

Agil systemutveckling

- en jämförelse mellan den agila och traditionella projektledaren

Kandidatuppsats, 10 poäng, i Informatik

Framlagd: 15 juni, 2007

Författare: Engwall, Emma
Jacobsen, Björn

Handledare: Steen, Odd

Examinatorer: Persson, Claus & Fernebro, Lars

Agil systemutveckling

- en jämförelse mellan den agila och traditionella projektledaren

© Emma Engwall, Björn Jacobsen

Kandidatuppsats framlagd: 15 juni, 2007

Omfång: 68 sidor

Handledare: Odd Steen

Resumé: Den här uppsatsen handlar om projektledning av projekt som tar stöd av agila systemutvecklingsmetoder. Från början skapades de agila metoderna, av utvecklarna själva, som ett svar på de traditionella metodernas rigiditet och, i deras mening, övertro på en dokumenterad process. Ordet "agil" betyder lättroilig och kärnan i de agila metoderna är att göra utvecklingen mer flexibel och lättare att manövrera. Det finns andra värderingar som är grundläggande i agil systemutveckling som påverkar projektledaren och dennes arbete för ett agilt projekt. Utgående ifrån en litteraturstudie kring traditionella projekt och den traditionella projektledaren, samt en empirisk studie av agila projekt och den agila projektledaren, är syftet med uppsatsen därför att visa på skillnaderna i projektledarens roll och arbete mellan traditionella och agila projekt. Problemet som vi försökt besvara är "Hur skiljer sig arbetet för en agil projektledare och det föreskrivna arbetet för en traditionell projektledare åt?", med följdfrågan "Förväntas en agil projektledare ta en annan roll och i så fall på vilket sätt?". Utgående ifrån en riktningvisare (Briner, Geddes & Hastings, 1999), som visar vilka riktningar en projektledare måste vända sig i sitt arbete, har den agila projektledarens arbete och roll jämförts med den traditionella projektledarens. Resultatet av undersökningen och den sammanfattade bilden av den agila projektledaren är en befattning som har en starkare PR- och utbildande roll än den traditionella, resultatansvaret har lyfts bort och ligger högre upp i organisationen och till sist arbetar den agila projektledaren främst som möjliggörare för ett självorganiserande team som projektledaren inte ingår i eller har bestämmanderätt över. De personliga och formella egenskaper som Lööv (1999) lyft fram som viktiga för en traditionell projektledare gäller även för den agila projektledaren. Men en starkare betoning på politiska egenskaper märks för den agila projektledaren. I anslutning till studien av den agila projektledaren studerades även agila metoder och hur de förhåller sig till traditionell projektledning. Den jämförelsen gav, utgående ifrån Lööv (1999), ett antal fallgropar och framgångsfaktorer för agila projekt.

Nyckelord: Systemutvecklingsmetod, projektledning, traditionell projektledare, agil projektledare, Scrum

Tack!

Under uppsatsens gång har vi stött på problem som vi inte riktigt förutsåg från början. Det skulle visa sig att ämnet agil systemutveckling är omgivet av starkare känslor än vad vi hade en aning om. Därför vill vi rikta ett tack till Ulf Atles på Appium AB för att han ledde in oss på detta intressanta område till en början och sedan för fortsatt hjälp. Tack till alla de tillmötesgående personerna på de intervjuade företagen som vi har fått kontakta i tid och otid med fler frågor efter intervjuerna. Hurra för Jens Norin på Softhouse som kommit med glada tillrop när vi själva hade svårt att le. Tack också till mamma och pappa Engwall som med föräldrars subjektiva ögon granskade och sa snälla saker. Sist men inte minst vill vi rikta ett stort tack till vår handledare Odd Steen.

Index

1	INLEDNING	1
1.1	Allmänt	1
1.2	Problemperspektiv	3
1.2.1	Problemformulering	4
1.2.2	Syfte.....	4
1.3	Intressenter	4
1.4	Begreppsdefinition.....	4
1.5	Avgränsningar.....	4
2	TILLVÄGAGÅNGSSÄTT.....	6
2.1	Sammanfattning	6
2.2	Val av metod.....	6
2.3	Urval av teori och källkritik	7
2.4	Urval av respondenter	8
2.5	Metodreflektion.....	9
2.6	Intervjuernas genomförande	10
3	TRADITIONELL PROJEKTLEDNING.....	12
3.1	Projekt	12
3.1.1	Projektets ramar	12
3.1.2	Uppdragsgivaren	12
3.1.3	Projektdeltagare	13
3.1.4	Övriga intressenter i projekt.....	13
3.2	Projektledaren.....	13
3.2.1	Framgångsfaktorer och fallgropar	16
4	SYSTEMUTVECKLINGSMETODER.....	17
4.1	Bakgrund	17
4.2	Historisk systemutveckling	18
4.3	Agila metoder	19
4.4	Disciplinära vs Agila metoder.....	21
5	SCRUM – METODEN.....	23
5.1	Aktörerna.....	24
5.1.1	Scrum teamet.....	24
5.1.2	Produktägaren.....	24
5.1.3	Scrum Master.....	24
5.2	Arbetsgång.....	25
5.3	Demonstration och utvärdering.....	26
6	FÖRETAGEN OCH RESPONDENTERNA.....	27
7	EMPIRISKT RESULTAT	28
7.1	Scrum i användning.....	28
7.2	Fördelar och nackdelar med Scrum.....	30

7.3	Scrum Master.....	30
8	DISKUSSION	33
8.1	Förhållandet mellan Agil & Traditionell projektledning	33
8.2	Scrum i användning	35
8.3	Den agila projektledaren	36
9	SLUTSATSER	41
10	FÖRSLAG TILL VIDARE FORSKNING.....	43
11	BILAGA A: INTERVJUFORMULÄR.....	44
12	BILAGA B: BACKLOG.....	45
13	BILAGA C: TRANSKRIBERINGAR	46
14	BILAGA D: AGILA MANIFESTET & PRINCIPER.....	59
15	KÄLLFÖRTECKNING	61

Index: Figurer och tabeller

Figur 3-1	Projektledarens riktningvisare	15
Figur 5-1	Scrum sprint.....	23
Figur 5-2	Burn down chart	26
Tabell 4-1	Agila och disciplinär metodens "home ground"	21
Tabell 4-2	Fem skillnader mellan agila och disciplinära metoder.....	22

1 Inledning

Som titeln avslöjar så handlar den här uppsatsen om projektledning. Mer specifikt, ledning av projekt som tar stöd av agila systemutvecklingsmetoder.

1.1 Allmänt

Grundtankarna till den här uppsatsen erhöles under kandidatkursens inledning då uppsatsförfattarna skrev en artikel som i grunden skulle handla om en systemutvecklarens vardag och metodanvändning. Genom kontakter, men också som något av en slump, halkade vi då in på agil systemutveckling. Ämnet kändes väldigt intressant att undersöka vidare även efter artikelns avslutande. Det tycktes någonstans finnas en diskrepans mellan det vi som informatikstudenter får läsa i böckerna om systemutvecklingsmetoder och det vi såg på de företag som intervjuades för artikeln.

Projekt som verksamhet har en över 60 år lång historia, från början kopplat till stora militärindustriella projekt och senare bygg- och anläggningsindustrin. Idag är projektarbetsformen en väldigt stor del av organisationers och människors vardag vilket har lett till att företagsledningar fokuserar på att effektivisera projekten. Det har också blivit allt viktigare att vara först ut på marknaden med nya produkter och att snabbt kunna förändra organisationen för att möta nya omvärldskrav. (Norrgrén, 2006)

Fram till 1980-talet dominerades projektledning av att duktiga praktiker systematiserade sina erfarenheter på projekt och därmed försökte utveckla metoder för projekt och läroböcker i ledning av projekt. Då grundtanken med projektledning har varit att styra, stödja och kontrollera processen har projektledaren traditionellt fått en väldigt auktoritär roll. (Norrgrén, 2006)

Trots den ökade fokuseringen på utbildning och sofistikerade metoder kvarstår många av problemen inom projektledning. Projekten blir fortfarande ofta försenade och överskrider tänkt budget. I princip ända fram tills nu har man försökt möta dessa problem med att förädla beprövade metoder, man reducerar osäkerhet genom en rigorösare planering eller utser "tung" personer med traditionella chefsegenskaper till projektledare. (Norrgrén, 2006)

Informationssystem utvecklas inte i ett vakuum utan omvärlden påverkar både utveckling och användning. En omvärld vars komplexitet och dynamik bara ökar (Avison & Fitzgerald, 2002). Enligt Christensen och Kreiner (1997) är det inte längre ett realistiskt antagande att företag kan handla opåverkade av varandra, utan vi utgör alla varandras omgivning och reagerar på de möjligheter och hot som omgivningen representerar. En handling förändrar villkoren för alla andra och konsekvenserna av ett handlande blir därmed svåra att förutse då de snarare blir ett resultat av en komplicerad interaktion mellan flera olika aktörer.

Att företag idag tvingas arbeta i projekt ständigt kan ses som ett symptom på den osäkerhet, turbulens och de snabba förändringar som präglar verkligheten. Förändringarna innefattar både marknad och teknik, samt kvalifikationer, värderingar och attityder bland medarbetare.

Christensen och Kreiner (1997) menar vidare att detta leder till en attitydförändring på många företag. Speciellt företag som befinner sig i snabba branscher. Den största förändringen ligger enligt Christensen och Kreiner (1997) i synen på ordning respektive oordning. Tidigare stod systemet för ordningen genom hierarki, uppgiftsstruktur, strategiskt planerande och auktoritära ledarfigurer. I den förändrade verksamheten har systemet fått den motsatta rollen genom att placera individen i ett nätverk av externa och interna relationer. Ett ömsesidigt beroende växer fram och blir tydligare och som individ krävs en stor förmåga till situationsanpassning för att kunna fungera. En auktoritär ledarfigur i ett projekt ges inte makt eller utrymme att kunna fungera i en sådan miljö utan en ledningsfilosofi som inspirerar till individuellt tänkande och agerande välkomnas hellre.

Det finns idag ett stort antal olika metoder för systemutveckling och projektledning som grundar sig på lite olika filosofi. En del av dem bygger på forskning andra på praxis, en del av dem har funnits med väldigt länge andra är relativt nya. Generellt kan man säga att de metoder som baserar sig i forskning och är lite äldre står för en större formell struktur än vad de praktiskt framtagna metoderna gör.

Agil betyder lättroblig och kärnan i de agila metoderna är att försöka göra utvecklingen mer flexibel och lättare att manövrera. Grundtanken är att i en föränderlig värld krävs utvecklingsmetoder som hanterar förändring som en del av verkligheten, inte sådana som blundar för förändringar eller försöker reglera bort dem. Flexibilitet istället för rigiditet efterfrågas. (Rozenberg, 2007) Mot bakgrunden av att utvecklingen ofta körde fast eller tog för lång tid så att den levererade produkten var obsolet redan vid leverans, fokuserar man mindre på att dokumentera processen än vad som är normalt i de traditionella metoderna och satsar på det man kallar ett direktare värdeskapande (iterativ utveckling) i nära samarbete med kunden. Attityderna kring planering och vad som går att förutse är annorlunda och man väljer att gå en annan väg för att hantera osäkerheten i omvärlden. (Agile Alliance, 2006)

De agila metoderna har vuxit fram underifrån, bland utvecklare och programmerare, som ett svar på de traditionella systemutvecklingsmetodernas formalism. "I en programmerares värld är det roligare att skriva kod än att fylla i olika dokument". Avsaknaden av regler och dokumenterad process fick omvärlden att kalla metoderna för hackermetoder som dessutom saknade vetenskaplig legitimitet. (Avison & Fitzgerald, 2002)

År 2001 bildades The Agile Alliance, en allians som marknadsför och verkar för användning av agila metoder. (Agile Alliance, 2006) "Agile" har blivit ett paraplybegrepp för en uppsättning värderingar, attityder och principer och det finns ett antal utvecklingsmetoder inom den agila familjen (Rozenberg, 2007). Alliansen har medverkat till att dokumentera ett antal agila metoder, med samma grundvärderingar och syn på utveckling, men främst tas de upp i familjen då upphovsmannen bekänner sig till de agila värderingarna. Scrum är en av de bäst dokumenterade agila metoderna och bland de företag som valt ett agilt arbetssätt i Sverige är den även den vanligaste (Lundgren BILAGA C [89]). Det är även denna metod vi valt att inrikta uppsatsen mot.

Utvecklingen av metoder för projektledning och för systemutveckling har inte gått hand i hand men de har följt varandra. De är utvecklade under samma era och med samma

värderingar. De agila metoderna bryter mot de värderingarna och man anser att de traditionella metoderna är för tungrodda samt lägger för mycket fokus på icke värdeskapande arbete och att det är på grund av detta som projekten blir försenade och drar över budget (Fowler 2000, Cockburn, 2004).

1.2 Problemperspektiv

De agila metoderna har inte sin grund i akademisk forskning och har därmed inte blivit granskade på samma sätt som många andra metoder. Bakgrunden har till och med gjort att den akademiska världen fram tills nu verkar ha tagit lite avstånd från dem, de står nämligen för en ganska provocerande motbild till det som varit norm. Det största problemet man har sett är att de agila metoderna inte är kompletta som metoder utan mer lösa tekniker och verktyg, något som kan fungera för en programmerare men inte för en systemerare eller en hel verksamhet.

The Agile Alliance marknadsför de agila metoderna väldigt hårt. Känslan av att slåss underifrån samt viljan att vinna mark har lett till en ibland ganska onyanserad marknadsföring där de traditionella metoderna beskrivs väldigt negativt och de agila enbart positivt. Den mesta informationen kring agila metoder finns att hitta på Internet, bland de företag som använder och utbildar i agila metoder. Detta kan leda till ytterligare snedvridning av informationen då dessa företags syfte är att sälja in och tjäna pengar på metoderna och inte beskriva dem alltför utförligt. Just nu har det gått lite mode i att utveckla agilt och till och med företag som är företrädare för traditionella metoder hoppar på tåget och vill visa hur man kan arbeta agilt inom ett formellt ramverk (IBM 2005).

För många är agil utveckling fortfarande väldigt nytt, kunskapen om det har inte riktigt nått till de generella läroböckerna inom systemvetenskap än och därför tas det upp bara väldigt kort under utbildningen. För oss, som studenter i informatik, väcktes nyfikenheten på att veta mer om den värld som agil systemutveckling utgör och som verkade vara "common knowledge" då vi intervjuade företag för artikeln, men som vi, genom institutionen, inte kunde få veta mer om.

Högskolorna ger kurser i projektledning där den traditionella synen på och egenskaperna för en projektledare lyfts fram. En syn som de agila förespråkarna inte verkar dela oförbehållet.

De agila metoderna bygger på en annan hierarkisk struktur, även om de vidareutvecklats för att passa hela verksamheter så ligger deras grund trots allt hos utvecklarna själva, därför nämns orden ledning, styrning och kontroll väldigt lite. Ordet "projektledare" förefaller värdeladdat och inom Scrum kallar man rollen som ligger närmast projektledarens för Scrum Master. Men valet att inte kalla det för projektledare vittnar även om att det finns något som skiljer en Scrum Master och en traditionell projektledare åt. Denna skillnad har bara inte konkretiserats för den allmänna betraktaren.

Därför kändes det relevant att undersöka hur synen på projekt och projektledning skiljer sig åt mellan den traditionella och den agila skolan.

1.2.1 Problemformulering

Ovanstående diskussion leder till följande problem att undersöka:

Hur skiljer sig arbetet för en agil projektledare och det föreskrivna arbetet för en traditionell projektledare åt? Förväntas en agil projektledare ta en annan roll och i så fall på vilket sätt?

1.2.2 Syfte

Syftet med uppsatsen är att, ur projektledarens perspektiv, visa på skillnaderna i projektledarens roll och arbete mellan traditionella och agila projekt.

Arbetet utgår ifrån en litteraturundersökning kring hur den traditionella projektledaren förväntas agera och en empirisk undersökning av agila projekt, med ett ytterligare syfte att kunna ge ett teoretiskt bidrag till litteraturen kring projektledning.

1.3 Intressenter

Uppsatsen skrivs för Informatikinstitutionen vid Ekonomihögskolan i Lund och riktar sig främst till lärare och studenter på institutionen. Företag eller projektledare med funderingar kring agil systemutveckling och ledning av projekt i en sådan miljö kan vara ytterligare intressenter.

1.4 Begreppsdefinition

Det engelska ordet "agile" betyder lättroilig. I Sverige verkar man inte ha enats om en gemensam översättning för "agile methods" utan man hittar benämningar som lättroiliga metoder, lättviktsmetoder och agila metoder. I intervjutranskripten används den benämning som företaget själva valt. I övriga uppsatsen används benämningen agila metoder.

Då man inte talar så mycket om projektledare inom Scrum utan istället talar om Scrum Master, är benämningen "agil projektledare" egenkomponerad, men syftar till ledaren av ett agilt projekt, vilken kan vara en Scrum Master.

Som motpol till agil projektledning och agil metod har vi i uppsatsen valt att använda oss av benämningen traditionell projektledning och traditionell metod. Ord som formell och disciplinär som motsatsord till agil är vanligt förekommande i litteraturen. De orden reflekterar syn och åsikter hos förespråkarna för respektive skola och kommer därför att användas oförändrat då det är den synen som ska komma fram.

Om inget annat anges kommer systemutvecklingsmetoder fortsättningsvis att benämnas i kortform som "metoder", detta för att skapa ett bättre flyt i språket.

1.5 Avgränsningar

Det är enbart rollen som agil projektledare, eller Scrum Master, som undersökts och inga andra roller i ett projekt, även om övriga roller kommer att omnämnas vid tillfälle då vi först tittar på hur det generellt är att arbeta med agila metoder. Med roll avses kombinationen av

konkreta bestämda uppgifter som ligger i tjänsten som agil projektledare tillsammans med ett övergripande beteende och personliga egenskaper denne bör ha.

Den empiriska undersökningen innefattar bara agil projektledning och ställs mot den teoretiska bild som litteraturen ger av den traditionella projektledaren.

2 Tillvägagångssätt

I detta kapitel görs en genomgång av metoden, som sedan leder in på en motivering av vald metod. Kapitlet tar även upp källkritik, validitet och reliabilitet.

2.1 Sammanfattning

Efter kandidatkursens inledande del, då en artikel om agila systemutvecklingsmetoder skrevs, var grunden till ämnesområdet lagd och även en del kontakter knutna. För att skapa en stabilare grund för det fortsatta arbetet genomfördes en litteraturstudie för att få en uppfattning om vad som var relevant att undersöka vidare inom området. Hjälp togs även av Ulf Atles på Appium AB för att få en företagsvinkel på vad som kunde vara intressant att titta på. I en diskussion med institutionens handledare snävades ämnet senare in till att behandla projektledaren. Uppsatsen har sedan skrivits helt och hållet för informatikinstitutionen och utan krav från andra företag.

Då det inte finns mycket litteratur som tar upp både projektledning och systemutvecklingsmetoder i en och samma bok, beslutades det att studien skulle vara explorativ. En större litteraturstudie genomfördes med fokus på två områden; projektledning och systemutvecklingsmetoder. Detta mynnade ut i uppsatsens teoretiska del och låg även till grund för den efterföljande empiriska undersökningen i form av underlag för intervjufrågor osv.

Det empiriskt utvalda materialet ställdes därefter mot befintliga teorier från litteraturen i en jämförelse för att hitta meningsfulla skillnader mellan det traditionella och det agila. Därigenom utkristalliserade sig ett antal slutsatser och tillägg till befintlig teori kring projektledning. Framtida studier inom de agila systemutvecklingsmetoderna kan förhoppningsvis bekräfta eller bygga ut dessa teorier ytterligare.

2.2 Val av metod

I denna uppsats är det viktigt att skilja på forskningsmetod och systemutvecklingsmetod. Den förra är ett redskap som kan vara olika väl lämpad för forskning inom olika problemområden. Alla undersökningar förutsätter att forskaren på något sätt samlar in, bearbetar, analyserar, tolkar och presenterar data. Bryman (2001) tar upp kvantitativa och kvalitativa forskningsstrategier. Det som framförallt skiljer de två strategierna åt är att den kvalitativa inriktar sig mer på ord än siffror. Forskare som utnyttjar kvalitativa undersökningsmetoder anser att den sociala verkligheten är i ständig förändring och påverkas av individernas skapande och konstruerande (Bryman, 2001). Tyngden inom kvalitativ forskning läggs på generering av teorier och det som söks är individers tolkning och uppfattning av sin verklighet. Detta tar helt avstånd från den naturvetenskapliga modellens normer och tillvägagångssätt där man istället söker ett totalt perspektiv och fullständig förståelse för det man ämnar undersöka. Vid val av metod är det därför viktigt att man utgår från det problemområde man rör sig inom. (Holme & Solvang, 1997)

Eftersom det fenomen vi ämnar undersöka är relativt nytt och outforskat, har vi för avsikt att finna mönster och indikationer. Syftet med uppsatsen är inte att kvantifiera data om den agila projektledaren, utan istället att få den implicita kunskapen explicit. Detta gör att den mest lämpade metoden är den kvalitativa.

