

Magisteruppsats i Informatik
Thesis work/Master thesis in Informatics

REPORT NO. 2008:007
ISSN: 1651-4769

Department of Applied Information Technology or
Department of Computer Science

Öka dina chanser att lyckas med Scrum

Ett ramverk för inkrementellt införande av Scrum i en organisation

Increase your chances of success with Scrum

A framework for incremental adoption of Scrum in an organisation

CARLJOHAN CARLSSON
Handledare: Kjell Engström



Abstrakt

Scrum är en projektstyrningsmetod som har blivit mycket populär de senaste åren. Scrum är en förhållandevis tunn metod och fokuserar på ett starkt iterativt moment. Med iterationerna som grund försöker Scrum, till skillnad från traditionella planbaserade metoder, att anpassa sig till rådande omständigheter istället för att försöka förutse dem. Även om Scrum beskriver sig som en lätthanterlig och tunn projektstyrningsmetod innebär ett införande av Scrum i en organisation en förändring och förändringar kan medföra problem.

I syfte att minska friktionen vid införandet av Scrum har jag genom litteraturstudier och med stöd av en fallstudie på Stendahls.net AB arbetat fram ett ramverk som kan användas som underlag vid införandet av Scrum i en organisation.

Ramverket visar på risker och vilka positiva effekter de olika delarna i Scrum innebär.

Innehåll

| | |
|---------------------------------------|-----------|
| ABSTRAKT | 2 |
| INNEHÅLL..... | 3 |
| 1. INTRODUKTION | 4 |
| BAKGRUND..... | 4 |
| SYFTE OCH FRÅGESTÄLLNING | 6 |
| PRAKTISK OCH TEORETISK RELEVANS | 6 |
| DISPOSITION | 7 |
| 2. METOD | 8 |
| ANGREPPSSÄTT | 8 |
| TILLVÄGAGÅNGSSÄTT | 9 |
| LITTERATURGRANSKNING | 10 |
| KÄLLKRITIK | 10 |
| INSAMLING AV DATA..... | 10 |
| METODKRITIK | 11 |
| 3. TEORETISKT RAMVERK..... | 13 |
| BEGREPP OCH DEFINITIONER | 13 |
| AGILA METODER..... | 14 |
| SCRUM | 16 |
| 4. FALLSTUDIEN | 28 |
| PROJEKTET | 29 |
| 5. RESULTAT | 34 |
| RESPONDENTPRESENTATION | 34 |
| ÖVERGRIPANDE..... | 35 |
| FLÖDE | 35 |
| ROLLER | 37 |
| ARTEFAKTER | 38 |
| RISKER VID INFÖRANDET AV SCRUM..... | 41 |
| SLUTSATS..... | 44 |
| 6. REFERENSER..... | 45 |
| 7. BILAGOR | 47 |
| INTERVJUUNDERLAG..... | 47 |
| SCRUM..... | 48 |

1. Introduktion

Bakgrund

Under 1990-talet skapades, oberoende av varandra, ett antal nya systemutvecklingsmetoder. Gemensamt för dessa metoder var att de var s.k. agila, lätttrörliga. Dessa nya metoder var en motreaktion mot de metoder man uppfattade som alltför byråkratiska och omfattande (www.agilesweden.org). Enligt Fowler (2005) är uppkomsten av det agila tänkandet och agila metoder en av de största förändringar i hur systemutvecklingsarbete bedrivs i organisationer de senaste åren.

Scrum är en agil projektstyrningsmetod. Termen Scrum kommer från en studie av Takeuchi och Nonaka (1986) som blev publicerad i Harvard Business Review. I studien uppmärksammar Takeuchi och Nonaka att projekt som använder små, tvärfunktionella projektgrupper historiskt producerar bäst resultat. De beskriver vidare att dessa högpresterande projektgruppen närmast kan beskrivas som Scrum-formationen i Rugby. Jeff Sutherland använde studien som bas när han utvecklade en systemutvecklingsprocess hos Easel Corporation 1993. Han använde också deras Scrum-analogi som namn för processen som helhet. Ken Schwaber formaliserade senare processen i en artikel publicerad 1995 (Sutherland 2006).

Scrum fokuserar, liksom andra agila metoder, på ett starkt iterativt moment. Scrum bryter ned utvecklingsprojektet i 30-dagars iterationer, kallade sprints, och ökar insikten i projektet med dagliga möten och uppföljningar (Fowler, 2005; Deemer & Benefield, 2006; Sutherland 2006). Scrum strävar också efter att prioritera arbete efter affärsvärde, vilket leder till att mer affärsvärde blir levererat tidigt i projektet (Sutherland 2006).

Stendahls.net AB är ett IT-företag med 25 anställda. Stendahls.net AB befinner sig just nu i en mycket expansiv fas. Med detta innebär att det finns en vilja att strukturera systemutvecklingsarbetet och införa en gemensam process för företagets alla utvecklingsgrupper. Försök har gjorts tidigare med tillämpning av olika utvecklingsmetoder såsom t.ex. RUP. Dessa har dock misslyckats eller lagts på is av olika anledningar. Framförallt uppfattades RUP som alltför rigid och byråkratisk av både utvecklarna och kunderna.

Stendahls.net vill nu således utvärdera en mer lättviktig process och det stora intresset bland utvecklarna har gjort att valet har fallit på Scrum.

Införandet av Scrum eller vilken annan godtycklig systemutvecklingsmetod innebär förändring för en organisation. (Little & Speyd, 2007) Förändringar inom en organisation är svårt och möter ofta motstånd. Detta motstånd är en rationell reaktion och kan ha flera orsaker t.ex rädsla för det okända och ändrade maktförhållande. (Jacobsen & Thorsvik, 1995). Detta motstånd kan leda till att införandet av metoden misslyckas.

I flera artiklar (Streibek, 2006; Benefield, 2007) beskriver företag lyckade tillämpningar av Scrum med ett inkrementellt angreppssätt. Med ett inkrementellt införande kan projektgruppen och organisationen gradvis vänja sig vid de nya rutiner och förändringar

som ett införande av Scrum innebär. Jacobsen & Thorsvik (1995) beskriver inkrementell utveckling såsom att utvecklingen sker i små, sammanhängande steg. Ur det många små förändringarna växer en mer omfattande förändring fram. En inkrementell utveckling medför att förändringen verkar mer som en naturlig utveckling än som en plötslig förändring. Little och Speyd (2007) beskriver detta angreppssätt som en mycket viktig del i införandet av agila metoder i en organisation. Little och Speyd menar även att det är viktigt att börja med de delar som innebär minst risk att misslyckas och sedan utifrån det gå vidare i införandet.

I syfte att göra Scrum enklare att införa i utvecklingsgrupper hos framförallt Stendahls.net AB försöker den här studien beskriva ett ramverk som kan användas vid inkrementellt införande av Scrum i en organisation.

Syfte och frågeställning

Syftet med denna uppsats är belysa problematiken kring införandet av Scrum i en organisation. Detta kommer att göras med utgångspunkten att införandet sker inkrementell där Scrums delar införs bit för bit.

Med studien vill jag visa på vilka positiva effekter Scrum delar har på en projektgrupp men även vilka risker de innebär. Detta ramverk kan användas som ett stöd och bakgrundsinformation till organisationer som är i begrepp att införa Scrum.

Frågeställningen för uppsatsen blir därmed:

Vilka positiva effekter har de olika delarna i Scrum på en projektgrupp och utgör några av dessa delar en högre risk att misslyckas i sitt införande?

Den följs av frågan:

Vilka delar inom Scrum lämpar sig att självständigt införas i en projektgrupp?

Genom att svara på de båda frågorna kommer jag kunna visa på ett ramverk med vilka delar, eller grupper av delar, som med faktorerna positiva effekter och risk för misslyckande kan ligga till grund för ett inkrementellt införande av Scrum i en organisation.

Praktisk och teoretisk relevans

Uppsatsen är av praktiskt relevans då den kan fungera som bakgrundsinformation och tillföra kunskap till företag och organisationer som är i begrepp att införa Scrum som projektstyrningsmetod.

Ytterligare hoppas jag på att studien skall resultera i ett bidrag till den fortsatta forskningen kring Scrum och agila metoder och i synnerhet införandet av dem i en organisation.

Agila metoder och Scrum är mycket populära systemutvecklingsmetoder. Dock anser jag att det inom litteraturen saknas studier där införandet av Scrum och faktiskt som en organisationsförändring och bör implementeras försiktigt.

I avseende av det lilla utbudet av litteratur inom detta område kommer studien utgöra en viktig beståndsdel i den framtida forskningen.

Uppsatsen kan även anses vara ett komplement till studierna: ”Ssh! We are adding a process...” (Streibeck, M, 2006) och “CIO Playbook for adopting Scrum” (Leffingwell D., Smits H., 2005).

Disposition

Kapitel 1: Introduktion

I denna delen beskriver jag primärt vad studien handlar om. Jag beskriver syftet med studien, dess avgränsningar och målgrupp.

Kapitel 2: Metod

Kapitlet metod beskriver och motiverar hur studien är genomförd. Vilket vetenskapligt tillvägagångssätt jag valt för att angripa problemet

Kapitel 3: Teoretisk ramverk

I detta kapitlet presenteras och definieras agila metoder och Scrum. Syftet är att ge läsaren nödvändig bakgrundsinformation och förståelse om vad Scrum och agila metoder är.

Kapitel 4: Fallstudien

Denna delen syftar till att beskriva förutsättningarna och genomförandet av fallstudien som ligger till grund för denna studien.

Kapitel 5: Resultat och analys

I det fjärde kapitlet redovisar jag resultatet av de fyra intervjuerna jag genomfört på Stendahls.net. I detta kapitlet presenteras även det slutgiltiga ramverket.

