



GÖTEBORGS UNIVERSITET

Generella riktlinjer vid distribuerad Scrum

**En kvalitativ studie av hur ett distribuerat projekt
bedrivs med hjälp av Scrum**

General guidelines for distributed Scrum

A qualitative study of how a distributed project conducted using Scrum

**JOAKIM TWEDMARK
DAVID JANSSON**

Kandidatuppsats i informatik

**Rapport nr. 2013:051
ISSN:1651-4769**

Abstrakt

I dagens samhälle är företag globala och kan ha kontor på olika ställen i världen. Det blir då allt mer vanligt att projekt bedrivs på olika platser, så kallade distribuerade projekt, dessa kan vara allt från olika städer inom länder till över nationers gränser. Det används både traditionella metoder så som Vattenfallsmodellen till nya flexibla metoder som till exempel agila metoder. Hur påverkas arbetet när ett projekt är distribuerat, alltså att en projektgrupp är utspridd på olika platser? Detta försvårar möjligheten till att skapa bra projekt, något vi belyser och försöker hitta lösningar på. Aspekter som kan spela in är svårigheter med kommunikation och interaktion samt tidsmässiga, språkliga och kulturella problem.

Kommunikationsföretaget Riksnet är i behov av en ny webbplats. De är intresserade av det agila tillvägagångssättet då de idag inte arbetar med någon speciell metod inom företaget. De vill implementera den agila metoden Scrum vid skapandet av denna webbplats. Scrum lyfter fram individer och interaktion framför processer och verktyg vilket underlättas vid fysisk interaktion. Att Riksnet befinner sig i Umeå och vi i Göteborg leder till att ett sådant projekt blir distribuerat. Detta ger oss följande frågeställning: *Vilka problem uppstår vid distribuerad Scrum och hur påverkar de det agila manifestets riktlinjer?*

För att besvara frågeställningen genomfördes litteraturstudier, intervjuer samt en fallstudie där vi arbetar med Scrum i ett distribuerat projekt. På fallstudien bedrivs aktionsforskning. Utifrån de upptäckter som gjordes i fallstudien ser vi till problem som kan uppstå vid distribuerad Scrum samt om det agila manifestet inte följs. De upptäckter som gjorts ställs mot befintlig litteratur samt intervjuer och utgör rapportens resultat. Upptäckterna ledde fram till generella riktlinjer som är tänkta att fungera för företag liknande Riksnet.

Rapporten är skriven på svenska.

Nyckelord: Scrum, distribuerad, artefakt, aktiviteter, Riksnet, agilt.

Abstract

Today many companies are global and can exist all over the world. It is more common that projects are conducted in different locations, from various cities in a country to nations in different parts of the world. To manage projects both traditional practices such as the Waterfall method and new flexible methods known as agile methods are used. How is the process affected when a project is distributed, when the team is scattered in different places? This makes it difficult to create a good project, something we will discuss and try to find solutions for. Aspects that can affect the project are difficulties with communication, interaction, timing, language and cultural problems.

The company Riksnet which deals with communication is in need of a new website. They are interested in the agile approach as they are not working with any particular method within the project. They want to implement the agile method Scrum for the creation of their website. Scrum emphasizes individuals and interactions over processes and tools, which is facilitated by physical interactions. Riksnet who are located in Umeå while we are located in Gothenburg means that the project will be distributed. This gives us the following question: *What problems arise in distributed Scrum and how do they affect the guidelines of the Agile Manifesto?*

To answer the question we conducted literature reviews, interviews and a case study in which action research was conducted. Based on the finds made in the case study we look at problems that can arise in distributed Scrum and if the Agile Manifesto were broken. The discoveries that were made were put against existing literature and interviews and are the result of this report. The discoveries are leading to general guidelines that are supposed to work for companies like Riksnet.

The report is written in Swedish.

Keywords: Scrum, distributed, artifact, activities, Riksnet, agile.

TACK

Vi vill tacka Shayan Rad och
Anton Lindgren på Riksnet för gott
samarbete och hjälp under fallstudien.

Vi vill också tacka informanterna
som ställde upp och hjälpte oss få svar på våra frågor.

Vi vill även tacka Fredrik Wendt,
Professional Scrum Trainer på Squeed för all hjälp med vår uppsats.

Slutligen vill vi tacka vår handledare
Henrik Sandklef för all hjälp
och motivation under svåra stunder.

AGILA MANIFESTET

Vi finner bättre sätt att utveckla programvara genom att utveckla själva och hjälpa andra att utveckla. Genom detta arbete har vi kommit att värdesätta:

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

Det vill säga, medan det finns värde i punkterna till höger, värdesätter vi punkterna till vänster mer.

(Manifestet för Agil systemutveckling, 2001).

Innehållsförteckning

1. Introduktion.....	8
1.1 Bakgrund	8
1.2 Problem	8
1.3 Syfte och frågeställning	9
1.4 Definition och avgränsning.....	9
1.5 Undersökningens upplägg.....	9
2. Teori.....	11
2.1 Vattenfallsmodellen	11
2.2 Agila metoder	11
2.3 Scrum.....	12
2.4 Artefakter i Scrum	13
2.4.1 Product Backlog.....	13
2.4.2 Sprint Backlog.....	15
2.4.3 Sprint Burndown	15
2.5 Roller i Scrum.....	16
2.5.1 Product Owner	16
2.5.2 Scrum Master	16
2.5.3 Scrum Teams	16
2.6 Planering och möten enligt Scrum	16
2.6.1 Sprint Planning Meeting.....	16
2.6.2 Daily Scrum.....	17
2.6.3 Sprint Review Meeting	17
2.6.4 Sprint Retrospective Meeting.....	17
3. Metod	18
3.1 Aktionsforskning.....	18
3.2 Intervjuer.....	18
3.3 Litteraturstudier	19
3.4 Urval	19
3.4.1 Presentation av urvalsgruppen	20
4. Fallstudie	21
4.1 Riksnet	21
4.2 Fallstudiens upplägg.....	21
4.2.1 Utbildning	21
4.2.2 Sprint 1	21
4.2.3 Sprint 2-5	22
5. Resultat.....	23
5.1 Identifiera problem	23

5.1.1 För många verktyg gör det rörigt	24
5.1.2 Problem med kommunikation.....	26
5.1.3 Product Backlog blir bortglömd.....	26
5.1.4 Problem med Daily Scrum	27
5.1.5 Mejlkonversationen blir lätt rörig	27
5.1.6 Allt tog lång tid till en början	27
5.1.7 Inte våga fråga eller uttrycka sig	28
5.1.8 Sprintverktyget glömdes bort.....	28
5.2 Distribuerad Scrum och det agila manifestet.....	29
5.2.1 Interaktion	29
5.2.2 Individer.....	30
6. Resultatanalys	31
6.1 Förändringar utifrån identifierade problem.....	31
6.2 Agila manifestet.....	32
6.3 Riktlinjer	33
7. Slutsatser	35
7.1 Studiens relevans och överförbarhet	35
7.2 Förslag till fortsatt forskning	35
Referenser	37

Bilaga 1 - Intervjufrågor

1. Introduktion

Introduktionen ger en bakgrund till det undersökta problemområdet. Vi lyfter här fram problem, vårt syfte och vår frågeställning. Kapitlet avslutas med definition och avgränsning samt undersökningens upplägg.

1.1 Bakgrund

Utveckling av nya system samt vidareutveckling av befintliga system är något som ständigt är aktuellt. För att göra detta finns det olika metoder att tillgå, från traditionella metoder som exempelvis Vattenfallsmodellen till nyare flexibla metoder som går under namnet agila metoder. Ett större intresse av att göra utvecklingen av system snabbare och att göra mer med mindre medel resulterar i att agila metoder så som Scrum attraherar många organisationer som vill hitta sätt att utveckla system mer effektivt (Sliger, 2006). Andra fördelar med agila metoder är att de ger ett snabbt värde och är bra på att svara på förändring i projekt (Boehm & Turner, 2003). Vattenfallsmodellen i sin tur är överskådlig och lätt att förstå och på så sätt sparar man tid då projektgruppen förstår hur modellen fungerar.

Idag används båda metoderna även om Vattenfallsmodellen får uppleva kritik medan agila metoder får beröm. Vattenfallsmodellen kan upplevas som trög, leda till stora förseningar och höga kostnader medan agila metoder förespråkas av många som flexibla, ger ökad produktivitet, och lägre kostnader (Björkholm & Brattberg, 2010).

Intresset för global systemutveckling växer snabbt på grund av att mjukvaruindustrin upplever ökad globalisering (Herbsleb & Moitra, 2001). Detta leder till att utvecklingen av system kommer ske på olika geografiska platser vilket kan leda till skillnader i tidszoner, kulturer och svårigheter att kommunicera. Som följd av detta krävs att organisationer idag vet hur man skall bedriva projekt då man sitter på olika platser, det vill säga distribuerat.

1.2 Problem

För att få ökad produktivitet i ett agilt projekt bör alla medlemmar arbeta tillsammans på ett ställe (Rasmusson, 2010). Man kan med detta i åtanke säga att Scrum bygger på daglig interaktion vilket blir svårt om projektgruppen inte är samlad. Agila metoder är utformade att passa små, samlokaliserade och samarbetsvilliga team (Sliger, 2006). Agila projekt kan dock ske på olika platser vilket gör att direkt kommunikation inte är möjligt. Det kan vara allt från olika städer inom ett land till olika länder i olika världsdelar. Scrum-projekt kan då bli svåra att genomföra när man inte sitter samlad och dagligen kan stämma av med samtliga medlemmar i teamet, vilket är en viktig faktor i Scrum (Björkholm & Brattberg, 2010).

Att arbeta med Scrum ställer krav på gruppen och att alla i gruppen vet hur man arbetar med metoden för att lyckas uppnå dess fördelar. Arbetar man med Scrum distribuerat leder detta således till att det ställs ännu högre krav på teamet för att det skall fungera, då möjlighet till interaktion öga mot öga blir begränsat.