Bryman (2001) tar upp olika intervjutekniker som kan användas inom den kvalitativa forskningen. Det är framförallt två olika tekniker som är gångbara, den ostrukturerade respektive den semistrukturerade intervjun. I den ostrukturerade ger frågeställaren respondenten mer frihet genom att man använder sig av en lägre nivå av kontroll och det är inte tvång på att teman följs och besvaras. I den semistrukturerade använder forskaren sig av en viss mall som denne förhåller sig till, på denna finns olika teman nedtecknade som skall undersökas. Med denna teknik har forskaren större möjlighet att styra intervjun och kan genom mallen fördjupa sig mer inom en viss fråga genom att ställa följdfrågor.

Då vår studie hade som mål att få fram så mycket som möjligt som kan förklara hur den agila projektledare skall vara och verka, samt även få ett mer hållbart frågeformulär, valde vi att utgå från den semistrukturerade intervjun. Genom fyra djupintervjuer önskade vi få fram det material som krävdes.

2.3 Urval av teori och källkritik

Teorin som återfinns i uppsatsen har valts utefter problemområdet. Initialt valdes till viss del annan teori, men då problemområdet ändrat karaktär under uppsatsens gång har även teorivalen behövt revideras.

Vi bestämde oss för att dela in teorin i tre delar, projektledning ur en traditionell synvinkel, systemutvecklingsmetoder och Scrum.

Teoriavsnittet inleds med en genomgång av projektledning i kapitel 3. Detta är också det teorikapitel som används mest då vikten i uppsatsen ligger på att jämföra den agila projektledaren med den traditionella. Kapitel 4 och 5 ger kött på benen och en viktig bakgrund till läsaren. För att komma in i tankegångarna och lättare kunna hantera den agila utvecklingsteorin såg vi ett behov av att ge en bild av traditionell systemutveckling, detta görs i kapitel 4. Där finns även en extensiv jämförelse mellan agil och traditionell systemutveckling, även detta för att kunna ge läsaren ett bättre grepp om de viktiga skillnaderna mellan traditionellt och agilt synsätt. Teorierna som berör Scrum, kapitel 5, är också viktiga då de ökar kunskapen och minimerar tolkningsutfallen som berör empirin.

Källkritik är ett slags systematiserat förnuft där man söker svar på frågor som: Vem är källan? Vilka möjligheter har källan att förstå och tolka ursprungsdata? Vilka intressen har källan? och Vad har hänt med budskapet på vägen?. Poängen med källkritiken är att kunna bedöma hur sannolikt det är att uppgifterna som fåtts är riktiga, utifrån vetskapen om källan. (Holme & Solvang, 1997) Då det gäller de traditionella och de agila systemutvecklingsmetoderna valde vi, efter en extensiv litteraturundersökning, att vända oss till de största namnen (de namn som återkom vid sökning) för referenser. Då vår litteraturundersökning var så pass omfattande fick vi också en väldigt bra bild av området och kunde se vilka källor som erbjöd bra och korrekta beskrivningar. Det finns en oerhörd

mängd litteratur kring traditionell projektledning, den mesta praktiskt inriktad och menad som guide och stöd åt projektledaren. Men de flesta tar trots det upp karakteristika för en projektledare. Under litteraturundersökningen fann vi en stor likriktning i den information som ges kring projektledaren. De källor som vi har valt att referera till i uppsatsen är de som gett en väldigt bra sammanfattande bild av vad de flesta andra böcker kring projektledning också skriver. De har alla haft specifika avsnitt som enbart handlar om projektledaren. Boken Projektledaren av Briner, Geddes och Hastings (1999) är även den bok som används vid institutionen för Informatik i projektledningskursen, därför kändes den relevant och viktig att ha med. Speciellt till den källan kan vi även utvidga bilden av projektledaren för studenterna som läst kursen.

2.4 Urval av respondenter

Vår urvalsprocess av respondenter startade som ett snöbollsurval, där vi placerade ut ett antal förfrågningar till personer i vår närhet för att se om de kunde vägleda oss till agila projektledare (Scrum Masters) som använde sig av Scrum. Vi startade även ytterligare en process där vi gick igenom ett större antal företags hemsidor för att få fram vilka systemutvecklingsmetoder som användes av dessa. Genom att enbart rikta in oss på deras "lediga jobb" kunde vi på de flesta företag få fram vilken metod de använde sig av då kompetens på området eftersöks – Citerus är till exempel ett av dessa företag. Detta genererade ett tiotal möjliga företag som vi kunde kontakta. Efter denna eftersökning skickades ett e-mail ut till alla presumtiva respondenter. En del av dessa valde att inte svara alls på vår förfrågan, andra avböjde efter en första kontakt att svara på våra frågor. Tre stycken svarade att de kunde ställa upp, men svaren inkom för sent och vi hade ingen tidsmöjlighet att bearbeta det stora material som en intervju producerar. Målet var att få fram de som var mest lämpliga att intervjua, alltså de som kunde ge mest information inom området. Men att de även var intresserade av att ställa upp samt att de hade tid med en intervju.

Holme och Solvang (1997) tar upp två olika typer av personer som går att använda sig av för att få det största möjliga informationsinnehållet från intervjuer, respondent respektive informant. En respondent är en person som tillhör gruppen som undersöks och en informant är en person som står utanför gruppen men som har mycket information att tillföra.

För att skapa en så stor bredd på uppsatsen som möjligt har vi ägnat mycket tid åt att finna några respondenter som är skickligare (seniora användare) i den föreskrivna användningen av Scrum som metod, och några som kanske är mindre kunniga i den föreskrivna användningen men som nyttjar metoden och har fått den att fungera. Detta för att se om olika tolkningar av metoden om möjligt kunde ligga till grund för något problem. Utan några projektledare med stor föreskriven kunskap, kunde det finnas en risk att de respondenter vi fått fram på något sätt hade misstolkat metoden på liknande sätt.

Det främsta målet har varit att de personer vi intervjuar, utan tvivel, ska vara representativa som projektledare inom metoden Scrum samt dess användning.

Vidare hade vi en stor önskan att även få tag på en informant som kunde upplysa oss mer om hur metoden är tänkt att användas samt hur det är tänkt att agila projektledare skall vara

och verka. Detta går även i viss mån att läsa sig till men vi ansåg att vi skulle få en mer nyanserad bild genom att intervjua någon som utbildar i agila metoder.

Slutligen hade vi kontakt med fyra lämpliga personer att intervjua, tre respondenter och en informant.

2.5 Metodreflektion

Med validitet menas hur giltig den insamlade informationen är. Problem med validitet minskar då man använder sig av en kvalitativ metod eftersom undersökaren får en närhet till respondenten och oklarheter kan förtydligas. En nackdel är att det är forskarens förutsättningar och föreställningar som påverkar tolkningen av det insamlade materialet. (Holme & Solvang, 1997)

Reliabilitet är sedan ett mått på i vilken utsträckning tillvägagångssättet, givet samma förutsättningar, ger samma resultat och att det kan generaliseras. Inom den kvalitativa metoden leder reliabiliteten till en stark sammankoppling mellan omständigheterna kring intervjuerna och resultatet. Det är därför av vikt att man inte påverkar respondenterna med personliga åsikter utan försöker hålla sig neutral till frågor och svar. Allt för att inte styra och påverka respondenten till att säga saker som denne inte står för. (Holme & Solvang, 1997) Kvale (1997) tar även upp betydelsen av att använda samma intervjuare då olika personers kunskap och känslighet för ämnet kan ge olika resultat.

Vårt samlade intryck från undersökningen är att det är ett omsusat område med starka förespråkare i två motstående läger. För att uppnå så hög validitet och reliabilitet som möjligt har vi utöver det kvalitativa metodvalet försökt släppa våra egna förutfattade meningar och ställa oss lite vid sidan av och titta objektivt på traditionella respektive agila metoder och projektledningsteorier. I grund och botten är vi väldigt nyfikna på de agila metoderna och efter den inledande artikeln kändes det som ett viktigt område att studera mer. Det fanns så mycket som tydde då på att ett alternativt tankesätt kring systemutveckling hade vuxit fram och verkligen började få fäste på företagen, men inte i den akademiska världen. Det var ingen "buzz" och det blev därmed relevant att ta upp. Det har varit väldigt svårt att inte påverkas av en yrkeskår som är så odelat och öppet positiv till sitt eget sätt att arbeta. Vår roll är dock inte att vara förespråkare för de agila metoderna, men inte heller granska och kritisera de agila metoderna som sådana. Vi studerar projektledarrollen och vill på ett neutralt sätt belysa hur den rollen förändras med agila projekt. Till den biten av problemområdet: Agila vs Traditionella systemutvecklingsmetoder, är det lättare att förhålla sig kritisk och inte påtvinga läsaren av uppsatsen med åsikter åt endera eller andra hållet. Vi har även valt att använda oss av mindre värdeladdade ord, ibland tvingats skapa egna ord, för att komma ifrån en situation som kan tolkas positiv eller negativ åt något håll. Men för att skildra verkligheten så som den ser ut använder vi ibland oss av de ord som författarna eller respondenterna använder. Därtill har vi haft uppsikt över vilka personer vi intervjuat samt använt oss av ett grundläggande intervjuformulär som riktar in svaren på ett gemensamt område.

Intervjuerna står för all empiri till uppsatsen och i kapitel 7 Empiriskt resultat finns en sammanställning av alla intervjuerna. Man får en oerhörd mängd information under en

intervju, sammanställningen tar därför bara upp det som vi anser vara viktigt för att kunna föra resonemanget vidare i Diskussion och Slutsatser. Det som tas upp i Empiriskt resultat är otolkat, men att vi valt att sortera i den information som vi fick fram från intervjuerna är en tolkning i sig. För att undvika validitetsproblem återfinns intervjuerna i sin helhet i Bilaga C: Transkriberingar. I likhet med intervjuerna har det även betydelse vem det är som transkriberar intervjuerna. Det finns en osynlig mall i huvudet på den som transkriberar och för att uppnå likformighet mellan de olika intervjutranskripten är det samma person som skött alla transkriberingarna. Alla fyra intervjuade har i efterhand haft möjlighet att läsa transkriberingarna och kunnat komplettera eller göra förtydliganden där de tyckte att så behövdes.

Det går att kritisera vårt tillvägagångssätt. Vi jämför inte med en traditionell projektledare i arbete utan har koncentrerat den empiriska undersökningen till agila projekt. Detta är inte optimalt utan det hade varit bättre att ta in även empiri från traditionell projektledning, för att styrka den teoretiska referensramen. Vårt val grundar sig dock i att vi anser att litteraturen kring traditionell projektledning är väldigt omfattande och den bild som sitter i huvudet på folk är den bild som skildras i litteraturen. Även om projektledning inte är något man kan läsa sig till så finns det en tro på att man kan det (annars hade det inte funnits projektledarutbildningar). Vi vill inte förändra den bilden utan bara lägga till ytterligare teorier kring agil projektledning då det har upptäckts ett behov för detta.

2.6 Intervjuernas genomförande

Den miljö i vilken intervjuerna genomförs är viktig att betänka (Holme & Solvang, 1997). I flera fall är det lämpligt att hålla intervjuerna på en plats vald av respondenten för att inte utsätta denne för stress. Respondenterna/informanten fick därför välja på vilket sätt de ville att intervjun skulle utföras, via möte, e-post eller telefon.

För att respondenterna skulle känna sig trygga, men även för att se om de var villiga att svara på våra frågor, hörde vi av oss i förväg och informerade om vad vårt arbete handlade om. Detta var i linje med Kvales (1997) råd att redogöra för respondenten vad som är syftet med intervjun samt hur den är tänkt att dokumenteras. Ulf Atles, Appium och Michael Rozenberg, Softhouse fick båda bestämma var och när intervjuerna skulle äga rum. Båda uppsatsförfattarna var närvarande vid de två intervjuer då de mötte respondenterna. Kroppsspråket hos intervjuaren vid ett personligt möte är viktigt att beakta (Holme & Solvang, 1997), det är centralt att intervjuaren är så öppen som möjligt och inte sitter med armar och ben i kors, vilket kan tolkas som slutenhet. Följaktligen hölls intervjuerna i en ganska avslappnad form på ett café respektive mötesrum med soffgrupp. För att undvika förvirring valde vi att enbart en av oss förde talan medan den andre antecknade. Vi upplevde även att detta gav ett bättre flyt i diskussionen då en person kunde koncentrera sig helt på att anteckna.

Henrik Sebring svarade via e-post på våra frågor, vi har även ställt följdfrågor via e-post där vi kände att det saknades information.

Informantintervjun med Mikael Lundgren genomfördes via telefon, det hade naturligtvis varit önskvärt om vi hade kunnat mötas för att kunna bedöma ansiktsuttryck och

kroppsspråk hos Lundgren. Men det geografiska avståndet gjorde detta svårt, istället fick vi fokusera på tonfallet. Det finns fördelar med att intervjua via telefon, en är att intervjuareffekten minskar, en annan att respondenten inte påverkas av intervjuarens kroppsspråk och ansiktsuttryck.

Under alla fyra intervjuerna arbetade vi utifrån en frågeguide (Bilaga A: Intervjuformulär) som i stort innehöll de frågor som togs upp under intervjun. I väldigt stor utsträckning talade respondenterna fritt, medan en av oss ställde följdfrågor för att få en bättre bild av respondentens/informantens uppfattning. Intervjuerna pågick i cirka en timme vardera.

Först använde vi oss av en demografisk fråga för att klargöra intervjupersonens personliga data, samt lättare kunna analysera svaren och göra en jämförelse. Resterande frågor hanterar områden som beteende, attityder, kunskaper och upplevelser. Alla frågor utformades på ett sätt så det var enkelt att ställa följdfrågor och få igång ett samtal. Det går att generalisera frågorna i två grupper:

***Scrum i användning** – som går ut på att ta reda på hur Scrum används av respondenternas respektive företag. Vilka fördelar och nackdelar ser man med metoden osv. Även för att kunna jämföra resultatet från intervjuerna var vi tvingna att ta reda på hur de använde sig av metoden.*

***Scrum master** – projektledarens påverkan av metoden samt projektledarens arbetsområde– ex vad som tar upp mest tid. Vad krävs av en projektledare inom Scrum – vilket stöd får projektledaren genom metoden, vilka egenskaper krävs osv.*

3 Traditionell projektledning

Kapitlet "Traditionell projektledning" inleder uppsatsens teoridel och behandlar det litteraturen har att säga om traditionell projektledning och befintlig teori kring projektledarrollen.

3.1 Projekt

Ett projekt brukar definieras som en uppgift som är avgränsad i tid och omfattning. Målet med uppgiften ska vara klart definierat och uppgiften ska kunna planeras med avseende på tid, resurser och kostnader. Organisationen som bedriver arbetet är temporär och upplöses då uppgiften är löst. Projekt varierar i storlek och tid, från små projekt på några veckor med ett fåtal inblandade till flerårsprojekt med hundratals personer inblandade. I många företag har projektarbetsformen blivit det naturliga. Detta är vanligt i företag som säljer olika typer av uppdrag eller systemlösningar. (Larsson & Larsson, 2005)

Då projektarbetsformen är en naturlig avgränsning av ett uppdrag är det enkelt att följa upp kostnader, tid, funktionalitet och kvalitet. Detta gör det enkelt att ha kontroll över ett enskilt uppdrag. Arbetsformen ger även stora möjligheter att lära för framtiden då uppföljningen av tid, kostnader, kompetenser, samarbeten med mera skapar en bas för utvärdering av projektet inför framtida projekt. (Larsson & Larsson, 2005)

3.1.1 Projektets ramar

Innan beslut om ett projekt tas beskrivs projektets förutsättningar grundligt i form av en bakgrund, omfattning, mål, milstolpar, budget, sluttidpunkt med mera. Denna beskrivning är projektets förutsättningar och definierar projektets ramar. Projektförutsättningarna ses som ett förarbete innan projektet startar men ser väldigt olika ut för olika företag. Syftet med det är trots det alltid att sätta upp de ramar som projektet är tänkt att hålla sig inom. (Larsson & Larsson, 2005)

3.1.2 Uppdragsgivaren

Projektledaren får sitt uppdrag av en uppdragsgivare. Denne har till uppgift att definiera projektets mål samt de ekonomiska ramarna. Uppdragsgivaren får löpande in rapporter från projektledaren om projektets framgångar och motgångar. För projektledaren är det viktigt att veta vem det är som styr eventuella förändringar i projektförutsättningarna. Det kan uppstå problem under projektets gång och förändringar bör ske i till exempel innehåll, tidsramar, budget eller kompetens. Beslut om detta tas av uppdragsgivaren då denne ska bevaka att projektet håller sig till projektförutsättningarna. (Larsson & Larsson, 2005)

Ofta är uppdragsgivaren även projektets beställare. I politiskt styrda organisationer står ledningsgruppen eller styrelsen vanligen som beställare men uppdragsgivaren är en s.k. *styrgrupp* som utsetts av styrelsen. Vid projekt gentemot utomstående kund är det kunden som är beställare och till exempel försäljningschef eller konsultchef som är uppdragsgivare. Alla projektledare måste ha klart för sig vem det är som fattar vilka beslut vid projektet. (Larsson & Larsson, 2005)

3.1.3 Projektdeltagare

Projektdeltagarna har tid avsatt för att utföra en viss uppgift inom projektramen. Ofta är de specialister och bidrar med sin kompetens på ett specifikt område i samverkan med de andra projektdeltagarna. Projektdeltagarna är involverade i projektet så länge deras kompetens behövs, vilket leder till att olika deltagare är engagerade under olika lång tid och vid olika tillfällen i projektet. (Larsson & Larsson, 2005)

3.1.4 Övriga intressenter i projekt

Andra aktörer i projektet är alla de som på något sätt har intressen i det. Dessa personer kan vara direkt involverade eller bara fungera som rådgivare i periferin. Det finns normalt flera projektaktörer som vill säga sitt vid ett projekt. Beroende på hur de agerar kan det påverka projektet i olika riktning vilket projektledaren måste ta hänsyn till. (Larsson & Larsson, 2005)

3.2 Projektledaren

Projektledaren har till främsta uppgift att leda och styra projektet från start till mål. Detta inkluderar att leda både de involverade människorna i projektet samt själva uppgiften. (Larsson & Larsson, 2005)

Det är vanligt att den person som kommit med projektförslaget även blir projektledare. Projektledaren ska arbeta för att projektgruppen utför uppgifterna så att projektets mål nås. Det är viktigt att projektledaren kan entusiasmera projektdeltagarna, styra, planera och driva projektet framåt. Viktiga egenskaper hos en projektledare är förmågan att hålla många bollar i luften, att vara flexibel, entusiasmerande och beslutsmässig. Vilka egenskaper som är viktigast varierar för varje projekt. (Löow, 1999)

Projekt är specifika och befinner sig utanför den vanliga linjehierarkin. Det kan även ligga vid sidan om den normala ansvars- och befogenhetsmatrisen. Det finns inte så många vare sig formella eller informella regler för hur en projektledare ska agera uppåt eller nedåt. Att projektet befinner sig utanför den "normala" hierarkin ger visst svängrum för projektledaren vad gäller att kräva resurser och medel för att styra projektet, vilket ofta sker genom egenbyggda kontaktvägar, men kan även bli en nackdel då en del kollegor kanske tycker att det stör ordningen eller det bestående. En projektledares arbete går på många sätt utöver "chefens" eller "specialistens" då denne har slutgiltigt ansvar för projektets framgång eller misslyckande. Det gör det möjligen lättare att stå på sig då det gäller krav vid tilldelning av resurser, men det innebär även hantering av många ofta motstridande intressen som rör projektet. Projektledaren måste hela tiden arbeta för att övertyga sin omgivning och skaffa sig inflytande. (Briner, Geddes & Hastings, 1999)

Det finns inga absoluta svar men vissa riktlinjer och principer kan skönjas för att underlätta och definiera projektledarens uppgifter. Det finns gemensamma kännetecken, som är utmärkande för projektledarens roll, som inte förekommer i andra chefsroller. (Briner, Geddes & Hastings, 1999)

En projektledare är (Briner, Geddes & Hastings, 1999):

- Ansvarig för att projektmålen nås. Dessa är begränsade men kräver en mycket tydlig och dynamisk aktivitet. Projektledarens roll i projektet kan liknas vid VD-rollen i ett företag.
- Den ytterst ansvarige och en person som inte kan gömma sig. Projektledarrollen är en roll som innebär stora risker.
- Begränsad när det gäller direkta befogenheter. Vanligen är det nödvändigt att förhandla om resurser och stöd från ett stort nät av kontakter både inom och utom företaget.
- Tvingad att arbeta över organisatoriska gränser, bryta gamla vanor och beteenden och ta till det okonventionella. Det kan vara väldigt krävande att ta hand om motstånd och kritik från övriga inblandade och intressenter.
- Ofta verksam inom nya områden för företaget. Det kan handla om ny teknik, nya marknader eller nya problemlösningsmetoder. Det finns ofta ett visst motstånd i organisationen mot det okända eller oförutsägbara. Projektledaren måste därför arbeta för att bygga upp trovärdigheten i projektet.