2. Metod

I detta kapitlet beskrivs i detalj det totala tillvägagångssättet vid observationerna. Syftet med den detaljerade beskrivning är replikation och evaluering. Med replikation menas att en utomstående person skall kunna upprepa studien med exakt samma resultat. Evaluering innebär evaluering av det empiriska förfarandet och metodens bärkraft för de slutsatser och resultat som diskuteras senare.

Angreppssätt

I allmänhet börjar utforskningen av angreppssätt i studiens förhållande till positivism. Positivismen menar att all verklig kunskap vi inhämtar är sprungen ur observationer eller erfarenheter av verkliga fenomen. Detta innebär att positivistisk forskning ämnar visa fakta som inte kan bli ifrågasatt. Vidare menar positivisterna att all fakta som finns inte innefattar några sociala värden eller koder och är tidlösa. (Cornford & Smithson, 1996). Sådana fakta, menar Cornford & Smithson, kan uttrycka observerade återkommande fenomen och utifrån dessa föreslå eller validera hypoteser. Ett anti-positivistiskt ståndpunkt beskriver snarare att fakta och sociala värden och koder är sammanflätade och oskiljaktiga. Det menar vidare på att all kunskap, även vetenskaplig, är socialt konstruerad och anpassad till det sammanhang där de är härledda ur (Cornford & Smithson, 1996).

Denna uppsats syftar till, som beskrivet, att nå en högre förståelse för tillämpning av Scrum samt svara på frågan om vilka positiva effekter och risker de olika delarna i Scrum innebär för en projektgrupp.

Med detta syfte har ett induktivt arbetssätt passat studien. Ett induktivt arbetssätt innebär att forskaren samlar in kvalitativ data inom ramen av det fenomen som skall undersökas. Utifrån dessa data försöker forskaren skapa sig en bild av problemområdet som i sin tur skall leda till en eventuell slutsats.

Motsatsen till det induktiva arbetssättet är det deduktiva. Då utgår forskaren från en hypotes. Hypotesen testas sedan mot den insamlade datan där forskaren försöker hitta mönster och sammanhang för att utröna om hypotesen stämmer eller ej.

Denna uppsats tar sin empiriska grund i ett antal semistrukturerade intervjuer och observationer utförda under fallstudien. Med dessa intervjuer försöker jag belysa problematiken med delvis införa Scrum i en projektgrupp.

Detta leder till att jag kan klassificera min uppsats som i huvudsak kvalitativ.

Kvalitativa metoder

Kvalitativa metoder kännetecknas ofta av att resultat från början inte är förutbestämt. Man börjar istället i empirin, den insamlade datan, för att där efter formulera frågeställningar och hypoteser (Backman, 1998). Ett kvalitativt angreppssätt litar, till skillnad från ett kvantitativt angreppssätt, inte på nummer i syfte att beskriva ett fenomen. En kvalitativ ansats innebär enligt Cornford och Smithson som forskning som baseras på ord.

Backman beskriver den kvalitativa forskningsprocessen som en process som innehåller ett stort mått av flexibilitet och dynamik. Ofta kan olika moment interagera med varandra och pågå samtidigt. I korthet innebär den:

- Uppsatsen inleds med en fråga.
- Litteraturgranskning
- Val av analysenhet
- Problem/frågeställning
- Observation
- Analys
- Tolkning

Fallstudier

Den kvalitativa forskningen visar enligt Backman på en förkärlek till fallstudier som empirisk grund i studier. En fallstudie undersöker ett fenomen i sin realistiska miljö eller sin kontext. Svårigheterna uppstår när man skall avgöra vart gränserna i det observerade fenomenet.

Fallstudier anses passa särskilt bra där studieobjekten är komplexa. T.ex. när man försöker förstå eller förklara delar av en organisation.

Jag anser att införandet av en systemutvecklingsprocess i en organisation, med tillhörande tillvägagångssätt, värderingar och attityder, med råge uppnår en hög grad av komplexitet varav jag valt att beskriva fenomenet med hjälp av en fallstudie.

Backman skiljer också mellan explorativ och deskriptiva avsikter med fallstudien. Fallstudien i denna uppsats kan klassificeras som deskriptiv då jag *förklarar* hur Scrum implementeras hos Stendahls.net i syfte att kunna formulera en hypotes utifrån den insamlade datan.

Tillvägagångssätt

Val av analysenhet

I och med att jag de senaste två åren varit anställd hos Stendahls.net förföll det sig naturligt för mig att utföra min studie där. Framförallt för att Stendahls.net AB har varit i begrepp att införa Scrum som generell systemutvecklingsmetod men också för närheten till det fenomenet jag faktiskt skulle studera.

Valet av analysenhet började med att jag frågade den kundansvariga på Stendahls.net om det fanns något eller några projekt hösten 2007 som han ansåg vara särskilt tillämpliga för en fallstudie om Scrum. Intresset för Scrum visades sig stort på såväl utvecklarnivå som på ledningsnivå.

I syfte att minska riskerna för ett projekt som använde Scrum för första gången valdes ett projekt ut med följande kriterier:

- Liten utvecklingsgrupp (3-5 personer)
- Erfaren utvecklingsgrupp. Majoriteten har varit anställda hos Stendahls.net i över 5 år.

- Avgränsad, relativt stabil och tydlig kravspecifikation. Projektet har en tydlig målbild.

Litteraturgranskning

I syfte att skaffa mig lämplig bakgrundsinformation till detta arbete har jag genom litteraturstudier inhämtat information om systemutvecklingsmetoder, agila metoder och i synnerhet Scrum och införandet av Scrum i en organisation. Studierna har gett mig en bred överblick av området. Det litterära materialet har i huvudsak bestått av böcker om Scrum men också till stor del av artiklar, debattinlägg och forskningsrapporter.

Studierna har gett mig bra bakgrundsinformation som jag dels kunnat använda mig av i det teoretiska ramverket, men jag har även kunnat använda mig av den i utbildningen av projektmedlemmarna i egenskap av min roll som Scrummaster.

Scrum och agila metoder är ett relativt ungt ämne (jmf The Agile Manifesto skrevs 2001) och jag har därför haft vissa svårigheter att hitta större mängder tidigare akademisk forskning inom ämnet. Dock är agila metoder ett aktuellt ämne och det har funnits ett otal böcker och artiklar av varierande kvalitet att tillgå.

Källkritik

Ett varningens tecken skall höjas mot i synnerhet Scrum som är den metoden som nått längst i kommersialiseringsprocessen. En uppsjö av företag säljer idag olika produkter, Scrummaster certifieringar och tjänster rörande Scrum. Därav har jag märkt under arbetets gång att det ofta ligger kommersiella intressen bakom många publikationer. Många av dem pekar på svårigheter inom Scrum och erbjuder avslutande tjänster och certifieringar för att överkomma dessa. Denna problematik har varit svår att lösa och det enda hjälpmedlet jag haft är att helt subjektivt försöka utröna litteraturens opartiskhet.

Då Scrum är en relativt ny metod har det också varit svårt att hitta vetenskapligt förankrad information om Scrum. Beskrivningen av Scrum under det teoretiska ramverket skall därför endast betraktas som min tolkning av Scrum utifrån den granskade litteraturen.

Insamling av data

Enligt Gunnarsson (2006) finns det tre grundläggande metoder att samla in data, dokumentation, observationer och intervjuer. Jag kommer i min studie använda mig utav dessa tre tillvägagångsätt.

Dokumentation

Sekundär data i form av mötesprotokoll, planering- och designdokument kommer att användas i syfte att skapa en initial bild av fallstudien.

Observation

Under projektets gång har jag verkat som Scrummaster i projektet. Ur studiens perspektiv finns det två anledningar varför jag valt att göra så. Observationerna har främst dokumenterats i en dagbok.

Daily meetings

Möten som hålls varje dag på morgonen.

Retrospective meetings

Möten som hålls internt efter en sprint i syfte att utvärdera den.

Review meetings

Externt möte med kund och/eller product owner

Semistrukturerade djupintervjuer

Tillskillnad från t.ex frågeformulär är intervjuer som datainsamlingsteknik en metod som tillåter intervjuaren att penetrera ett ämne på djupet.

Intervjuer är vanligtvis klassificerade i termer av den underliggande strukturen. Från helt ostrukturerade intervjuer till helt strukturerade där intervjuaren har alla frågor förberedda och där de ställt i en förutbestämd ordning. Båda dessa extremer har sina fördelar och nackdelar. (Cornford & Smithson, 2001)

Jag har använt mig av en semi-strukturerad intervjuteknik. Med ett semi-strukturerat angreppssätt menar Cornford & Smithson att intervjuaren förbereder ett antal frågor inklusive ett antal nyckelfrågor för att använda som guide genom intervjun. Dock utan intention att följa den strikt.

Intervjuerna har tagit plats på Stendahls.net kontor under december månad. Intervjuerna dokumenterades med elektronisk inspelningsutrustning.

Urval av respondenter

Eftersom fallstudien endast rörde ett projekt med begränsat antal deltagare föreföll sig urvalet av respondenter naturligt.

Fakta om intervjuerna:

- Alla respondenter är svenskar och arbetar på Stendahls.net AB i Göteborg.
- Alla intervjuer varade ungefär 1 timma.
- Alla respondenter är mer eller mindre involverade i det studerade projektet.
- Totala antalet intervjuer var 5 st.

Metodkritik

Först och främst skulle mer material ha kunnats samlas in. Från fler intervjuer till att jag skulle ha behövt delta i fler projektmöten och vara allmänt mer aktiv inom projektet.

