
| 1. | Nuvarande titel och vad du gör?  |
| 2. | Jag är operativ produktägare och mitt jobb är att ta features ifrån förstudiefas eller tidig systemeringsfas hela vägen ut till slutverifiering och leverans och produktpaketering alltså möjlighet till lansering hos kund. Det är de jag gör.  |
| 3. | Och det är någon slags produktägare?   |
| 4. | Det är en produktägare i mjukvarudelen. Man har ett ansvar för en hel feature det är alltså någon form av funktionstillväxt inom befintlig mjukvarubas. Hur många Scrumteam har du? Det varierar från varje feature jag har hand om. För tillfället jobbar jag med en väldigt stor feature, en framittdsatsning och där har jag för tillfället två team. Den normala featuren man gör ligger väl någonstans 1-4 teams, vi försöker ju inte hålla det på en så hög nivå som 3-4 team men 1-2 då har man styckat featuren tillräckligt. Bytar man team från ett till ett annat så att vid en ny feature får man ett nytt team? Ja så kan det vara. Vi har valt att arbeta så att vi som är produktägare är feature produktägare, så vi äger featuren vi äger inte teamet. På några sighter är man team OPO. SÅ man har ett team och sen slussar man ner uppgifter till de teamen statistiskt.  |
| 5. | Har du jobbat med Scrum sen tidigare?  |
| 6. | Jag har jobbat med Scrum i tre år ungefär. tre och ett halvt. Har det varit någon annan roll i det? Nä faktum är att när vi började var jag fortfarande i den gamla strukturen. Där vi jobbade med props vår vattenfallsmodell. Och då började vi göra den agila transformationen och då hade jag en roll i transformationsprojektet och den var mer inriktad i integration som continuous integration och dom bitarna, batch delivery. Men min projektledarroll på den tiden var fortfarande som mjukvaruprojektledare. Och då var jag ansvarig för ett arkitekтуellt område inom mjukvarustacken. Men sen 2013 i juni då när vi gjordes om till modems har jag varit produktägare först area produktägare på modems vilket innebär att jag var produktägaren för alla team på en avdelning. Dessa ägde jag backlogen till men jag jobbade inte operativt med scrumteamen med deras sprint i sprint prioritering eller liknande. utan jag satte den övergripande featureprioriteringen mellan features. Nu jobbar jag på en lägre testnivå i featuren gör jag aktiva val. |
| 7. | Men vad är de uppenbara skillnaderna mot props då?   |
| 8. | Största skillnaden framförallt är väl team-ivolveringen. Tidigare jobbade man med ett antal tekniska experter som man intergreerade med systemmänskorna och så systemerade man en ny lösning så fick de här seniora mänskorna som man hade på den tekniska stegen högt upp sitta och fundera och tänka hur ska interfacen se ut hur ska signaleringen se ut på just de här punkterna och sen så lämnade du något som var rätt sår nerbrytet till bara ett antal individer i princip som gjorde var sin del. Det var inget sanck om parprogrammering eller sådana saker en person kunde en domän och en annan person kunde en annan domän och sen så plockade man bara ihop dom på någon slags vertikal så man fick ihop kedjan utan att tänka på att det skulle vara ett samarbete och integration mellan de individerna.  |

|     |  |
|-----|--|
| 9.  | Finns det skillnader inom organisationen? Stora skillnader? eller ungefär samma tank?  |
| 10. | Idag jobbar teamen varandra väldigt likt. I alla hus. De är involverade och de vi försökt jobba in, nu kan jag motsäga mig själv i jämförelse med den tidigare metodologin men vi har någonting som vi kallar för systemspearhead, någon eller några som går inte lite föra alla team medlemmar med system. För att vara litegrann en informationskälla tillbaka till teamet så de kommer upp snabbare på banan. Men i övrigt sen så är det hela teamet som är involverade i nedbrytningen, den detalerade designen och allting som kommer löpande.  |
| 11. | Så det är mer helhets ansvar för teamen nu för tiden?  |
| 12. | De har ju inte längre någon att säga att de var de som skrev för studien färdigt sen ska vi bara implementera utan aktiva beslut måste tas av teamet isäg. Det är inte längre bara hur ska en if-sats skrivas utan hur ska de här hela entiteterna prata med varandra i olika user cases. Det är en jätteförändring. Trivs du med? Jag tyckte det känns bra det blir ju såklart mer högljudda diskussioner och det blir kanske en mer engagerad dialog som sker. Tidigare var det mer så att experterna har systemetiserat upp det här nu gör vi det, stöter vi på en riktig roadblock då får vi bara hoppa över den eller ta in experterna igen. Nu är det så att man ska beta ner dom gemensamt så att alla har en förståelse för varöfr vi gör det. |
| 13. | Följer ni scrumprincipfast?  |
| 14. | Vi är lite mer flexibla skulle jag säga. Vi har teams som har jobbat med kanban i vissa sprintar. Oftast när man är väldigt tidigt i en feature och förstudien är så pass omogen eller man har inte så mycket konkret material än så länge att man kan börja implementera. Då kör man kanske kanban i stället men vid en punkt så vill man ändå säga nu går vi över till en vanliga Scrumkadens. För nu kan vi refinera ut user stories som vi kan sätta exit kriterier på och nu kan vi börja jobba enligt dom. Så att nä det är väl en viss dynamik.   |
| 15. | Det finns en produktägare som är för en viss antal team, och sen en Scrummaster för varje team. Och den är fast i det laget och som jobbar för det som Scrummaster och utvecklare? eller är man unik endast som?   |
| 16. | Nä. Generellt sett tror jag inte det finns något team som har en unik Scrummaster som bara har den rollen som agil facilitator. Dom är allihopa medlemmar i teamet.  |
| 17. | Finns det skillnader mellan avdelningar i Lund? Vad gäller att följa Scrum principfast?  |
| 18. | Jag skulle inte säga avdelningar i sådana fall är det nog team till team. Vissa team är mer agilt principfasta och ifrågasätter mer hur de ska göra. Mycket av det här retrospekt tänket att man varje sprint ska våga utmana sig själv och att göra det lite annorlunda för att se om man kan göra det lite bättre och lära av misstagen. Och det tror jag att teamen är på väldigt olika nivå.   |
| 19. | Så det är vad man trivs bäst med? Ja Så det är inte så här givet att jobbar man med det här så   |