Enligt Briner, Geddes och Hastings (1999) finns det sex riktningar som projektledaren måste vända sig i; *uppåt, utåt, framåt, bakåt, nedåt* och *inåt*. Projektledarens roll kan delas in i tre dimensioner nämligen hanteringen av; *intressenterna, projektets livscykel* samt *prestationerna*.

De två första aspekterna, uppåt och utåt, behandlar hanteringen av projektets intressenter:

Uppåt- Projektledarens chef står som projektets mentor och skyddar projektet om det behövs. Det är viktigt för projektledaren att vara medveten om vilka personliga skäl mentorn haft för att be projektledaren åta sig projektet. Detta för att det är väsentligt att förstå bakgrunden till projektet för att undvika att trampa på någons tår senare och omöjliggöra ett lyckat slutresultat för projektet. Att hela tiden ha blicken uppåt och hålla en god kommunikation med mentor är därför en viktig del i projektledarens arbete.

Utåt- Alla projekt har en beställare i form av ledning, kund eller klient, en slutanvändare samt eventuellt ett antal övriga intressenter. De har alla förväntningar på projektet som projektledaren måste försöka infria på ett eller annat sätt.

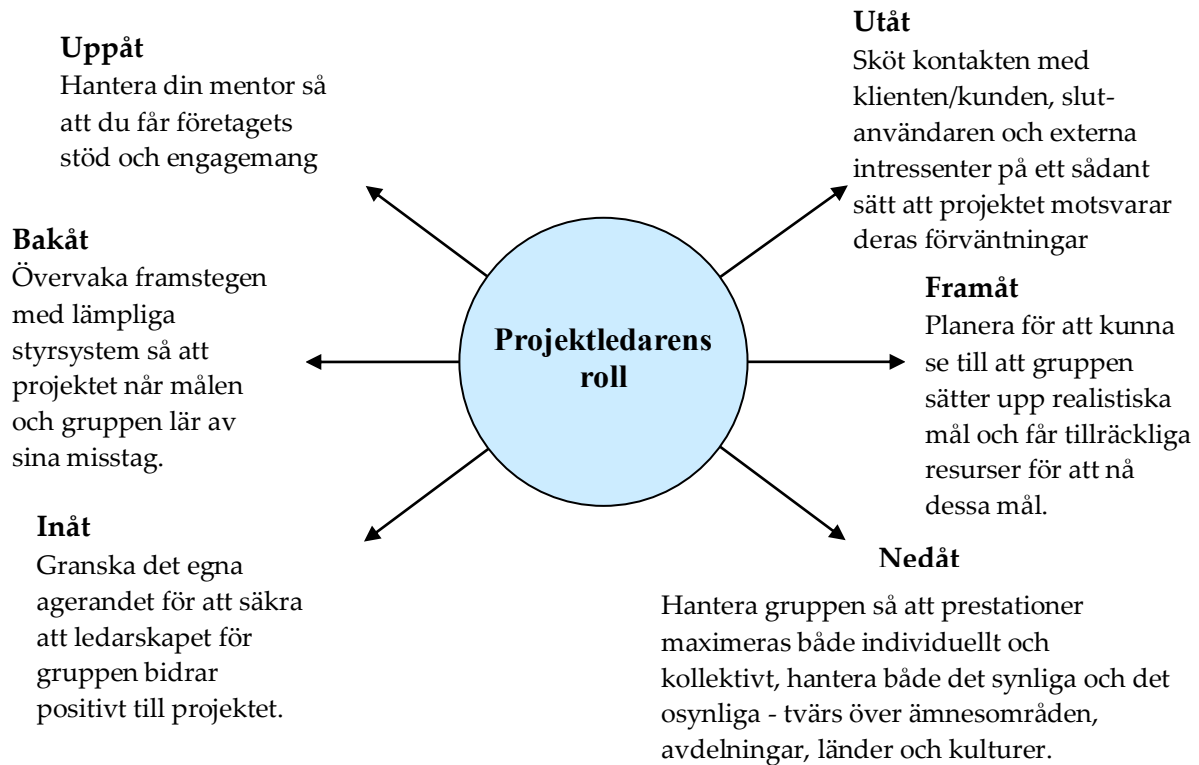
Aspekterna framåt och bakåt gäller hanteringen av projektets livscykel:

Framåt och bakåt- Riktningarna bildar en gemensam funktion. Projektledaren behöver ha ett lämpligt sätt för att följa upp tidigare arbete och lära sig av sina misstag, samt se till att målen blir uppnådda. För att kunna blicka bakåt måste man först se framåt, skapa realistiska planer, skaffa nödvändiga resurser, utnyttja ett ändamålsenligt system för övervakning och rapportering.

De två sista aspekterna av projektledarens roll handlar om hantering av prestationer:

Nedåt- Projektledaren är ansvarig för dels hela gruppens men även de individuella deltagarnas prestationer.

Inåt- Projektledaren har mycket på sin agenda och hamnar lätt i en situation där de dagliga uppgifterna tar mycket tid och den egna prestationen glöms bort. Trots att den har en stor inverkan på projektets helhet.



Figur 3-1 Projektledarens riktningssvisare (Briner, Geddes & Hastings, 1999, sid 33)

För att kunna uppfylla sin roll som projektledare på ett bra sätt har Lööw (1999) hittat ett antal personliga och formella egenskaper som verkar krävas:

- Entusiasmerare
- Förhandlare
- Diskussionsledare
- Samordnare
- Problemlösare
- Bollplank
- Ledare/coach
- Ombud för projektet
- Informatör

Några vanligt förekommande arbetsuppgifter för projektledaren (Lööw, 1999):

- Leda och fördela arbetet
- Sammankalla till möten
- Göra projektplan tillsammans med projektdeltagarna
- Ansvara för att projektet når målen

- Avrapportera till styrgrupp
- Följa upp, stämma av och agera även mellan projektmöten

Projektets resultat hänger inte helt och hållet på projektledaren. Genom de andra i gruppen kan projektledaren leda projektet till ett lyckat resultat. Det är ett krävande arbete och det behövs en ofta brinnande energi för att hålla projektet igång även vid motgångar. (Löow, 1999)

3.2.1 Framgångsfaktorer och fallgropar

Det finns ett antal faktorer som påverkar projektets utgång. Flera utav dessa är faktorer som projektledaren är ansvarig för eller har möjlighet att påverka och styra i rätt riktning. Beroende på grundförutsättningarna i projektet så ser framgångsfaktorerna olika ut men en del har de gemensamt (Löow, 1999):

- Klar struktur på arbetet
- Helst heltidsengagerade projektledare
- Tydliga direktiv
- Entusiastiska medarbetare
- Gemensamma mål för alla inblandade
- Tydliggjorda förväntningar, roller och tydlig arbetsfördelning
- Hänsynstagande till deltagarnas värderingar
- God planering/rätt sak på rätt sätt
- Målen nedbrutna i etappmål
- Revidering av målen i förekommande fall
- Kontinuerlig information och förankring
- Väl genomtänkta beslutsunderlag
- Uppföljning av resultatet

De vanligaste fallgroparna enligt erfarna projektledare (Löow, 1999):

- Undermålig planering
- För lite tid avsatt för att skapa vi-anda i projektgruppen
- Projektet är för otydligt
- Projektet kör på, utan avstämning i styrgruppen eller projektplanen
- Projektledare som inte kan skapa entusiasm eller motivera projektgruppen
- Projekten har inga eller för få avgränsningar
- Projektledaren har svårt att säga nej
- Sammansättningen i projektgruppen innebär alltför lika personlighetstyper
- För stora projekt

Det är bara projektledaren själv som kan avgöra hur mycket tid som ska läggas på varje del och samordningen av projektet. Kontakten med uppdragsgivaren skall hållas men frågan om eventuella korrigeringar bör göras måste projektledaren själv ställa. Ofta kommer reaktionen för sent vilket leder till akuta uttryckningar och för att lösa problem som kunde ha förebyggts genom tillämpning av en mer produktiv strategi. Tid för reflektion krävs så att prioriteringarna i slutändan blir de rätta. (Briner, Geddes & Hastings, 1999)

4 Systemutvecklingsmetoder

Kapitlet inleds med att beskriva vad en systemutvecklingsmetod är, för att sedan ta upp metodernas framväxt och användning ur ett historiskt perspektiv. Därefter följer en mer utvecklade definition av de agila metoderna. Kapitlet avslutas med att beskriva skillnaderna mellan agila och disciplinära metoder, där iterativa och vattenfallsmodell baserade metoder står för en vanlig generalisering för de disciplinära. I kapitlet likställs ordet metod med Metodologi, då vi liksom Fitzgerald, Russo och Stolterman (2002,) tycker att det är en bättre användning av språket.

4.1 Bakgrund

Människan skapar verktyg för att utöka sin mentala och fysiska styrka. Det naturliga är kanske att tänka på fysiska verktyg, som hammare och såg. Men i dagens informationssamhälle används verktyg anpassade för verksamheten. Utvecklare av informationssystem (IS) behöver en verktygslåda. En sådan består oftast av metoder, tekniker och verktyg för att ge stöd från början till slutet av utvecklingsprocessen (Beynon-Davies, 2002). Ett av målen (för dessa utvecklare) med att använda metoder är att hantera utvecklingen effektivt och kompetent (Avison & Fitzgerald, 2002). Metoder utvecklades framförallt för att kunna hantera mantrat "inom kostnad och tid" men det bör även tilläggas "med den efterfrågade kvaliteten och funktionaliteten".

Det finns idag en mängd olika metoder även om antalet varierar kraftigt mellan olika källor. Marginella skillnader och tolkningen av ordet metod skapar den stora variationen i beräkningen. Många företag hävdar att de använder en utvecklingsmetod, men forskning visar att de används sparsamt. Detta kan bero på flera orsaker, Avison och Fitzgerald (2002) har bland annat funnit att metoderna inte uppfyller utvecklarnas behov eller att flera metoder är teoretiska produkter och därför inte alltid fungerar som det är tänkt.

Enligt Avison och Fitzgerald (2002) är en metod ett sammanhängande och systematiskt tillvägagångssätt att angripa ett problem (baserat på vilken speciell filosofi av systemutveckling som används) som guidar utvecklarna i vilka steg som skall tas, hur dessa steg bör utformas och varför dessa steg är viktiga i utvecklingen av ett informationssystem. Systemutvecklingsmetoden är ett medel att säkerställa kvalitet och erbjuder en gemensam notation för de involverade. Metoden syftar till att styra, planera, hantera och utvärdera IS och projekt. Den grundläggande idén med metoder är att hjälpa den enskilde utvecklaren att komma på rätt spår, att ta bort lite av ansvaret från individen. Ju mer utförlig metoden är, desto mindre ansvar får utvecklaren och vice versa. Denna spänning mellan lite ansvar och mycket ansvar är viktig för att skapa motivation inom organisationen. (Avison & Fitzgerald, 2002)

4.2 Historisk systemutveckling

Under de första åren av systemutveckling arbetade de flesta utvecklare utan en direkt plan att följa, man utvecklade ad hoc. Under 1950 talet började IS att förflyttas in i affärsområdet och då krävdes mer och mer av utvecklarna (Fitzgerald, Russo & Stolterman, 2002). Tyngdpunkten låg dock fortfarande i programmeringen och på att lösa tekniska problem, underhåll samt att få systemen att fungera som det var tänkt, trots att nya system skulle ha kunnat svara bättre på vad användarna efterfrågade.

“It was a major achievement to get a program to run at all in the early 1960s”
Fitzgerald, Russo och Stolterman (2002, s 21).

Tillvägagångssättet var oftast upp till den individuella utvecklaren och baserades på personliga erfarenheter och kunskaper. Detta tillförde mycket problem vid estimering av tid och kostnad för utvecklingen. Hela 70-talet och början av 80-talet kännetecknades av en systemutveckling som var hårt styrd av faser och steg, vattenfallsmodellen (SDLC, The System Development Life Cycle). Vattenfallsmodellen, tillsammans med många andra modeller och metoder, kan spåras tillbaka till 30-talets ingenjörskonst. Den era som rådde under 70- och 80-talet fick bort många av de problem som tidigare funnits. Det blev enklare att beräkna tid och kostnad för utveckling. Dock existerade det fortfarande problem, nu till stor del beroende på den hårt styrda miljön i de vattenfallsbaserade metoderna. Utvecklingsmiljön blev oflexibel vilket kunde leda till missnöjda kunder. (Avison & Fitzgerald, 2002)

Som ett svar på problemen förökade sig metoderna snabbt under 80-talets senare hälft. De kom fram antingen genom praktisk utveckling eller genom teoretisk forskning. De metoder som tagits fram genom praktisk utveckling kallas kommersiella och är de mest spridda och använda. Många av dessa metoder passerar med dagens kriterier inte längre kraven för att kallas metod utan innehåller ofta bara det vi brukar benämna verktyg. (Avison & Fitzgerald, 2002)

Det var oftast så att någon utvecklare fann sitt sätt att utveckla som det bästa och därför ville få ut det till andra utvecklare, således uppkom en ny metod. Men det dröjde inte länge förrän det kom fram sammansatta metoder ofta utvecklade in-house på någon konsultfirma som tröttnade på att alla konsulter arbetade med olika verktyg och därför inte nådde samma resultat. Organisationer investerade pengar och tid i sina metoder för att bland annat få dem nedskrivna, sammanhängande, marknadsmässiga och underhållna. (Avison & Fitzgerald, 2002)

Att utveckla system utan metod gick bra så länge system, organisationer och projekt var småskaliga och inte alltför komplexa. Men när detta förändrades krävdes metoder för att åstadkomma stabilitet. En av dessa metoder är Rational Unified Process (RUP) (Avison & Fitzgerald, 2002). Målet med denna var just att göra mjukvaruutveckling mer förutsägbar och effektiv och detta åstadkoms genom en stark fokusering på planering. Men den har kritiserats för att vara för byråkratisk, att förlänga utvecklingstiden och refereras ofta som tung (Fowler, 2000) eller disciplinerad (Cockburn, 2004). Som en reaktion till att de traditionella metoderna bara utvecklades till att få en mer och mer tungrodd process har det

uppkommit flera så kallade lättviktsmetoder (agila) under de senaste åren (Schuh, 2005) Dessa metoder försöker skapa en kompromiss mellan alltför mycket mjukvaruprocesser och inga alls.

4.3 Agila metoder

Även om vissa av de Agila metoderna har funnits uppemot 20-30 år så är termen agil metod aningen nyare. Den 17 februari 2001 möttes sjutton personer som var antingen utvecklare eller starka förespråkare för det som då kallades de lätta metoderna. De möttes för att se om det fanns något gemensamt inom de olika metoderna (Koch, 2005). Koch (2005) tar upp fyra grundläggande punkter som dessa var överrens om:

1. Det finns ett behov för metoder som kan svara mot förändringarna i omvärlden, därför namngavs samlingen metoder till agila istället för lätta. Men namnet agilt var inte enbart till för att få ett samlingsnamn utan var även ett marknadsföringsknep, då de 17 förespråkarna såg en svårighet att få väldigt tekniska personer att vilja använda en metod som gick under ett samlingsnamn som "light" (lättviktsmetod).
2. Ett manifest togs fram (se Bilaga D: Agila Manifestet & Principer) som fångar nyckelvärderingarna som alla agila metoder bygger på, och det lyder:

We are uncovering better ways of developing software by doing it and helping others to do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan.

That is, while there is value in the items on the right, we value the items on the left more (for example individuals and interactions). (Avison & Fitzgerald, 2006)

Det finns en hel del kritik mot det agila manifestet, bland annat att de agila metoderna genom manifestet ger alltför mycket makt åt de delar som står till vänster vilket bara underbygger hackermentaliteten än mer. Cockburn (2002) anser att det agila manifestet enbart är en nedskrivning av redan använda principer. Boehm (2002, se Avison & Fitzgerald, 2006, s 145) anser att man istället kanske skall fokusera på att få balansen någonstans mittemellan den högra och den vänstra spalten i manifestet, inom både agil och disciplinär utveckling. På IBM, som till stor del utvecklat RUP, anser man inte heller att det behöver vara en tävling mellan blocken av traditionella och agila utvecklingsmetoder utan ser möjligheter för hur metoder kan vara komplement till varandra (IBM, 2005).

3. Den tredje överenskommelsen var tolv agila principer som går igenom och delar upp manifestet i mindre beståndsdelar (se Bilaga D: Agila Manifestet & Principer).

4. Den sista nivån var en ännu mer detaljerad aktivitetsöverenskommelse som gav spelrum för varje individuell metod att få utveckla sina egna karaktärsdrag, så länge som den förhöll sig väl till tidigare överenskommelser.

Det är en vanlig feltolkning att säga att agila metoder är emot processer, och ett flertal utvecklare anammar felaktigt agila metoder för att slippa följa en förbestämd process. Koch (2005) menar att varje enskild agil metod har sin egen process, och om fördelar skall kunna dras av metoden är det väldigt viktigt att processen följs.

Vidare anser Koch (2005) att de agila metoderna har mycket gemensamt men framförallt rörligheten (agility), förändring, planering, kommunikation och lärande.

Rörlighet (Agility) är det område som mest skiljer sig från den traditionella systemutvecklingen, emedan den traditionella utvecklingen gör allt för att följa en plan, genom att kontrollera oförutsedda händelser och påverkan från omgivningen, så förändrar sig istället den agila utvecklingen mot omvärldens förändring och påverkan.

Förändring är således välkommen inom de agila utvecklingsmetoderna, för att ge mer information så att en bättre produkt skall kunna skapas. De traditionella metoderna ser ofta förändring som en fiende som måste stävjas, de accepterar således att förändring är något nödvändigt ont, men att det inte är något som gagnar varken utveckling eller organisation. Inget av ovanstående arbetssätt fungerar alltid och ska inte ses som ett optimalt arbetssätt. Beroende på olika situationer är de olika lämpade vid skilda tillfällen.

Planering är lika viktigt i agila som i traditionella metoder, men de behandlar planering olika, särskilt när det kommer till avvikelser från planen. När det sker förändringar från planen i den agila utvecklingen, läggs den nya informationen in i projektet. Meningen är att få planeringen att anpassas till verkligheten. Traditionell utveckling görs genom att försöka få verkligheten att anpassas till planen, i extrema fall tas det fram en ny plan. Inte heller under denna punkt är något av sätten att föredra framför den andre.

Kommunikation är inom de agila metoderna mest muntlig, den enda gång som skrivna dokument egentligen skall användas är när det kan anses tillföra information till den talade. Även om de agila metoderna inte har som avsikt att eliminera den skrivna dokumentationen så vill de absolut inte ha dokumentation som enbart har som syfte att arkiveras. Traditionella metoder argumenterar inte emot kommunikation, men de anser att kommunikation uppstår genom de planerade aktiviteterna och genom dokument, men detta skapar ofta missförstånd senare i projekten. Vilket sätt som är mest lämpat är inte alltid enkelt att bestämma. Den talade kommunikationen är inte alltid bra, ibland glöms viktiga delar bort, och två personer kan ha olika minnesbilder och uppfattningar om en sådan kommunikation. Koch (2005) anser att om något är värt att prata om så är det även värt att dokumentera.

Lärande är en viktig del inom agila metoder, då varje projekt ses som en erfarenhetsutökning. Metoderna skapar en kopiös mängd kommunikation mellan intressenterna för att accelerera lärandet. Denna nya kunskap resulterar i förändringar i kravbilderna för det nya systemet, en iterativ process. De traditionella metoderna ser detta på ett lite annorlunda sätt. Under planeringsfasen anser de att all information för att slutföra

projektet finns tillgängligt, att det under kravinsamlingen går att finna alla krav osv. när detta inte stämmer får utvecklarna rätta till felen. Verkligheten är att alla inblandade i alla projekt lär sig något under projektets livscykel, men det är inte för det säkert att det är viktig kunskap. Varje ny del av information ska utvärderas och integreras enbart om den ger värde.

Ett flertal personer anser att de agila metoderna enbart är till för att ge en "license to hack" (Wiegers, 2001 se Avison & Fitzgerald, 2006, s 145). Andra påpekar svårigheten att applicera metoden på stora projekt, då framförallt för att möjligheterna till en god kommunikationsmiljö förändras.