I min mening skulle också projektdeltagarna fått mer utbildning om Scrum. Under den här studien har projektdeltagarna endast fått en 2 timmars genomgång av Scrum. Detta är enligt min mening alldeles för lite för att förstå den verkliga innebörden av Scrums värderingar och principer. Till försvar kan sägas att många projektdeltagare självständigt sökt upp mer information och material angående Scrum.

De eventuella fördelarna med att inte veta för mycket om metoden skulle kunna vara att deltagarna då hade varit mer flexibla och anpassningsbara med metoden än att snarare följa metoden till punkt och pricka.

Min utgångspunkt som Scrummaster har varit att inte styra precis *hur* deltagarna gör Scrum, bara att dem faktiskt i någon utsträckning gör den. Att jag själv har varit

Scrummaster har självklart påverkat studien. Fördelen är att jag kunnat lära ut och förmedla Scrum till deltagarna och att jag kunnat observera projektet ”inifrån”. Nackdelarna är naturligtvis att jag, i viss utsträckning, har påverkat fenomenet jag studerat. I vilken utsträckning detta faktiskt har skett är omöjligt för både mig och projektdeltagarna att veta.

3. Teoretiskt ramverk

I detta kapitlet har jag för avsikt ge en grundläggande beskrivning om agila metoder och Scrum. Då framför allt om teorierna bakom Scrum, kritiken mot Scrum och införandet av Scrum i en organisation. Jag ger också en kort beskrivning om dimensionerna av förändring i en diskussion i syfte att läsaren en grundläggande förståelse.

Begrepp och definitioner

Scrum använder en mängd begrepp och definitioner för de olika delarna i metoden. Dessa uttryck är på engelska. För att förenkla har jag översatt dessa termer till svenska enligt följande ordlista:

| | |
|-------------------------------------|-----------------------|
| Flöde | |
| <i>Daily Scrum Meeting</i> | Dagligt möte |
| <i>Sprint Planning Meeting</i> | Planeringsmöte |
| <i>Sprint Retrospective Meeting</i> | Retrospektivt möte |
| <i>Sprint Review Meeting</i> | Granskningsmöte |
| | |
| Roller | |
| <i>Product Owner</i> | Produktägare |
| <i>Scrum Master</i> | Scrummaster |
| <i>The Team</i> | Utvecklingsgrupp |
| | |
| Dokument | |
| <i>Product backlog</i> | Produktspecifikation |
| <i>Sprint backlog</i> | Sprintspecifikation |
| <i>Release backlog</i> | Versionsspecifikation |
| <i>Burndownchart</i> | Burndowngraf |

Agila metoder

I februari 2001 samlades konsulter, processexperten och ledarna för dessa nya metoder under en helg i Snowbird, Utah. Under helgen tog "The Agile Manifesto" form.

Manifestet är signerat av samtliga ledarna och är ett symboliskt dokument som beskriver värderingar, principer och attityder kring "den nya metoden" som övergripande beskrivs som agil (lätttrölig). (agilemanifesto.org).

Fowler pekar på i huvudsak två viktiga faktorer i det agila tänkandet:

- **Agila metoder är anpassningsbara istället för förutsägbara.** Tidigare systemutvecklingsmetoder har, liksom de flesta ingenjörsmässiga metoder, försökt förutsäga och planera systemutvecklingsprocessen. Detta fungerar bra tills någonting under processen vill förändras. Det blir således naturligt för dessa metoder att motverka förändring under arbetets gång. Agila metoder välkomnar istället förändring och försöker anpassa sig till de rådande omständigheterna, även till den grad att de t.o.m. anpassar sig själva.
- **Agila metoder är fokuserade på människor istället för processer.** Om målet för ingenjörsmässiga metoder är att definiera en process som skall fungera oavsett vem som använder den anser agila metoder att den viktigaste faktorn är utvecklingsgruppen, eller människorna som faktiskt använder den. Processens ansvar blir då att *understödja* dem som faktiskt använder den.

Principerna bakom det agila synsättet är (fritt översatt från agilemanifesto.org):

1. Att göra kunden nöjd genom tidiga och regelbundna leveranser av värdeskapande programvara.
2. Anpassning till förändrade krav och förutsättningar är naturligt, även i ett sent skede; utnyttja förändring till kundens fördel.
3. Leverera användbar programvara ofta, helst med bara några veckors mellanrum.
4. Verksamhetskunniga och utvecklare arbetar tillsammans dagligen.
5. Självgående och ansvarstagande individer är den främsta framgångsfaktorn. Med nödvändigt stöd och förtroende kommer de att lösa uppgiften.
6. Kommunikation ansikte mot ansikte är det bästa sättet att förmedla information, både till och inom teamet.
7. Funktionalitet är det främsta måttet på framsteg.
8. Agile verkar för uthållighet; teamet skall kunna upprätthålla jämn arbetsbelastning så länge som behövs.

9. Kontinuerlig uppmärksamhet på teknisk elegans och bra design stärker anpassningsförmågan.
10. Enkelhet - konsten att göra rätt saker, varken mer eller mindre - är grundläggande.
11. Grupper som organiserar sig och sitt arbete själva, ger bäst problemförståelse, arkitektur och design.
12. Gruppen utvärderar och anpassar regelbundet sitt arbetssätt för att förbättra sin effektivitet.

Scrum

Scrum är en agil projektstyrningsmetod med övertygelsen om att de flesta systemutvecklingsmetoder har en felaktig uppfattning om systemutvecklingsprocessen.

Traditionella metoder arbetar enligt devisen att systemutvecklingsprocessen är väl förstådd och att den framgångsrikt går att planera, tidsuppskatta och genomföra. Alla misslyckade projekt, felaktiga system, missade deadlines och överskridna budgetar är enligt traditionella metoder ett bevis på att processen behöver vara mer rigid och följas av utvecklare, projektledare och kunder.

Scrum påstår däremot att systemutveckling är en komplicerad process vilket gör den svår, för att inte säga omöjlig, att planera och tidsuppskatta.

Empirisk process kontroll

Ingenjörsmässiga discipliner har alltid lagt stor tonvikt vid planering och design innan konstruktionsfasen. Utifrån dessa designdokument och ritningar kan man sedan härleda vad som behöver göras, hur saker passar ihop och vilka aktiviteter som är beroende av varandra. Design beslut, som hur belastningen på en bro skall hanteras, tas undertiden dessa dokument skapas. Dokumenten lämnas sedan över till en annan grupp (ibland till ett annat företag) för att verkställas. Om dokumenten verkligen motsvarar en komplett design kan konstruktionsgruppen utan problem bygga produkten. De kan faktiska bygga flera exemplar av produkten utan mer inblandning av de som faktiskt designade den.

Vad man ser här, enligt Fowler, är en separation av två olika aktiviteter. *Design* och *planering* som är svåra att förutsäga då de innefattar kreativa aktiviteter och *konstruktion* som är enklare att förutsäga. (Fowler)

Systemutvecklingsmetoder har genom tiderna alltid haft en kraftig slagsida åt detta ingenjörsmässiga angreppssätt. Först designdokument (i form av t.ex. UML-diagram) och av dessa dokument en konstruktionsplan. (Fowler)

Detta synsätt har dock på senare blivit kritiserat. J. Reeves menar t.ex att programmering som aktivitet inte kan klassas som en konstruktionsaktivitet utan snarare som en *design aktivitet*. Konstruktionsfasen enligt Reeves är brukandet av en kompilator och en länkare. Denna attityd mot systemutveckling och programmering får självklart stora konsekvenser inom ramen av ett ingenjörsmässigt angreppssätt.

På senare tid har det pågått en debatt rörande processkontroll inom systemutveckling. I ena läget finns Software Engineering Institute (SEI) med deras Capability Maturity Model (CMM). SEI och CMM betraktar systemutveckling som en definierad process.

Med en definierad process menas att varje aktivitet inom processen är förstådd. Samma samling av definierade ingångsparametrar kommer att resultera i exakt samma utgångsparametrar.

SEI menar att systemutvecklingsprocessen definieras genom gradvis och fortlöpande förbättring. Det är enligt SEI varje organisations ansvar att förbättra sin process genom att klättra på CMM-stegen.

I det andra läget finns en skara människor som inte tror att systemutvecklingsprocessen kan definieras i ett antal steg (däribland förespråkare för agila metoder).

Jeff Sutherland, en av grundarna av Scrum) ger sin åsikt om definierade plandrivna processer:

"Our attempts to use the Waterfall Methodology, an approach which assumes a fully constrained development environment, has been a fools errand. We have been embarked on a mission to do the mathematically impossible! It is not surprising that a third of development projects are canceled before completion and that those which are completed are, on the average, 189% over budget."

Ken Schwaber beskriver hur han lät en grupp processexperter vid DuPont inspektera flera olika systemutvecklingsprocesser:

"I have rarely provided a group with so much laughter. They were amazed and appalled that my industry, systems development, was trying to do its work using a completely inappropriate process control model. They said systems development had so much complexity and unpredictability that it had to be managed by a process control model they called 'empirical'. They said this was nothing new, and all complex processes that weren't completely understood required the empirical model."