|     |   |
|-----|---|
|     | kommer man också ligga åt det här hållet? Utan trivs laget med det här så fortsätter man med det?   |
| 20. | Ja och där är ju en viss risk att man fastnar i ett känt mönster för det är bekvämt men det är upp till varje linjefefer att utmana dom och va en agilcoach på deras nivå och facilitera det.   |
| 21. | Så det finns någon som ändå kan komma in och säga att nu kanske det är dags att pröva på det här..?   |
| 22. | Det tycker jag absolut att det gör det är linjefeferens roll att göra det sen kommer jag rätt ofta med såna här indikationer och tips också och försöker påverka från mitt håll om man ser att något går i stöpet.  |
| 23. | Och linjefeferen är den som håller i en avdlening?  |
| 24. | Nä det finns ju linjefefer på olika nivåer men här syftar jag på en sektionchef som har två teams.  |
| 25. | Vad är svårast att hantera i din roll?  |
| 26. | Det är den tekniska balansen helt enkelt, vi måste hela tiden göra detta just in time och inte översystemisera för tidigt. samtidigt måste vi ha så pass mycket förståelse om det vi ska göra i nästkommande sprint för att kunna sätta rimliga exit kriterier på varje user story och definiera det på en sådan nivå att vi faktiskt kan exikvera på den utan större analysstop som vi kallar det under sprinten. Så man måste vara en man i vågen titta lite längre fram men samtidigt vara med här och nu för att se till att vi gör rätt saker i rätt tid.  |
| 27. | Det är din roll som ska ha det längre perspektivet då?  |
| 28. | Det är absolut mitt ansvar att titta lite längre fram för att se att vi går steg för steg rätt och sen kan jag lägga över det på teamet och säga till dom att dom ska ta ansvar för den nästkommande sprint och se till att de är nöjda med user storisen för det är ändå dom som sätter sin stämpel på att de levererar det.   |
| 29. | Kan det finnas intressekonflikter där? Mellan teamet och dig? I med att ni har olika perspektiv?  |
| 30. | Absolut kan det finnas det, jag har ju många krav som jag måste uppfylla och teamets intresse kanske finns i att de vill göra implementationen och sen så göra någon slags verifiering och sen är vi nöjda där men för mig handlar det om ett antal olika produkter jag faktiskt ska leverera sen kunddokumentationen, vi ska göra sekvensdiagramen klara liknande. Vi har en kundorganisation vi har sådana här guardians för koden vi ska hålla allting i balans sen ska vi ha feedback på våra system sen måste jag också rapportera till programmet som i slutändan köper den här featuren av mig på något sätt. Det är dom som har gett mig resurserna att utveckla den så jag måste hela tiden ha material nog för att föda dom. Där kan ligga vissa intressekonflikter i det |
| 31. | Hur hanterar man en sådan konflikt?   |