4.4 Disciplinära vs Agila metoder

Av de iterativa och vattenfallsbaserade metoderna kan de flesta anses vara disciplinära. Tabell 4-1 sammanfattar vad Boehm och Turner (2003) kallar för "Home Ground" och den tar upp de grundläggande skillnaderna mellan agila och disciplinära metoder.

Kännetecken	Agila	Disciplinära
Användning		
Primära mål	Ger ett snabbt värde, att svara på förändringar	Förutsägbar, stabil, hög säkerhet
Storlek	Mindre team och projekt	Större team och projekt
Omvärld	Turbulent, stor förändringsgrad, projekt fokuserad.	Stabil, liten förändringsgrad, projekt/organisations fokuserad.
Ledning		
Kund relationer	Engagerade (involverade) kunder, fokus på prioriterade tillägg.	Kontakt när det behövs, fokus på överenskommelser i kontrakt.
Planering och kontroll	Intern plan, kvalitativ kontroll.	Dokumenterade planer, kvantitativ kontroll
Kommunikation	Tyst kunskap som skapar kommunikation.	Väl dokumenterad kunskap
Teknisk		
Krav	Prioriterade informella historier och test journaler, genomgår oförutsägbar förändring.	Formaliserade projekt, möjligheter, samverkan, kvalitet, förutsägbara utvecklingskrav.
Utveckling	Enkel design, korta etapper, förändringar anses vara billiga	Omfattande design, längre etapper, att förändra anses dyrt.
Test	Körbara test definierar krav	Dokumenterade testplaneringar och procedurer.
Personal		
Kunder	Engagerade, CRACK* utövare	CRACK* utövare
Utvecklare	Stor kunskap krävs, där utvecklaren vågar bryta mot regler i metoden och anpassa till en ny situation.	I ett tidigt skede av utvecklingen krävs det ganska mycket av utvecklaren, ju längre det går desto mindre kunskap krävs.
Kultur	Trivsel och självbestämmande genom flera olika grader av frihet (mår bra i kaos)	Trivsel och självbestämmande via ramverk av politik och procedurer (mår bra under ordning)
* Collaborative, Representative, Authorized, Committed, Knowledgeable		

Tabell 4-1 Agila och disciplinära metodernas "home ground" (Boehm & Turner, 2003)

Efter att ha skapat en gemensam grund för säregna kännetecken mellan agila och disciplinära metoder så tar Boehm och Turner (2003) fram de balanserade kriterierna. Tabell 4-2 tar upp fem faktorer som avgör om det bäst lämpar sig att använda agila eller disciplinerade metoder i ett specifikt projekt. Dessa faktorer är storlek, kritisk faktor, dynamism, personal och kultur. Ett projekt som passar bra i fyra av dessa fem faktorer är ett projekt som kan kräva en riskutvärdering och kanske det bäst lämpar sig att använda en mix av både agila och disciplinära metoder för att få fram en bra produkt.

Faktorer	Agila	Disciplinära
Storlek	Välanpassad till små produkter och team. Beroende av tyst kunskap och har begränsad skalbarhet.	Metoder utvecklade för att hantera stora produkter och team. Svår att använda på mindre projekt.
Kritisk faktor	Otestad på säkerhetskritiska produkter. Potentiella svårigheter med enkel design och avsaknad av dokumentation.	Svår att använda på projekt där man inte behöver vara så kritisk. (läs snabbare/enklare projekt.)
Dynamik	Enkel design och ständiga förändringar är utmärkt för en dynamisk miljö men kan leda till kostsamma omarbetningar i en stabil miljö.	Detaljerade planer och stor Design som kan visas snabbt fungerar utmärkt i en väldigt stabil miljö, men kan bli en källa för omarbete i en dynamisk miljö.
Personal	Kräver ständig närvaro av personer som klarar av att anpassa metod och lösning till nya problem och situationer. Riskfyllt att använda icke agila personer som inte klarar av att följa standarder etc.	Kräver personer med stor kunskap på metod under definitionsfasen men kan minimera dessa ju längre in i projektet man kommer, om inte omgivningen är dynamisk. Kan för det mesta använda sig av mindre kunniga utvecklare då det generellt krävs mindre av varje person.
Kultur	Frodas i en kultur där personalen känner sig bekväm och självbestämmande genom att de har mycket frihet.	Frodas i en kultur där personalen känner sig bekväm och självbestämmande genom att de har sina roller definierade där politiken är tydlig och genom klara procedurer.

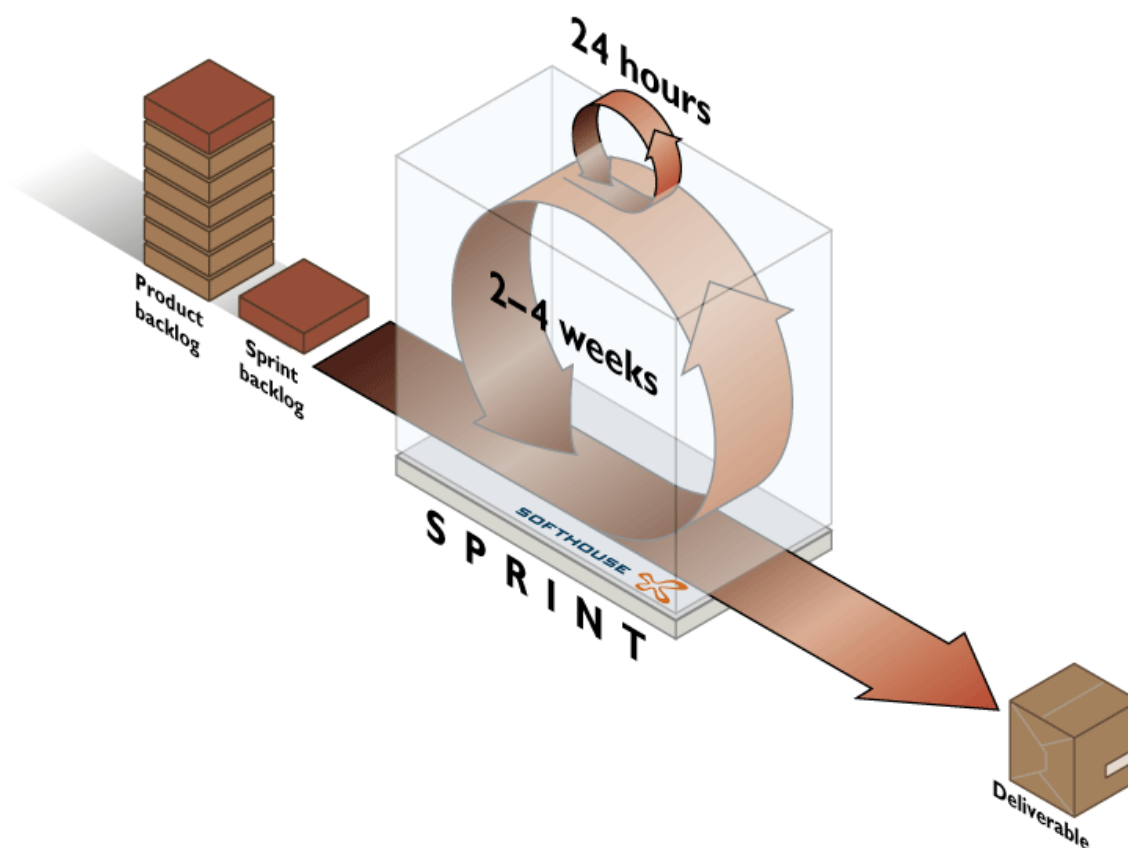
Tabell 4-2 Fem skillnader mellan agila och disciplinära metoder (Boehm & Turner, 2003).

5 Scrum - metoden

Kapitlet går igenom en av de agila metoderna på ett djupare plan och tar även upp grundläggande fakta kring Scrum Mastern

Scrum tillhör de agila utvecklingsmetoderna (Avison & Fitzgerald, 2006). Denna metod är inte enbart framtagen för mjukvaruutveckling utan är en mer generell metod för att hantera produktutveckling oavsett teknik. Det sätt metoden används på idag växte fram i Japan under mitten av 80-talet som en vidareutveckling av metoden Sashimi¹, Scrum är en term inom sporten rugby och betyder ungefär att föra tillbaka bollen i spel. Likt de flesta andra agila metoderna bygger inte Scrum så mycket på processen utan mer på filosofin kring arbetssättet (Koch, 2004).

Scrum är baserat på det som kallas sprintar, en sprint är oftast en 30-dagars iteration där den samlade ansträngningen fokuseras mot ett bestämt mål. Varje sprint skall öka produktens värde och ge nya funktioner och förbättringar som kan levereras till kunden. (Softhouse, 2006)



Figur 5-1 Scrum sprint (Softhouse, 2006)

¹ Sashimi kommer från en modifiering av vattenfallsmodellen där japaner försökte få bort det som de ansåg vara dåligt med modellen och anpassade den för att bättre passa in i deras utveckling. Till att börja med reducerades antalet steg till fyra - krav, design, prototyp och acceptans - utan att ta bort några aktiviteter, vilket resulterade i en överlappning av vattenfallsmodellens faser (Controlchaos, 2007).

5.1 Aktörerna

Det finns tre olika roller inom Scrum; team medlem, Scrum Master och produktägare. De resterande är personer som har något intresse i projektet men som inte åtagit sig att göra något. Men det är ändå viktigt att de i ett senare skede av projektet involveras och ger sin syn på den utvecklade produkten (Schwaber, 2004). På de dagliga mötena är alla välkomna att lyssna men det är endast de som har blivit tilldelade en roll som får lov att tala.

5.1.1 Scrum teamet

Teamet är de som utför det egentliga arbetet som problemlösare och konstruktörer. Ett team består oftast av 5-9 personer. Detta anses, enligt erfarna forskare, vara en optimal storlek för denna typ av arbete (Koch, 2005). Hur arbetet sen går tillväga och vem som gör vad bestäms helt och hållet av teamets medlemmar. Under sprinten har alla medlemmar sin roll men i övrigt saknas fasta roller i projektet. Tanken är att alla ska kunna byta uppgifter med varandra, eller vem som helst i ett team ska kunna ta sig an en uppgift. Detta hindrar dock inte att enstaka medlemmar är specialister på ett område och det därför inte går att förflytta dem. Ett team ska innehålla all kunskap som krävs för att klara av uppgiften som ligger till hands under sprinten. Så om det under perioden behövs specialkunskaper som inte innehas av den från början sammansatta projektgruppen så ska denna person vara med under hela sprinten. Detta betyder att de inte bara kommer utanför gruppen som expertkonsulter, utan att de antingen utför arbetet eller att de agerar mentorer för att lära upp någon annan. (Schwaber, 2004)

5.1.2 Produktägaren

Produktägaren (product owner) är beställare och har till uppgift att se till att Scrum teamet arbetar med rätt saker ur ett affärsmässigt perspektiv och se till att beställarorganisationen får något tillbaka på sin investering. Det är alltså personen som representerar beställarens intressen, men måste inte vara kund utan kan tillhöra den egna organisationen. Produktägaren administrerar en "product backlog" där alla krav på produkten radas upp utefter hur lönsamma de bedöms vara. Detta leder till ett dokument som är tillgängligt för hela organisationen så alla vet vad som kommer i den nya releasen. Uppgiften kräver breda kunskaper om teknik, marknad och affärsprocesser. (Schwaber, 2004)

5.1.3 Scrum Master

Scrum Master, kan i andra metoder/projekt närmast jämföras med projektledare, är den person som ansvarar för att utvecklingsprocessen går åt rätt håll. Scrum Mastern har tre uppgifter. För det första ska denne ta ansvar för att lära projektmedlemmarna hur de ska använda metoden tills de klarar det själva. För det andra ska Scrum Mastern hålla modet uppe och agera ledare och guide. För det tredje ska denne se till att reglerna följs. (Schwaber, 2004)

Det är Scrum Masterns huvuduppgift att få teamet att följa processen oavsett vad som sker. Varje dag träffar Scrum Mastern teamen i korta möten, dagliga Scrums. I de fall någon utanför projektet har något viktigt ärende att ta upp med teamet, försöker Scrum Mastern se till så att utvecklarna störs så lite som möjligt i sitt arbete. Fokus ska alltid ligga på att ge

teamen bästa möjliga förutsättningar att nå de uppsatta målen för sprinten. Efter varje sprint håller Scrum Mastern ett utvärderingsmöte med Scrum teamet, sprintåterblick, där de erfarenheter som har gjorts och de lärdomar som har dragits går igenom. Syftet är att höja teamets kunskapsnivå och öka motivationen inför nästa sprint. (Schwaber, 2004)

5.2 Arbetsgång

Första dagen skapas en product backlog, detta är en prioriterad (produktägaren gör prioriteringen över vad som är viktigast för hans organisation att få levererat först – för det mesta det som ger mest värde) lista över vad projektet ska förmå att få fram i produktväg, denna lista innehåller även uppskattade tider för när de olika delarna ska vara klara. Product backlogen ska utvecklas och förändras om affärsvillkor eller tekniker förändras och det är produktägaren som ansvarar för product backlogen. Alla idéer, både stora och små, ska finnas med och det läggs till nya idéer under projektets fortgång (Schwaber, 2004).

Vid början av varje ny sprint ska product backlogen gås igenom för att se till att den är uppdaterad. En sprint varar oftast 20 arbetsdagar eller 30 kalenderdagar, men kan variera. Den första dagen på en ny iteration avsätts till att skapa en sprint backlog. När gruppen är överens om uppgifter och tidsåtgång släpper produktägaren taget. Från och med nu jobbar Scrum teamet under eget ansvar. Om gruppen har rätt sammansättning kommer arbetet att vara självorganiserande. Teamet fortsätter sitt arbete och betar av uppgifterna som finns i sprint backlogen. Denna backlog kan ses som en del av product backlogen fast på djup detaljnivå. Listan ska innehålla ansvarsfördelning och vara uppdelad i arbetsuppgifter. Meningen är att det ska leda till att en uppgift i product backlogen avklaras under varje sprint. Varje dag som går under sprinten ska det hållas ett Scrum möte, under detta möte som maximalt ska vara 15 minuter rapporteras framsteg och hinder till Scrum Mastern. Var och en av deltagarna ska på något sätt svara på tre frågor: (Schwaber, 2004)

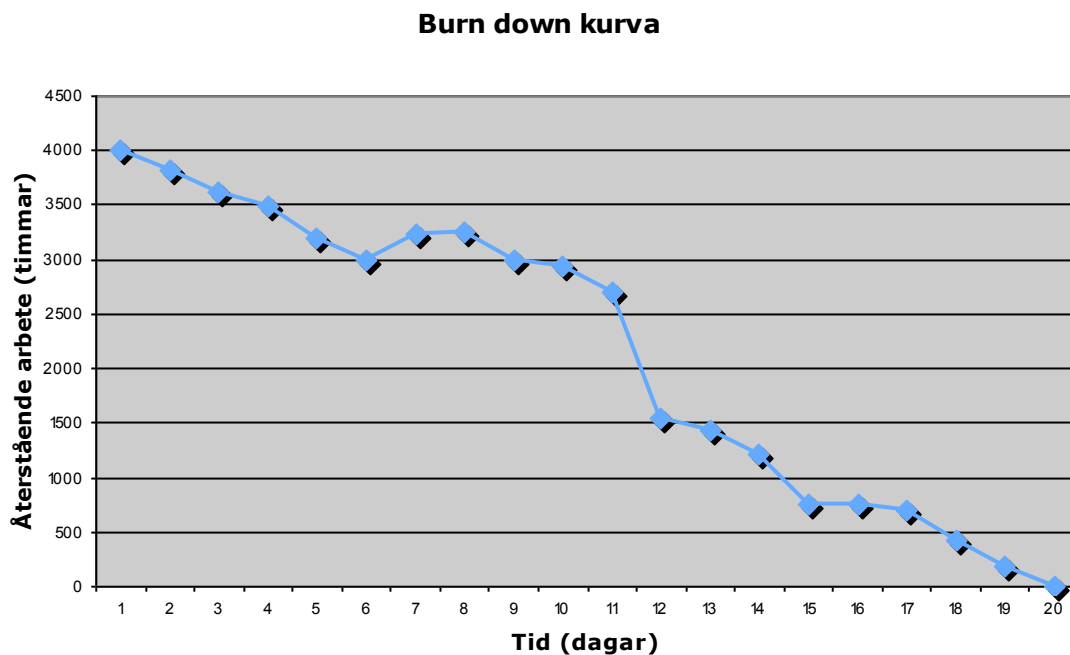
Vad har du gjort sedan förra mötet?

Vad tänker du göra inför kommande möte?

Är det något som hindrar dig från att uträtta ditt planerade arbete?

De två första frågorna ger mötesdeltagarna full insyn i hur projektet fortskrider. Den tredje frågan ger underlag för problemlösning – det kan vara allt från en ny datormus till organisationsförändringar på företaget. Vem som helst, får vara med och lyssna på mötet, men det är bara Scrum Mastern och team medlemmarna som får tala (Softhouse, 2006).

Sprint burn down (se Figur 5-2) är en tabell som visar hur det går för team medlemmarna. Tabellen är en estimering av återstående tid för att nå mål och ger en föräning om det går att förverkliga visionen med sprinten (Schwaber, 2004).



Figur 5-2 Burn down chart

5.3 Demonstration och utvärdering

Efter varje sprint körs en demonstration där en fungerande programvara exekveras inför en större grupp som förutom produktägaren omfattar t.ex. användare och representanter för företagsledningen (Softhouse, 2006). Detta ligger till grund för ett möte där gruppen tillsammans reflekterar över vad som kunde ha gjorts bättre inför nästa sprint, för att gruppen ska bli mer produktiv och få trevligare på arbetet (Schwaber, 2004).

6 Företagen och respondenterna

Nedan följer en beskrivning av de fyra företag som är med i undersökningen samt en kort presentation av respondenterna och informanten.

Appium AB är sedan 1992 aktiva inom telekombranschen och är idag en ledande leverantör av applikationsplattformar. Företagets huvudprodukt Appium Xway används idag av flera ledande operatörer världen över såsom BT, FT/Orange, TIM, Hi3G, Connex. Huvudkontoret återfinns i Malmö, med försäljningskontor på olika platser i Europa, USA och Asien.

Ulf Atles är ekonom i grunden och är idag ansvarig för "Products" på Appium, med en grupp på 20 personer. Atles har lång erfarenhet av processer och produktutveckling och är certifierad Scrum Master.

Massive Entertainment, lokaliserat i Malmö, grundades 1997 och producerar spel och interaktiva mjukvaruprodukter till en internationell marknad. Mest känt är PC-spelet Ground Control som lanserades år 2000. Företaget har även utvecklat flera integrerade spel för mobiltelefoner. År 2002 köpte Vivendi Universal Games (VUG) upp Massive Entertainment för att göra företaget till en stark utvecklare utanför Nordamerika, tillsammans arbetar de nu för att skapa topptitlar till nöjesindustrin.

Henrik Sebring har läst Människor Datateknik Arbetsliv (160p) i Ronneby. Han har arbetat på Massive Entertainment sedan 1997 och har varit med och utvecklat tre spel med ett fjärde på väg. Tidigare har han arbetat som "Lead Game Designer", ansvarig för ett litet team av game designers. Sedan 2006 arbetar han som projektledare och "Lead Singleplayer Designer" för ett team på tio personer som arbetar med konceptutveckling.

Softhouse AB är en svensk mjukvarutillverkare och levererar lösningar för nästa generations nätverkskommunikation. Man erbjuder även konsultkompetens inom området. Företaget är fokuserat på Mobile Multimedia och Software Business Management. Softhouse är medsamordnare av "Øresund Agile", en årlig sammankomst av agila utövare från Skandinavien och norra Europa.

Michael Rozenberg arbetar på Softhouse sedan hösten 2006, har tidigare läst till civilingenjör i InfoCom och haft eget företag. Han är nu certifierad Scrum Master.

Citerus AB grundades i Uppsala 1996, finns idag även i Stockholm, och är Sveriges ledande konsultbolag inom lätttrörlig mjukvaruutveckling. De håller i utveckling åt kund (bland annat Tre, Lantmäteriet osv.) kan anordna mentorskap, samt utbildningar. Citerus blev 2004 världens första företagsmedlem i Agile Alliance som verkar för att sprida kunskap om lätttrörliga utvecklingsmetoder.

Mikael Lundgren har teknisk utbildning från Uppsala och arbetar idag som konsult på Citerus. Han är en av endast två personer i Sverige som innehar rätten att certifiera Scrum Masters.

7 Empiriskt resultat

Följande avsnitt är ett sammandrag av de fyra intervjerna, de finns i sin helhet i BILAGA C: Transkriberingar. Mikael Lundgren på Citerus har svarat på frågorna i rollen som informant, d v s han har beskrivit metoden Scrum som sådan utan större koppling till det egna företagets utvecklingsverksamhet. Övriga är respondenter och har beskrivit situationen på det egna företaget. Orden "task" och "feature" förekommer i denna del då respondenterna använde sig av dessa uttryck. Task kan närmast liknas vid uppgift och feature en funktion eller funktionalitet i en del av en produkt.