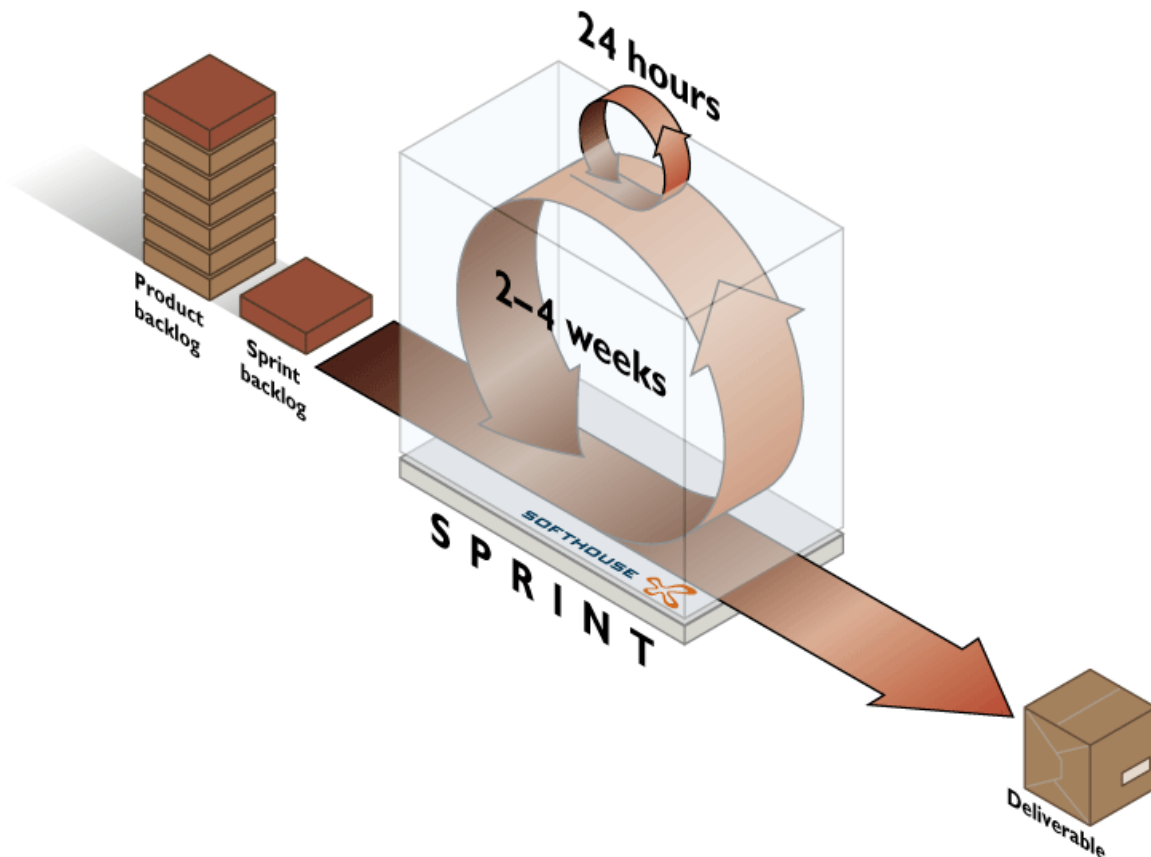
För att det skall vara möjligt att hantera en föränderlig process måste det finnas tillgång till empirisk data om processens status. Scrum tillvägagångsätt för insamling av denna data är frekventa observationer av projektet. Scrum säger sig erbjuda en hög grad av projekt transparens, dvs. lättillgänglig information om projektets verkliga status.

Iterativ utveckling - kontinuerligt leverens av affärsvärden

Enligt Scrum mäts inte ett projekts framåtskridande med rapporter eller projektdokumentation. Rapporter och otestad kod kan enligt Scrum innehålla mängder med felaktigheter som inte upptäcks. Däremot när man faktiskt sitter och arbetar med det verkliga systemet blir felaktigheterna uppenbara både i termer av buggar och framförallt i termer av missupfattade krav.

Enligt Scrum mäts projekts framgång med levererade affärsvärden.

Ett sådant förhållningssätt kräver en iterativ utvecklingsmodell. Iterativa utvecklingsmodeller är ingen ny ide utan har funnits länge under olika namn (inkrementell, evolutionär, spiral) (Fowler, 2005). Scrum tar dock det iterativa tänket en bit längre genom att hålla de iterativa perioderna korta (2-4 veckor) och lägger till förutsättningen att dessa alltid skall leverera testbara affärsvärden (delar av fungerande system).



Figur 3-1: Scrum flow (Softhouse, 2006)

Kommunikation och information

Dagligt möte

Varje dag möts utvecklingsgruppen under 15 minuter i ett dagligt möte. Mötet syftar till att synkronisera informationen bland projektdeltagarna. Varje person i utvecklingsgrupper svarar under mötet på tre frågor:

- Vad har du gjort i det här projektet sen senaste mötet?
- Vad kommer du att göra tills nästa möte?
- Finns det några hinder som motverkar dig att uppnå sprintens mål?

Det är viktigt att inte diskussioner halkar in på tekniska detaljer angående projektet. Fokus skall under mötet vara på sprint backloggen och på utvecklingsgruppens framsteg. Yip (2007) pekar också på ett ytterligare hjälpmedel för att verkligen försöka hålla mötet under 15 minuter, nämligen att samtliga mötesdeltagare står upp under mötet.

Planeringsmöte

Ett sprint planning möte består i realiteten utav två möten. I det första mötet där utvecklingsgruppen möter Scrummastern, produktägaren och användare för att

bestämma *vad* som skall göras i följande sprint. Detta görs genom att en delmängd ur produktspecifikationen väljs ut för att implementeras under nästkommande sprint (se Figur 3-1: Scrum flow (Softhouse, 2006))

I det andra mötet, där endast utvecklingsgruppen är närvarande, bestäms *hur* sakerna skall implementeras. I detta möte bryts aktiviteterna och funktionaliteten ner i mindre delar (vanligtvis kallade *tasks*) och tidsuppskattas. Dessa mindre delar utgör det som behöver göras för att uppnå sprinten mål.

Granskningsmöte

Vid ett sprint planning möte sätter utvecklingsgruppen upp mål de antas kunna uppnå under sprinten. Vid slutet av en sprint anordnas ett sprint review meeting. Syftet med mötet är att utvecklingsgruppen skall presentera det som de uppnått under sprinten. Vanligtvis med en demonstration av det delar som implementerats. Utvecklingsgruppen berättar också om deras erfarenheter under sprinten.

Ledningen, kunder och användare skall vara med på ett sprint review möte i syfte att kunna ta välinformerade beslut om vad som skall göras härnäst i projektet. (Scwhaber & Beedle, 2001).

Retrospektivt möte

Efter en sprint samlas utvecklingsgruppen för ett retrospektivt möte. Syftet med det retrospektiva mötet är diskutera föregående sprint. Vad fungerade? Vad fungerade inte? Vad kan vi göra för förändringar inför nästa sprint? Produktägaren, scrummaster och utvecklingsgruppen deltar alla i det retrospektiva mötet. Det retrospektiva mötet är en viktig del i Scrum och bidrar till ett ständigt förbättrande av processen inom projektgruppen. (Deemer & Benefield, 2006)

Roller

Produktägare

The Product Owner is responsible for representing the interests of everyone with a stake in the project and its resulting system. (Schwaber, 2004)

Produktägaren ansvarar för de övergripande kraven på systemet och systemets return on investment¹. Insamlingen och uppfångandet av dessa krav sker genom att produktägaren representerar alla personers intressen i systemet.

Dessa krav samlas in i produktspecifikationen som beskrivs närmare senare i uppsatsen. Produktägare använder sig utav produktspecifikationen för att försäkra sig om att den mest värdefulla och prioriterade funktionaliteten implementeras först i systemet.

Scrummaster

The Scrum Master is responsible for the success of Scrum (Schwaber, 2004)

Scrummastern är en ny typ av roll som närmast kan jämföras med den traditionella projektledaren. Scrummastern ansvarar för att utvecklingsprocessen går åt rätt håll och att utvecklarna störs så lite som möjligt i deras arbete genom att undanröja eventuella hinder som dyker upp. (Engwall & Jacobsen, 2007).

Scrummastern är ytterst drivande för att Scrum som metod efterföljs av alla projektdeltagare och kan ses som en brygga mellan utvecklarna och ledningen. Under de dagliga mötena observerar och lyssnar Scrummastern på utvecklingsgruppens framgångar eller motgångar (Schwaber, 2001).

Utvecklingsgrupp

A team commits to achieving a Sprint goal. The team is accorded full authority to do whatever it decides is necessary to achieve the goal. (Schwaber 2001)

Utvecklingsgruppen är ytterst ansvariga för att möta de mål som sätts upp på Sprint Planning mötet. Hur mycket funktionalitet de avser att implementera under sprinten är helt upp till dem själva.

¹ Return on investment (ROI) är en affärsmodellanalys. ROI är intäkten från ett agerande dividerat med den uppskattade kostnaden för agerandet.

Scrum understryker vikten av att en utvecklingsgrupp skall vara tvärfunktionella. En utvecklingsgrupp skall således bestå utav alla kunskaper som behövs för att uppnå de mål som satts upp inför sprinten. Detta kan vara t.ex. utvecklare, designers, testare och domänexperter. Synergieffekterna av dessa tvärfunktionella grupper blir enligt Schwaber ökad produktivitet i gruppen då t.ex. en testare kan hjälpa en designer att producera kod som enklare gå igenom testfasen etc.

Enligt Scwhaber är utvecklingsgrupper i Scrum självorganiserande. Scrummastern skall, så långt det är möjligt, undvika att blanda sig i konflikter inom gruppen. Schwaber menar att så fort en Scrummaster försöker lösa konflikter eller på annat sätt blanda sig i hur gruppen löser sin uppgift tar denne en liten bit av utvecklingsgruppens ansvar. Utvecklingsgruppen är anförtrodda att lösa målen och dem kommer själva att fundera ut hur dem skall lösa det på bästa sätt.

Dokumentation

Produktspecifikation

Product Backlog is an evolving, prioritized queue of business and technical functionality that needs to be developed into a system. (Schwaber, 2001)

Som inledningstexten ovan antyder är produktspecifikationen en lista som beskriver vilka funktioner, teknologier, förbättringar som utgör förändringarna som kommer att utföras på ett system i framtiden. Allt aktiviteter som skall göras skall ingå i produktspecifikationen.