1.3 Syfte och frågeställning

Syftet med uppsatsen är att identifiera de problem som kan uppstå vid distribuerad Scrum samt att se om det agila manifestet följs. Litteraturstudier och intervjuer ger oss tidigare identifierade problem, vilket gör att vi kan ta med dessa i vår planering av fallstudien. Genom vår fallstudie kommer vi belysa befintliga och nya problem som kan uppstå och sedan utvärdera dem för att komma fram till lösningar som mynnar ut i riktlinjer för hur mindre distribuerade projekt inom ett lands gränser kan bedrivas med Scrum. Detta resonemang leder fram till vår frågeställning:

Vilka problem uppstår vid distribuerad Scrum och hur påverkar de det agila manifestets principer?

1.4 Definition och avgränsning

Med distribuerat menar vi projekt som är utspridda på geografisk olika platser. I vårt fall handlar det om ett projekt inom ett lands gränser. Vi kommer bortse från tidsskillnader och kulturella skillnader i denna uppsats. Projektet bedrevs på ett mindre svenskt företag med en liten projektgrupp. Studien får därför mindre relevans för projekt med stora projektgrupper där Scrum redan är implementerat i organisationen.

Agila metoder är ett samlingsnamn för en viss typ av systemutvecklingsmetoder. Exempel på sådana metoder är Scrum och Extreme Programming. De skiljer sig från traditionella metoder genom att bland annat snabbt svara på förändring och ge ett snabbt värde (Boehm & Turner, 2003). I denna studie har vi valt att arbeta med Scrum för att bedriva ett distribuerat projekt.

Studien genomfördes på Riksnet som är ett mindre svenskt företag beläget i Umeå. Företaget hade en önskan om att ta fram en ny webbportal och såg det som lämpligt och intressant att göra detta med Scrum. Vi kommer tillsammans med två anställda i Riksnet bedriva distribuerad Scrum, där de arbetar i Umeå och vi i Göteborg. Studien pågick i sex veckor.

Något att ta hänsyn till är att alla i projektet var nya på Scrum, en faktor som kan påverka resultatet. Precis som Rayhan och Haque (2008) valde vi att fasa in medlemmarna under en vecka innan första sprinten då fokus låg på utbildning, samt under tiden som projektet fortlöpte. Vi fokuserade på litteraturen och kunde på så sätt utbilda de andra deltagarna.

På Riksnet kände man till vissa aspekter av Scrums arbetssätt och hade arbetat med dessa i begränsad omfattning utan att veta något om Scrum. Hänsyn tas även till att samtliga deltagare var positiva till att arbeta med Scrum och viljan att arbeta med metoden var stor. Detta gör inte att vi kan räkna bort det faktum att alla är nya på Scrum med det minskar risken att detta kommer få en betydande avgörelse på projektets framgång.

1.5 Undersökningens upplägg

I kapitel två börjar vi med att gå igenom teorin. Detta kapitel ger en beskrivning av vad agila metoder är och gör en djupdykning i Scrum och dess beståndsdelar. Detta kapitel är till för att

ge läsaren en grundförståelse för Scrum som är viktig för vidare läsning av uppsatsen. I kapitel tre går vi igenom valda metoder för att samla in data. Valda metoder är litteraturstudier, intervjuer och aktionsforskning. Intervjuer sker med personer med erfarenhet av Scrum och distribuerade projekt. Aktionsforskningen bedrevs under projektets sex veckor. Fjärde kapitlet ger en kort beskrivning av fallstudiens upplägg.

Uppsatsens resultat presenteras i kapitel fem. Insamlad data presenteras i form av de problem som uppstår vid distribuerad Scrum samt hur detta påverkar det agila manifestets principer. I kapitel sex presenterar vi vår analys där vi redogör för framkomna riktlinjer. Dessa är till för att fungera som ett ramverk för hur organisationer skall lyckas uppnå framgångsrika distribuerade Scrum-projekt. Avslutningsvis lyfter vi fram vår slutsats samt studiens relevans och överförbarhet och även förslag till fortsatt forskning i kapitel sju. Detta görs genom att belysa hur studien är relevant för organisationer liknande Riksnät.

2. Teori

För att få en djupare och bättre förståelse av problemområdet finns det vissa begrepp som behöver definieras. Fokus i vår uppsats kommer att ligga på Scrum och vi kommer i teorikapitlet att ge en omfattande beskrivning om vad agila metoder är och framförallt om vad metoden Scrum är och hur den fungerar.

Litteraturen vi valt är hämtad utifrån sökningar i artikeldatabaserna IEEE.org, ACM.org och Chalmers och Göteborgs Universitets respektive artikeldatabas. För att få fram bra och relevanta artiklar har vi sökt på följande ord: *Scrum, Agile, Global software development, Action Research, Schwaber*. Vi har även hämtat litteratur från universitetsbibliotek i Göteborg.

Teoridelen får relativt mycket utrymme i vår uppsats då en förståelse för agila metoder och Scrum med alla dess begrepp är viktiga för läsaren. Det är av största vikt att man förstår begreppen för att på bästa sätt kunna ta del av framkommet resultat.

2.1 Vattenfallsmodellen

Vattenfallsmodellen är en systemutvecklingsmetod som kom fram redan på 70-talet och är en traditionell metod som grundar sig i följande steg: analys, design, konstruktion, testning, integrering och underhåll.

Metoden är uppdelad i olika faser och varje fas ska vara genomförd innan nästa påbörjas. Skulle ett fel uppstå försöker man lösa detta innan man går vidare. Vattenfallsmodellen är inte agil vilket gör att den heller inte är flexibel. Ett projekt som använder sig av specifika uppgifter blir svårare att ändra under projektets gång även om omvärlden förändras och andra krav tillkommer. Vattenfallsmodellen förutsätter att kraven inte kommer att ändras under projektets gång (Winston, 1970).

I vattenfallsmodellen sker testningen av projektet i de sista faserna vilket innebär att om en bugg skulle uppstå skulle projektet gå bakåt. Att gå tillbaka till redan avklarade faser tar både tid och pengar.

2.2 Agila metoder

Det blir allt mer vanligt att arbeta i projekt i företag och organisationen. Det finns olika sätt att genomföra projekt på och många utvecklingsmetoder man kan använda sig av när man arbetar i projekt.

Ordet ”agile” kommer från engelskan och betyder lättrorlig. Det centrala i agila metoder är att göra utvecklingen mer flexibel, lättare att göra förändringar och lättare att styra. ”Agile” är också ett samlingsnamn för flera olika metoder man kan använda sig av när man ska utveckla produkter (Björkholm & Brattberg, 2010). Förutsättningarna i ett projekt förändras hela tiden och det krävs då utvecklingsmetoder som kan hantera dessa. I dag krävs det att man som systemutvecklare är flexibel eftersom omgivningen förändras. Ett system ska hela tiden kunnas förbättras och förnyas allt eftersom både företaget eller omvärlden förändras.

Det finns många olika agila metoder som exempelvis Scrum, Extreme Programming och Lean software development som används vid systemutveckling. De agila metoderna har mycket gemensamt och då framförallt genom rörlighet, förändring, planering och välkoordinerad (Björkholm & Brattberg, 2010). Rörlighet är det område som skiljer sig mest från Vattenfallsmodellen, som är en traditionell metod, där man gör allt för att följa den plan som satts upp innan projektets start. Det gör man genom att ha kontroll på oförutsedda händelser och andra ting som kan påverka projektet.

Inom de agila metoderna är planering viktig och när det sker förändringar som inte följer planen läggs den nya informationen in i projektet och planeringen anpassas efter verkligheten (Holcombe, 2008). Det gör man inte med de traditionella metoderna där man måste kunna förändra målet under projektets gång för att hela tiden utveckla och förbättra projektet. Lärande är också en viktig del inom agila metoder då varje projekt ses som en ny erfarenhet och ny kunskap. Denna nya kunskap resulterar ofta i förändringar i kravbilden för det nya systemet vilket ofta resulterar i ännu bättre och utvecklade system.

2.3 Scrum

Scrum är en av de agila metoderna och är en smidig metod för att hantera produktutveckling. Om man jämför de andra agila metoderna bygger inte Scrum så mycket på processer utan mer på hur man arbetar i projektet. När man arbetar med Scrum får man en god överblick av projektet vilket gör det enkelt att gå tillbaka i arbetet och antingen ändra eller förbättra. Scrum är byggt på det som kallas Sprint och är oftast tre veckor där det samlade arbetet koncentreras mot ett tydligt mål (Kniberg, 2007). Varje Sprint skall ge nya funktioner och förbättringar som kan levereras till kund samt att produkten utvecklas (Björkholm & Brattberg, 2010).

Första dagen på projektet skapas en Product Backlog, vilket är en lista över vad projektet ska klara av att få fram i produktväg. Denna lista innehåller även tider för när de olika delarna ska vara färdiga. Product Owner gör prioriteringen över vad som är viktigast för dennes organisation att få levererat först. En Product Backlog ska utvecklas och förändras om vissa förutsättningar eller tekniker förändras under tiden. Alla idéer som gruppmedlemmarna kommer på ska finnas med och det läggs till nya idéer under hela projektets gång. Vid början av varje ny Sprint ska Product Backlog gås igenom för att se till att den är uppdaterad och att alla medlemmar i projektgruppen är uppdaterade (Björkholm & Brattberg, 2010).