7.1 Scrum i användning

Appium [51] är ett produktbolag och har därför interna beställare och produktägare. Man har olika kundprojekt men bara ett produktprojekt. Det är samma övergripande produkt som förädlas hela tiden men kraven från de olika kunderna kommer in efterhand. För att passa verksamheten och företagets verklighet har man valt att förändra Scrum och använder bara metoden som bas [52]. Appium hade svårt att följa Scrums krav på demoklara leverabler² i slutet av varje sprint och släppte därför det kravet [52], nu har man ofta 7-8 iterationer innan release [53]. Kodstoppet för att göra en demo klar resulterade i frustration hos utvecklarna och de kände att de hamnade i en vattenfallsliknande process där för mycket tid ägnades åt onödiga saker. För att vara fortsatt flexibla mot produktägare och göra det möjligt att få snabb respons från produktägare och beställare behöll man arbetet i en två veckor lång sprint [53].

Även Massive Entertainment har förändrat Scrum efter verksamheten. Istället för Scrum team bestående av olika discipliner har de valt att indela grupperna efter bransch, till exempel programmerarna för sig [32]. Man använder sig inte heller av titeln Scrum Master utan den rollen sköts av företagets Leads [32]. Iterationen hålls till tre veckor [33], men ibland itererar man om vissa "features" för att de inte håller företagets kvalitetsnivå, inte hunnit bli klara eller inte är tillräckligt roliga [36].

På Softhouse har man filosofin att det är meningen att utvecklingsmetoden ska förändras [3]. Därför blir det också olika i varje projekt. Till exempel har det inte alltid varit möjligt att ha produktägaren närvarande på plats och då har man fått lösa kommunikationen under Scrummöten på annat vis. Utöver Scrummötet väljer man vid tillfälle att lägga till ett tekniskt möte då gruppen ändå är samlad [3].

En iteration på Appium startar med en titt på vad som redan är känt samt marknadskraven [54]. Detta ska ge en initial kravlista. Ibland, då tekniken är okänd, går inte en kravlista att upprätta från början utan man genomför istället en förstudie för att veta vad man har att arbeta med [54]. Cykeltiden kan därmed variera i början och först när projektet är igång använder man utsatt sprinttid. På Massive Entertainment gör man en grovskiss, med enklare tid och resursuppskattning, med större "features" som ska vara med i projektet. Uppskattningen tar hänsyn till vilken grupp det är som ska utföra arbetet [33]. Varje

² **Leverabel** – "(deliverable) - produkt som är färdig för leverans (och som fungerar som planerat)" (IDG, 2007)

”feature” bryts sedan ner i mindre ”tasks” på 4-16 mantimmar och för den kommande sprinten får de en noggrannare detaljplanering [33].

De dagliga Scrummötena används av Softhouse [3] och Massive Entertainment [34]. På Massive Entertainment får alla i teamet redogöra för vad de gjorde igår, vad de ska göra idag och om det finns något som hindrar dem från att utföra sina uppgifter [34]. På Appium har man uppföljningsmöte varannan vecka och kontrollerar då ”var projektet befinner sig nu” och ”hur många iterationer är det kvar” [54]. Appium använder sig inte heller alltid av burn down charts [54] medan man på Massive Entertainment varje dag uppdaterar teamet om hur många timmar det är kvar på uppgifterna och om ”tasks” är redo att skickas till testning [35].

Vid sidan av iterationsplaneringen har Massive Entertainment även en långsiktig planering med ett antal fasta datum. Anledningarna kan vara att vissa saker måste bli klara i tid för att man ska hinna med, som till exempel röstinspelningar som måste bokas i god tid i förväg [37]. Även på Softhouse upplever man att man måste ha en långsiktig planering utanför den kortsiktiga. Rozenberg anser att Scrum kan bli för fokuserat på bara en sprint framåt och vill därför hålla en något längre tidshorisont. Det handlar då ofta om testmiljöer som måste bokas i förväg eller synkning med andra projekt [8]. Atles anser, på grund av att man arbetar med okänd teknik, att en fyra veckor lång sprint skulle bli för lång för Appium då man redan idag har problem att lista de krav som ska vara med de kommande två veckorna [53].

Både Softhouse och Citerus har tagit in tankar kring Lean Development³ i sin utveckling. På Softhouse handlar utvecklingsmetoden om just Lean och vad det bär med sig i form av frekventa möten och att man sitter tillsammans vilket underlättar kommunikationen. Sen kan man kalla metoden Scrum eller vad man vill, det är tankesättet bakom som är det viktigaste.

På Appium har man valt att beroende på var projektet befinner sig någonstans tidsmässigt använda inslag från andra metoder. I början av ett projekt då allt är ganska osäkert liknar utvecklingen XP⁴ (eXtreme Programming) men efterhand som kravbilderna växer fram formaliserar man metoden och i slutet innan release kan man köra väldigt strikta iterationer i en RUP-liknande process och tar då även vid tillfälle in Gantt-schema i tillägg till de verktyg som ges i Scrum (för verktyg se kap 5 Scrum – Metoden).

Även Citerus kan tänka sig att ta in Gantt-schema eller använda sig av Use Cases om man vill visualisera saker på ett annat sätt eller finner det lämpligt.

³ **Lean Development** - är ett koncept taget från Toyota. 2003 släpptes boken *Lean Software Development: An Agile Toolkit for Software Development Managers* av Mary och Tom Poppendieck, där de förklarar hur Lean Development går att applicera på mjukvaruutveckling. De tog även fram sju principer för detta: Eliminate Waste, Learn Constantly, Delay Commitment, Deliver Fast, Build Integrity In, Empower the Team, See the Whole (Citerus, 2006)

⁴ **XP** (eXtreme Programming) – är en agil systemutvecklingsmetod skapad av Kent Beck. Det är en kodkoncentrerad process och baseras på fyra grundvärderingar; kommunikation, enkelhet, återkoppling och mod. (Jeffries, 2001)

7.2 Fördelar och nackdelar med Scrum

På Appium ser man att den största fördelen med Scrum är den tydliga rollfördelningen och att alla arbetar parallellt. Det parallella arbetssättet ger att även all dokumentation är klar när det står att något är klart [56]. Rozenberg på Softhouse anser att den tidiga feedbacken är en stor fördel [11] man har hela tiden koll på situationen och vet var man befinner sig i projektet, en statusrapport tar inte mer än fem minuter att skriva [7]. Sebring på Massive Entertainment anser att metoden ger honom möjlighet att fokusera på kvaliteten i projektet [38].

Scrum ligger på team och organisationsnivå [11][59] och har därför inga "engineering practises" enligt Atles [58]. Detta kan enligt Rozenberg försvåra arbetet för programmerarna då det inte finns några direkta riktlinjer och koden dessutom ska hållas så generell som möjligt [11]. En fördel blir därför, enligt hans åsikt, att teamet är blandat och det ingår seniorer med bättre kännedom om arkitekturen [11]. Nedbrytningen i mindre bitar blir även den positiv då allt blir tydligt och strukturerat. Det blir enklare att diskutera det som står på tavlan med alla.

Scrum säger inte heller något om HUR ett problem ska lösas, utan lyfter bara upp problemen till ytan [10]. Detta är en lucka som enligt Lundgren är medveten, speciellt vid införandet av Scrum [75]. Plötsligt börjar man ifrågasätta saker i den befintliga verksamheten, flaskhalsar och problem syns på ett annat sätt och det är jobbigt när det inte finns en klar lösning på dem [75]. Då det inte finns några tumregler för hur problem ska tas omhand i Scrum finns det inte heller något om konflikthantering, det är bara Scrum Masterns uppgift att lösa dem [9].

Sebring upplever det som relativt svårt att visa projektets status gentemot externa kunder, speciellt i de lägen då de inte tar sig tid att själva kolla igenom den senaste builden. Den yttre världen är enligt Lundgren full av personer med sekventiellt tänkande som strider mot Scrums grundtankar. I ett agilt projekt kan du inte fråga om en exakt release på samma sätt som i ett traditionellt projekt. En fördel med Scrum i den situationen menar Lundgren är att Scrum är lätt att förstå, oavsett position i företaget, och kan användas på en hel verksamhet.

7.3 Scrum Master

Scrum ställer, enligt Lundgren, högre och annorlunda krav på ledaren jämfört med andra metoder. Det finns inte så mycket föreskrivet utan som Scrum Master måste man hitta sina egna vägar. Ledaren inom Scrum kallas för Scrum Master för att betona att det inte är fråga om en vanlig projektledare, de två är väsensskilda från varandra [76]. Som Scrum Master är man en coach och inte en projektledare i traditionell mening. Scrum Mastern arbetar som möjliggörare där denne ser till att teamet har förutsättningarna för att lösa problemet, planera och följa upp sitt eget arbete. Scrum Mastern är inte en medlem i teamet på samma vis som en projektledare deltar i projektgruppen, därför har Scrum Mastern inte mer makt än resterande medlemmar i teamet [5].

Scrum Mastern arbetar på två nivåer, dels övergripande på business och sprintnivå där kraven från olika intressenter spelar roll, dels på lågnivå med själva teamet som arbetar mer på beskrivningsnivå och bara har arbetet på listan att prioritera [59].

Som Scrum Master blir man en representant för metoden och det är Scrum Masterns uppgift att föra all kommunikation utåt gentemot andra intressenter, ta initial kontakt och lösa alla eventuella konflikter och hinder utåt [7]. Det blir något av en PR-roll där det gäller att marknadsföra projektet, få produktägare och andra med på noterna. Internt är det teamet som är ansvarigt för utvecklingen och Scrum Mastern har inget leverans eller resultatansvar. Rollerna i Scrum ger en annan ansvarsfördelning och resultatansvaret ligger högre upp hos produktägaren. Softhouse sitter ofta ute hos kund som använder sig av andra utvecklingsmetoder. För att göra det möjligt att köra Scrum fullt ut har det blivit viktigt att förklara hur processen fungerar. Förväntningarna på Scrum Mastern blir ofta samma som på den traditionella projektledaren från dem som inte är insatta. De har svårt att ta till sig att det är teamet som är gemensamt ansvarigt och att det inte finns en projektledare att gå till med alla frågor. Det är därför enligt både Atles och Rozenberg en grundförutsättning att alla, även ledningen, är införstådda med processen och Scrum för att det ska fungera.

En stor del av tiden och energin går enligt Rozenberg åt till att ställa om personer med annat tankesätt. Kommunikationen, både internt och externt, små vardagliga problem som att se till att personer som borde prata med varandra också gör det, lösa teamets långsiktiga och kortsiktiga problem, kolla status på projektet, göra rätt prioriteringar och se till att teamet arbetar med rätt saker varje dag är annars sysslor som kräver mycket tid.

Internt kan det i början kännas stressande för teammedlemmar som inte är vana vid eget ansvar att göra egna nedbrytningar i mindre "tasks" och tidsestimeringar. Det gäller enligt Rozenberg för Scrum Mastern att få dessa personer över tröskeln. Det är svårt att lära folk att uppskatta den tid det tar för en "task" att bli klar även om man koncentrerar sig på den tid som är kvar istället för den totala tiden. Vissa personer vantrivs dock något enormt, situationen med eget ansvar för alla teammedlemmar eller känslan av fråntaget ansvar från traditionella projektledare, gör att Scrum inte passar alla enligt Rozenberg och Lundgren. På Citerus undviker man ofta att utbilda gamla "projektledarrävar" till Scrum Masters då de har svårt att släppa rollen som totalt ansvariga. Ett ansvar som Atles anser är för tungt för den traditionella projektledaren och leder till en stress som går ut över resten av teamet. Därför är rollen som Scrum Master egentligen behagligare, man sätter mer ansvar på gruppen att planera och förbinda sig till projektet. Jobbet ligger snarare i att vara trollkonstnär och kunna parera.

Scrum stödjer Scrum Mastern i sitt arbete genom att erbjuda ganska mycket formalism som gör det möjligt att mäta och följa upp dagligen. De dagliga mötena ger enligt Lundgren en oerhört viktig statuscheck på projektet, Sebring anser vidare att man genom mötena snabbt kan åtgärda de problem som annars riskerar att bromsa hastigheten. Lundgren menar att verktygen i Scrum ofta ger en bättre uppdaterad information än vad något annat verktyg i de disciplinära metoderna ger. Man kan få ut siffrorna direkt.

För att ett projekt ska lyckas så krävs det enligt Sebring att Scrum Mastern själv följer metoden och reglerna. Det går inte att lägga ansvaret på gruppen och sen peka med hela

handen och bestämma exakt vad de ska göra i alla fall. Även Atles anser, trots att man ska förhålla sig flexibel till metoden, att det är väsentligt att man följer den metod man valt oavsett vilken metod det gäller. Det är vidare viktigt att metoden anpassats till företagets verksamhet och inte tvärtom, allt annat leder till frustration. Projekt har i Atles mening aldrig fungerat och det blir därför svårt att diskutera vad som krävs för att uppnå ett lyckat resultat. Man har alltid stött på problem vid projekt, skillnaden är att man tidigare skötte problemen under ytan och all dokumentation såg fin ut. Men man följde sällan den dokumenterade arbetsmetoden.

På en annan nivå menar Sebring att det är viktigt för Scrum Mastern att vara fokuserad på de problem som kommer upp under mötena med teamet och att kunna fungera utan en detaljerad plan och att kunna arbeta proaktivt, för att projektet ska lyckas. Scrum Mastern måste ständigt hålla fokus på processen, mycket kan efter ett tag bli vana och det kan leda till slarv. Det är då enligt Rozenberg lätt att man tappar hela processen. För ett lyckat resultat ska även en förmåga att överblicka verksamheten och förmågan att jämkä olika åsikter och viljor till [88].

Alla intervjuade fick möjlighet att nämna de viktigaste egenskaperna för en Scrum Master, utfallet av denna fråga såg ut på följande vis:

- Social, bollplank
- Lyhörd
- Prestigelös
- Pådrivande
- Förtroendeingivande
- Domänkunnande
- Handlingskraftig
- Diplomatisk
- Erfaren
- Kreativ
- Modig

För att vara aktuell som Scrum Master ska man enligt Lundgren besitta coachegenskaper och hellre ha en meritlista för ledarskap inom sport eller ideell verksamhet, snarare än ha en fin projektledarutbildning på papper. Rozenberg anser att man måste förstå, anamma och förespråka Lean och agila principer, inte tro att man kan ta saker direkt ur en bok utan situationsanpassning. På Massive Entertainment vill man inte snäva in sig till en typ av ledare utan ser att flera olika ledartyper kan passa in så länge det passar företagets arbetskultur. Erfarenhet är något som väger tungt på Appium och man anser där att en projektledare från RUP-miljö lätt kan anpassas till deras utvecklingsprocess. Vidare är det viktigt att man som projektledare har styrka och integritet att säga nej [68].

8 Diskussion

För att kunna gå in på hur arbetet ser ut för en agil projektledare börjar vi med att titta på likheter och skillnader mellan traditionella projekt och de agila metoderna. Därefter diskuteras företagets användning och syn på Scrum i förhållande till teorierna. Med bakgrund i det diskuteras sedan på vilket sätt en agil projektledares arbete skiljer sig från den traditionella projektledarens.

8.1 Förhållandet mellan Agil & Traditionell projektledning

Ett projekt definieras som en uppgift som är avgränsad i tid och omfattning. Målet med uppgiften ska vara tydligt definierat och uppgiften ska vara möjlig att planera med avseende på tid, resurser och kostnader. Redan i den definitionen skiljer sig agila projekt från de traditionella. Det är svårt att prata om hur lång tid en utveckling med hjälp av agila metoder kommer att ta då det slutliga målet inte blir definierat på samma sätt som i den traditionella utvecklingen. Därför blir det också svårt att i början tala om kostnader för utvecklingen.

Tonvikten på vad som är viktigt att ta upp i beskrivningen av projektets förutsättningar förändras därmed också. Syftet med beskrivningen; att sätta upp de ramar som projektet är tänkt att hålla sig inom, förändras helt. I en agil utveckling vet man inte på förhand vilken omfattning projektet kommer att ha i sin helhet. I Scrum finns en product backlog, som produktägaren ansvarar för, men den är inte på något sätt definitiv från start utan förändras under projektets gång. Därför är det mer på sin plats att jämföra den traditionella projektbeskrivningen med Scrums sprint backlog, där de väsentligaste skillnaderna är tidsperspektivet som är mycket kortare i Scrum och sättet att se på tidsåtgång, förverkad tid för de traditionella metoderna kontra tid kvar för de agila.

Löow (se kap. 3.2.1) har tagit fram ett antal framgångsfaktorer och fallgropar för traditionella projekt. En del av fallgroparna, som att projektet kör på utan avstämning eller undermålig planering, undviker de agila metoderna med de återkommande mötena och den muntliga kommunikationen. Andra fallgropar, till exempel att projektet är för otydligt eller har inga eller för få avgränsningar, påverkar det agila projektet tydligare men ses inte som fallgropar då man välkomnar snabba förändringar i såväl specifikation som omfattning.

Något som ofta påpekas kring agila metoder är svårigheten att applicera metoden på stora projekt. Framförallt för att möjligheterna till en god kommunikationsmiljö förändras. Boehm och Turner (se kap. 4.4) tar också upp att agila metoder klarar sig bäst när de används av små utvecklingsteam. Scrum är inget undantag utan menar att 5-9 utvecklare är en optimal storlek. Möjligheten att snabbt förändras är direkt inverkan på denna faktor då det genom små team är enklare att muntligt få fram vad som behövs. En daglig förändring i ett team med 100 utvecklare skulle vara svårt att få till stånd. Att projekten blir för stora är därför en fallgrop som man inom de agila metoderna bör se upp med ännu mer och vara aktsam för felaktig skalning av metoden.

Lööw (se kap. 3.2.1) säger att en av de vanligaste fallgroparna är för lite avsatt tid att skapa vi-anda i projektgruppen. Detta är en aktivitet som normalt försiggår utanför projektgränserna men som borde hjälpas upp av ett tätare samarbete inom ramen för projektet. Återigen spelar den nära muntliga kommunikationen, som ofta är mindre officiell, i de agila metoderna en roll. Att ha tid för att sammansvetsa gruppen är fortsatt viktigt men blir genom metodens utformning inte av samma vikt inom de agila projekten som de traditionella.

I ett Scrum team ska vilken medlem som helst kunna ta sig an en uppgift i den föreliggande sprinten. Trots det är tanken inte att hindra vissa individers specialistkompetens. Men detta kan leda till en viss likriktning av kompetensen i teamet. Vidare är det, enligt intervjuerna, så att Scrum eller det agila tänkandet inte passar alla och dessa personer därmed sorteras bort från teamet. Resultatet av det kan bli ett team sammansatt av personer med väldigt lika personlighet och även kompetens. Ett team med den sammansättningen är något Lööw (se kap. 3.2.1) varnar för och bör därför beaktas i de agila metoderna så att likriktningen inte blir för stor.

En fallgrop som tillkommer med användandet av agila metoder är brist på kännedom kring arbetssättet bland projektets deltagare och intressenter. Intervjuerna gav att det är en grundförutsättning att alla involverade förstår metoden, inte bara själva utvecklingsteamet utan hela organisationen och kunden.

Genom att frånga det linjära tänkandet och bedriva utvecklingen i iterationer kan man i de agila metoderna fånga många av de faktorer som Lööw (se kap. 3.2.1) menar krävs för framgång. Det finns en tydlig vilja att revidera målen, man får en kontinuerlig information, förankring och uppföljning nästan per automatik. Beslutsunderlagen blir på många sätt väl genomtänkta eftersom man skjuter fram beslutet till den tidpunkt då man faktiskt vet. Lundgren ser den formella delen i Scrum som en stor fördel, den ger möjlighet för en kontinuerlig feedback och genom dessa kan man enkelt få ut siffror på saker som sker. Atles talar om de tydliga rollerna som en av Scrums främsta fördelar. Tydliggjorda förväntningar, roller och tydlig arbetsfördelning är en framgångsfaktor även enligt Lööw (se kap. 3.2.1). Rozenberg ser nedbrytningen av uppgifterna som en inneboende fördel med Scrum, samtidigt som Atles tar upp problemet med ny teknik och att det då kan vara svårt att göra korrekta estimat och därmed nedbrytningar. Grundverktygen för nedbrytning finns inom Scrum men sen är det upp till den enskilde teammedlemmen att utnyttja dem rätt. Men nedbrytning är en framgångsfaktor som är väsentlig även inom agil utveckling.