En produktspecifikation skall alltid vara en prioriterad lista. Aktiviteter och funktioner som är viktiga är högt prioriterade är således viktiga att utföra så snart som möjligt. (Schwaber, 2001). Endast en person är ansvarig för att produktspecifikationen hålls uppdaterad. Denna person är Produktägaren. Schwaber understryker det faktum att det inte skall vara en grupp av människor som ansvarar för produktspecifikationen. En grupp av människor kan ha motsägelsefulla mål med projektet och product backloggen kan bli otydlig vilket kan medföra förvirring för dem som faktiskt skall implementera den.

Detta medför inte att Produktägaren kan väljas att influeras eller få råd ifrån en grupp av människor (t.ex. en kommitté), men ansvaret att faktiskt uppdatera och underhålla den är Produktägarens själv.

Sprintspeifikation

Sprint Backlog consists of the tasks that the Scrum Team has devised for att Sprint. These tasks are work to transform the selected Product Backlog into the Sprint goal. (Schwaber, 2001)

Sprint backlog är en lista på funktionalitet eller aktiviteter som skall göras under sprinten. Under Sprint Planning meeting bestämmer utvecklingsgruppen vilka delar av product backloggen de tror att de hinner med under nästkommande sprint.

Dessa delar tidsuppskattas, vanligtvis i timmar, och utgör sedan sprintens backlog. Utvecklarna själv är sedan ansvariga för att uppdatera uppskattningarna allteftersom arbetet fortlöper. Detta fortgår under hela sprinten tills det att alla aktiviteter antingen är gjorda eller avskrivna från sprinten.

Burndowngraf

The trend of work remaining across time in a Sprint, a release or a product. (Schwaber 2004)

Burndowngraf är ett kvantitativt visualiseringsverktyg som gör det möjligt att följa ett projekts eller en sprints framgång.

En burndowngraf visar mängd arbete på X-axeln och kalenderdagar på Y-axeln.

En linjär linje ritas upp för att visa den normativa utvecklingen. Varje dag under sprinten uppdateras tidsuppskattningarna i sprintspecifikationen av utvecklarna. Antalet återstående timmar används sedan i grafen för att identifiera utvecklingsgruppens framgång under sprinten.

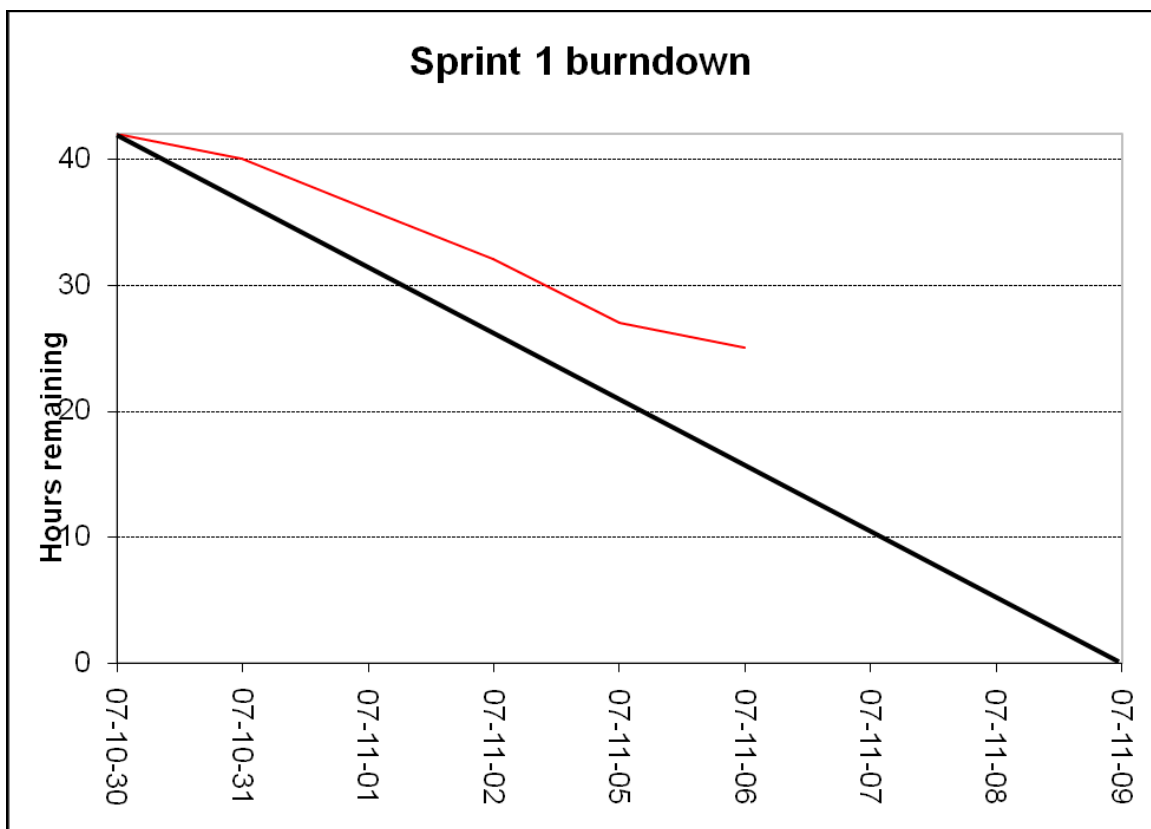


Figure 3-2: Exempel på burndown graph. Här visualiseras att utvecklingsgruppen ligger efter tidschemat. Kanske något behöver tas bort ifrån sprint backlog?

Som Scwhaber menar i inledningstexten till detta stycke kan detta kvantitativa verktyg även appliceras på product backloggen eller release backloggen för att öka visualiseringen även på högre nivåer.

Kritik av Scrum och agila metoder

Marginaliserar problemlösare

Skowronski (2004) menar i sin artikel ”Do Agile Methods Marginalize Problem Solvers” att Isaac Newton hade varit ett misslyckande som agil utvecklare.

Med detta påstående menar Skowronski att agila metoder kan verka negativt på de bästa problemlösarna i och med att de uppmuntrar ett ständigt samarbete inom gruppen.

Författaren beskriver vidare att sådana metoder inte tillåter individuella tankeprocesser. Det är inte heller säkert att metoder som brainstorming hjälper en grupp att lösa ett problem. Istället menar Skowronski att det kan behövas kunskap och erfarenhet som tillförs gruppen utifrån.

Agila metoder menar tvärtom att när människor arbetar tillsammans i en bra miljö med god kommunikation når en högre nivåer och förbättrar effektiviteten.

Scrum förstår inte den mänskliga psykologin

Kevin Brady (2006) menar att Scrum ser bra ut i teorin men misslyckas i praktiken. Detta beror enligt Brady på att Scrum och agila metoder inte till fullo förstår vad Brady kallar ”den mänskliga faktorn”. Brady pekar på ett antal aspekter av den mänskliga naturen som han menar att agila metoder inte har tagit i betraktelse (fritt översatt från artikeln):

- Människor sätter alltid sina egna intressen i första hand framför gruppens.
- Människor är självcentrerade
- Kommersiella beslut är baserade på rationella förväntningar
- Du kan aldrig få mer än 5 personer att komma överens om något.

Brady menar att Scrum och de agila synsättet förvandlar projektledare till projektadministratörer utan verklig auktoritet (agila utvecklingsgrupper är självstyrande) att styra över projektet. Med denna utveckling menar Brady att agila projekt kan helt komma sakna planer för riskhantering, kvalitetskontroll, budget och rapportering.

Vidare beskriver Brady att agila utvecklingsgrupper ofta ”tas över” av en person i utvecklingsgruppen med stark personlighet. Denna person, ofta en senior tekniker, tar över allt intressant utvecklingsarbete och lämnar det mindre intressanta till de övriga i gruppen. Detta, tillsammans med avsaknad av dokumentation, leder till att denna person får total kontroll över systemet och kommer att bli den enda personen i gruppen med tillräcklig kunskap att underhålla och vidareutveckla det. Vidare leder detta till högre status för personen och möjligtvis en lön över marknadsnormen.

Scrum och agila metoder är inte vetenskapliga

Gustaf Juell-Skiels pekar i sin artikel i Computer Sweden på bristen av vetenskapliga undersökning gällande Scrums framgångsfaktorer. Juell-Skiels menar att det inte finns några belägg för att agila metoder fungerar bättre vid införandet av ett standardpaket och affärssystem annat än anpassningar av dessa system. Juell-Skiels beskriver vidare att Scrum saknar kritiska moment i systemutvecklingsprocessen såsom analys av nuläge och affärsprocesser.

Vidare kritiserar författaren Scrum i avseende att metoden är härledd ur Schwabers och Beedles högst subjektiva uppfattningar om systemutveckling och inte ur solid systematisk forskning.

Införande av Scrum i en organisation innebär förändring

*”... what will I do and where will I fit into the new organization”
(Leffingwell & Smiths. Sid 9)*

Under Scrums första 15 år har den givande trenden varit att de flesta Scrum implementationer har varit s.k. bottom-up² drivna (Leffingwell & Smiths). Bottom-up drivna implementationer av Scrum karaktäriseras av att utvecklarna anammar metoder och driver på införandet inom projektet eller organisationen. Detta faller väl in med Scrums principer och värderingar om självbestämmande utvecklingsgrupper.

På senare tid, menar Leffingwell & Smiths, har företag och organisationer, som ett steg i att öka företagets produktivitet, visat mer intresse av att införa Scrum top³-down, dvs. att ledningen driver på införandet. Författarna beskriver att införandet av Scrum kan te sig enkelt på ytan men menar på att Scrum för med sig betydande organisationsförändringar. Dem föreslår vidare att en ”förändringsagent” eller lokal sponsor skall utses med primärt ansvar för att avlägsna hinder som dyker upp under ett införande av Scrum i en organisation.