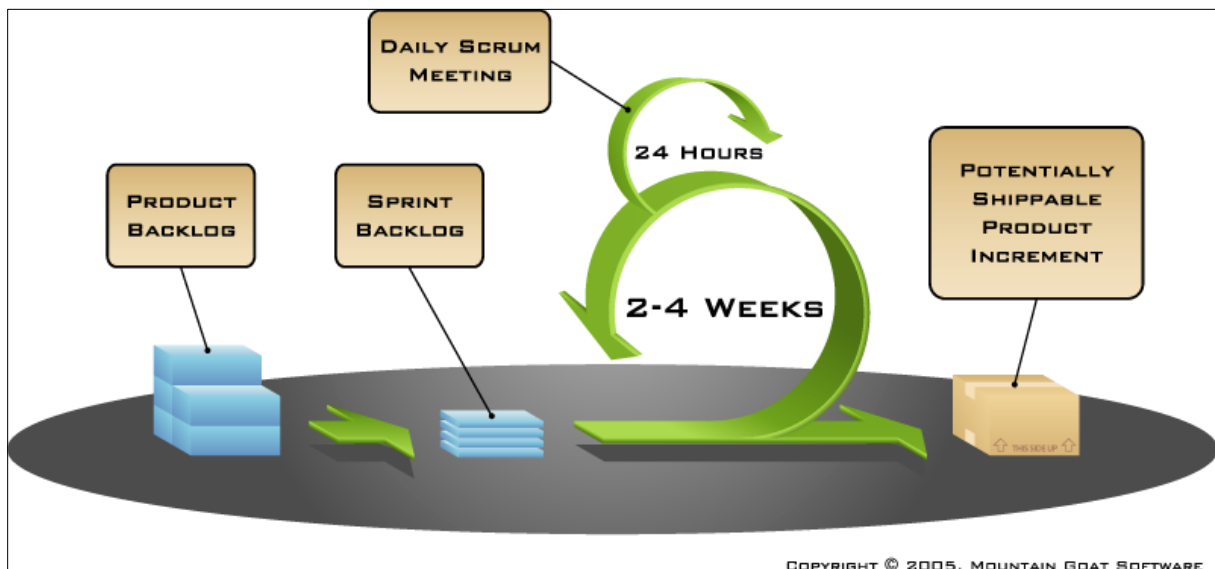
Den första dagen på en ny Sprint skapar man en Sprint Backlog, vilken är gruppens aktivitetsplan (Björkholm & Brattberg, 2010). När gruppen har bestämt vilka uppgifter som måste göras och hur lång tid det kommer ta släpper Product Owner taget. Från och med nu jobbar projektgruppen under eget ansvar. Gruppen fortsätter sitt arbete och betar av uppgifterna som finns i planen.

Sprint Backlog kan ses som en del av Product Backlog fast på en mer detaljerad nivå (Schwaber, 2010). Listan ska innehålla vem som har ansvaret för vad och vara uppdelad i arbetsuppgifter. Meningen är att det ska leda till att en uppgift i Product Backlog avklaras under varje sprint. Det hålls även regelbundna möten under varje Sprint där man pratar om

vad som går bra och om det finns några problem som måste rättas till. Målet med Scrum är att få en effektivare organisation men samtidigt en ökad motivation och handlingskraft hos medarbetarna på företaget.

De här punkterna har Craig Larman (2005) tagit fram för att få en inblick i hur man planerar en Sprint.

1. Längden på en Sprint bestäms och oftast är den mellan 2-6 veckor. Generellt gäller kortare desto bättre.
2. Ett planeringsmöte hålls. Detta brukas ofta ske i slutet av varje Sprint exempelvis på en fredag om nästa Sprint börjar på måndagen därefter.
3. En lista med mål presenteras där målen rankas i prioriteringsordning. Listan kommer oftast från både kunden och chefen.
4. Varje medlem i projektgruppen säger hur många timmar de kan arbeta med projektet, om de ska ha semester osv.
5. Ett mål beskrivs och frågor ska besvaras. Därefter frågas gruppmedlemmarna om de har några idéer av de mer specificerade uppgifterna för målet.
6. Steg 5 repeteras tills tillräckligt med arbete har valts. Om arbetet fungerar med de givna resurserna och deadlinesdatum för en Sprint är mötet färdigt.



Figur 1 visar hur Scrum är uppbyggt (Chon, 2005).

2.4 Artefakter i Scrum

2.4.1 Product Backlog

Product Backlog är produktens att-göra-lista. Denna lista är strikt prioriterad där endast en punkt är prioritet ett, en punkt är prioritet två och så vidare. Punkterna på listan kallar man för Backlog Items och kan vara en ny funktion, en teknisk förbättring eller en buggfix. Gäller det en ny funktion brukar man oftast göra detta via User Story. En User Story beskrivs utifrån kundens perspektiv och inte hur den ska lösas rent tekniskt.

Det är Product Owner som äger listan och är ansvarig för den (Schwaber, 2010). Det är Product Owner som bestämmer vad som ska läggas dit och i vilken ordning varje punkt ska vara. Product Backlog blir aldrig komplett utan är alltid levande och när som helst kan Product Owner lägga till nya saker och omprioritera ordningen på listan (Björkholm & Brattberg, 2010). Björkholm och Brattberg (2010) listar fyra saker som kan vara bra att tänka på när man ska prioritera en Product Backlog:

1. Vilket värde tillförs?
2. Vad är kostnaden?
3. Minimeras någon risk genom att den utvecklas?
4. Kan vi lära oss något genom att få den utvecklad?

I Knibergs (2007) exempel på Product Backlog (se figur 2) har varje User Story ett ID som visar att varje sådan är unik och förhindrar att det blir dubletter. Under rubriken Name finns en kort beskrivning av varje User Story. Imp står för importance och visar hur viktig varje User Story är. Est står för Initial Estimate och innebär att teamet har uppskattat hur svår uppgiften är jämfört med de andra uppgifterna. How to demo är en utförlig beskrivning av hur en User Story ska bli demonstrerad under Sprint Review. Kolumnen Notes innehåller övrig information.

PRODUCT BACKLOG (example)					
ID	Name	Imp	Est	How to demo	Notes
1	Deposit	30	5	Log in, open deposit page, deposit €10, go to my balance page and check that it has increased by €10.	Need a UML sequence diagram. No need to worry about encryption for now.
2	See your own transaction history	10	8	Log in, click on "transactions". Do a deposit. Go back to transactions, check that the new deposit shows up.	Use paging to avoid large DB queries. Design similar to view users page.

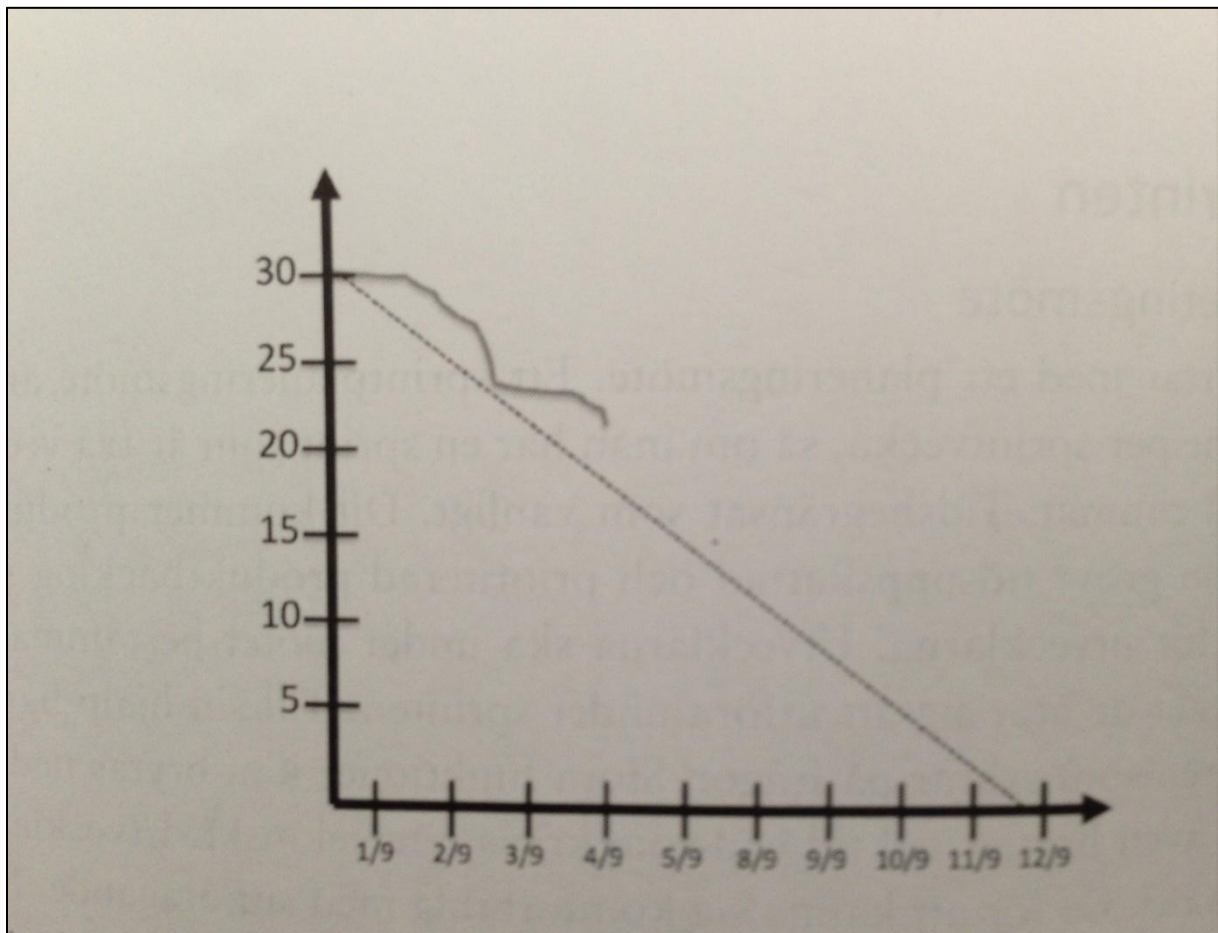
Figur 2 visar exempel på hur en Product Backlog kan se ut (Kniberg, 2007).

2.4.2 Sprint Backlog

Sprint Backlog är den del av Product Backlog som teamet tar på sig att färdigställa under en Sprint och det brukar vara de högst prioriterade sakerna som ska göras. Sprint Backlog skapas i början av Sprint Planning Meeting varje vecka där Scrum Master tillsammans med sitt Scrum Team går igenom vad som ska göras i kommande Sprint (Björkholm & Brattberg, 2010). Innehållet i en Sprint Backlog är till skillnad från Product Backlog helt låst. De enda som kan ändra en Sprint Backlog är utvecklingsteamet (Schwaber, 2007).

2.4.3 Sprint Burndown

Vid varje Sprint summeras det arbete som teamet har tagit på sig. Det kan vara summan av antalet timmar för arbetsuppgifter som är kvar eller bara antalet arbetsuppgifter som är kvar. Siffran ger en punkt på y-axeln (se figur 3) och x-axeln utgör dagarna som ingår i sprinten. Varje dag räknas en ny siffra fram och med tiden syns förhoppningsvis en kurva som korsar x-axeln vid slutet av en Sprint. Grafen är till för visa hur det har gått och grafen handlar inte bara om att få till en bra kurva utan att teamet tidigt kan se en trend och meddela produktägaren om det finns risk att några punkter inte blir färdiga i tid (Björkholm & Brattberg, 2010).