|     |  |
|-----|--|
| 32. | Ja vanligtvis får vi ju ha en tydlig prioritet och det är ju mitt yttersta ansvar att sätta en tydlig prioritet och en dagsordning för den och när de väl går in i sprinten är teamet självbärande och självorganiserande och då får ju jag egentligen inte pilla på något sånt utan de ska bara lyssna in och när de ber om hjälp för någon riktig impediment och då ska jag bara göra allt för att få det att försvinna. Men där i mellan och till näst kommande och näst näst kommande måste jag jobba väldigt aktivt med att se till att vi gör rätt saker i rätt ordning för att möta dessa intressenternas krav.   |
| 33. | Vad upplever du som den största fördelen med Scrum?  |
| 34. | Jag skulle säga att de är ju ansvarstagandet, att du delar ut det på så många individer och du skapar inte längre en hierarkisk kedja där man alltid kan skylla på ovanstående led utan du blir på något sätt den yttersta ansvarig från första stund. Så kan man säga att när allt är färdigt, visst är det jag som produktägare som står för allting men teamet är fortfarande de som gjort hantverket och de vilar tungt på deras axlar att det som kommer ut är en fullgod produkt och att vi möter alla krav som ställs. Det är en stor skillnad. Tidigare om man jobbade i småprojekt som gick upp i störreprojekt. Då satt jag i ett storprojekt och hade 5-6 småprojekt under mig. Då var det varje sån delprojektledare som jag pratade med och sen assigna han/hon bara ett antal individer och sen så var det den delprojektledaren som egentligen höll ihop det som kom och sa nu går det inte så bra eller nu går det jättebra och var så god här är min lillabit och sen ska jag bara baka ihop kakan på toppen. Men nu är det verkligen så ax till limpa. Det ska komma från teamet och vanligtvis finns det inte så många från sidan som hjälper till eller gör något liknande utan de kommer med det. |
| 35. | Finns det negativa aspekter då?  |
| 36. | Det är klart att eftersom det är en sådan omvälvande metodik förändring finns det ju folk som har åsikter gällande vad ger det oss här och nu och vad är min roll i det hela jag vill bara sitta vid min dator och programmera.  |
| 37. | Som jag varit inne på här nu, så ställer vi ett krav för att du ska va med hela tiden, du ska ta ett ansvar både för systemiseringen och för veriferingen du kan inte bara komma in mitt och göra en implementation och sen släppa allting. Så det är klart att det möter ett visst, inte motstånd, men det blir en liten pedagogisk utmaning att konvergera det budskapet.  |
| 38. | Men då är det mer på en individnivå nästan motivering i varför man ska involvera i andra delar?  |
| 39. | Ja och sen så att man ska se det som att teamet inte är en kedja i sig där du har en sysmtemperson en implementatör och en testare utan istället kan alla göra allting och alla tar ansvar för allting.  |
| 40. | Finns det någon form av förändring du skulle vilja göra?   |
| 41. | Det är mycket det här att man önskar så klart alltid det här att teamet ska ta ett större ansvar själva, att de ska våga ta den här rollen och steppa fram. För ofta är det så att många av oss som jobbar som produktägare eller som scrummaster vi har varit med i en stor organisation som krympt till något väldigt litet och det ahr kokat ner till att många haft högre roller och är vanna vid något annat ett tempo som är mycket annorlunda än vad vi har idag. Och då kan det bli lite grann att folk tenderar att dra åt vilja köra detaljstyrning. Det är farligt i sig. Det märker man tydligt, släpper man och låter teamet agera själva, och  |

|     |  |
|-----|--|
|     | låter de vara självorganiserade, får vi ut mycket bättre resultat om någon är nere och pillar i detaljerna. Så att det skulle man vilja önska generellt. att teamen steppar och säger sluta pilla vi sköter detta själva. Jag tycker vi har en mognadskurva som är väldigt lovande för teamen. Åtminstone de jag jobbar med.   |
| 42. | Hur många är det som ingår i scrumteamen?  |
| 43. | Generellt skulle säga mellan 6-8 personer 8 är ovanligt men 6-7 är väl standard talet.   |
| 44. |  |
| 45. | Det känns lagom?   |
| 46. | Det tycker jag, för mig är det så att, vi hade något team som var fem innan och då blir det lite riskfyllt ifall man har lite sjukdomsperioder och sådana saker. Kan ju sluta illa. Är du uppe i 6-7 personer kan du fortfarande ta att 2 personer är borta och få progress och dom som är här behöver inte jobba övertid och sådant.  |
| 47. | Du ansvarar för backloggen helt enkelt?  |
| 48. | Precis.  |
| 49. | Om man tänker intern kommunikation. Har ni någon formell kommunikation? Kanske möten med dagordning?   |
| 50. | Nä det är mest ceremonierna som är formella i sådana fall. Du har backlog refinement, sprint planering och retrospektives det är mest det som är formellt. De är fasta enligt ett schema. Och de som är formaliserade kan i sådana fall vara att man har mycket interaktion med ett prestudie team, alltså systemledningen, för att göra en ny feature kanske du har stående agendor med dom men det är också en kadens som löper. Och den informella kommunikationen? Den tycker jag fungerar väldigt bra. Där är det ju så att vi sitter på olika siter vi kör LYNC, ringer varandra, sätter upp spontana möten. |
| 51. | Teamen här är alla på samma plats?   |
| 52. | Ja.  |
| 53. | Men man får ändå kommunicera med andra på andra platser?   |
| 54. | Ja, vi jobbar ju med många teams ifrån Kista just nu. Då jobbar vi ju mycket med Lync och sånt dela skärm , ibland åker vi dit. Ibland måste man tyvärr åka en del fysiskt också.  |
| 55. | Det finns väl ett mer planerat steg än att knacka axeln på grannen liksom?   |