Sutherland (2006) beskriver vidare i sin föreläsning på Google tre nivåer av Scrum. Sutherland menar att företag kan börja på lägsta nivån och sedan klättra uppåt och använda sig utav mer och mer avancerade Scrum tekniker och verktyg.

| | A - Projekt | B - Produkt | C – Organisation |
|-----------|--|---|---|
| Dokument | Produktspecifikation Sprints specifikation Burndown graf | Versionsplanering Produktspecifikation Sprints specifikation Burndown graf | Resursplan Versionsplanering Produktspecifikation Sprints specifikation Burndown graf |
| Cermonier | Sprintplaneringsmöte Dagligt möte Sprint review | Multilevel planering Dagligt möte Scrum of Scrums Sprint review | Multilevel planering Dagligt möte Scrum of Scrums Sprint review |
| Roller | Produktägare Scrummaster Utvecklingsgrupp | Produktägaransvarig Scrummasteransvarig Flera produktägare Flera Scrummasters Flera utvecklingsgrupper | Produktägaransvarig Scrummasteransvarig Flera produktägare Flera Scrummasters Flera utvecklingsgrupper Meta Scrum |

Tabell 1

² Bottom-up är en term från organisationsteorin som beskriver förändringsstrategi.

³ Se bottom-up.

Som figuren visar beskriver Sutherland tre typer av Scrum.

- **Typ A Scrum**
Få utvecklingsgruppen att fungera ihop. Fokus skall vara på att få fungerande kod i slutet av varje sprint. Sutherland beskriver det som en projektrytm.
- **Typ B Scrum**
Fokuserar på flera produkter och flera utvecklingsgrupper och projekt som måste synkronisera för att skapa bra produkter.
- **Typ C Scrum**
Scrum av typen C beskriver Sutherland som en variant av Scrum där metoden är implementerad i hela organisationen. Ett begrepp som MetaScrum införs där alla projekt ingår i s.k. metaiterationer där resurser och gemensamma planer kan förändras över organisationen t.ex. varje vecka.

Det bästa sättet att bereda vägen för en ny systemutvecklingsmetod är att testa metoden i en relativ ostörd miljö. (Lunell, 2003, Leffingwell & Smiths 2005). Målsättningen med ett pilotprojekt är att under projektet noggrant observera och dokumentera metodens fördelar och nackdelar.

Inkrementellt införande

4. Fallstudien

I detta kapitel ges en sammanställning av fallstudien på Stendahls.net. Kapitlet inleds med bakgrundsinformation om Stendahls.net och dess organisation och nuläge. Vidare beskriver jag det aktuella projektet organisatoriskt och tekniskt. Kapitlet avslutas med att jag ger en kort överblick över de faktiska sprintarna som utfördes under studien.

Stendahls.net AB

Stendahls.net AB bildades 1999 och är ett dotterbolag till Stendahls AB (reklambyrå). Stendahls.net AB har 23 medarbetare. Företaget verkar framförallt som IT-konsulter men har också en uppsättning egna produkter och stöd för webbaserade marknadsplaner, trycksaks-produktion och bildhantering. Stendahls.net arbetar även med att utveckla webbsajter och webbkampanjer för ett antal större företag med global verksamhet.

Organisation

Stendahls.net AB växer kraftigt. De senaste sex månader har mellan fem och tio personer anställts och de ökande trycket från företagets kunder kommer att kräva ytterligare förstärkningar.

Företagets organisationsform kan kategoriseras som "organisk". En organisk organisationsform kännetecknas av följande egenskaper (Jacobson & Thorsvik):

- Nätverksstruktur för auktoritet, kontroll och kommunikation
- Arbetsuppgifterna omdefinieras ständigt och anpassas efter behov.
- Den enskildes roll är generellt definierad.
- Kommunikationen är både vertikal och horisontell - allt efter behov.

Denna organisationsform kallades i början för ad hoc-kрати men valdes senare att betecknas som den *innovativa organisationen* (Jacobson & Thorsvik).

Stendahls.net har tidigare gjort försök med att strukturera sina systemutvecklingsprocesser. Pilottester har gjorts med bl.a. Rational Unified Process (RUP). RUP är en välkänd och etablerat utvecklingsprocess. RUP är också, tillskillnad från Scrum, en kommersiell produkt skapad av Rational Software Corporation.

Dock var RUP en alldeles för tung process att implementera på ett framgångsrikt sätt inom Stendahls.net och efter ett halvår lades försöket ner.

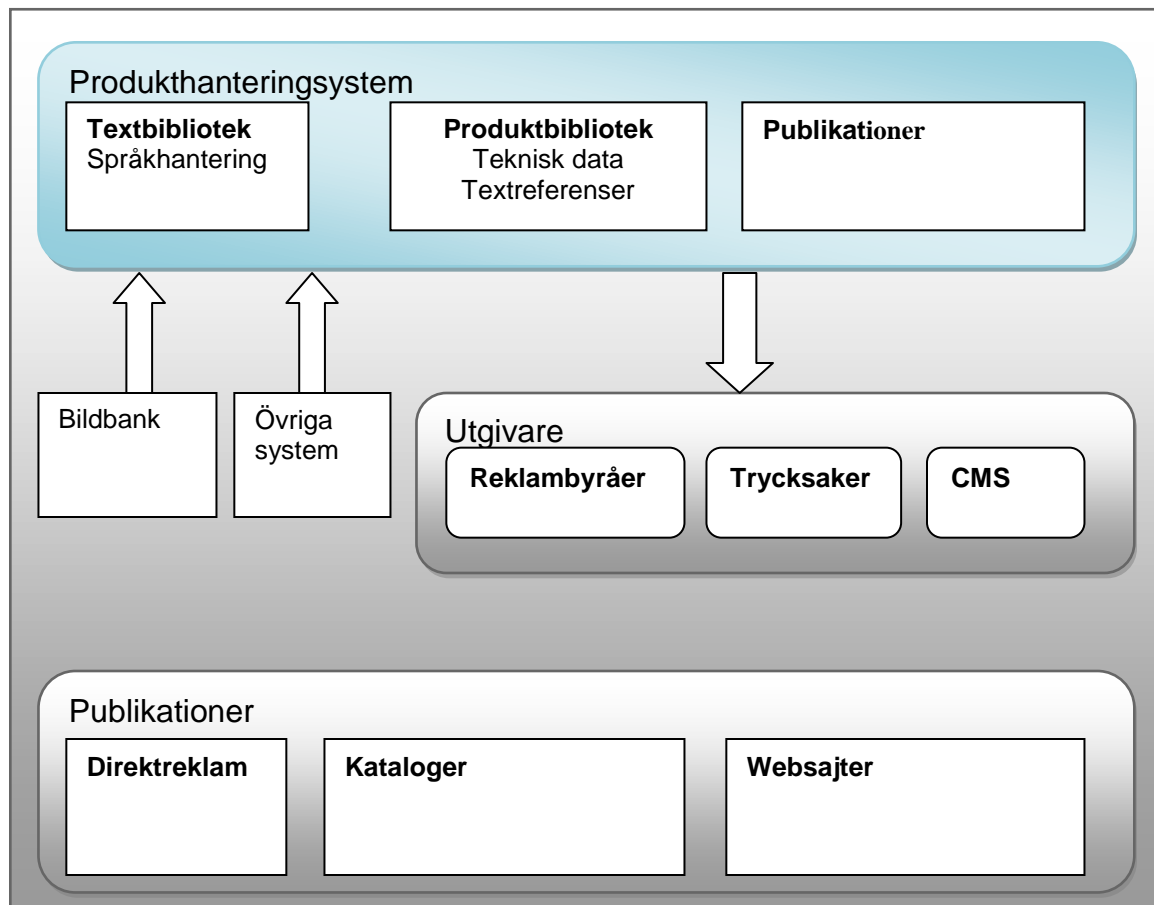
Projektet

Introduktion

Projektet jag valt att följa är ett vidareutvecklingsprojekt av ett produkthanteringssystem till ett globalt verkstadsföretag. Syftet med produkten är att konsolidera all marknadsföringsinformation rörande alla produkter och artiklar inom företaget. Antalet artiklar uppskattas till runt 16,000 i dagsläget. Även tillbehör och reservdelar innefattas av systemet. System utvecklades i en första version 2005 och har sedan dess vidareutvecklats kontinuerligt.

Teknik

Systemet bygger på en klient/server arkitektur där klienten har implementerats i Java Swing⁴ och servern i Microsoft .NET. Klienten och servern kommunicerar med varandra genom Web Services. Motivationen för att inte välja ett webbaserat gränssnitt är snabbheten och möjligheterna till avancerade gränssnittsfunktioner såsom drag-and-drop etc



⁴ Swing är ett API som gör det möjligt för Java-program att tillhandahålla ett grafiskt gränssnitt.

Utbildning och arbetsmiljö

Projektet inleddes med en introduktion om Scrum till projektdeltagarna. De grundläggande komponenterna i Scrum gick igenom och diskuterades. Utbildningsmaterial om Scrum sattes upp på en whiteboard för att snabbt kunna slå upp ord och begrepp som Scrum använder sig utav.

En plats för Scrummöten etablerades och projektdeltagarna flyttade ihop sina arbetsplatser till ett och samma rum för att enkelt kunna kommunicera med varandra.

Ett manuellt system med ett stort papper och PostIt lappar etablerades för att få ett övergripande bild av projektet.