Figur 3 visar exempel på hur en Sprint Burndown kan se ut (Björkholm & Brattberg, 2010)

2.5 Roller i Scrum

Scrum har tre roller: Product Owner, Scrum Team och Scrum Master . Personer som är utanför dessa roller men ändå har intressen kallas för intressenter (Schwaber, 2007).

2.5.1 Product Owner

Product Owner är den person som har ansvar för att utvecklingen av produkten ger ett så stort värde till kunden som möjligt. Product Owner ansvarar också för innehållet i Product Backlog dess tillgänglighet och prioritering (Schwaber, 2007). Product Owner måste ha bra koll på alla intressenter (kunder, företagsledning, support, drift med flera) och deras önskemål. Rollen som Product Owner är en viktig roll och kan många gånger vara en tung roll att ha och det är då viktigt att Product Owner har ett nära samarbete med sitt team (Björkholm & Brattberg, 2010).

2.5.2 Scrum Master

Scrum Master fungerar som en lagkapten för teamet och han eller hon ska se till att medlemmarna i teamet kan jobba strukturerat, ostört och effektivt (Björkholm & Brattberg, 2010). Scrum Master ansvarar för att den bestämda processen följs. Björkholm och Brattberg (2010) menar att en bra Scrum Master är en person som upptäcker och löser problem utan att göra en stor sak av det.

2.5.3 Scrum Teams

Scrum Team eller utvecklarteam ses som en roll i Scrum (Björkholm & Brattberg, 2010). Individerna i ett team har inte egna roller i Scrum förutom Scrum Master. Anledningen till det är att teamet ska gemensamt ta på sig ansvaret att arbeta och leverera utifrån Product Backlog och Sprint Backlog.

Självorganiserade team är vanliga inom Scrum och Björkholm och Brattberg (2010) menar att om teamen själva får ta ansvar och organisera sig för att uppnå ett optimalt resultat blir medlemmarna mer drivna och aktiva samtidigt som kreativiteten ökar. Författarna menar också att ett Scrum Team oftast brukar sitta tillsammans på samma plats eftersom ett team fungerar bäst när medlemmarna sitter samlade.

2.6 Planering och möten enligt Scrum

Scrum består av fyra möten: Sprint Planning Meeting, Daily Scrum, Sprint Review Meeting och Sprint Retrospective.

2.6.1 Sprint Planning Meeting

Varje Sprint startar med ett planeringsmöte. Ett Sprint Planning Meeting är ungefär en timme per Sprint-vecka, vilket innebär att om man har en Sprint som är två veckor är mötet två timmar. På detta möte kommer Product Owner med en Product Backlog som de visar för utvecklarna. Sedan bestämmer utvecklarna hur mycket det kan åstadkomma under kommande Sprint (Björkholm & Brattberg, 2010).

2.6.2 Daily Scrum

Daily Scrum är ett möte som teamet och Scrum Master har varje dag för att synkronisera gruppen. Mötet är max 15 minuter och är på samma tid varje dag. Mötet går till så att utvecklarna berättar:

1. Vad de gjorde igår, sedan förra mötet?
2. Vad de planerar att göra idag, fram till nästa möte?
3. Vad som hindrar dem och vilka problem de har?

Fokus på detta möte är vilka hinder som finns och efter mötet diskuteras hur de kan lösa dessa hinder (Rising & Norman 2000)

2.6.3 Sprint Review Meeting

När en Sprint är slut ska utvecklarna leverera fungerande funktioner som ska demonstreras för intressenterna. Till detta möte är alla välkomna för att se vad som gjorts och för att föreslå hur man kan vidareutveckla produkten. Eftersom leveransen är något som kan testas är det enkelt för intressenterna att förstå hur produkten kommer se ut och fungera och tidigt rätta till eventuella missförstånd (Björkholm & Brattberg, 2010).

2.6.4 Sprint Retrospective Meeting

När demonstrationen är färdig har teamet och Product Owner ett möte för att förbättra sättet att arbeta. Men går igenom vad som fungerat bra och vad som behöver förbättras till nästa Sprint. Ett sätt är att brainstorma fram vad som varit bra, vad som varit mindre bra och vad som kan göra för att lösa det som varit mindre bra. Mötet resulterar i en prioriterad lista med förbättringsförslag och denna lista har Scrum Master ansvar för att punkterna blir gjorda (Björkholm & Brattberg, 2010).

3. Metod

Vårt syfte med undersökningen var att identifiera problem som uppstår vid projekt som bedrivs med distribuerad Scrum och om det agila manifestet inte följdes, samt att med hjälp av aktionsforskning ta hänsyn till dessa problem och förbättra arbetssättet. För att identifiera befintliga problem har vi använt oss av litteraturstudier och intervjuer. Detta gav oss en tillräcklig grund för att planera vår fallstudie.

3.1 Aktionsforskning

För att se hur det fungerade att i praktiken arbeta med distribuerad Scrum valde vi att genomföra en aktionsforskning på en projektgrupp där vi tillsammans med Riksnet tog fram deras nya webbportal. Med grunden från litteraturstudierna och intervjuer har vi kunnat planera projektet med hänsyn till befintliga problem vi visste skulle uppstå.

Valet föll på aktionsforskning då gruppen är på fyra personer vilket gör att vi aktivt kan delta samtidigt som vi utvärderar arbetet. Resultatet hade inte blivit relevant för oss om vi bara hade valt att observera, då projektet endast hade bestått av två anställda från Riksnet och på så sätt inte varit distribuerat. Aktionsforskning är hur vi kan göra saker bättre (Ferrance, 2000). Aktionsforskning som metod är lämplig då den är en kraftfull och befriande form av professionell utredning, på grund av att utövarna själva kan undersöka sin egen utövning när de finner sätt att leva mer fullt ut i riktning mot deras utbildningsvärden (McNiff, 2002).

McNiff (2002) menar även att vi på så sätt är en del av den situation vi undersöker och kan då ställa frågor som ”Hur går vårt arbete?” och ”Hur kan vi utveckla vårt arbete?”. Hur arbetet går och vad vi kan göra bättre till nästa gång är viktiga aspekter i vår fallstudie för att vi skall kunna besvara frågan hur man driver ett distribuerat projekt med hjälp av Scrum. Lewin (1958) förklarar förändringsprocessen med tre steg:

1. Unfreezing
2. Changing
3. Refreezing

Först identifierar vi de problem som uppstår och bryter det negativa mönster dessa ger upphov till. Vi försöker genom förändring få till ett bättre arbetssätt som vi sedan provar och utvärderar. Gynnar detta nya sätt projektet lär vi oss av det och implementerar det i vår organisation. Vi följer Lewins tre steg i vår aktionsforskning för att ändra beteenden som påverkar vårt projekt negativt.

3.2 Intervjuer

För att få en bra grund för att lyckas planera vår fallstudie valde vi att efter litteraturstudier genomföra intervjuer. De intervjuade var två personer med insikt i hur det är att arbeta med Scrum och med vana att arbeta distribuerat. En intervju skedde över mejl, p.g.a. geografiska skillnader, och den andra öga mot öga under en halvtimme. Frågorna var ostrukturerade då vi

var intresserade av hur informanten med egna tankar och ord upplevde distribuerade projekt och agila metoder.

De inledande tre frågorna var generella och syftade till att peka på fördelar och nackdelar. De två sista frågorna styrde vi mot att innefatta hur man bäst bör göra för att arbeta med Scrum i en distribuerad miljö. På detta sätt fick vi värdefull information att ta med oss in i fallstudien. Tanken var att vi skulle vara medvetna om eventuella fallgropar för att kunna lägga märke till dessa direkt och på så sätt lyckas jobba mer effektivt.

En fördel med data som samlas in genom ostrukturerade intervjuer är att de genererar rik data som är inbördes relaterad och komplex, alltså data som ger en djupare förståelse för ämnet (Sharp et. al. 2011). Sharp et. al. (2011) menar även att data från ostrukturerade intervjuer tar tid att analysera. Detta till trots utgick vi ifrån att ostrukturerade frågor var det bästa sättet för oss att få fram den information som var nödvändig för att ha en bra grund att stå på inför fallstudien.

Intervjun som skedde öga mot öga spelades in och transkriberades medan intervjun som skedde över mejl sparades och strukturerades upp i textdokument, detta för att tydligare kunna se på likheter i vår analys av de olika intervjuerna.

3.3 Litteraturstudier

Vi började med litteraturstudier för att få en grundläggande förståelse för området. Detta för att lyckas ställa de frågor som är relevanta och ge oss den informationen vi behöver få med oss in i fallstudien. Litteratur som varit aktuell har berört områdena agila metoder, Scrum och hur man bedriver distribuerade projekt. Litteraturen gav oss även förslag på befintliga problem som finns vid distribuerade projekt.

3.4 Urval

Scrum är uppbyggt av aktiviteter och artefakter och det krävs att man har en förståelse för dessa begrepp för att projektet skall bli lyckat. Därför valdes informanterna ut efter sin kunskap rörande Scrum och distribuerade projekt. Att informanternas kunskap är relevant för oss är avgörande för hur väl vår fallstudie kommer gå. En av oss antar rollen som Scrum Master och måste klara av att lära övriga projektmedlemmar hur man arbetar med Scrum.

Eftersom deltagarna i projektet var nya på Scrum måste vi ta reda på vad det var som påverkar utfallet. Är det för att det var nya i Scrum eller var det för att Scrum bedrivs distribuerat. För att avgöra detta tog vi hjälp av en certifierad Scrum Master som känner igen problem som uppstår då Scrum implementeras som systemutvecklingsmetod. Vi kunde då avfärda vissa problem som är specifika för införandet av Scrum i organisationer.