|     |   |
|-----|---|
| 56. | Nä det krävs ju lite grann men det har gått rätt så bra nu det senaste året. Man bestämmer med ett annat team, sätter upp ett möte kanske kör en videokonferens, vi har en del videokonferensrum, den formella kommunikationen är väl mer att jag som produktägare tar ansvar för ett program och sen har vi milestones som jag måste placera för att visa mognadsgrad och hur långt vi kommit. Då är det mer så här en checklist som man fyller i. Men det är ju mest jag som jobbar där. Teamet i sig jobbar väldigt mycket med det informella.   |
| 57. | Det här på andra platser det är fortfarande i Sverige?  |
| 58. | Ja det är. Visst vi har någon i Ottawa och vi har även i Kina och lite i Polen och så.  |
| 59. | Är det svårare, med språkskillnader eller tidzoner?   |
| 60. | Tidzoner är nog det stora problemet men på någon sätt alla länder har sina sidor men vi är vana vid det. Vi har jobbat många år med det vi hade en stor site i Indien förr i tiden och en i arty bay i Carolina så att det är lite det att man är van vid det. Hur förmedlas övergripande mål och strategier vi gick in lite på att du har lite överblicken fram i tiden. men det kan ju finnas strategier som går ännu högre upp och kommer ner och sådär.   |
| 61. | Vem har ansvar att förmedla dom och hur förmedlas dom till teamen?  |
| 62. | När det gäller såna här saker som roadmaps och var vi är påväg inom R&D vad ska vi göra vilka är slutprodukterna vilken tidshorisont har de här. Då är det klart att det finns budskap som inte jag är den som ska förmedla. Det måste komma från programmet, mjukvaruprogrammet som ska leverera ut i kundorganisationen eller att det ska komma från de högre linjecheferna. Man har ju stormöten eller typ teccdays, alltså föreläsningar via videolänk och såna saker ibland även lokalt. När man pratar om ett visst område elelr en väg att gå. Då brukar vi ha en 3 års horisont, ungefär om 3 år ska detta va gjort, det här är vart världen är på väg. Vi jobbar mycket enligt en spec TGP heter specen. |
| 63. | Men måste folk dyka upp på dem?   |
| 64. | Nä det skulle jag säga är rätt så frivilligt. Vill du vara här och nu så kan du vara här och nu, men du har också en möjlighet att titta framåt. Allt vi gör är via webcast, så det är on demand, så det är när du har tid och lust kan du titta på de.   |
| 65. | Tycker du det är någonstans den här interna kommunikationen brister?  |
| 66. | Inte mer än så att, det klart att det är så om man ska jobba agilt att det är en fördel, kanske om man ska vara hård, att man ska sitta co-located. På samma ställe allihopa, om man ska ha en riktigt bra dynamik i det. Så visst men det är något som är extremt svårt att överbrygga, hur ska vi få alla att sitta på ett och samma ställe. Då hade hade vi väl inte suttit i Lund tyvärr.   |
| 67. | Om man tittar mellan avdelningar, finns det samma typ av internkommunikation? Behövs det?   |

|     |   |
|-----|---|
| 68. | Man försöker ju stycka det ganska väl bara för att folk ska kunna vara självgående i parallell med varandra. Men sen kommer man till en punkt när man måste ha en dialog. Och eftersom vi jobbar med de här featurena hoppas man att man kan enkapsulera det i dessa och de här personer är allokerade att göra samma sak vi gör, då är det helt okej att gå och fråga. Men man kanske får en felrapport och känner att man måste prata med någon som har en annan domänkompetens kanske det är ett lite längre steg. Men då tror jag det är den mänskliga naturen, den personen hade kunnat sitta en våning upp och det hade ändå varit lite jobbigt att fråga. Så det vet jag inte om det har med det agila arbetssättet att göra. Det upplevde jag väldigt tydligt förr också, det blir lite svårt.                                    |
| 69. | Så när man lämnar de vanliga strukturerna så...   |
| 70. | Ja precis. Utanför komfort zonen så är du lite mindre fri.  |
| 71. | Om man tittar mer på extern kommunikation, kommunikation med intressenter, har ni representanter?   |
| 72. | Nä vi har många filter, eller många led. Visst, där kan man ju också se, men det tror jag är en kvarleva från gamla strukturen att jobba mer med vattenfallsdelen, nån gör en förspecifikation där det tolkas vad kunden egentligen har sagt, i ett första läge. Sen tolkas den i ett andra läge, när vi får in den och systemerar upp den. Där är så att vi möter kunden med ett ansikte och sen är det ett helt annat ansikte på dem som gör jobbet.  |
| 73. | Märker man att det tolkas många gånger?   |
| 74. | I sig kan man kanske man kan tänka sig det, men jag tror ju framförallt det är så att originalkravet inte var tillräckligt tydligt från allra första början. Vi har det här problemet, kanske vi kan lösa det på något sätt. man har ingen riktigt fast idé och då börjar folk formalisera ner det och koka ner det och så implementerar man något utifrån det. Tillslut säger man ja det här var ju bra men det var ju inte riktigt det problemet vi skulle lösa, men det här var också intressant. Men vi jobbar rätt mycket efter den här specen, TGPP, och det innebär att saker och ting som är standardiserade är, universella ska jag inte säga, men de är globala. Då vet man att det är de här kraven som har kommit in, och den här featuren som diskuteras, och det här vi ska stödja nästa release på och göra det gemensamt. |
| 75. | Kan det vara så att det ni jobbar med är så svårt att det teamen gör på detaljnivå är omöjligt för kunden att ha en idé om?   |
| 76. | Ja de får ju bara extrahera upp det och säga det här är det övergripande problemet eller det här är ett usecase som vi tror kunderna hade nappat på, vi hade gärna jobbat i det här fältet.   |
| 77. | Hur prioriteras motstridiga krav från intressenter? Finns det en tydlig prioritering?   |
| 78. | Det är olika backloggar. Varje kund, om den är tillräckligt stor har ju en egen pengapåse knuten till det. Där behövs det inte bli en prioritering kring det, utan det finns resurser för båda jobbet. Men motstridiga ur en teknisk synpunkt försöker vi mitegrera det på ett tidigt studiestadie när vi systemerar. Det är väll så vi bemöter det.  |