De grundläggande punkterna som alla agila metoder har gemensamt är enligt Koch (se kap. 4.3) rörlighet, förändring, planering, kommunikation och lärande. På dessa punkter skiljer sig de agila metoderna mot de disciplinära och därmed även mot traditionell projektledning. Inom traditionell projektledning ska du följa planen och skydda den från yttre påverkan, i nödfall sker förändringar men om det krävs beror det på dålig planering. Mycket sker via dokumentation och lärandet ligger mycket i att följa samma dokument från gång till gång. Detta stämmer inte överens med de agila metoderna där lärandet ligger i den muntliga kommunikationen och dokumentation enbart utförs när den kan tillföra något som inte den muntliga informationsöverföringen kan. En generell fördel och en möjlig framgångsfaktor

med det, som de intervjuade anger, är att man i de agila metoderna inte döljer problem bakom gedigen dokumentation utan lyfter upp dem i dagern.

8.2 Scrum i användning

Beroende på hur man tolkar det så verkar det kunna finnas en lätt motsägelse i den agila världen. Vissa anser att man ska följa metoden till punkt och pricka om den ska fungera, medan man fortfarande anser att rörligheten är viktig. Då rörligheten går ut på att det ska gå att förändra och anpassa utvecklingen efter omvärldens krav, kan det i vissa fall betyda att metoden inte går att applicera rakt av utan förändringar. Alla de intervjuade företagen har på ett eller annat sätt förändrat metoden eller ställt sig positiva till att göra förändringar vilket tyder på att det är relativt vanligt att frångå metoden och anpassa den efter behov. Att följa en metod till punkt och pricka bör därför tolkas som "följ den metod ni valt, egenkomponerad eller inte, till punkt och pricka". Det är viktigt att följa reglerna och det är viktigt att inte släppa processen för tydlighetens skull, något som de intervjuade företagen även betonar. Man lägger även vikt vid att det är metoden som ska anpassas till företagets verklighet och inte tvärtom. Att tvinga in företagets verklighet i metoden skulle leda till frustration. Därför handlar rörligheten i de agila metoderna kanske mer om en inbyggd rörlighet. Det är flexibiliteten gentemot kunden och anpassningen till hur marknaden ser ut, att göra det möjligt att förändra och ta in nya krav under utvecklingens gång, som är den grundläggande avsikten med rörligheten.

Boehm och Turner (se kap 4.4) drar resonemanget kring val av metod vidare och menar att man kan använda sig av deras balanserade kriterier; storlek, kritisk faktor, dynamism, personal och kultur, för att reda ut om ett projekt lämpar sig bäst för en agil eller disciplinär metod. Författarna ställer sig inte främmande till en mix av agila och disciplinära metoder utan ser att alla företag har olika förutsättningar vad gäller användning av metoden, ledning, teknik och personal. Atles på Appium använder i princip samma resonemang, men ser att det kan finnas behov av olika metoder även inom ett projekt, beroende på vilket skede man befinner sig i. Därför använder sig Appium också av en utvecklingsmetod som blir formellare efterhand som kunskapen och säkerheten i projektet ökar. Alla företagen använder sig också i någon situation av, eller efterfrågar, verktyg eller tekniker som gör metoden mer formell. Även Citerus, som utbildar i Scrum, kan se en nytta av detta, liksom IBM (se kap. 4.3).

Företagen vi har varit i kontakt med verkar inte ha valt metoden Scrum som sådan, för att det är just den metoden, utan för att det är en av de mer frekvent använda utav de agila metoderna och den som man generellt har bäst kännedom om i Sverige. Lundgren menar att Scrum är lätt att förstå, går att använda på en hel verksamhet och språkbruket är detsamma oavsett nivå. Företagen verkar dock betona att det inte är metoden som är det viktiga utan det bakomliggande agila tänkandet, med frekventa möten och att man sitter tillsammans vilket underlättar kommunikationen, som är det väsentliga. Sen kan man kalla metoden Scrum eller vad man vill.

De intervjuade företagen förefaller ha en stor medvetenhet kring metoder, inte bara bland utvecklarna utan på företagen som helhet. Detta ter sig egentligen inte konstigt då det för det första understryks hur viktigt det är att ledningen är införstådd med hur processen fungerar,

något som kanske inte är lika vanligt i traditionell utveckling, och för det andra då det finns en stor betydelse i att utvecklarna själva förstår metoden och klarar av att anpassa den inför nya situationer. Enligt Boehm och Turner (se kap. 4.4) så krävs det större kunskap och förståelse för metodrelaterade funktioner om man ingår i ett agilt projekt. Detta torde ha sin största förklaring i att det inom agila metoder är teamet som helhet som sköter utvecklingen och varje enskild utvecklare har ansvar för sin del. Men medvetenheten kring metoden verkar på de intervjuade företagen ha lett till att de vågar kringgå den föreskrivna och tillverka en egen metod. För att skapa en gemensam notation, tydlighet utåt samt kvalitetssäkring så väljer man att kalla metoden Scrum, snarare än något annat som inte är lika bekant och vedertaget.

8.3 Den agila projektledaren

Enligt Briner, Geddes och Hastings (se kap. 3.2) så finns det gemensamma kännetecken som är utmärkande för projektledarens roll som inte förekommer i andra chefsroller. Vid en första anblick så stämmer dessa kännetecken även in på den agila projektledaren så när som på de två första punkterna; "Ansvarig för att projektmålen nås" samt "Den ytterst ansvarige och en person som inte kan gömma sig". Projektledarrollen för ett agilt projekt ska inte jämföras med en chefsroll på samma vis som den traditionella projektledarrollen kan jämföras med VD-rollen. Det verkar först och främst vara ett semantiskt misstag att använda ord som "chef" och det går emot den agila kulturen. Men sen är inte den agila projektledaren ytterst ansvarig för projektet och inte heller för att projektmålen nås. Att projektledaren är "begränsad då det gäller direkta befogenheter" och "tvingad att arbeta över organisatoriska gränser" gäller fortfarande i högsta grad och kanske ännu mer i ett agilt projekt. Den agila projektledaren har inte bestämmanderätt över teamet och arbetet inom organisationen och utåt mot kunden har fått en ökad betydelse. Som projektledare är man ofta verksam inom nya områden för företaget, sättet att hantera osäkerheten vid ny teknik, nya marknader och nya problemlösningsmetoder, är annorlunda inom de agila metoderna. Osäkerheten bemöts med flexibilitet och projektledaren får arbeta med att få upp acceptansen för det okända inom organisationen.

Briner, Geddes och Hastings (se kap. 3.2) menar att det finns sex riktningar som projektledaren måste vända sig åt och att projektledarens roll kan delas in i tre dimensioner. Vi använder oss av dessa riktningar och dimensioner för att redogöra för den agila projektledarens roll och arbete i relation till den traditionella projektledaren. Inom det ramverket kommer vi även att diskutera vanliga arbetsuppgifter för en agil projektledare i relation till de arbetsuppgifter Löow (se kap. 3.2) tagit upp för den traditionella projektledaren.

De två första aspekterna, uppåt och utåt, behandlar hanteringen av projektets intressenter:

Uppåt- Den person som står det agila projektet närmast uppåt är produktägaren. Denna person är ansvarig för product backlogen och därmed även för produkten i sin helhet. Produktägaren är inte samma person som den traditionella projektets mentor, men kommunikationen uppåt är fortfarande väldigt viktig. Den agila projektledaren arbetar uppåt med att få även de ovanför i organisationen att förstå utvecklingsmetoden. Alla de intervjuade företagen har förändrat metoden samtidigt som vikten av att hela organisationen

är införstådd betonas, vilket leder till att det blir den agila projektledarens uppgift att informera. Det är även den agila projektledarens uppgift att knyta rätt kontakter, få in rätt personer och se till att personer som bör kommunicera också gör det. Genom kommunikation och genom att lösa eventuella konflikter och hinder är det den agila projektledaren som ska möjliggöra ett lyckat resultat för projektet. Den öppna kommunikationen ska även reda ut vilka eventuella personliga skäl som olika personer kan ha för projektet. En viktig skillnad jämfört med det traditionella projektet är att den agila projektledaren inte har något resultatansvar utan det ligger högre upp i organisationen. Att den agila projektledaren blir avlastad på det ansvaret anser Atles leder till en minskad stress och press som annars kunde ha påverkat projektet och utvecklarna negativt. Detta borde även göra ett kritiskt objektivet synsätt kring projektet och utvecklingsmetoden enklare. Det blir mindre laddat att framföra brister och fel om man inte blir "hängd" för det senare, vilket kan vara fallet i traditionella projekt där en syndabock ofta krävs. Inom de agila projekten faller den biten bort vilket i slutändan borde leda till en mer kvalitativ produkt.

Utåt- Precis som i traditionella projekt så finns det flera intressenter som har förväntningar som den agila projektledaren på ett eller annat sätt måste infria. Eftersom man väljer att ta in kunden närmare i agil utveckling så blir arbetet på businessnivå och hantering av krav från olika intressenter en stor del i den agila projektledarens roll. Det är dennes uppgift att sköta all kommunikation och ta hand om alla problem, lösa konflikter och hinder utåt. Allt för att teamet ska kunna arbeta ostört. Förväntningarna från intressenterna är ofta de samma på en agil projektledare som på en traditionell projektledare. Det är en grundförutsättning att alla är införstådda med den utvecklingsmetod man arbetar med, därför blir den agila projektledarens roll ofta en marknadsförande sådan där det gäller att få intressenterna med på noterna och förstå. Även, som i Softhouse fall, då man sitter som konsult hos kund med en annan utvecklingsfilosofi blir det oerhört viktigt att förklara den agila metoden om man vill kunna köra den fullt ut. I ett agilt projekt går det inte att tala om kostnader, exakta tidpunkter för slutförande och projektomfattning på samma sätt som i ett traditionellt projekt. Frågor som alla intressenter är vana vid att kunna få svar på. Därtill kan det vara svårt att ta till sig att teamet är gemensamt ansvarigt och det inte finns en traditionell projektledare att gå till med alla dessa frågor. Det är den agila projektledarens roll att klargöra situationen för alla.

I jämförelse med den traditionella projektledarens roll uppåt och utåt så tar sig arbetet för en agil projektledare snarlikt för det är samma intressenter man tvingas hantera. Men tonvikten i arbetet är annorlunda, utöver att försöka tillgodose intressenternas krav har den agila projektledaren fått en PR- och utbildande roll som är starkare än tidigare och man arbetar mer politiskt. Den väsentligaste skillnaden är dock avsaknaden av resultat- eller leveransansvar vilket ger en öppnare miljö för kommunikation där brister och fel kan påpekas, som i sin tur kan inverka positivt på slutprodukten.

Aspekterna framåt och bakåt gäller hanteringen av projektets livscykel:

Framåt och bakåt: Riktningarna kan bilda en gemensam funktion även för agila projekt. En större del av arbetet sker via muntlig kommunikation än dokumentation i de agila projekten. Därför blir det också viktigare att ha en frekventare kommunikation och tätare möten så att inget glöms bort. Detta uppnås i Scrum genom de dagliga mötena där alla får redogöra för status av sin del av projektet. Nuvarande status för ett projekt blir därmed tämligen enkelt att redovisa för intressenterna. För att kunna blicka tillbaka och lära av misstagen används mötena i slutet av varje sprint, där iterationens arbete sammanfattas och diskuteras. För planeringen framåt används product och sprint backloggarna, där alla uppgifter finns med, samt burn down charts där återstående tid för utvecklingen syns. Dock verkar verktyg för framförhållning för projektet på längre sikt saknas, verktygen koncentreras till den pågående sprinten. Detta har lett till att de agila projektledarna har tagit in, eller efterfrågat, verktyg (t ex Gantt-schema) för att kunna planera för en längre period framåt. Det finns yttre omständigheter, som till exempel testmiljöer som ska bokas i förväg, vilket gör att en längre tidshorisont krävs.

Det är upp till den agila projektledaren att skapa ett ändamålsenligt system för övervakning och rapportering, precis som i de traditionella projekten. De verktyg som finns för tillbakablick och kontroll av projektets nuvarande status verkar fungera tillfredsställande medan det för den framtida planeringen ofta tas in formellare verktyg än vad den agila metoden ger vilket leder till en, i vissa fall, mer traditionell process.

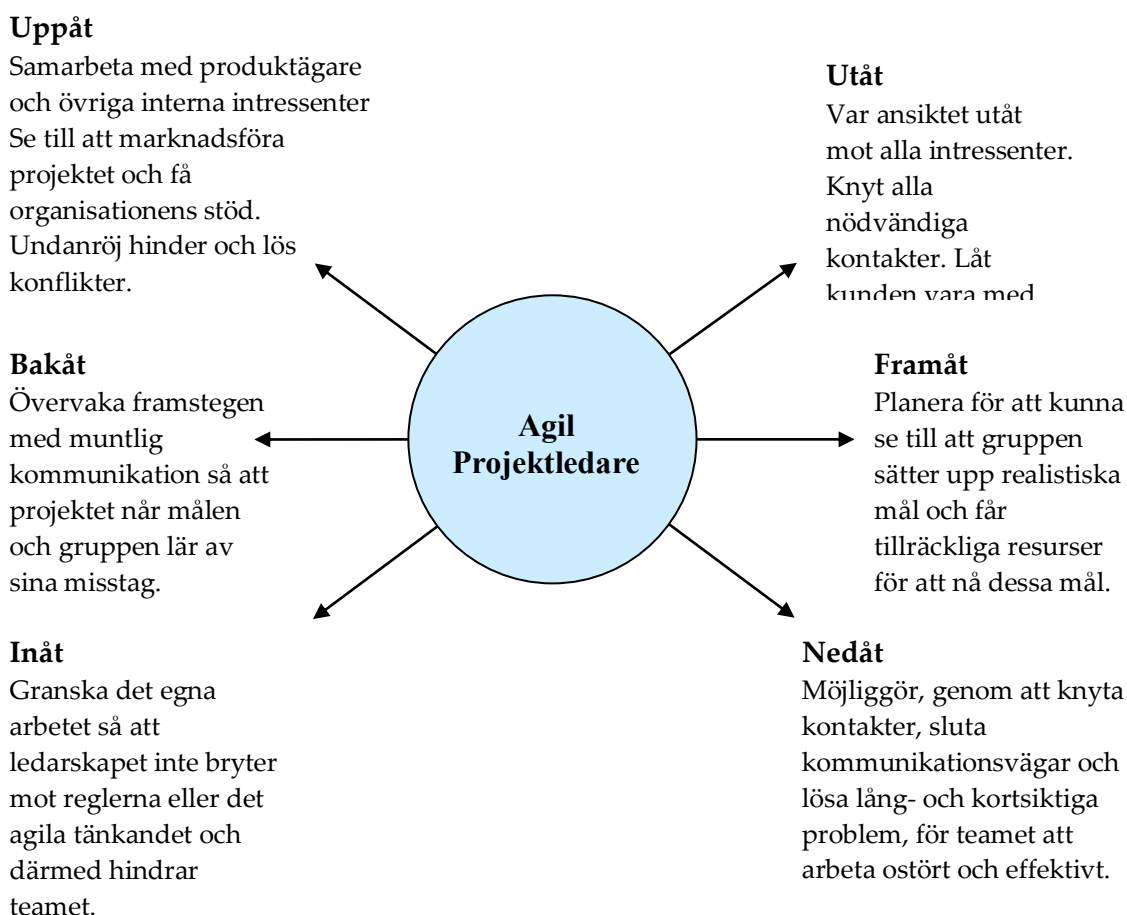
De två sista aspekterna av projektledarens roll handlar om hantering av prestationer:

Nedåt – I motsats till de traditionella projekten så är inte den agila projektledaren ansvarig för teamets eller de individuella medlemmarnas prestationer. Teamet är gemensamt ansvarigt för arbetet i den föreliggande sprinten. Den agila projektledaren är inte en egentlig medlem i teamet och sitter därför inte som den enda personen i gruppen med mer makt än de andra. Detta förhindrar att den agila projektledaren kan gå in och styra exakt vad någon i teamet ska göra. Att ansvaret ligger på den enskilda teammedlemmen är något som verkar kunna ställa till problem i de fall då man inte är van vid att tänka i de banorna eller arbeta på det viset. Mycket av arbetstiden går därför åt till att få alla personer trygga och bekväma med ansvaret och göra den del av arbetet som tidigare låg på projektledaren men som nu ligger på teamets medlemmar, som nedbrytningar i mindre "tasks" och korrekta tidsestimeringar. I likhet med kommunikationen uppåt är den agila projektledaren ansvarig för kommunikationen nedåt. Denne ska mest verka för att teamet kan utföra de uppgifter de är ålagda att göra och agera möjliggörare, det vill säga undanröja hinder och lösa konflikter. I de dagliga sysslorna ingår även en ständig kontroll av projektets status, se till att göra rätt prioriteringar i backloggen och se till att teamet arbetar med rätt saker.

Inåt – I de agila projekten så har projektledaren en väldigt proaktiv roll. Denne ska vara förmedlare mellan teamet och projektets intressenter, ha en förmåga att överblicka verksamheten och kunna jämkna olika åsikter och viljor. Detta är egenskaper och förmågor som är viktiga även i det traditionella projektet men i det agila projektet är förutsättningarna något annorlunda. Det finns inte samma mängd dokumentation att luta sig tillbaka på som annars kunnat ge en känsla av "gott utfört arbete för dokumentationen är klar". Reflektionen av det egna arbetet blir därmed direkt hänvisad till en annan nivå, men är sedan som vanligt upp till projektledarens förmåga att objektivt analysera sitt eget arbete. För att trivas som

ledare för ett agilt projekt så bör man vara en person som trivs med förändring och dessutom vill driva den, vara bekväm med att inte arbeta med en lång tidshorisont och ha fasta planer, samt inte ha behovet att punktstyra hela projektet utan ha en coachande inställning. Rollen som traditionell projektledare är väldigt social men det går att vara projektledare även utan den egenskapen. I ett agilt projekt, där så mycket baseras på muntlig kommunikation, är det väsentligt att projektledaren är social, lyhörd, diplomatisk och förtroendeingivande, dessa egenskaper går inte att klara sig utan.

De självorganiserande och självbestämmande teamen gör rollen som agil projektledare annorlunda. Man ingår inte i teamet och styr inte längre neråt på samma sätt utan gruppen är ansvariga gemensamt. Den agila projektledaren arbetar bara som möjliggörare för att teamet ska kunna sköta sitt arbete. För att kunna agera adekvat som agil projektledare bör man därför ha en coachande inställning, förmåga att ta människor, vara säker i det agila tänkandet och trivas med förändring.



Figur 8-1 Den agila projektledarens riktningvisare – omarbetning av Briner, Geddes och Hastings, (se kap. 3.2)

Löow har tagit fram ett antal personliga och formella egenskaper som verkar krävas för att en projektledare ska kunna uppfylla sin roll som projektledare på ett bra sätt. Dessa togs upp i kapitel 3.2 Traditionell projektledning och är:

- Entusiasmerare
- Förhandlare
- Diskussionsledare
- Samordnare
- Problemlösare
- Bollplank
- Ledare/coach
- Ombud för projektet
- Informatör

Med tanke på hur de intervjuade företagen beskriver arbetet för en agil projektledare så är ovanstående egenskaper gällande även för dessa. Då respondenterna fick frågan om vilka personliga egenskaper som de ansåg vara viktiga sammanföll, med lite modifikation och tolkning av ord, till och med flera av deras svar med listan:

- Social, bollplank
- Lyhörd
- Prestigelös
- Drivande
- Förtroendeingivande
- Domänkunnande
- Handlingskraftig
- Diplomatisk
- Erfaren
- Kreativ
- Modig

De skillnader man kan se är att respondenterna trycker ytterligare på formella politiska egenskaper som prestigelöshet, diplomati, handlingskraft och mod, samt vill se en person med kunskaper inom både metod och produkt för att inge förtroende. De politiska egenskaperna kan man tänka sig behöver framhävas ytterligare hos en agil projektledare då arbetet handlar så mycket om kommunikation och utbildning över organisationsgränser. Att en person med breda kunskaper inom både metod och produkt efterfrågas har dels med att man inte dokumenterar metoden att göra, dels med att man inom agil utveckling inte sätter gränser på samma vis och därför behöver någon som kan de tekniska gränserna. Två av fallgroparna i projekt (Löow, se kap. 3.2.1) innefattar projektledaregenskaper. De handlar om att projektledaren ska kunna entusiasmera och motivera gruppen samt kunna sätta gränser och säga nej. Även i det agila projektet är det viktigt att projektledaren, genom sin uppgift att jämka samman viljor, verkar motiverande på teamet, något som de under intervjuerna nämnda egenskaperna även vittnar om. Tillika bör man som agil projektledare ha styrka och integritet att säga nej, även det en egenskap som kom fram under intervjuerna.

9 Slutsatser

Mot bakgrunden av att den traditionella projektledaren redan är välbeskriven i litteraturen har vi ställt det litteraturen har att säga om den traditionella projektledarens arbete och roll mot det resultat vi fick fram empiriskt om den agila projektledaren.