Med den begränsade tid vårt projekt bedrevs under ansåg vi att valet av litteratur och valet av informanter gav oss en bra grund för att bedriva aktionsforskning. Avgränsningen till trots, där vi bara använde oss av två informanter, ansåg vi att vi kommer lyckas identifiera problem

som uppstod vid distribuerad Scrum, vilka principer i det agila manifestet som bryts samt ett framtagande av våra riktlinjer.

3.4.1 Presentation av urvalsgruppen

Följande är en kort presentation av de två informanterna. Ålder och kön presenteras inte då detta inte är av relevans för resultatet.

Informant 1: Certifierad Scrum Coach. Arbetar på ett företag som använder sig av agila metoder och har stor erfarenhet av Scrum.

Informant 2: Forskat och observerat distribuerade team.

Vi ansåg att dessa informanter levde upp till de krav vi hade för att lyckas besvara frågeställningen.

4. Fallstudie

För att få möjlighet att jobba distribuerat med Scrum valde vi att bedriva aktionsforskning på det Scrum Team som var tänkt att ta fram en webbportal åt Riksnet. Teamet bestod av oss två och två anställda på Riksnet. Vi var stationerade i Göteborg och de anställda på Riksnet i Umeå. Nedan följer en beskrivning av företaget och fallstudiens upplägg.

4.1 Riksnet

Riksnet är ett företag som förser kunder med bredband, telefoni och TV. Kunder är fastigheter, bostadsrättsföreningar och samfälligheter. Man erbjuder även att projektera och bygga kraftfulla nätverksanslutningar till fastigheter eller områden. Företaget drivs av ett team med marknadens längsta erfarenhet av bredband och som tidigare framgångsrikt byggt upp exempelvis Bostream.

4.2 Fallstudiens upplägg

Fallstudien som bedrevs med Scrum planerades upp i fem sprintar, där varje Sprint varade i en vecka. Innan Sprint ett startade började vi med att utbilda de andra medlemmarna i projektet i Scrum. Veckan innan Sprint ett vigdes helt åt utbildning. När sprint ett startade hade deltagarna en tillräcklig grund för att börja arbeta med Scrum. Alla fem sprintar bestod av Sprint Planning Meeting, Daily Scrum, Sprint Review Meeting och Sprint Retrospective Meeting.

Rollfördelningen i projektet blev som följande:

- Scrum Team: Joakim, David, Shayan (Riksnet), Anton (Riksnet)
- Product Owner: Shayan
- Scrum Master: Joakim

Då projektet bestod av enbart 4 deltagare beslöt vi oss för att Product Owner och Scrum Master skulle ingå i teamet.

4.2.1 Utbildning

Utbildningen skedde innan Sprint ett och bestod av att ta del av aktuell litteratur och utbilda de anställda på Riksnet i Scrum. Utbildningen skedde distribuerat. Vi använde oss av telefon och mejl för att utbilda den andra delen av teamet som satt i Umeå.

4.2.2 Sprint 1

Sprint ett Inleddes med att arbeta fram User Story tillsammans i teamet och skapa en Product Backlog. Vidare följde framtagning av uppgifter för de User Story som var aktuella i den första sprinten. I samband med detta skapades även Sprint Backlog för Sprint ett. Daily Scrum hölls på samma tid och på samma plats varje dag för att minska komplexiteten (Scrumguiden, 2011). Sista dagen i sprinten bestod av en Sprint Review Meeting och Sprint Retrospective Meeting där vi demonstrerade vad som gjorts och gick igenom vad som gått bra respektive

mindre bra under sprintens gång. Det var här vi fick reda på problem som uppstått och möjligheten att utvärdera problemen för att få fram lösningar vi kunde prova i efterföljande Sprint för att uppnå förbättring.

4.2.3 Sprint 2-5

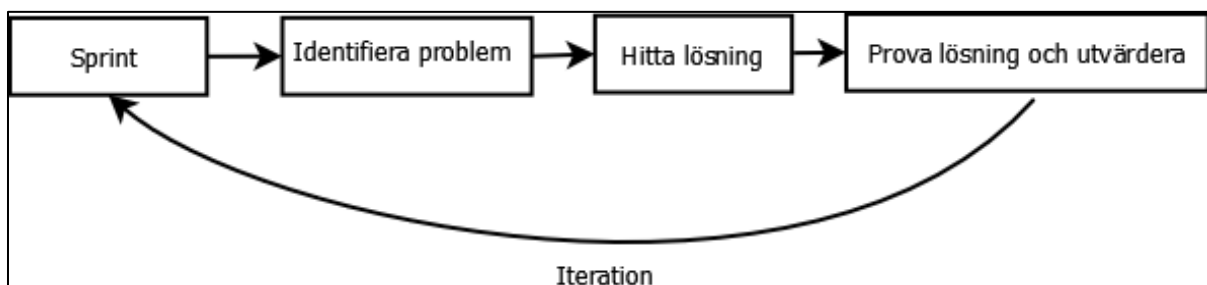
Varje Sprint inleddes med Sprint Planning Meeting. Med hjälp av Product Backlog valdes User Story som vi bröt ner till uppgifter och gav upphov till Sprint Backlog. Daily Scrum skedde i slutet av dagen. Sprintarna avslutades med Sprint Review Meeting och Sprint Retrospective Meeting. Vi fick då fyra tillfällen på oss att finna problem, utvärdera dessa och komma fram till lösningar för att effektivisera vårt distribuerade Scrum-projekt.

5. Resultat

I resultatet lyfter vi fram de problem vi identifierat med hjälp av våra metoder. Vi kommer även visa på vilka av det agila manifestets principer, vilka är själva grundstenarna i agil systemutveckling, som bryts vid distribuerad Scrum. Förslag till hur man arbetar på bästa sätt med distribuerad Scrum, alltså de riktlinjer vi kommit fram till presenteras först i nästa kapitel.

Genom att ha arbetat med distribuerad Scrum kommer vi kunna presentera de problem som uppstår i sådana projekt. Problem som uppstod analyserades och bearbetades under studiens gång för att på så sätt kunna undgås längre fram i projektet. Med hjälp av litteratur och intervjuer fick vi med oss en bra bild av möjliga problem vi kunde stöta på i projektet, medan vi i aktionsforskningen själva fick stöta på dessa problem och hantera dem.

5.1 Identifiera problem



Figur 4 visar hur vi har gick till väga för att komma fram till våra riktlinjer.

Figur 4 visar vilket tillvägagångssätt vi hade när vi kom fram till våra riktlinjer och identifierade problem som uppstod i projektet. I första sprinten identifierade vi problem som uppstod under sprintens gång. Ett exempel på problem kunde vara att telefonmötena vid Daily Scrum tog lång tid och i början var det lätt att avbryta en annan person som talade samt att det var lätt att koncentrationen kunde sjunka när man inte såg varandra. Ett annat problem vi hade i början var att uppdatera Trello, som är ett grafisk verktyg för att sköta Daily Scrum, och flytta "lapparna" till rätt plats. Dessa problem diskuterades vid Retrospective Meeting där vi kom fram till lösningar på problemen. Efter mötet provade vi lösningen och på slutet av veckan utvärderade vi om lösningen blivit bättre eller sämre än innan. Detta itererades sedan i varje sprint för att projektet skulle utvecklas och bli bättre.

Innan projektets start fick vi genom intervjuer och litteraturstudier fram vilka problem som kan uppstå vid distribuerade projekt. De befintliga problemen var:

1. Problem med kommunikation och koordination av vardagliga saker. Det kan uppstå missförstånd.
2. Distribuerade projekt är inte lika snabba och effektiva som vanliga projekt.
3. Det kan bli problem om vi inte ses i början av projektet.
4. Kan bli svårt att skapa sociala strukturer, det vill säga en miljö där olika team kan arbeta tillsammans på ett sätt de känner sig trygga i.

5. Det kan bli problem om man inte tidigt etablerar kommunikationskanaler.

De problem som vi identifierades under studien är:

1. För många verktyg vilket gör det rörigt
2. Problem med kommunikation
3. Product Backlog blir bortglömd
4. Problem med Daily Scrum
5. Mejlkonversationen blir lätt rörig
6. Allt tog lång tid i början
7. Inte våga fråga eller uttrycka sig
8. Sprintverktyget glömdes bort

5.1.1 För många verktyg gör det rörigt

Att vi använde oss av många olika verktyg för att lösa interaktionen visade sig göra det hela rörigt. Vi använde oss av fyra olika kanaler för att kommunicera med varandra vilket gjorde att det var svårt att veta var viss information fanns. Daily Scrum skedde över telefon, eller via mejl de gånger telefonkontakt inte var möjligt. De andra kommunikationskanalerna var chatt och sms. Även om många verktyg gör det rörigt och har sina nackdelar, så har de även fördelar som man tidigt bör utnyttja för att maximera interaktionen. De verktyg vi använde oss av var telefon, mejl, Facebook, Dropbox och Trello.

Telefon

Vi bestämde från start att telefon är det verktyg som vi i första hand skall använda för att sköta alla möten i de olika sprintarna. Vi använde oss av ett telefonkonferensnummer dit alla ringde och loggade in med en pinkod. Detta verktyg gav oss mest direkt kommunikation. Det bästa verktyget vid Daily Scrum.

Fördel: Snabb feedback och snabbt beslutstagande.

Nackdel: Tar lång tid jämfört med direkt kommunikation som sker ansikte mot ansikte. Lätt att man avbryter en annan talare. Fokus kan minska när man inte har möjlighet att se varandra.

Mejl

Mejl var den kommunikationskanal vi använde allra mest till en början för att skicka dokument till varandra samt för att utbyta information. Användes även som verktyg för att hantera Daily Scrum de gånger då telefonmöten inte var möjligt.

Fördel: Förmågan att uttrycka sig och förklara sig grundligt. Även möjlighet till reflektion är stor då du har texten framför dig. Enkelt att skicka med dokument.

Nackdel: Ingen omedelbar respons. Blir lätt rörigt då det kan bli många mejltrådar.

Facebook

Detta var en kanal som kom att användas mer under projektets gång. Att använda chattfunktionen blev populärt då vi skulle ställa frågor till varandra samtidigt som vi kunde sitta och arbeta. Ett verktyg som tillåter direkt kommunikation då vi ofta är online när vi jobbar.