|     |  |
|-----|--|
| 79. | Är det svårt att hålla alla stakeholders nöjda?  |
| 80. | Ja det klart, det är alltid en utmaning. Det finns ju ofta så att, ja du driver åt olika håll. Har du en marknadsmässig aspekt där det handlar om pengar som ska in i slutändan och du har några som jobbar mycket mer tekniskt och är intresserade av att hålla en snygg mjukvaruarkitektur så är dem två intressena tämligen svårförenliga. Det ena kräver tid och det andra kräver snabbhet och där i uppstår en skärningspunkt som man försöker flytta till ett lämpligt ställe.   |
| 81. | Är det långsiktiga kvalitét mot att implementera nästa feature?  |
| 82. | Långsiktig kvalitét mot kortsiktig lösning. Det är väll så.  |
| 83. | Det kan var svårt för kunden att se fördelen med den ena framför den andra?  |
| 84. | Ja eftersom det inte är den specifika kunden som får gagn av långsiktiga arkitekтуella arbetet så är det ju väldigt svårt att sälja det. Den här blir tre månader sen för att vi ska kunna göra en väldigt snygg implementation som kommer hjälpa oss i framtiden som företag. Den blir svår att konvergera. Men det klart vi försöker ta höjd för det eller så mycket som vi kan.   |
| 85. | Är det överlag svårt att hantera technical debt?   |
| 86. | Nä jag skulle säga att vi har ett rätt väl utvecklat system för vad vi gör. Vi dokumenterar det och sen ser vi fall det är lämpligt att hantera det redan i den featuren som, innan man är färdig med feature implementationen och kan skeppa den till kunden. Kan man ta tag i technical debten eller behöver det inte vara att det har skapat den utan bara att de upptäckt den här är ett hål, här måste vi täppa till. Utan det kan ju vara så att de jobbar och är 80% färdiga och då kommer ett annat team fritt och säger vi ska ändå jobba i detta området, det verkar intressant. Då gör vi den här product caren och fyller igen det hållet. |
| 87. | Det finns en kontinuerlig hantering av technical debt?   |
| 88. | Ja vi har våra sånna product guardians och vi har en tämligen hög, nuförtiden, resurstilldelning, till sånt jobb. Ett antal team som faktiskt enbart jobbar med detta.   |
| 89. | Men detta har man upptäckt att om man inte gör något aktivt åt det så kommer det falla?  |
| 90. | Det kom nog med det agila tänket. Förr hade du domänansvar och de hade sitt lilla område och då var dem ju så stenhårda på att ska det se ut, det ska vara på detta viset, händer något så får vi investera i att fixa det. Men när du lägger ut det på en massa team att du ska jobba, inte över allt, men lite här och var och då blir det lite annorlunda. Så därför har vi det övergripande guardiannätet som ska hålla koll på mjukvaran.   |
| 91. | Har ni någon form av formell kommunikation med era stakeholders?   |

|      |   |
|------|---|
| 92.  | I slutändan när man ska leverera måste man göra en kunddokumentation. Den kommer ut. Den ska godkännas i flera led. De som har systemerat den, de har beställt den. Det klart, den finns där. När jag säger beställt det då menar jag den interna beställaren, inte slutkunden i sig. |
| 93.  | En representant?  |
| 94.  | Ja. Så det finns många delar. Sen vill vi gärna att den ska gå att testa i kundens nät och godkännas även där.  |
| 95.  |   |
| 96.  | Går kommunikationen mot intressenterna på båda hållen? Det kan även vara från teamen och upp? En dialog?  |
| 97.  | Det är ju det som är grejen med att det finns flera lager. Ibland kan det vara lite riskfyllt att öppna upp en sådan kanal. Man försöker skicka upp något och så får man tre saker tillbaka. Det klart att vi försöker få ut det men vi gör det på ett mer internt plan.              |
| 98.  | Har ni underleverantörer som skriver kod?   |
| 99.  | Inte här i Lund. Än så länge i alla fall. På andra siter finns det ju, man hyr in konsulter. Vi hade mycket konsulter förr, men nuförtiden har vi inte det här.   |
| 100. | Även när ni jobbade med Scrum?  |
| 101. | Ja.   |
| 102. | Gick det att involvera dem i processen eller?   |
| 103. | Ja, dem var integrerade medlemmar i teamen. Vi jobbade med väldigt långa konsultkontrakt. Och nu förtiden är det så att vi hyr in hela team. Hela konsultteam. Så gör dem egna jobb och så har de egna produktägare i princip. Helt autonoma.   |
| 104. | De blir en integrerad del av processen?   |
| 105. | De blir en integrerad del av processen, verkligen som ett autonomt team.  |
| 106. | Men om de levererar något som inte håller, faller det tillbaka på dem då?   |
| 107. | Absolut. Man har ett helhets ansvar ut till att det är launchat i nätet.  |