I jämförelse med den traditionella projektledarens roll uppåt och utåt så ter sig arbetet för en agil projektledare snarlikt för det är samma intressenter man tvingas hantera. Men tonvikten i arbetet är annorlunda, utöver att försöka tillgodose intressenternas krav har den agila projektledaren fått en PR- och utbildande roll som är starkare än tidigare och man arbetar mer politiskt. Den väsentligaste skillnaden är dock att en agil projektledare inte har något resultat- eller leveransansvar vilket ger en öppnare miljö för kommunikation där brister och fel kan påpekas, som i sin tur kan inverka positivt på slutprodukten.

Det är upp till den agila projektledaren att skapa ett ändamålsenligt system för övervakning och rapportering, precis som i de traditionella projekten. De verktyg som finns för tillbakablick och kontroll av projektets nuvarande status verkar fungera tillfredsställande medan det för den framtida planeringen ofta tas in formellare verktyg än vad den agila metoden ger vilket leder till en, i vissa fall, mer traditionell process.

De självorganiserande och självbestämmande teamen gör rollen som agil projektledare annorlunda. Man ingår inte i teamet och styr inte längre neråt på samma sätt utan gruppen är ansvariga gemensamt. Den agila projektledaren arbetar bara som möjliggörare för att teamet ska kunna sköta sitt arbete. För att kunna agera adekvat som agil projektledare bör man därför ha en coachande inställning, förmåga att ta människor, vara säker i det agila tänkandet och trivas med förändring.

I relation till de formella och personliga egenskaper som Lööw (se kap. 3.2) anser att en traditionell projektledare bör inneha fann vi att de är de samma även för en agil projektledare. Men de traditionella projektbegreppen styra, stödja och kontrollera har en annan, lite negativ, innebörd i de agila projekten. Därför tryckte respondenterna ytterligare på formella politiska egenskaper som prestigelöshet, diplomati, mod och handlingskraft. I listan nedan ses Lööws (se kap. 3.2) efterfrågade egenskaper och inom parentes de egenskaper som de intervjuade framhävde men som med fördel kan vägas in i Lööws begrepp:

- Förhandlare (*Diplomatisk, lyhörd, prestigelös*)
- Diskussionsledare (*Drivande*)
- Samordnare (*Handlingskraftig*)
- Problemlösare (*Domänkunnande, kreativ, modig*)
- Bollplank (*Social*)
- Ledare/coach (*Förtroendeingivande, erfaren*)
- Ombud för projektet
- Informatör

Övriga slutsatser som drogs i diskussionen rör det agila projektets fallgropar och framgångsfaktorer snarare än projektledarens. Två av fallgroparna, "Projektledare som inte kan skapa entusiasm" och "Projektledaren har svårt att säga nej", som Lööw (se kap. 3.2.1) tagit fram rör endast projektledaren dock. Att ha styrka och integritet att säga nej samt kunna verka motiverande på teamet är viktigt även som agil projektledare. Resterande av Lööws (se kap. 3.2.1) fallgropar behandlar bara projektet. Där togs en del fallgropar bort i diskussionen då de inte har samma inverkan på agila projekt och en del nya fallgropar som bara rör agila projekt kom till.

Fallgropar för det agila projektet:

- Brist på kännedom kring metoden i hela organisationen
- Oaktsam eller felaktig skalning av metoden
- För lite tid avsatt för att skapa vi-anda i projektgruppen
- Projektledare som inte kan skapa entusiasm eller motivera projektgruppen
- Projektledaren har svårt att säga nej
- Sammansättningen i projektgruppen innebär alltför lika personlighetstyper
- För stora projekt

Framgångsfaktorerna för det agila projektet är de samma som för traditionella projekt. Men en del inneboende egenskaper i de agila projekten underlättar uppfyllandet av framgångsfaktorerna. Den framgångsfaktor som tillkommer är att problem inte döljs bakom gedigen dokumentation utan problem lyfts upp i dagern i de agila projekten.

Framgångsfaktorer för det agila projektet:

- Klar struktur på arbetet
- Helst heltidsengagerade projektledare
- Tydliga direktiv
- Entusiastiska medarbetare
- Gemensamma mål för alla inblandade
- Tydliggjorda förväntningar, roller och tydlig arbetsfördelning
- Hänsynstagande till deltagarnas värderingar
- God planering/rätt sak på rätt sätt
- Målen nedbrutna i etappmål
- Revidering av målen i förekommande fall
- Kontinuerlig information och förankring
- Väl genomtänkta beslutsunderlag
- Uppföljning av resultatet
- Dölj inte problem bakom gedigen dokumentation

Vårt kunskapsbidrag är ett tillägg till befintliga teorier kring projekt och speciellt till projektledning. Vi har genom att studera den agila projektledaren empiriskt funnit en annan roll och beskrivit den i relation till den traditionella projektledaren. Vi har även studerat det agila projektet och tagit fram framgångsfaktorer och fallgropar för detta som bekräftar, reviderar eller utökar teorierna kring traditionella projekt.

10 Förslag till vidare forskning

Områdena agil systemutveckling och projektledning är två väldigt stora områden att forska på och i högsta grad intressanta för informatiker.

I direkt relation till den här uppsatsen skulle även den traditionella projektledaren med fördel kunna undersökas empiriskt, för att nyansera och göra tillägg till våra slutsatser.

Mycket av den agila projektledarens arbetstid går idag åt till metodundervisning. Får de agila metoderna ett bättre fäste och den allmänna kunskapen ökar, försvinner en del av det arbetet. Det hade även varit intressant att titta vidare på den agila projektledarens arbete då, eftersom rollen kommer att förändras.

Genom den teoretiska studien, samt när respondenter skulle hittas, fick vi en känsla av att de agila metoderna är på frammarsch. Medvetenheten kring metoden på de företag som arbetar agilt har vi diskuterat. Men det hade varit intressant med en vidare diskussion kring metodval där attityder på företag som arbetar med traditionella metoder tas med. Just medvetenheten kring metodval skulle kunna bli en väldigt stor fråga för företagen som helhet då vi ser ett scenario där de förskapade metoderna som ska användas rakt av kommer att sjunka i popularitet. Det har blivit ett marknadsföringskrig mellan olika metoder från olika skolor och alla anser de sig vara bäst. En ökad medvetenhet skulle leda till en bättre sällning bland metoderna och fler modifieringar för att passa verksamheten.

Ett annat område som påträffats genom empirin är modifieringen av metoder. Detta skulle i längden kunna göra att Scrum inte alls ser likadan ut hos olika utvecklingsföretag, beroende av hur långt modifieringen tillåts att dra. Hur fungerar då de gemensamma vokabulärerna och hur skulle detta speglas utåt, i synnerhet då om två bolag blir anlitade av ett annat, och båda bolagen säger sig använda Scrum.

11 Bilaga A: Intervjuformulär

Semistrukturerat intervjuformulär som medger tillägg i formulär under intervjun.

- 1) **Frågor om respondenten t.ex. namn, utbildning, arbetsplats, nuvarande arbetsposition osv.**
- 2) **Om ni arbetar med en annan/modifierad variant av Scrum, beskriv gärna den.**
- 3) **Hur påverkar systemutvecklingsmetoden dig i ditt arbete som projektledare?**
- 4) **Vad upptar mest av din tid under ett projekt i rollen av projektledare?**
- 5) **På vilket sätt stödjer metoden dig i ditt arbete som projektledare?**
- 6) **Hur skulle du vilja ha stöd där du anser att metoden ger ett dåligt/obefintligt stöd?**
- 7) **Vad krävs framförallt av dig för att ett projekt ska uppnå ett lyckat resultat?**
- 8) **Vad är viktigast att tänka på när man skall välja/anställa en projektledare?**
- 9) **Har du arbetat med andra metoder tidigare?**
 - a. **Vad tycker du saknas från den metoden?**
 - b. **Vad var bättre?**
 - c. **Hur skulle detta kunna föras in i den agila metoden?**
- 10) **Vilka fem egenskaper anser du vara viktigast för en projektledare?**

12 Bilaga B: Backlog⁵

Days Left in Sprint		15	13	10	8	
Who	Description	7/22/2002	7/24/2002	7/26/2002	7/31/2002	F
Total Estimated Hours:		554	458	362	270	0
-	User's Guide	-	-	-	-	-
SM	Start on Study Variable chapter first draft	16	16	16	16	
SM	Import chapter first draft	40	24	6	6	
SM	Export chapter first draft	24	24	24	6	
Misc. Small Bugs						
JM	Fix connection leak	40				
JM	Delete queries	8	8			
JM	Delete analysis	8	8			
TG	Fix tear-off messaging bug	8	8			
JM	View pedigree for kindred column in a result set	2	2	2	2	
AM	Derived kindred validation	8				
Environment						
TG	Install CVS	16	16			
TBD	Move code into CVS	40	40	40	40	
TBD	Move to JDK 1.4	8	8	8	8	
Database						
KH	Killing Oracle sessions	8	8	8	8	
KH	Finish 2.206 database patch	8	2			
KH	Make a 2.207 database patch	8	8	8	8	
KH	Figure out why 461 indexes are created	4				

⁵ Mountain Goat Software

13 Bilaga C: Transkriberingar

Softhouse

Michael Rozenberg – 3 maj, 2007

Företagsinformation och information om den vi har intervjuat.

- [1]. Softhouse är en svensk mjukvarutillverkare och levererar lösningar för nästa generations nätverkskommunikation. Man erbjuder även konsultkompetens inom området. Företaget är fokuserat på Mobile Multimedia och Software Business Management. Softhouse är medsamordnare av Øresund Agile, en årlig sammankomst av agila utövare från Skandinavien och norra Europa. Michael Rozenberg arbetar på Softhouse sedan hösten 2006, har tidigare läst till civilingenjör i InfoCom och haft eget företag. Han är nu certifierad Scrum Master.
- [2]. Softhouse kan nu hålla i internutbildning men hittills har man åkt till Köpenhamn för att utbilda sig inom Scrum. Ofta är det Jeff Sutherland som hållit i utbildningarna. Rozenberg är ganska nyutexaminerad och känner att skolan saknar förankring till företagen. På LTH läste han inte speciellt mycket om projektledning, processer eller metoder. Det agila tänkandet nämndes lite i form av en kurs i XP.

Hur Softhouse har förändrat Scrum för att passa verksamheten

- [3]. Softhouse anser att det är meningen att man ska ändra i Scrum. Därför blir det olika för varje projekt. Exempel finns där produktägaren inte kan vara närvarande on site mer än en dag i veckan. Då får man sköta kommunikationen på annat vis. Utöver 15 minuters Scrummöte har man ibland även ett tekniskt möte efteråt när folk ändå är samlade. Dessa möten hålls också korta, mötet är bland annat till för att repetera demos innan presentation för kund.
- [4]. Softhouse arbetar även med Lean Development och anser att det är tankesättet i Scrum som är det viktigaste. Scrum handlar mest om Lean och Agile, frekventa möten och att man sitter tillsammans vilket underlättar kommunikationen. Sen kan man kalla det vad man vill, Scrum eller vad som helst.

Hur projektledaren påverkas av metoden och vilka fördelar man ser med Scrum

- [5]. Man blir avlastad av metoden när man ingår i ett team, det är ett sätt man påverkas av metoden. Det är bra med nedbrytningen, allt blir väldigt tydligt och strukturerat. Allt står på tavlan och det blir enkelt och tydligt att diskutera problem med alla.

Scrum Mastern ingår inte i teamet och är egentligen inte heller medlem. Medlemmar i Scrumteamet är de personer som faktiskt arbetar med projektet. Scrum Master är deras facilitator som hjälper till att möjliggöra deras uppgifter. Projektledarens uppgift är att

driva projektet. Att ha en projektledare i teamet innebär stora risker eftersom den personen har mer makt än resterande teammedlemmar.

- [6]. Ibland kan det kännas stressande att ha en överväldigande mängd saker på tavlan att göra. Speciellt i början av ett projekt eller sprint då allt på tavlan finns i kolumnen "opåbörjat", när allt dagen innan fanns i kolumnen "avslutat". Då kan det bli tungt att starta om på nytt men i slutändan är det bra att allt faktiskt finns på tavlan.

På vilket sätt projektledaren får stöd från metoden och var saknas det stöd

- [7]. Som Scrum Master så blir man en representant för metoden. Man får stöd från metoden genom att det blir väldigt enkelt att skicka ut statusrapporter externt. Man har hela tiden kontroll på situationen och vet var man befinner sig i projektet varför statusrapporterna bara tar 5 minuter att skriva. Internt är det teamet som arbetar och SM har inget leveransansvar.
- [8]. I grundläggande Scrum upplever Rozenberg att det saknas ett fokus på release. En release kan ta ett halvår, därför behöver man ibland ha en längre horisont än vad Scrum förespråkar. Man måste kunna synka med andra projekt ibland eller boka en testmiljö flera månader i förväg. Man måste kunna prata med produktägaren om vart man är på väg. I Scrum kan det bli för fokuserat på bara en sprint framåt.
- [9]. Vidare så säger Scrum inget om konflikthantering eller på vilket sätt problem ska lösas, bara att SM är ansvarig för att lösa dem.

Fördelar och nackdelar med metoden

- [10]. Scrum löser inte alla problem eftersom den inte säger något om HUR problemen ska lösas. Det enda Scrum gör är att lyfta upp problemen till ytan.
- [11]. Scrum ligger på team och organisationsnivå. Det finns inga direkta riktlinjer för programmerare. Kan vara svårt för dem då koden ska vara så generell som möjligt. Bra att seniorer är med i teamet som kan arkitekturen. Skönt med tidig feedback i Scrum. Har man gjort fel så får man reda på det på ett tidigt stadium innan man hunnit koda iväg.
- [12]. Softhouse sitter ofta ute hos kund som använder sig av andra utvecklingsmetoder. Ska de köra Scrum fullt ut där blir det viktigt att förklara hur Scrum processen fungerar. Förväntningarna på Scrum Master blir ofta de som ligger på en traditionell projektledare för att ledning och andra inblandade inte är insatt i hur det fungerar. De har svårt att ta till sig att det är teamet som är gemensamt ansvarig och det inte finns en projektledare att gå till med alla frågor. På företag som arbetar med Scrum är det därför väldigt viktigt att ledningen är införstådda med hur processen fungerar.

Vad som tar upp den mesta av tiden

- [13]. Det som tar mest tid och energi är att ställa om folk med annat tankesätt. Det kan kännas stressande att behöva vara ansvarig själv för nedbrytning och tidsestimering i början och man måste få folk över den tröskeln och känna sig bekväma med det.
- [14]. Metoden passar helt enkelt inte alla. Vissa vantrivs något enormt med att styra över sig själv för att situationen känns ovan eller obekvä. Gäller att coacha alla till att gilla metoden och går det inte så ska de personerna som inte vill inte heller vara med i teamet.

Viktiga egenskaper i projektledarrollen

- [15]. Scrum Mastern måste hela tiden hålla fokus på processen. Saker blir lätt rutin och man börjar slarva. Men det är lätt att man tappar hela processen då. Det krävs också en förmåga att ladda om. Det kan vara svårt att starta om, luften går ur en lite vid en release och det nya kan kännas tungt att ta tag i.
- [16]. Scrum Master är en väldigt social roll.

De viktigaste egenskaperna för projektledaren anser Rozenberg vara:

- [17]. Lyhörd – Allt sägs kanske inte rakt ut på retrospektive och daily Scrum, måste lyssna runt
- [18]. Drivande – Varje dag följa upp och ha energi att ta tag i det som kommer upp
- [19]. Domänkunnande – Man känner till teamets vardag och förstår det man jobbar med tekniskt
- [20]. Social – måste kunna prata med folk. Den initiala kontakten med produktägare och externa intressenter tas av SM
- [21]. Duktig på att delegera ansvar
- [22]. Själv vara övertygad om att Lean är bra
- [23]. Teamet måste ha förtroende för dig – du har ingen egentlig makt och inget resultatansvar så det gäller att inte missbruka relationen man har till teamet.

Vad krävs för att uppnå ett lyckat projekt

- [24]. För att projektet ska uppnå ett lyckat resultat så krävs det att Scrum Mastern själv följer reglerna. Han kan inte säga att ansvaret ligger på teamet och sen börja bestämma exakt vad de ska göra i alla fall. Man måste vara drivande utåt se till så att konflikter löser sig. Kan inte teamet lösa ett problem är det SMs ansvar att ta diskussionen med rätt instans utanför.

Anställning av projektledare:

- [25]. En ny projektledare eller Scrum Master är en person som:
- [26]. Kan ta egna initiativ
 - [27]. Förstår, förespråkar och anammar Lean och agila principer

- [28]. Förstår att man inte kan ta saker direkt ur en bok utan allt måste situationsanpassas
- [29]. Är öppen för nya olika situationer

Rozenbergs förtydligande av Agile och Lean:

Agile är ett synsätt gemensamt för en grupp av lättrorliga metoder. Grundtanken med Agile är att i en föränderlig värld krävs utvecklingsmetoder som hanterar förändring som en del av verkligheten, inte sådana som blundar för förändringar eller som försöker reglera bort dem. Fler regler kommer inte ge oss fler lyckade mjukvaruprojekt. Vi behöver flexibilitet - inte rigiditet.

Agile är ett paraplybegrepp. Det är en uppsättning värderingar, attityder och principer. Inom Agile finns ett antal olika utvecklingsmetodiker som anses vara lättrorliga. Detta görs främst då metodikens upphovsman bekänner sig till de agila värderingarna.

Agile manifest:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Lean är en samling principer som utifrån kundens bedömning av värde söker att strömlinjeforma den värdeskapande processen. Lean har sitt ursprung hos Toyota och användes ursprungligen för produktionssystem och har sedan 1980-talet används framgångsrikt inom en rad branscher.

De viktigaste principerna är:

- Eliminera slöseri etc.
- Kontinuerlig förbättring
- Lärande organisation
- Synliggöra processen och status

Massive Entertainment

Henrik Sebring – 28 mars, 2007

Företagsinformation och information om den vi har intervjuat.

- [30]. Massive Entertainment, startades 1997, och producerar spel och interaktiva mjukvaruprodukter till en internationell marknad. 2002 köpte Vivendi Universal Games (VUG) Massive Entertainment för att göra företaget till en stark utvecklare utanför Nordamerika, tillsammans arbetar de nu för att skapa topptitlar till nöjesindustrin. Företaget är lokaliserat till Malmö.
- [31]. Henrik Sebring, har läst Människor Datateknik Arbetsliv (160p) på BTH i Ronneby 93-97. Har jobbat på Massive Entertainment sedan 97 och varit med och utvecklat tre spel med det fjärde på väg just nu. Har arbetat tidigare som Lead Game Designer (ansvarig för ett litet team av game designers). Arbetar sedan mitten av 2006 som projektledare för Team 2 samt som tillförordnad Lead Singleplayer Designer på Massive. Team 2 består idag av 10 personer som arbetat med konceptutveckling men förväntas utökas under hösten och bli ett fullt produktionsteam.

Följande svar utgår ifrån hur Sebring arbetat i konceptgruppen och i level design teamet.

Hur Massive Entertainment har förändrat Scrum för att passa verksamheten

- [32]. Till skillnad från traditionell Scrum har vi inte idag några "Scrum" grupper bestående av olika discipliner utan våra grupper är helt branschindelade (t.ex. programmerare för sig). Vi har heller inte några Scrum masters utan deras sysslor sköts av våra leads.

Arbetsätt

- [33]. Vi gör en grov planering av vilka bitar som skall ingå i projektet. Dessa större "features" får en enklare tid- och resursuppskattning. Uppskattningen tar hänsyn till om det skall göras av en programmerare, designer eller grafiker. För de närmaste tre veckorna gör vi en detaljplanering där varje feature bryts ned i ett antal tasks. En task består av 4-16 mantimmar. Färre timmar är ointressant och större antal gör uppskattningen för luddig. Varje team ansvarar för att ta fram och uppskatta tidsåtgång för sina egna uppgifter (tasks).
- [34]. Varje morgon hålls ett stående möte med teamet där alla får svara på tre frågor:
Vad gjorde jag i går?
Vad skall jag göra idag?
Finns det någonting som hindrar mig från att utföra mina uppgifter?
- [35]. Varje dag uppdaterar teamet hur många timmar som är kvar på sina uppgifter och om tasks är klara skickas de på testning. Som projektledare sammanställer Sebring dessa

varje dag för att få fram ett "burn down chart" som visar hur mycket de har kvar att arbeta denna tre veckors period samt i vilken hastighet teamet arbetar.

- [36]. Vid slutet av varje period visar teamet upp vad som blivit färdigt och resultatet utvärderas. Under tiden prioriteras features om vid behov. Markerar dem som färdiga eller att de behöver mer arbete för att nå Massives kvalitetsnivå. Slutligen planeras nästa treveckors period. Ibland itereras vissa features för att de inte blivit tillräckligt "roliga" eller färdiga.
- [37]. Vid sidan av denna typ av planering har de även vissa fasta datum som kommer av diverse anledningar. Till exempel tider som de måste vara färdiga med vissa bitar för att hinna med. Detta kan till exempel röra sig om röstinspelningar som görs i USA och behöver bokas långt i för tid.