Fördel: Direkt kommunikation. Minskar obehaget för att ställa frågor. Ger möjlighet att snabbt hjälpa varandra genom uttryck i både text och bilder.

Nackdel: Svårigheter med tydligheten då fler än två är med i en konversation.

Dropbox

Det verktyg som användes för att samla, hantera och lagra material. Vi lagrade bilder, designförslag och kodfiler. Dropbox underlättar möjligheten att dela dokument med varandra direkt.

Fördel: Alla kan med lätthet ta del av innehållet då det lagras i molnet. Man slipper oroa sig för att dokument skall försvinna vid krascher. Du har alltid tillgång till den senaste versionen.

Nackdel: Svårigheter att samtidigt redigera då man kan spara över den andres arbete. Strukturen kan lätt bli rörig då alla har möjlighet att påverka den.

Trello

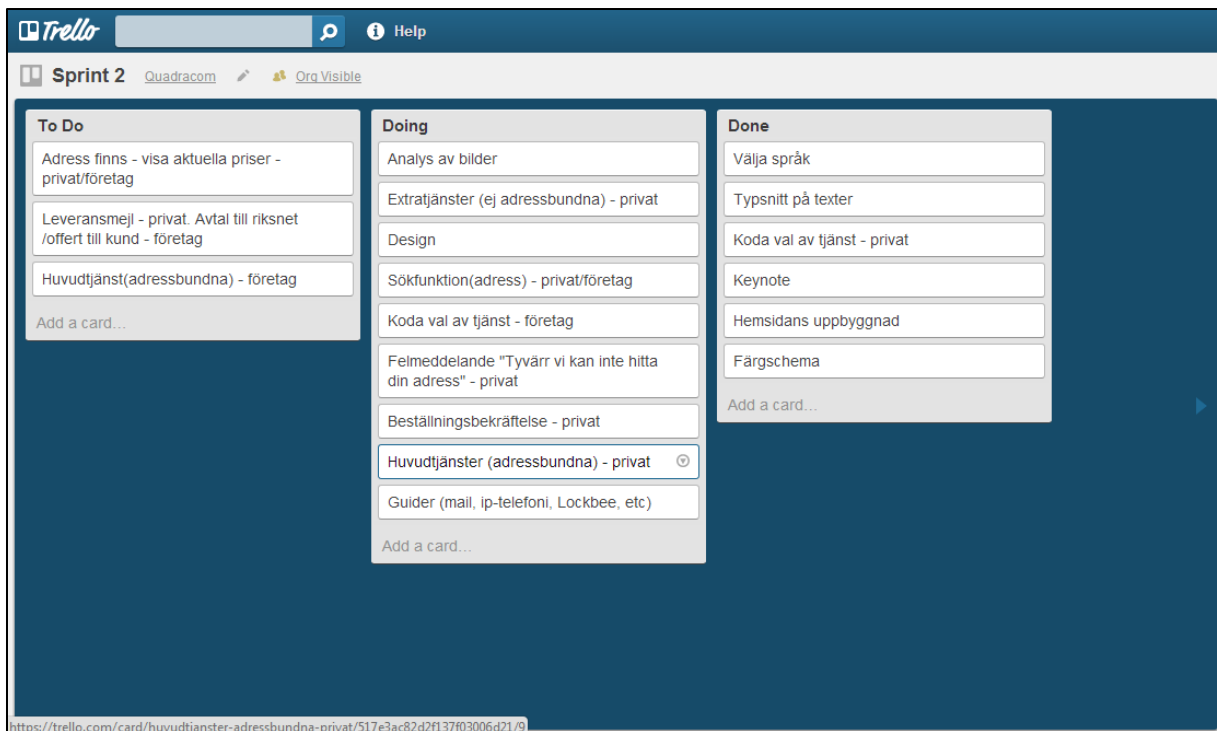
Det verktyg som användes för att teamet skall kunna se hur arbetet går och om man är på väg att klara av sprintmålet. Det mest grafiska verktyget som är tänkt att stötta teamets arbete för att nå sprintmålet.

”Man bör inte börja med ett elektroniskt verktyg, utan man skall försöka börja med papper och penna” (intervjuinformant 1).

Som vi kan se enligt citatet ovan bryts den agila principen från start. Här går verktyg och processer före individer och interaktion. Men vi upplevde att detta behövdes göra för att klara av att arbeta distribuerat med Scrum. Man bryter helt enkelt principen för att kunna sköta processer och interaktion så bra det bara går.

Fördel: Ger en bra överblick. Lätt att se hur man ligger till i sprinten genom kolumnerna ”to do”, ”doing” och ”done”.

Nackdel: Teamet kan glömma bort att flytta ”lappar” mellan kolumnerna.



Figur 5 visar hur en sprint ser ut i Trello.

5.1.2 Problem med kommunikation

“Problem som främst kommunikation och koordination av vardagliga praktiska saker. Kan få allvarliga konsekvenser då vissa aktiviteter har stort genomslag på framtida framgång, exempelvis kan missförstånd av krav ha stor effekt på projektet” (intervjuinformant 2).

Ett problem som uppstod under projektets gång var problem att följa vissa av Scrums regler och riktlinjer, vilket härstammade från utbildningen i början av projektet. Utbildning är något som inte är specifikt för distribuerad Scrum. Vet man inte hur man skall arbeta med Scrum spelar det ingen roll hur projektet är uppdelat. Men utbildning är en av de praktiska saker som kan få allvarliga konsekvenser på framtida framgång. I vårt fall blev detta tydligt vid skapandet av user stories. Har inte alla i projektet förstått vad som skall göras och lyckats få ner detta i tillräckligt många bra User Story kommer det få effekt på resultatet.

5.1.3 Product Backlog blir bortglömd

“Håll Product Backlog levande genom att ha möte varje Sprint” (intervjuinformant 1).

Vi blev medvetna om att vår Product Backlog som skapades i ett excelark och sparades ner i Dropbox lätt kunde bli bortglömd. Vi menar inte att den efter att ha skapats inte öppnades igen förrän i slutet av projektet, utan mer att den kom i skymundan. Det hade varit bättre för teamet och projektet om den hade funnits synlig. Jämför med ett samlat projekt där man kan ha Product Backlog på en Whiteboard, konstant synlig för projektets medlemmar.

Vi lyckades trots detta hålla den levande genom att i slutet av varje sprint gå igenom den och uppdatera när user story blivit avklarad.

5.1.4 Problem med Daily Scrum

“Daily Scrum är hjärtat i Scrum. Sköts ej det så är det inte Agil systemutveckling” (intervjuinformant 1).

“Daily Scrum skall vara regelbundet” (intervjuinformant 1).

“Scrum säger inget om hur Daily Scrum sker på bästa sätt” (intervjuinformant 1).

I början av projektet upplevde vi svårigheter med Daily Scrum. Under tiden blev Daily Scrum bättre och flöt på smidigare. Vi lärde oss hur vi skulle bete oss när vi alla fyra var samlade. Att tala tydligt, att låta folk prata till punkt samt att bara svara på de tre frågorna som hör Daily Scrum till gjorde att mötena blev effektivare. Vi upplevde problem med att uttrycka oss utan att kunna visa saker grafiskt, svårigheter med att veta när den andre pratat klart samt upprepningar då det kunde vara störningar på linjen. Vi upplevde också att det tog lång tid när vi hade Daily Scrum via mejl och det var svårt att uppfatta om vi tolkat allt rätt. Detta gjorde att Daily Scrum tog mycket längre tid än vad det gjorde när teamet var samlat på samma ställe. Under de tre dagar vi var samlade fungerade Daily Scrum bättre och med högre effektivitet.

Eftersom Scrum inte säger hur Daily Scrum sker på bästa sätt valde vi att lägga mötena i slutet av dagen. Verktygen vi använde oss av var telefon. När det inte är klart uttalat hur man gör på bästa sätt fick vi genom egen erfarenhet komma fram till hur genomförandet sker på bästa sätt.

5.1.5 Mejlkonversationen blir lätt rörig

Då mycket kommunikation skedde över mejlen blev det lätt rörigt i och med att vi inte hade någon uttalad tråd att skriva i. Vile vi kontakta någon enskild person i teamet så skrev man ett mejl och skickade till den personen. Mottagaren kanske svarade och ställde en ny fråga som rörde en annan i teamet och la till denna i tråden. Så kunde det hålla på tills det blev väldigt många trådar och man hade svårt att hålla reda på var specifik information fanns och det var något som tog tid och skapade frustration.

5.1.6 Allt tog lång tid till en början

“Distribuerade projekt är ej lika snabba och effektiva säger de agila principerna” (intervjuinformant 1).

I början av projektet tog interaktionen lång tid. Detta är inget specifikt problem för distribuerad Scrum, men något som man löser på ett annat sätt. För oss gick interaktionen bättre när vi arbetat en Sprint och utvärderat vad som fungerade bra och mindre bra med valda verktyg. Det var också viktigt att vi var tydliga med vad som skulle göras när vi hördes över telefon så det inte blev några förskjutningar.

Det blir viktigt att man under Sprint Retrospective ställer rätt frågor för att interaktionen skall maximeras. En ytterligare faktor som påverkade effektiviteten var möjligheten att träffas och

lära känna varandra. Detta gjorde att många spärrar släppte, vilket leder oss in på nästa problem.

5.1.7 Inte våga fråga eller uttrycka sig

“Försök att ses en gång i början av projektet för att lära känna varandra”
(intervjuinformant 1).

“Svårt att skapa sociala strukturer, det vill säga en miljö där olika team kan arbeta tillsammans på ett sätt de känner sig trygga i” (intervjuinformant 2).

Innan vi träffades och lärde känna varandra upplevde vi att det fanns en rädsla att uttrycka sig felaktigt eller på ett sätt där den andra parten kunde ta illa upp. Efter att teamet samlats i Göteborg och vi umgicks öga mot öga släppte denna rädsla. Att lära känna de andra personerna samt att med kroppsspråk och ansiktsuttryck påverka det som sades, gjorde att det i framtiden blev lättare att ställa besvärliga frågor när vi arbetade distribuerat.