|      |  |
|------|--|
| 108. | Hur ser det ut med så här generiska lösningar?   |
| 109. | Så vi jobbar så generellt sätt att interface ska vara bakåt kompatibla, tillägg ska vara standardiserade så vi förstår dem. Vi har ju inom RAN(Radio access network), det finns ett par interface för att kunna använda olika entiteter från olika tillverkare inom samma nät, då måste det vara avstämt och koordinerat. Så där finns det ju generella lösningar. Och sen så även inom koden, gör vi en statemaskin så ska den vara så snyggt byggt att den inte kommer stoppa. |
| 110. | Men mer, kan man använda lösningar igen?   |
| 111. | Rent kumulativt?   |
| 112. | Ja.  |
| 113. | Jo men absolut det försöker vi ständigt, jag har inget konkret exempel. Vi har en modulariserad arkitektur och i den ingår att interface och funktioner i den ska ju vara så att om man gör en utbyggnad eller tillägg så ska man inte behöva riva eller behöva bygga nått helt nytt utan då ska man bara kunna göra det, adderat. Absolut.  |
| 114. | Så det finns inte det här med att uppfinna hjulet igen?  |
| 115. | Nä den försöker vi verkligen undvika. Det gör ont när man måste titta på realtidssystem för det tar exekveringstid och processorkraft. Optimeringar är dock alltid intressant, lite snabbare, lite billigare.  |
| 116. | Saknar folk struktur till planering av projekt?  |
| 117. | Jag tror det handlar om vilken roll man hade tidigare, vill roll man trivs mer med. Ifall du är nån som vill göra en liten del av kakan och sen veta hur den börjar och slutar, så såklart kommer du att frestas av det. Och det går i alla led, från en testare/utvecklaren till den högsta chefen. Vissa kommer alltid tycka att det är jättebra om man gör sekventiella bitar. Istället för parallelliserade.   |
| 118. | Det blir en plattare organisation, kan det vara så att det är svårt att se sin egna karriärstege eller ha någon att hänvisa till?  |
| 119. | Jo men det är sant. Det kan man se som en liten painpoint. Det är inte lika tydligt längre att du kan gå det här steget, för du är en utvecklare som vill bli en delprojektledare, du vill bli projektledare och du vill jobba i programmet och så vidare. Det är ju inte lika klart. Det finns inte dem rollerna. Du måste sätta mycket större stolthet i att vara en bra utvecklare med lång erfarenhet här. Absolut. Det håller jag fullständigt med om.                      |
| 120. | Hur mäter ni kvalitén på den ni gör? Hur vet ni att ni gör rätt?   |

|      |  |
|------|--|
| 121. | Ja, haha, det är en bra fråga. Det är tyvärr lite mer komplext. Du kan ju skapa tusen KPI:er som ligger och pumpar på de olika legacy regressions tester eller du har code coverage tal som du måste nu upp till. Och vi jobbar med lite olika delar där och ser till att allt detta kokar ner till nånting som känns övergripbart. Men i slutändan är det så att det är folk som har koll på arkitekturen och koden samtidigt som förstår lite grand vart vi är på väg, det är vi måste lita oss till. Sen är det ju sant, vi jobbar i ett realtidssystem med starka hårdvaruberoenden där vi faktiskt säljer en total låda med mjukvara och hårdvara. Vi måste se till att vi håller ström��parnivåer, vi håller exekveringstider, vi följer specar. Så dem sakerna kommer vi aldrig från, att det är rena tal som måste uppnås. Men i övrigt är det mycket regressionstester, man måste se att det byggs på på rätt sätt. |
| 122. | Har ni nått sätt att mäta farten på teamen?  |
| 123. | Velocityn, ja, vi har ju en burn up eller burn down. För mig som produktägare brukar jag ha en burnup för att bränna upp alla poäng tills featuren är färdig. För varje sprint har teamet en burn down där de bränner ner sina timmar i sprinten.  |
| 124. | Hur ser det ut med projekten som görs? Blir dem klara i tid?   |
| 125. | Jag skulle säga att dem är mer i tid nu än vad dem blev tidigare.  |
| 126. | Att det är en erfarenhetsgrej? Att man lär sig estimerar?  |
| 127. | Nja, framförallt, där är ju som jag sagt, du har ju fler personer som är med från början. Och dem som ska utföra jobbet, om du tidigare hade ett estimat från en systemperson som bara satt och systemerade och skulle utföras av en helt annan människa så kommer ju dem inte harmonisera i deras bild av hur stort arbetet är. Men när du faktiskt har teamet inne från början som ska slutföra arbetet så kommer man närmre iallafall. Sen tycker jag generellt mjukvaruutveckling är lite stokastiskt, du har en bild och tror det kommer ta en viss tid, så tar det lite längre tid och ibland lite kortare.  |
| 128. | Så tidsestimeringen är någorlunda träffsäker?  |
| 129. | Jag skulle vilja säga att den är, den är rätt så bra. Men pricksäker och pricksäker, haha. Det finns nog inte ett mjukvaruprojekt som är pricksäkert, då har man nog buffrat för mycket tid på nått sätt.  |
| 130. | Men arbetsbelastningen då, om man tittar över en sprint finns det tydliga stressmoment i den?  |
| 131. | När det är mer ett sprintcommitment, hur moget är teamet. Ifall de sätter en sprintcommitment kommer de jobba gemensamt för att uppnå det. Gör dem det kanske de kommer jobba lite övertid men har ändå en balans i det. Men generellt sätt så brukar de inte behöva det, utan de commitar det de faktiskt klarar. Och det tycker jag har blivit väldigt bra. Sen så kan man som produktägare bli besviken, kan ni inte ta in en userstory till?   |
| 132. | Finns det inte en konflikt där?  |