Hur projektledaren påverkas av metoden och vilka fördelar man ser med Scrum

- [38]. Metoden ger Sebring möjlighet att fokusera mer på kvaliteten i projektet. Som spelprojekt har de ett speciellt krav – det skall vara kul. Även om en feature är inne och buggfri i spelet är detta inte en garanti för att den är klar. Vid varje build finns möjlighet att iterera processen. Dock kostar iterationer och man blir tvungen att stryka lägre prioriterade features.

På vilket sätt projektledaren får stöd från metoden

- [39]. Genom de dagliga uppdateringarna av tasks kan Sebring se hur mycket kvar som teamet tror de har kvar just nu, vilket även till en viss del tar hänsyn till kvalitet på tasks. Genom de dagliga morgonmötena med teamet kan Sebring snabbt åtgärda de problem som annars riskerar att bromsa ned hastigheten.
- [40]. Det är tufft att lära folk att uppskatta den tid det tar för en task att bli klar. Även om de fokuserar på "time remaining" till skillnad från en uppgifts "duration"
- [41]. Sebring upplever det som relativt svårt att visa projektets status gentemot externa kunder. Detta gäller speciellt kunder och intressenter som av en eller flera anledningar inte tar sig tiden att själva utvärdera den senaste tillgängliga builden.

Vad som tar upp den mesta av tiden

- [42]. Detta beror enormt stor del på vilken fas projektet befinner sig. Men den genomgående röda tråden är att kommunicera med teamet och se till att deras kortsiktiga och långsiktiga problem blir lösta. Det kan vara att se till att kommunikationen mellan två personer fungerar, att beroenden mellan team blir lösta.

De viktigaste egenskaperna för projektledaren anser Sebring vara:

- [43]. Prestigelöshet - gruppen går före den personliga "kolla vilket ledarskap jag har här"

[44]. Handlingskraftig

[45]. God Lyssnare - mycket viktigt att lyssna på gruppen för att få reda på vad som sker eftersom traditionell tidsrapportering inte finns

[46]. Coachande inställning - gruppen löser problem och det är projektledarens roll att se till att teamet klarar av sina uppgifter. Kräver många gånger individuell coaching för att uppnå full potential och förståelse för agile development

[47]. Sinne för detaljer

Vad krävs för att uppnå ett lyckat projekt

[48]. Att vara fokuserad på de problem som kommer upp under möten med teamet. Att även utan en detaljerad plan klara av att se framåt och arbeta pro aktivt.

Anställning av projektledare

[49]. Att anställa en projektledare som passar till arbetskulturen på företaget. Det finns många olika projektledarpersonligheter och lika många olika typer av ledare. Vid en anställning gäller det att utforska den tilltänkta projektledarens personlighet och ledarskapsstil lika noggrant som han/hennes erfarenhet av att leda projekt.

Appium AB

Ulf Atles – 27 Mars 2007

Företagsinformation och information om den vi har intervjuat.

[50]. Appium är sedan 1992 aktiva inom telekombranschen och är idag en ledande leverantör av applikationsplattformar. Huvudkontoret återfinns i Malmö, med försäljningskontor på olika platser i Europa, USA och Asien. Ulf Atles är ansvarig för Products, med en grupp på 20 personer, på Appium och är även certifierad Scrum Master.

Hur Appium har förändrat Scrum för att passa verksamheten

[51]. Då Appium är ett produktbolag är beställare och produktägare i projekten interna. Man har olika kundprojekt men bara ett produktprojekt. Det är samma övergripande produkt som förädlas hela tiden men krav från olika kunder kommer in i sprinten.

Arbetsätt

[52]. Appium arbetar med Scrum i grunden men metoden har anpassats för att passa företagets verklighet och förutsättningar. Enligt Atles fungerar Scrum bäst för känd teknik och enkla krav, då klarar man leverablerna vid varje sprintslut och kodstoppet genererar ingen större frustration. Man arbetar mycket med API (Application Programming Interface) och det är då svårt att följa Scrums krav på demoklara leverabler i slutet på varje sprint. När man försökte med en tvåveckors sprint upplevde utvecklarna att de hamnade i en vattenfallsliknande process. Kodstoppet kom för tidigt och därefter skulle tid läggas på att skapa en demo. Det blev för lite väsentlig kodning och för mycket planering och "administration" för att kunna komma framåt på så kort tid.

[53]. För att vara fortsatt flexibel mot produktägare och även få snabbare respons från produktägare och beställare behåller man arbetet i en två veckor lång sprint, men man har släppt kravet på demoklara leverabler efter varje sprint. Appium har ofta 7-8, ibland fler, iterationer innan release. En fyra veckor lång sprint skulle vara för svår att arbeta efter då man redan idag har problem att lista kraven för de kommande två veckorna.

[54]. En iteration startar med en titt på vad som redan är känt samt marknadskraven, för att få en initial kravlista. Ibland går inte en kravlista att upprätta, tekniken är okänd, utan det enda som kan bestämmas är att en förstudie måste genomföras. Därefter vet man vad som ska med i kravlistan. Cyklerna växlar i det läget med uppgiften. Tids estimeringar som finns som verktyg i Scrum används inte då heller. När projektet väl kommer igång håller man sprinttiden och kollar varannan vecka upp "var är vi någonstans?" samt "hur många iterationer har vi kvar på ett ungefär?". I projektets

slutskede, och tidpunkten för release närmar sig, formaliserar man processen och kör hårdare sprintar. Man lägger ibland in Gantt-scheman.

Vilka influenser har man från andra metoder och vilka fördelar har man sett med andra metoder

[55]. Beroende på säkerheten i hur man ska gå framåt i projektet kan man säga att företaget växlar från XP till RUP liknande processer, med Scrum i botten. I början XP-inspirerat med kortläggning där man tar ett kort i taget för att senare ta till sig fördelarna med RUP där det formella med en hårdare styrning behövs inför en release.

Hur projektledaren påverkas av metoden och vilka fördelar man ser med Scrum

[56]. Fördelarna man ser med att arbeta med Scrum är att alla arbetar parallellt och alla innehar tydliga roller. När något står att det är klart så är det verkligen klart, även all nödvändig dokumentation.

[57]. Projektledaren och övriga i teamet påverkas mycket av metoden. Även om man ska förhålla sig flexibel till den metod man valt, och inte vara statisk, så ska man hålla sig till den modell eller metod man valt att arbeta efter. Det är av högsta vikt att det är metoden som anpassas till företagets verklighet och inte tvärtom. Skulle man passa in företagets verklighet i metoden skulle det bara leda till frustration.

På vilket sätt projektledaren får stöd från metoden

[58]. Projektledaren får stöd från Scrum för att den är så tydlig i uppdelningen mellan rollerna. Förstår alla processen, vilket är en grundförutsättning, så fungerar det. Motsatsen skulle vara att man likt det traditionella tror att projektledaren vet allt, kan svara på allt, har mandat att göra allt osv. Det går inte att fråga om exakt release i ett agilt projekt. På nivåerna prioritering och beslut stödjer Scrum projektledaren bra, men det finns inga engineering practises.

Vad som tar upp den mesta av tiden

[59]. Projektledaren arbetar på två nivåer: dels övergripande på business och sprintnivå där kraven från olika kunder och säljare behandlas, dels på lågnivå med själva teamet som arbetar mer på beskrivningsnivå och bara har det arbete som står på listan att prioritera. För projektledaren gäller det att sammanjämka alla dessa resurser. Det som upptar den mesta av tiden är att kolla status på projektet, checka av mot förändringar, prioritera rätt och se till att teamet gör det viktigaste varje dag.

Lite allmänt spekulerande om skillnader

[60]. Traditionellt har projektledaren resultatansvar vilket leder till en orimlig press som går ut över utvecklarna. Det sker för mycket saker som inte går att råda över. I Scrum ger de tydliga rollerna en annan ansvarsfördelning, resultatansvaret ligger högre upp, Scrum Masterr sätter mer ansvar på teamet att planera och förbinda sig till projektet.

Det är på så vis en skönare uppgift att vara Scrum Master än traditionell projektledare. Projektledaren i ett agilt projekt har snarare som jobb att vara trollkonstnär och kunna parera.

- [61]. Kvaliteten på produkten är upp till produktägare och ledning. Projektledaren meddelar bara vilka risker som finns med olika beslut. Till exempel "ber ni oss göra en release om två sprintar så händer detta..", "beordrar övertid gör ni med följande konsekvenser", det är inte projektledaren som gör det.
- [62]. En väldigt viktig person enligt Atles är produktchefen. Någon som har övergripande ansvar för produkten och kan vara den som sorterar kraven från säljare så att inte den som skriker högst får in mest i kommande sprint.

Viktiga egenskaper i projektledarrollen

- [63]. Arbetet som projektledare i agila projekt kräver att man är ganska hård och kan tydliggöra roller. Man behöver se till att teamet trivs och vara pro aktiv gentemot övriga intressenter och inblandade i projektet. Rollen som ledare är mer social än tidigare, man har mindre strikta regler att följa.

De viktigaste egenskaperna för projektledaren anser Atles vara:

- [64]. Diplomati
- [65]. Social och kunna fungera som bollplank
- [66]. Drivande
- [67]. Erfarenhet
- [68]. Styrka och integritet att säga nej

Anställning av projektledare

- [69]. Appium anställer i stort sett bara erfarna projektledare. Det är egenskaperna som kommer av erfarenhet som Appium söker och det är inget man kan läsa sig till i en bok. Har man arbetat med andra projekt tidigare, i t.ex. RUP, kan man ta till sig Appiums sätt att arbeta.

Vad krävs för att uppnå ett lyckat projekt

- [70]. Projekt har i Atles mening aldrig fungerat utan är något av en illusion. Därför kan det vara svårt att diskutera vad som krävs för att ett projekt ska uppnå ett lyckat resultat. Man har alltid stött på problem vid projekt, skillnaden är att man tidigare skötte dem under ytan och all dokumentation såg fin ut. Men man följde sällan den dokumenterade arbetsmetoden.

Citerus

Mikael Lundgren – 26 april, 2007

Information om företaget och respondent

[71]. Citerus AB grundades i Uppsala 1996 och är Sveriges ledande konsultbolag inom lätttrörlig mjukvaruutveckling. De håller i utveckling åt kund (bland annat Tre, Lantmäteriet osv.) kan anordna mentorskap, samt utbildningar. Mikael Lundgren arbetar idag som konsult på Citerus och är en av endast två personer i Sverige som innehar rätten att certifiera Scrum Masters.

Hur man kan tänka sig att förändra metoden

[72]. Lundgren har erfarenhet av andra metoder sen tidigare. Det finns element i andra metoder, där Scrum har öppna luckor, som gör att han kan tänka sig att ta in dem i Scrum. Det handlar mycket om Lean Production, från XP är det testdriven utveckling och parprogrammering, från RUP kan man ta in visst dokumentstöd till exempel om man gillar att arbeta med use cases. Vill man presentera resultat på fler sätt så inför ett Gantt-schema.

Hur projektledaren påverkas av metoden och vilka fördelar och nackdelar man ser med Scrum

[73]. Systemutvecklingsmetoden påverkar arbetet som Scrum Master väldigt mycket. Metoden ger ett antal tumregler att följa, 15 minuters möte varje dag osv. Vid dessa möten är det SMs uppgift att sniffa upp eventuella problem och hinder, man jobbar mot hela verksamheten och man jobbar ständigt med förändring.

[74]. Scrum ställer höga och annorlunda krav på ledaren jämfört med andra metoder. Man måste som ledare hitta egna vägar. Men det är med flit man lämnat en del öppet. En jämförelse görs med XP som hade ganska mycket "tunga regler" men som blev baktung och företagen misslyckades med att arbeta med metoden.

[75]. En lucka, men som är medveten, är vid införandet av Scrum. Det handlar om förändring så när företagen ska införa Scrum så kommer det upp en massa frågor kring den befintliga verksamheten som: Varför gör vi det här på det här viset? Man hittar en massa flaskhalsar som det inte finns några direkta svar eller lösningar på vilket kan kännas jobbigt.

Allmänt spekulerande om skillnader

[76]. Man kallar det för Scrum Master inom Scrum för att betona att det inte är fråga om en vanlig projektledare, de två är väsensskilda. Som Scrum Master är man en coach och inte en projektledare i traditionell mening. Man kan jämföra med Bengan, den gamle handbollsledaren. Han är en modern ledare och på många sätt den första

Scrummästaren i Sverige. Tillsammans med laget så lade han upp riktlinjer som följde verksamhetsreglerna men det var laget som löste själva uppgiften. Bengan bokade hotellrum osv.

- [77]. Lite liknande är det med ett Scrumteam; Scrum Master hjälper teamet att bli bättre men det är teamet som löser problemet, planerar och följer upp sitt eget arbete.
- [78]. SM har lite av en PR-roll och är ansvarig för att marknadsföra sitt projekt, få produktägare och andra att förstå för de lever i en lite annan värld. En värld full av sekventiellt tänkande som strider lite mot Scrums grundtankar.
- [79]. Det klassiska är att räkna hem en affär när kontraktet är skrivet. Det Citerus gärna vill är istället att räkna hem det först när kunden är nöjd. Det är inte alltid kunden vet från start vad som gör den nöjd.

På vilket sätt projektledaren får stöd från metoden och var saknas det stöd

- [80]. Scrum stödjer projektledaren i sitt arbete med ganska mycket formalism som gör det möjligt att mäta och följa upp dagligen. Scrum är en empirisk process, d.v.s. den utvärderas ständigt för att resultatet ska bli bra. Motsatsen är en traditionell definierad process där man med samma input förväntas få samma output varje gång, men det är inte säkert att det är det rätta. Det formella i Scrum är de dagliga mötena som ger en oerhört viktig statuscheck, burn down, mm. Scrum ger i många fall mycket bättre uppdaterad information än vad något verktyg i de traditionella metoderna kan ge, du kan få ut mer mätetal direkt om du vill.
- [81]. Då det är Scrum Masters ansvar att rapportera fel, kostnadsineffektiviteter och andra saker som organisationen "gör fel" (det finns effektivare sätt) så hjälper inte Scrum Scrum Master att behålla sitt jobb. Är det fel personer i ledningen så kan en Scrum Master mycket väl förlora sitt jobb om han kommer med för många påståenden om ineffektivitet.

Vad som tar upp mycket tid

- [82]. Det som tar upp mycket tid är att lösa små vardagliga problem, se till så att människor som borde prata med varandra faktiskt också gör det. "Nä, men han har inte hört av sig" eller "Den och den har inte mailat".

De viktigaste egenskaperna för projektledaren anser Lundgren vara:

- [83]. Lyhördhet
- [84]. Förmåga att ta människor
- [85]. Kreativitet
- [86]. Mod
- [87]. Prestigelöshet

Vad som krävs för att uppnå ett lyckat resultat

- [88]. Det som krävs av Scrum Master för ett lyckat resultat är att kunna överblicka verksamheten och jämkta olika åsikter och viljor. Scrum är bra för det är den enda metod som går att använda för hela verksamheten. Alla kan förstå den oavsett position i organisationen. Samma ppt-presentation kan användas överallt.
- [89]. Lättillgängligheten och att det fungerar för en hel verksamhet är två av orsakerna till att Scrum har blivit den vanligaste utav de agila metoderna i Sverige.

Anställning av en projektledare

- [90]. För att vara aktuell som Scrum Master så ska man besitta coachegenskaper. Som person måste man vara villig att omvärdera sitt ledarskap, vara villig att förändra och se alternativa lösningar på problem. Man måste kunna prata med både tekniker och verksamhetsledare. Man är ansiktet utåt för projektet.
- [91]. Man har nytta av en projektledarutbildning men inte personer som har en vilja att styra hårdare.
- [92]. Gamla projektledarrävar är svåra att utbilda till Scrum Masters då de är vana vid att säga vad alla ska göra. Scrum använder en plattare struktur och det är teamet som arbetar med plan och uppföljning vilket känns som ett maktfråntagande.

14 Bilaga D: Agila Manifestet & Principer

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

© 2001, the above authors
this declaration may be freely copied in any form,
but only in its entirety through this notice.

Agile manifesto (2001).

Principles behind the Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Principles (2001).

15 Källförteckning

Agile Alliance (2006). *What Is Agile Software Development?*

Tillgänglig:

<http://www.agilealliance.org/show/2> (2007-05-28)

Agilemanifesto (2001). *Agile Manifesto*

Tillgänglig:

<http://agilemanifesto.org/> (2007-02-19)

Agile Principles (2001). *Principles behind the Agile Manifesto*

Tillgänglig:

<http://agilemanifesto.org/principles.html> (2007-02-19)

Avison, D., & Fitzgerald, G. (2002). *Information Systems Development: Methodologies, Techniques and Tools* 4th ed. London: McGraw-Hill Education.

Avison, D., & Fitzgerald, G. (2006). *Information Systems Development: Methodologies, Techniques and Tools* 4th ed. London: McGraw-Hill Education.

Beynon-Davies, P., (2002). *Information Systems – An introduction to informatics in organisations*. New York (N.Y): Palgrave.

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

Briner, W. & Geddes, M. & Hastings, C. (1999). *Projektledaren*. Smedjebacken: Fälth & Hässler.

Bryman, A. (2001). *Samhällsvetenskapliga metoder*. Malmö: Liber ekonomi.

Charvat, J. P. (2002). *Heavyweight vs. lightweight methodologies: Key strategies for development*

Tillgänglig:

<http://builder.com.com/5100-6315-5035285.html> (2007-03-13)

Christensen, S., & Kreiner, K. (1997). *Projektledning – Att leda och lära i en ofullkomlig värld*. Köpenhamn: Jurist- og Ekonomiförbundets Forlag.

Citerus (2006) *Lättrörlig utveckling - struktur utan byråkrati*

Tillgänglig:

<http://www.citerus.se/kunskap/omlatrorligutveckling.4.a939951052f77843580006986.html> (2007-05-10)

Cockburn, A. (2002). *Agile Software Development*. Boston: Addison-Wesley.

Cockburn, A. (2004). *Crystal clear : a human-powered methodology for small teams*. Boston: Addison-Wesley

Controlchaos (2007). *Scrum how it works*

Tillgänglig:

<http://www.controlchaos.com/about/how.php> (2007-03-20)

Fitzgerald, B., Russo, N. L., Stolterman, E. (2002). *Information systems development: Methods in action*. Maidenhead: McGraw-Hill Education.

Fowler, M. (2000). *The New Methodology*

Tillgänglig:

<http://www.martinfowler.com/articles/newMethodology.html> (2007-02-01)

Fowler, M. (2002). *Agile Manifesto*

Tillgänglig:

<http://www.martinfowler.com/articles/newMethodology.html#AgileManifesto> (2007-02-01)

Holme, I. M., Solvang, B., K., (1997). *Forskningsmetodik – om kvalitativa och kvantitativa metoder*, Lund: Studentlitteratur.

IBM (2005). *RUP in the dialogue with Scrum*.

Tillgänglig:

<http://www-128.ibm.com/developerworks/rational/library/feb05/krebs/> (2007-05-04)

IDG (2007) *Språkwebb*

Tillgänglig:

<http://cstjanster.idg.se/sprakwebben/ord.asp?ord=leverabel> (2007-05-24)

Jeffries, R. (2001). *Extreme Programming Installed*: Harlow UK: Pearson Education.

Koch, A., (2005). *Agile Software Development : Evaluating the Methods for Your Organization*. Norwood, MA, USA: Artech House Inc.

Kvale, S., (1997). *Den kvalitativa forskningsintervjun*. Lund: Studentlitteratur

Larsson, A-C. & Larsson, R. (2005). *Projektledarrollen i praktiken – Några grunder i att leda och delta i projekt*. Uppsala: Uppsala Publishing House AB, Halmstad Bulls Tryckeri.

Löow, M. (1999). *Att leda och arbeta i projekt*. Malmö: Liber AB, Kristianstads boktryckeri AB.

Mountain Goat (2005). *Software Scrum Overview*

Tillgänglig:

<http://mountaingoatsoftware.com/Scrum/Scrumteam.php> (2006-10-09)

Norrgren, F. (2006). *Projektledning*

Tillgänglig:

<http://www.chalmers.se/sections/forskning/professorer/beskrivningar/flemming-norrgren>
(2007-03-13)

Schuh, P., (2005). *Integrating agile development in the real world*. Hingham, MA: Charles River Media.

Schwaber, K. (2004). *Scrum methodology process guidance*

Tillgänglig:

<http://Scrumforteamssystem.com/ProcessGuidance/> (2006-10-10).

Softhouse (2006). *Scrum på fem minuter*

PDF-Format

Tillgänglig:

http://www.softhouse.se/Uploades/Scrum_broschyr.pdf (2007-05-06)