Det blev därför avgörande för projektets framgång att vi hade möjlighet att ses tidigt i projektet för att lära känna varandra. Som informant två säger så blir det svårigheter att skapa denna sociala struktur i början av projektet. Denna trygghet infinner sig snabbare om man träffar teamet och lär känna varandra. För vår del blev den sociala strukturen bättre och vi kände oss tryggare efter att vi hade lärt känna varandra och vågade ställa de frågorna som vi tidigare inte vågat.

5.1.8 Sprintverktyget glömdes bort

“Sprint Backlog är en plan för att leverera det som skall levereras” (intervjuinformant 1).

Vi valde att med hjälp av Trello visualisera vad som skulle göras i varje Sprint. Eftersom sprintverktyget inte var en uppgift för Scrum Master att sköta föll detta lätt mellan stolarna. I sprint ett och två uppdaterades Trello i slutet av sprinten. Problemet belystes i den andra sprintens Sprint Retrospective Meeting. Efter detta blev det bättre och uppgifterna flyttades allteftersom de blev klara.

Detta problem uppstod i samband med att teamet inte fullt ut förstod innebörden med verktyget.

”Att det skall vara en plan för att lyckas leverera i något i varje sprint måste varje medlem av teamet vara införstådd med” (intervjuinformant 1).

Det blir då viktigt att detta finns med i utbildningen av teamet. Något som redigerades genom en tydligare genomgång av Sprint Backlog i slutet av sprint två.

5.2 Distribuerad Scrum och det agila manifestet

Av det agila manifestets fyra principer är det den första, individer och interaktioner framför processer och verktyg, som bryts vid de problem vi identifierat i distribuerad Scrum. De tre andra påverkas inte. Både begreppet individer och interaktion påverkas då projektmedlemmarna inte befinner sig på samma plats. Individer genom att de inte träffa varandra och interaktion genom att den inte sker direkt, öga mot öga.

Den andra principen, fungerande programvara framför omfattande dokumentation, gäller lika väl vid ett distribuerat projekt som ett samlat. Att följa Scrums riktlinjer och i slutet av varje sprint leverera färdig programvara var något vi lyckades med. Under Sprint Review visades den programvara som hade skapats. . Således lades inte massa tid på att dokumentera allt. Något som gjordes klart från början, krav på dokumentation var inget som fanns från Riksnets sida.

Den tredje principen, kundsamarbete framför kontraktförhandling, påverkas inte heller i ett distribuerat projekt. Detta gjordes genom att vi hela tiden följde Scrums riktlinjer. Vi hade hela tiden nära kontakt med Product Owner, något vi lyckades med genom att i slutet av varje sprint ha en Sprint Review. Product Owner blev då medveten om hur projektet låg till och kunde på så sätt ta beslut om det var något som behövde ändras. På så sätt undviker man att projektet rullar på i fel riktning en längre tid. Denna princip bryts inte i ett distribuerat Scrum-projekt.

Den fjärde och sista principen, anpassning till förändring framför att följa en plan, påverkades inte den heller i ett distribuerat projekt. Efter att ha följt Scrums riktlinjer märker vi genom att inte fokusera på annat än det som sker i den aktuella sprinten sker en anpassning. Att efter ha avslutat en sprint och planera kommande sprint i ett Sprint Planning Meeting får man automatiskt en anpassning till förändring då man tar följande sprints utfall i åtanke. Detta till skillnad från att planera allt från början för att sedan få planera om.

5.2.1 Interaktion

”Det är viktigt att tidigt etablera kommunikationskanaler” (intervju informant 1).

”Hur tänker vi kommunicera på möten?” (intervjuinformant 1).

Det agila manifestet bygger på att fokus skall ligga på själva interaktionen och inte på verktygen som kan användas för att interagera. Detta är något som bygger på att interaktion individer emellan kan ske så smidigt som möjligt. Så smidigt som möjligt är då direkt kommunikation, gärna öga mot öga. Så när detta inte är möjligt blir det mer fokus på verktyg. Sett till de två citaten ovan och vad vi kommit fram till i vår fallstudie blir det ökat fokus på verktyg i ett distribuerat projekt. Men i vårt fall var det en avgörande faktor för att kommunikationen inte skulle bli bristfällig. Verktygen får då en direkt avgörande roll på hela projektet. Vi kommer därmed bryta principen för att lyckas arbeta med Scrum.

5.2.2 Individer

Att individer inte får chansen att träffa varandra spelar stor roll för projektets resultat.

”Se till att försök samla teamet och lära känna varandra” (intervjuinformant 1).

“Svårt att skapa sociala strukturer, dvs. en miljö där olika team kan arbeta tillsammans på ett sätt de känner sig trygga i” (intervjuinformant 2).

Att man aldrig träffar sina medarbetare leder till att många barriärer inte rivs. Att ha möjlighet att träffas och lära känna varandra leder till en bättre fungerande grupp. Detta var något vi märkte av egen erfarenhet. När de två övriga medlemmarna av teamet kom ner till Göteborg för att arbeta med oss, och lära känna oss i tre dagar skedde en stor förändring. Rädslan att uttrycka sig fel eller annorlunda släppte.

6. Resultatanalys

Vi kommer i detta kapitel presentera våra upptäckter som framkommit i samband med identifiering av de problem som uppstått i distribuerad Scrum samt vad det innebär att man inte följer det agila manifestet. Vi kommer diskutera hur vi initialt arbetade i projektet och gå vidare med att belysa hur identifierade problem påverkade gruppen och ledde till förändrade arbetssätt. Detta mynnar ut i generella riktlinjer som vi tycker man bör följa för att lyckas med distribuerade projekt. De generella riktlinjerna kommer delas upp i tre faser som kommer från sättet vi bedrev vårt projekt. Dessa tre faser är planeringsfas, uppstartsfas samt projekt.

6.1 Förändringar utifrån identifierade problem

Vårt sätt att jobba med distribuerad Scrum kom att ändras under våra fem sprintar. Innan projektet startade valde vi att utbilda teamet i Scrum. Detta skedde över mejl och telefon. Utbildning blir än mer viktigt då man inte sitter tillsammans. Något vi märkte när projektet Minsta otydlighet i utbildningen ledde till att aktiviteter i Scrum tog längre tid. Exempelvis framtagandet av user stories som påverkade projektet.

Till en början använde vi oss mycket av telefon och mejl för att sköta kontakten. Vi följde Scrumguiden och bestämde tider för våra möten, vilka vi var noga med att hålla. Vi upplevde att detta faktiskt minskade komplexiteten. Genom att vi hela tiden visste när kommunikationen skulle ske, slapp vi oroa oss för detta. Att vi tidigt bestämde vilket verktyg som skulle användas för Daily Scrum minskade också komplexiteten. Bröts någon av dessa punkter (exempelvis vid sjukdom eller annat förhinder) bestämdes snabbt ny tid eller nytt sätt att kommunicera på. Detta sätt att hantera Daily Scrum var det som utgjorde stommen i vårt sätt att bedriva distribuerad Scrum och något vi inte ändrade på under projektets gång.

Förändringar i kommande Sprintar ledde till förbättringar och ökad effektivitet i teamet. För att identifiera problem så var vi noga med att sköta Sprint Retrospective Meeting. Det var i detta möte som vi analyserade aktuell Sprint och fick fram de problem som uppstått och analyserade dem för att finna lösningar att prova i nästa sprint.

I tredje sprinten märktes detta bl.a. då Daily Scrum gick bättre. Daily Scrum gick effektivare och vi höll oss under utsatt tid (15 minuter enligt Scrum). Detta kom av att Scrum Master fick trycka extra noga på vad Daily Scrum innehåller och att det inte skall pratas om mer än det. Att vi gick ner till färre verktyg hjälpte också teamet att öka effektiviteten. Verktygens roll är att hjälpa teamet, inte skapa förvirring. Genom att använda vissa verktyg mer minskades förvirringen och försvann så småningom. Exempelvis så minskade mejlen i slutet av projektet då teamet kände lärde känna varandra bättre och kommunikationen gick över till att ske mer över telefon.

Att konversation över mejl minskade berodde även på att det här blev rörligt. Vi ändrade detta efter att problemet blivit uppmärksammat i andra Sprinten och kom fram till regler för hur konversation över mejl skulle ske. Dessa regler innehöll att man skulle använda sig av samma tråd för alla konversationer och att man skulle bifoga alla medlemmar i teamet direkt.

Efter en Sprint märkte vi även att Product Backlog inte fungerade som tänkt. Efter uppmärksammande av detta problem och analys kom vi fram till att det berodde på att våra user stories var för stora. Detta härstammar också från utbildning. Att ha för stora user stories leder till att user stories inte hinner bli klara i tid och att Product Backlog inte uppdateras på grund av detta. Detta problem är inte specifikt för distribuerad Scrum. Men vad som kan hända i distribuerade projekt är att Product Backlog glöms bort eftersom den inte är synlig på samma sätt som i ett samlat projekt. I vårt projekt ledde detta till att Product Backlog blev svår att uppdatera. Den fyller då inte sin funktion eftersom vi i teamet inte använde den för att arbeta efter. Genom uppmärksammandet av detta minskades vissa user stories samt att alla i teamet fick större medvetande var man kunde hitta Product Backlog, vilket ledde till att teamet kunde se framstegen som gjordes och en direkt påverkan på moralen i gruppen märktes.

Sprint Backlog var ett annat verktyg som även ledde till förändring. Problemet vid distribuerade projekt är att den inte är synlig på samma sätt som i ett samlat projekt (då den kan vara synlig för att medlemmar på en Whiteboard). Trello som är ett verktyg som alla i teamet gillade, just för att det påminde användandet av Whiteboard och post-it-lappar, nåddes via en internetadress. Då inte alla i teamet hade Trello uppe på sin dator under Daily Scrum blev det problem att uppdatera vad som görs fram till nästa möte och vad som är klart. Detta blev bättre efter att problemet uppmärksammats och teamet bestämde att den måste vara uppe varje Daily Scrum och att det är allas möjlighet att uppdatera den.