Scrum som metod beskriver egentligen inget av ovanstående tillvägagångssätt. Idéerna till detta kom igenom ett dokument skrivet av Henrik Kniberg (Scrum and XP from the trenches). Dokumentet är en samling erfarenheter Kniberg har tagit till sig när han arbetat med Scrum under 1 års tid.

Denna plats användes genom både sprint 1 och sprint 2 som mötesplats för de dagliga mötena.

Kravspecifikation blir produktspecifikation

Planeringen över kommande funktionalitet inom projektet har tidigare legat i en kravspecifikation. Dokumentet hade många likheter med en product backlog.

I huvudsak bestod den av en lista med funktionalitet som skall implementeras där varje del var tidsuppskattad och beskriven i icke-tekniska termer.

Arbetet med att ta fram en initial *product backlog* gick därför relativt enkelt.

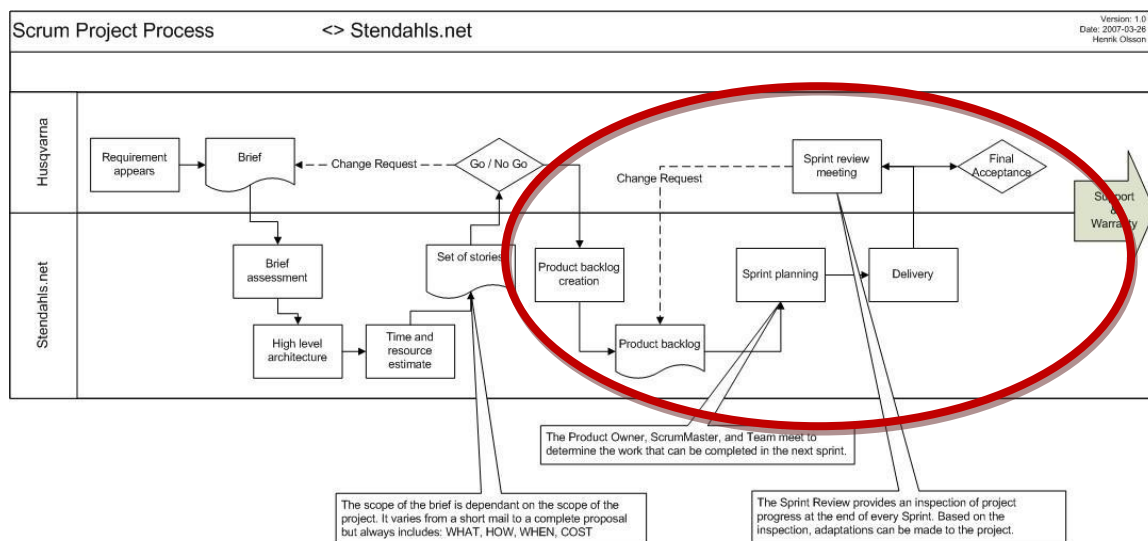
Ett kalkylblad upprättades och eftersom varje del redan var tidsuppskattad behövde de endast prioriteras. Detta gjordes av utvecklingsgruppen i nära samarbete med kund.

Tidigare har en utvecklare varit ansvarig för underhållet av kravspecifikationen men i och med att vi införde Scrum ändrade vi den ansvarig till projektledaren (produktägaren).

Kund och beställare

Beställaren och kunden av projektet notifierades tidigt i projektet att Scrum skulle användas som projektstyrningsmetod. Kunden ställde sig positiv till arbetssättet och metoden och gav klartecken till att använda den. En ny version av Stendahls.net generella projekt process togs fram (Figur 4-1). Med den modellen hjälptes kunden att förstå hur vi framtiden skulle hantera systemutvecklingsprojektet och interagera mot varandra.

Vi började också tidigt i projektet att använda oss utav Scrums vokabulär (*stories, sprintar, product owner* osv.) i projektdokumentationen för att skapa klarhet och undvika missförstånd som skulle ha kunnat uppkomma om termer och begrepp missförstås. En enkel introduktion till Scrum och Scrums begrepp och termer skickades därför över till kund.



Figur 4-1: Stendahls.net Project Process med Scrumdelen markerad.

Roller

Produktägare

Tidigt i projektet bestämdes det att projektledaren skulle axla rollen som produktägare tillsammans med kunden. I praktiken innebar detta att projektledaren agerade produktägare *under* sprinten, medans kunden har sista ordet i vad som skall prioriteras till sprintarna. Detta visades sig dock vara ganska svårt i projektet eftersom en ny projektledare tillsattes efter ungefär halva projektet. I praktiken var det faktiskt en utvecklare som faktiskt uppdaterade och prioriterade produktspecifikationen tillsammans med kunden.

Den absolut viktigaste egenskapen hos produktägarrollen är att den endast skall bestå av en person enligt Scrum (Schwaber, 2005). Det delade ansvaret runt produktspecifikationen visar på att projektet inte har använt sig utav Scrums definition av produktägaren. Detta var självklart ett medvetet val från Stendahls.net sida då det skulle kräva större organisations- och arbetssätfförändringar om produktägarrollen skulle tillämpas till fullo.

Scrummaster

Eftersom jag skulle studera projektet och att jag, efter litteraturstudien, hade relativt stor kunskap om Scrum bestämdes det att jag skulle axla Scrummaster rollen inom projektet. Eftersom jag inte har varit involverad i projektet innan var det underförstått att jag inte skulle agera Scrummaster i projektledarbemärkelse utan snarare som Scrum handledare och utbildare.

Rollen Scrummaster enligt Scrum definitionen var således frånvarande i detta projektet. Vissa delar (utbildning, handledning) hanterades av mig och projektledningsdelarna hanterades av den traditionelle projektledaren.

Utvecklingsgruppen

Utvecklingsgruppen bestod utav tre personer:

Utvecklare 1

Har varit med i projektet sen start. Har stor kunskap om verksamhetsområdet och den tekniska lösningen, främst på klient sidan.

Har hittills arbetat nära kund och tekniskt ansvarat för produkten.

Utvecklare 2

Har arbetet i mindre omgångar i projektet, främst då på klientsidan. Arbetar nu för första gången på serversidan då ordinarieutvecklare är pappaledig.

Utvecklare 3

Har till och från arbetat inom projektet främst som hjälp till utvecklare A. Har även utvecklats på serversidan då ordinarie utvecklare är pappaledig.

Sprint 1

Sprint 1 pågick endast en begränsad tid och under förhållandevis låg intensitet. Orsaken till detta var bland annat att delar av utvecklingsgruppen var på semester och ett visst frånfall pga. sjukdom. Målet för sprint 1 delades upp i två delar. Dels skulle en systemdesign valideras genom en minimal implementation och dels skulle en utvecklare som tidigare inte arbetat på serversidan komma igång med de programvaror och rutiner som erfordrades.

Sprinten löpte på bra (enligt burndown grafen) och alla delar slutfördes inom utsatt datum.

På grund av den dåliga närvaron och den låga intensiteten (endast en utvecklare under denna perioden) beslutades det att inte starta sprint 2 förens alla utvecklare fanns på plats. Under denna perioden arbetade dock den återstående utvecklare på inom projektet och slutförde saker på klientsidan som ej var beroende av servern.

Detta var naturligtvis inte positivt för studien med tyvärr sådant man får räkna med.

Detta till trots avslutades ändå sprint 1 med ett granskningsmöte där den nya funktionaliteten demonstrerades för produktägaren (i det här fallet projektledaren).

Sprint 2

Under sprint 2 fanns hela utvecklingsgruppen närvarande (tre personer). Sprinten startade med ett *sprint planning meeting* där utvecklingsgruppen valde ut den högst prioriterade funktionalitet, tillsammans med produktägaren, dem trodde sig skulle hinna med under nästkommande sprint.

Utvecklingsgruppen fortsatte mötet själva där de bröt ner varje del i mindre uppgifter som de tidsuppskattade. Uppgifterna kategoriserades efter:

- Prioritet
- Del av system (klient eller server).
- Inom vilken komponent.

Olika färger på lapparna som symboliserade uppgifterna användes för att enklare kunna skapa sig en överblick över projektet. Tidsuppskattningarna markerades på lapparna och uppdaterades dagligen under *daily scrum meeting*. I samband med detta uppdaterade även projektledaren *burndown graph*.

5. Resultat

I detta kapitel presenterar jag det resultat de fem intervjuerna har gett mig. Inledningsvis ger jag en kort presentation av respondenterna. Resultatet är uppdelat i tre områden: roller, artefakter och flöde.

Respondentpresentation

Samtliga respondenter i intervjuerna har varit delaktiga i projektet som har utgjort fallstudien i det här arbetet. Alla är, i skrivande stund, anställda hos Stendahls.net AB.

U1

U1 är systemutvecklare och har varit anställd hos Stendahls.net i 8 år. Har arbetat i det aktuella projektet sedan i höstas. Arbetet har intensifierat under november.

Då den ordinarie utvecklaren är pappaledig är U1 tillförordnad serveransvarig.

U2

U2 är systemutvecklare och systemarkitekt i det aktuella projektet. Tidigare har U2 verkat som både teknisk projektledare och övergripande projektledare. U2 har arbetat med programmering i 11 år och har varit med i projektet sen start, i snart 5 år.

U2 har främst arbetat med klientdelen och är ansvarig för gränssnittet.

U3

U3 har verkat som systemutvecklare hos Stendahls.net AB i 3 år och inom det aktuella projektet i 2 år. U3 har arbetat på både klient och server delen och framförallt i egenskap av avlastare åt både U1 och U2.

PL1

PL1 har verkat som projektledare hos Stendahls.net i 4 veckor. PL1 har tidigare arbetat som projektledare i 7 år.