|      |   |
|------|---|
| 133. | Jo, jag är en kravställare. Jag kan ju komma och fråga hur ser det här ut? Nä den får inte plats, tyvärr.   |
| 134. | Men då är det dem(teamet) som har sista ordet?  |
| 135. | Ja. De äger sin tid.  |
| 136. | Om man tittar på sprinten som så här, blir det snarare övertid i slutet av en sprint?   |
| 137. | Nä, jag tycker det känns som det är rätt balanserat. Redan i vecka två av tre så brukar det börja snackas, behövs det extra timmar?   |
| 138. | Och det blir ytterligare ett led i det här att man har ansvar själv?  |
| 139. | Ja, absolut.  |
| 140. | Vi är intresserade av det du sa med product guradians, var kom den idén ifrån?  |
| 141. | Generellt sätt så brukar man prata om erfarna personer som ska sitta i teamen men ändå ha ett arkitekturellt ansvar. Det är något som jag tror är rätt vedertaget för mjukvaruindustrin. Sen hur explicita för dom är för domän specifika delar, det låter jag vara osagt. För oss som jobbar med ett sånt komplext system så är det tyvärr ett måste att det måste sitta på en punkt. Men annars så tror jag man alltid måste man ha en nån slags sanitetscheck. Man kan inte testa dig ur alla problem, man måste se de här logiska mönstren i mjukvaran. |
| 142. | Är det svårt att leverera continuous? Att man levererar ständigt på något som är igång?   |
| 143. | Det är ju vår ambition, och vi försöker. Men jag tror alla som provar eller håller på med continuous integration, har man inte en hemsida eller en applikation där du kan göra saker oberoende av vartannat så är det väldigt svårt hålla det tåget löpande hela tiden. Spotifytåget är, inte ouppnåligt men rätt utopiskt. När du jobbar med såna här system så, du har du intrekatasytem, och det är svårt att komma runt det, tyvärr.  |
| 144. | Tror du man behöver göra mer modifieringar i Scrum för större projekt, än om du utvecklar bara en enkel webbapplikation, ett mindre projekt?  |
| 145. | Kanske inte med Scrum i sig men med continuous integration, det måste du definitivt. Jobbar du med en webbapplikation eller vilken OS applikation som helst så, så blir det ändå ett jobb som du kan ge, sköta genom att gå upp ett par steg, du behöver ha detaljkollen på det. Det går att leverera in det ändå. Men har du beroenden som är timingmässiga, interfacemässiga överallt, då blir ju varje sprint att du måste tajma, och se till att koden kommer in på rätt ställe så folk kan rebasea, gå ut på nya brancher och leverera in snabbt igen. |

---

|      |  |
|------|--|
| 146. | Så egentligen när det finns många olika beroende mellan många olika team som det behövs planering?   |
| 147. | Ja det krävs mer planering. Men det, jag tror inte det är så mycket våld på Scrum, däremot kräver det mycket av teamen också. Löpande dialog.    |
| 148. | Har du något mer att tillägga?   |
| 149. | Nä, nä, jag tittade igenom era punkter innan och det känns som om vi täckt det mesta. Har ni någon kompletterande fråga får ni gärna höra av er. |

### 7.8.1 Kompletterande frågor till intervju 5

Fråga: Tenderar Scrumlagens kompetens att bli väldigt generell? Dvs försvinner spetskompetensen eftersom alla måste göra lite av varje?

Svar: För att vara lite mer seriös så tenderar Scrumteamen att självreglera kompetenserna genom att välja att arbeta med områdena som de är individuellt intresserade av. Dock finns det självfallet en risk att de individerna inte får chansen att studera in och utveckla sina kompetenser inom specifika områden ifall den funktionalitet de ska utveckla i temaet inte berör "deras" delar av arkitekturen. Sen finns det alltid en möjlighet att vara deltids product guardian och ha allokerad tid till det, men i gengäld bara ha ex. ~50% av sin tid på att lägga på teamets arbete.