Att allt tog lång tid var ett problem som blev bättre och bättre ju längre in i projektet vi kom. Den avgörande faktorn här var att de kom ner till Göteborg och vi kunde sitta tillsammans och arbeta och umgås utanför projektet. Ju tidigare projektet hinner träffas desto tidigare kommer man lära känna varandra och skapandet av en bra social struktur sker snabbare. Uteblir denna träff kommer det ta längre tid att skapa en bra social struktur. Detta leder till att man vågar ställa frågor som kan vara obekväma och att kommunikationen blir mer effektiv. Det blir då direkt avgörande för projektets framgång eftersom man inte har tid att vänta på att en bra social struktur skapas då man måste börja leverera från Sprint ett.

En annan faktor som påverkade att projektet blev mer effektivt var att vi efter uppkomna problem tog fram regler för hur verktygen skulle användas. Detta är något som bör göras i uppstartsfasen eller i början av första sprinten, i samband med att vissa verktyg får en person i teamet som ansvarig för just det verktyget. Exempelvis blev det rörigt i vår Dropbox då alla i projektet kunde skapa nya mappar, vilket ledde till en komplicerad struktur. Något som hade kunnat undvikas genom en ansvarig person som hanterar strukturen.

6.2 Agila manifestet

För att arbeta distribuerat krävs det att man följer efter det agila manifestet. Vi kom fram till att vi följer detta manifest på alla punkter förutom en, nämligen individer och interaktioner framför processer och verktyg. Trots detta så menar vi att vi lyckas bedriva detta projekt med

den agila metoden Scrum. Genom att bryta första punkten lyckas vi arbeta med Scrum distribuerat, något som inte varit möjligt om annars.

Att verktyg och processer går framför individer och interaktion är något som måste göras då dessa är själva nyckeln för att hantera interaktion då projektet inte är samlat. Går inte dessa framför individer och interaktioner kommer kommunikationen bli komplicerad och påverka projektet. Vad vi vill lyfta fram är tydlighet. Vad vi menar är att alla i teamet skall veta vilka verktyg som används för att interagera med varandra på olika möten och om det finns bestämmelser för hur, så måste dessa följas. Detta uppnås inte om man inte lägger fokus på verktyg och processer vid projektets start.

6.3 Riktlinjer

Nedan presenterar vi våra framtagna riktlinjer som bör följas för att bedriva distribuerad Scrum så effektivt som möjligt. Våra riktlinjer lyfter fram aktiviteter som blir viktiga att genomföra för att lyckas med distribuerad Scrum, observera att dessa riktlinjer ses som kompletterande till övriga aktiviteter som behövs för att planera ett Scrumprojekt.

Skall vi bortse från saker som redan finns med vid vanlig Scrum, typ User Story?

Förberedelser

- Utbilda teamet
- Välj “rätt” verktyg för ditt projekt
- Välj kommunikationskanaler för att hantera Scrums aktiviteter
- Samla alla i teamet
- Skapa en social struktur

Lyckas man genomföra de tre första punkterna, i samband med att alla i teamet är samlade och lär känna varandra kommer man snabbare skapa en bra social struktur vilket leder till att teamet kommer kunna leverera redan från första sprinten.

Uppstart

- Välj inte för många verktyg
- Bestäm hur Daily Scrum skall ske
- Bestäm när Daily Scrum skall ske
- Se till att alla i teamet vet var Product Backlog finns
- Utse personer som är ansvariga för olika verktyg
- Ta fram vilka regler som gäller vid användande av olika verktyg

I uppstarten blir det viktigt att alla i teamet är medvetna om ovanstående punkter för att projektet skall kunna fungera direkt från första Sprinten.

Projektet

- Se till att samla teamet om det inte har skett än
- Sköt Sprint Retrospective Meeting
- Var noga med Daily Scrum
- Se till att alla har Sprint Backlog synlig under Daily Scrum

Det är hög tid att samla alla i teamet om man fortfarande inte gjort det. Det som blir extra noga är att man sköter Sprint Retrospective Meeting och Daily Scrum. Den förstnämnda för att identifiera problem som uppstår i ditt projekt och den sistnämnda för att teamet skall klara av att leverera det som skall levereras vid Sprintens slut.

7. Slutsatser

Identifieringen av problem och hur dessa påverkar det agila manifestet och som sedan mynnar ut i riktlinjer för hur man bör bedriva distribuerad Scrum är vårt huvudsakliga resultat och besvarar frågeställningen:

Vilka problem uppstår vid distribuerad Scrum och hur påverkar de det agila manifestets riktlinjer?

Våra viktiga upptäckter i denna uppsats är att vi endast bryter en av de fyra principerna i det agila manifestet. Den princip vi bryter mot är individer och interaktioner framför processer och verktyg. Både begreppet individer och interaktion påverkas då projektmedlemmarna sitter på olika platser i Sverige där individer påverkas genom att de inte träffar varandra och interaktion genom att den inte sker direkt, öga mot öga. Att det bara är en princip som bryts innebär att det går att arbeta med Scrum i distribuerade projekt. Viktigt blir att man tidigt lägger fokus på hur man kommunicerar och interagerar med varandra. Med hur menar vi att man skall bestämma vilka kommunikationskanaler och verktyg som skall användas. Viktigt att tänka på är att man inte använder sig av för många verktyg vilket lätt kan göra interaktionen rörig. Man bör även utse en som är ansvarig för att visst verktyg för att på så sätt hålla det uppdaterat.

Andra faktorer som spelar in var att samtliga i projektet var nya på Scrum och samtliga projektdeltagare var villiga att arbeta med Scrum och såg detta som något spännande. Även att projektdeltagare inte var bekanta med varandra och de anställda på Riksnet hade uppgifter vid sidan om samt att Scrum Master och Product Owner ingick i Scrum-teamet var faktorer som spelade in.

Vi upplevde även att en avgörande faktor är att man bör försöka ses minst en gång under projektet, gärna i uppstartsfasen av projektet. Detta leder till många fördelar som påverkar gruppen och projektet till det bättre.

7.1 Studiens relevans och överförbarhet

Denna undersökning och rapport skrevs i samband med utvecklandet av Riksnets webbportal. Rapportens riktlinjer är tänkt att vara användbara för organisationer av samma storlek som Riksnet och som tidigare inte har någon tidigare erfarenhet av Scrum och som vill arbeta med distribuerade projekt som sker utan kulturella skillnader och utan tidsskillnader. Detta är relevant då organisationer blir mer och mer utspridda, vilket kräver att det finns en kunskap inom organisationer om hur man arbetar distribuerat. Något vi hoppas uppnå med våra riktlinjer. Riktlinjerna skall fungera som generella riktlinjer för dessa organisationer.

7.2 Förslag till fortsatt forskning

Denna uppsats ser endast till projekt inom ett lands gränser i en mindre organisation. Tre områden som är intressanta, vilka inte berörs i denna uppsats, och som kan ha stor påverkan

på distribuerade projekt är kulturella skillnader, tidsskillnader och större organisationer och mer komplexa projekt. Vårt förslag till vidare forskning rör då dessa tre områden. Då organisationer går över nationers gränser är dessa områden viktiga för framtida forskning.

Vidare hade det även varit intressant att se hur distribuerad Scrum fungerar i projekt med team som är vana vid att arbeta med Scrum. Då dessa team redan vet hur viktigt det är med dagligt kommunikation kanske de redan från start kan undvika många problem.

Referenser

Tryckta källor

Björkholm, T., Brattberg, H. (2010), *Prioritera, fokusera, leverera – din snabbguide till Lean, Agile, Scrum och XP*, Vulkan.

Boehm, B., Turner, R. (2003), *Rebalancing Your Organization's Agility and Discipline*, Springer-Verlag Berlin Heidelberg

Ferrance, E. (2000), *Action Research*, LAB at Brown University

Herbsleb, J.D., Moitra, D. (2001), *Global software development*, IEEE Software, March/April

Holcombe, M. (2008), *Running an Agile Software Development Project*, Wiley

Kniberg, H. (2007), *Scrum and XP from the trenches – How we do Scrum*, Lulu.com

Larman C, (2005), *Applying UML and Patterns*, Prentice Hall

Lewin, K. (1958), *Group Decision and Social Change*. New York: Holt, Rinehart and Winston.

McNiff, J. (2002), *Action Research: Principles and Practice*, RoutledgeFalmer

Rasmusson, J. (2010), *The Agile Samurai: How Agile Masters Deliver Great Software*, The Pragmatic Programmers.

Rising, L., Norman, S.J. (2000), *The Scrum Software Development Process for Small Teams*, IEEE SOFTWARE July/August

Rayhan, S., Haque, N. (2008), *Incremental Adoption of Scrum for Successful Delivery of an IT Project in a Remote Setup*. Agile 2008 Conference

Schwaber, K. (2007), *The enterprise and Scrum*, Microsoft press

Schwaber, K., (2010), *What Is Scrum?*, Volaro

Sharp, H., Rogers, Y., Preece, J. (2011), *Interaction Design: Beyond human-computer interaction 3rd edition*. Wiley

Sliger, M. (2006), *Bridging the gap - Agile projects in the Waterfall Enterprise*, Better Software

Winston, W, R. (1970) *Manging the development of large software systems*, IEEE WESCON

Elektroniska källor

Chon, M. (2005), *An Overview of Scrum for Agile Software Development*.
<http://www.mountaingoatsoftware.com/scrum/overview> [2013-04-16]

Manifestet för Agil systemutveckling. (2001), <http://agilemanifesto.org/iso/sv/> [2013-05-13]

The Scrum Guide. (2011), <http://www.scrum.org>, [2013-05-22]

Bilaga 1 - Intervjufrågor

- Vilka problem tror du det finns med att arbeta med Scrum distribuerat?
- Vårt projekt kommer vara i 6 veckor och vår tanke nu är att köra två stycken 1-veckosprintar och två stycken 2-veckosprintar. Vad tycker du om det upplägget? Hur hade du gjort?
- Om vi inte kommer få ha dagliga möten med dem, hur påverkar det projektet?
-Hur viktig är den dagliga interaktionen när man använder sig av Scrum?
-Hur kommer det fungera att ha möten över skype eller telefon?
- Hur skapar vi en bra backlog och hur ser vi till att den uppdateras?
- Har du några tips på hur man ska arbeta distribuerat?
- Har du några tips på hur man kan använda sig av Scrums riktlinjer när man arbetar distribuerat?
- Finns det någon punkt du specifikt kan peka på som är avgörande för hur lyckat ett Scrumprojekt blir?