---

## 7.9 Referenser

1. Barlow, Jordan B. et al. (2011) Overview and Guidance on Agile Development in Large Organizations. *Communications of the Association for Information Systems*: Vol. 29, Article 2.
2. Codabux, Z. och Williams, B. (2013), Managing Technical Debt: An Industrial Case Study. *4th International Workshop on Managing Technical Debt, MTD 2013 - Proceedings, 2013*, s. 8-15.
3. Dybå, T och Dingsøy, T (2008) Empirical studies of agile software development: A systematic review. *Information and Software Technology 50 (2008)*. s. 833–859
4. Elbanna, A. (2014) "IDENTIFYING THE RISKS ASSOCIATED WITH AGILE SOFTWARE DEVELOPMENT: AN EMPIRICAL INVESTIGATION" . MCIS 2014 Proceedings. Paper 19. <http://aisel.aisnet.org/mcis2014/19>
5. Hossain, E. , Ali Babar, M. och Paik, H. (2009), Using Scrum in global software development: a systematic literature review. *Fourth IEEE International Conference on Global Software Engineering*, s.175-184
6. Hummel, M. (2014). State-of-the-Art: A Systematic Literature Review on Agile Information Systems Development. *47th Hawaii International Conference on System Science*, s. 4712-4721
7. Jacobsen, D. I. (2002): Vad, hur och varför. Om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen. Studentlitteratur, Lund
8. Joey Cho, Juyun, (2010). *An Exploratory Study on Issues and Challenges of Agile Software Development with Scrum*. Diss. UTAH STATE UNIVERSITY
9. Kollmann J, Sharp H, och Blandford A (2009), “The importance of identity and vision to user experience designers on agile projects,” *Proceedings of the 2009 AGILE Conference, Chicago, IL, USA, August 24–28*. IEEE Computer Society, 2009, s. 11–18.
10. Ktata, O. och Lévesque, G. (2009) Agile development: issues and avenues requiring a substantial enhancement of the business perspective in large projects *University of Québec at Montréal 201, avenue du Président-Kennedy Montréal, Québec, Canada*

11. Larson, Eric W. och Gray, Clifford F.(2011). *Project Management - The Managerial Process*. 6. Uppl. New York: McGraw-Hill Education.
12. Lei, H, et al., A statistical analysis of the effects of Scrum and Kanban on software development projects, *Robotics and Computer Integrated Manufacturing* (2015), <http://dx.doi.org/10.1016/j.rcim.2015.12.001>
13. Oliveira, F., Goldman, A. och Santos, V. (2015) Managing Technical Debt in Software Projects Using Scrum: An Action Research. *IEEE International Symposium on Information Theory (ISIT)*; s. 50-59.
14. Penttinen, M. och Mikkonen, T. (2012), Subcontracting for Scrum Teams: Experiences and Guidelines from a Large Development Organization. *IEEE Seventh International Conference on Global Software Engineering, 2012*, s. 195-199.
15. Pikkarainen, M. et al. (2008) The impact of agile practices on communication in software development. *Empir Software Eng* 13:303–337.
16. Ralph, P. och Shportun, P. (2013) Scrum Abandonment in Distributed Teams: A Revelatory Case . *PACIS 2013 Proceedings. Paper 42*.  
<http://aisel.aisnet.org/pacis2013/42>
17. Schwaber, K och Sutherland, J (2013) *The Scrum Guide - The definite Guide to Scrum: The Rules of the Game*. 2014 Scrum.Org och ScrumInc.
18. Scrum Alliance (2016). *Who uses Scrum and why?*. <https://www.scrumalliance.org/why-Scrum/who-uses-Scrum> (Hämtad 2016-04-14)
19. Standish Group International (2001). *Extreme Chaos*.  
[http://www.cin.ufpe.br/~gmp/docs/papers/extreme\\_chaos2001.pdf](http://www.cin.ufpe.br/~gmp/docs/papers/extreme_chaos2001.pdf) (Hämtad 2016-04-07)
20. Sureshchandra, K. och Shrinivasavadhani, J. (2008), Adopting Agile in Distributed Development. *2008 IEEE International Conference on Global Software Engineering*, s. 217-221.
21. Sutherland, J. och Schwaber, K. (1995) Springer Science + Business Media, LLC). *Business object design and implementation: OOPSLA '95 workshop proceedings*. London : Springer, cop. 1997.

22. Tanner M. och Mackinnon A. (2015). Sources of Interruptions Experienced During a Scrum Sprint. *The Electronic Journal of Information Systems Evaluation Volume 18 Issue 1 2015*, (s. 3-18)
23. Tom, E., Aurum, A. och Vidgen, R. (2013) An Exploration of Technical Debt. *The Journal of Systems and Software 86*, s. 1498–1516.
24. Turk, D., Rumpe, B., France R. (2002) Limitations of Agile Software Processes. *Third International Conference on Extreme Programming and Flexible Processes in Software Engineering, XP2002, May 26-30, Alghero, Italy, pg. 43-46, 2002.*
25. Wikipedia (2009) *The Scrum Process*.  
[https://en.wikipedia.org/wiki/Scrum\\_%28software\\_development%29#/media/File:Scrum\\_process.svg](https://en.wikipedia.org/wiki/Scrum_%28software_development%29#/media/File:Scrum_process.svg) (Hämtad 2016-04-28)