Övergripande

Jag tror att projektet faktiskt har gått bättre än om vi inte hade kört Scrum (U1)

Samtliga respondenter är överens över att Scrum har lyckats i det aktuella projektet. Vid närmare granskning av de aspekter respondenterna angivits som de viktigaste går det att urskilja vissa mönster. Överlag är respondenterna mycket välkomnande mot den struktur Scrum har infört i projektet. Detta måste beaktas i sitt sammanhang. Projektet har innan införandet av Scrum inte använt sig utav någon strukturerad metod eller process. Projektet har i stort styrts av kunniga, motiverade och drivande medarbetare, men hur ett metodperspektiv på ett relativt ostrukturerat sätt. Projektet har alltså, med Scrum, tagit ett steg mot ett *mer* strukturerat arbetssätt.

Utvecklarna är ganska ambivalenta såvida de känner sig mer eller mindre stressade när de använder Scrum. Vissa menar att det känner sig lite mer stressade i och med att varje del i sprinten är nedbruten i små tekniska uppgifter vilket kräver en daglig prestation av utvecklaren. Samtidigt menar också t.ex. U3 att denne känner sig mindre stressad av den anledningen att denne hela tiden känner av projektets status.

Flöde

Dagliga möten

Daily meetings ger den där vardagsstrukturen i projektet. Det är vad som ger oss mest. (U1)

De dagliga mötena har varit en av de delarna i Scrum som har uppskattats mest av både utvecklarna och av projektledare. Flera av responderna menar just att de dagliga mötena ger en struktur i arbetsdagen och att informationsutbytet som sker är ovärderligt. Precis som U1 menar ovan ger de dagliga mötena en vardagsstruktur för projektgruppen. Flera respondenter tycker att det var alldeles för mycket möten till en början, men menar att de positiva effekterna som uppstod under projektet ganska snabbt överskuggade detta.

U2 beskriver en intressant effekt av att de dagliga mötena är planerade till att vara på morgonen. U2 menar att detta medför att denne kommer igång snabbare med sitt arbete och visar, tidigt på dagen, både för sig själv och för övriga vad man skall prestera. I huvudsak utvecklarna, menar också på att det var lite problematiskt att få en bra struktur på mötena i början då de gärna skenade iväg i någon teknisk diskussion men att detta blev bättre med tiden.

Ibland har vi diskuterat samma sak två dagar i sträck. (U2)

Självklart är det svårt, till en början, att få till effektiva dagliga möten på det sättet som Scrum beskriver dem. Diskussioner kan som U2 menar lätt skena iväg, och aldrig komma till ett naturligt slut. Schwaber (2001) trycker tydligt på den punkten att det dagliga mötet inte är en designsession eller ett möte för tekniska diskussioner. Schwaber menar att diskussioner som dyker upp skall noteras och att istället ett möte för diskussionen skall bokas. Det är också viktigt att Scrummaster, eller projektledaren i vårt fall, försöker ha kontroll över mötet och ser till att den faktiska agendan efterföljs. Dock är det här viktigt att tänka på att dessa tekniska diskussioner måste få sin tid och om man inte tar dem när dem dyker upp är det lätt att glömma av dem. Kniberg (2004) visar på ett angreppssätt där utvecklarna skriver en kom-ihåg-lapp angående diskussionen och sätter upp den med övriga uppgifter att göra under dagen. På så sätt hålls det dagliga mötet snabbt och övriga diskussioner som dyker upp glöms inte bort.

U1 anser att gruppen skulle kunna förbättra uppföljningen av uppgifter som man tagit ansvar över. Just uppföljningsmomentet (frågan: vad har jag gjort sen det senaste mötet?) är en viktig del i det dagliga mötet men har i vårt fall kanske hamnat i lite i skymundan. Det finns en tendens att istället diskutera det som skall göras under dagen. Detta är givetvis ett problem och i det aktuella fallet är det en effekt utav att mötena inte har varit tillräckligt styrda av Scrummastern och projektledaren.

U1 anser även att det är för mycket ”vattentäta skott” mellan delarna (klienten och servern) i projektet och att detta minskar förutsättningarna för en diskussion kring eventuella hinder man har. U1 tror att en lösning på detta är, även om det känns onödigt för de andra, att diskutera igenom det delar man har för avsikt att implementera under dagen. U1 pekar här på en viktig detalj. Självklart kan det vara så att olika utvecklare inom ett projekt arbetar på helt olika delar i systemet. Det är ändå dock viktigt att varje utvecklare under det dagliga mötet försöker beskriva för de andra vad som gjorts. Scrummastern måste också vara aktiv och trycka på att få varje utvecklare att beskriva sitt arbete.

PL1 menar att de dagliga mötena ger projektet transparens och att det blir mycket svårare att ”gömma sig”. PL1 uttrycker också att det tidigt i projektet, via de dagliga mötena, är möjligt att se hur det faktiskt kommer att utveckla sig. Vad PL1 utvecklar här är ett av Scrums kärnvärden. Schwaber (2001) menar att genom att närvara på det dagliga mötena kan ledningen enkelt och snabbt få en snabb inblick i projektets verkliga status.

PL1 menar också att de dagliga mötena ger en chans åt projektledaren att känna sig på ”samma nivå” som övriga deltagare. Detta leder till, enligt PL1, att kreativiteten och engagemanget hålls uppe under projektet och projektledare slippar att kontrollera vad som gjorts föregående dag eftersom det med de dagliga mötena kommer naturligt.

PL1 har också upplevt att denne fått en godare teknisk inblick i projektet genom att närvara under de dagliga mötena.

Planeringsmöte

Planeringsmötet syftar till, som nämnts tidigare, att bryta ner de delar som skall implementeras under sprinten till mindre, tekniskt beskrivna, uppgifter.

U2 beskriver hur han känner en viss ovana att bryta ner dessa delar i så små (max 1 dag) delar. Men ser det endast beror på att han är ovan tillvägagångssättet och anser att detta kommer att bli bättre med tiden.

U2 menar också att denne har blivit mer effektiv med dessa mindre uppgifterna och de dagliga uppföljningarna.

När jag tar upp en lapp som är uppskattad till en timma så vet jag att jag har gott om tid, men att jag inte heller kan gå omkring och prata med folk hursomhelst (U2)

U1 ser en svårighet med att flera delar i sprint 2 inte har ett tydligt, eller inget, gränssnitt och att detta då blir svårt och demonstrera för kund på granskningsmötet. U1 menar att genom att konkretisera hur utvecklingsgruppen faktiskt skall demonstrera delarna i systemet tydliggör de också vad som faktiskt förväntas av dem.

Roller

Samtliga respondenter är eniga över att vi inte tillämpat rollerna i särskilt hög grad under projektet vilket också medfört att respondenterna överlag inte har så mycket att säga om dessa. De gemensamma är att tidigare roller (en projektledare, en utvecklingsgrupp) faktiskt har fungerat förhållandevis väl. Detta är intressant på så sätt att respondenterna har upplevt att Scrum i sin helhet har fungerat relativt bra även utan att strikt följa rollbegreppen som Scrum uttrycker.

Scrummaster

Scrummastern har enligt respondenterna varit i huvudsak periferi. Detta anser dock U1 som en positiv egenskap då utvecklingsgruppen skall vara tämligen självgående. Vidare anser U1 att Scrummastern inte skall vara endel av projektet i övrigt:

Jag tycker det är bra att det (Scrummastern) är någon utanför projekt som kan ha en "gubben i lådan" roll och kan ta obekväma beslut som utvecklare eller projektledare inte vill ta.

Flera av framförallt utvecklarna ansåg det också viktigt att ha en Scrummaster som motvikt mot affärssidan. Som kommentaren ovan menar på anses det viktigt att ha en person med formell auktoritet som kan ta obekväma beslut och få tyngd bakom teknikkraven.

Scrummastern har ju, som nämnts, på sitt ansvar att utbilda deltagarna på sitt ansvar vilket har gjorts löpande under projektets gång. Respondenterna ser inga brister i kunskapsnivån angående Scrum i projektet. Ingen ser sig själv dock som expert och förstår

metoden till fullo. Detta anser dock PL1 som något positivt. Det ger deltagarna en chans att själva ”finslipa” metoden och utveckla och anpassa den själva. Detta är en intressant synpunkt. Risken kan dock vara att utbildningen blir bristfällig och den anpassade metoder ”glider ifrån” Scrum-definitionen och mister endel av de värden som Scrum utlovar.

Produktägare

U1 anser att det är svårt att hitta en given produktägare i vår organisation. Detta kan till viss del bero på att projektet har bytt projektledare ett antal gånger under de senaste åren, men visar också på svårigheterna att byta arbetssätt och tillsätta nya roller inom en organisation.

Hittills har U2 varit den person inom projektet som närmast skulle kunna beskrivas som produktägare. Dock innebär det här, enligt U1, att man kan få sitta på dubbla stolar.

I en ideal situation skulle rollen naturligtvis axlats av kunden. Dock är detta svårt då kunden i det aktuella fallet inte alltid har tid att följa upp och kontrollera projektet i den grad som Scrum förordar. Vid en eventuell bredare tillämpning av rollbegreppen i Scrum skulle detta kunna lösas med vad Schwaber (2001) kallar för en kundproxy. Dvs. en person (t.ex. projektledaren) agerar produktägare å kundens vägnar. PL1 ser det inte så viktigt att följa rollbegreppen i Scrum i så här små projekt utan tycker att det är tillräckligt med att ha en traditionell projektledare.

Vad man kan utläsa av resultaten är att det är svårt i en organisation som Stendahls.net AB, som närmast kan beskrivas som en ad-hockrati, att börja strama upp ansvarsområden och införa rollbegrepp enligt Scrum. Risken med att införa rollbegreppen är ju naturligtvis att det inkräktar relativt mycket på organisationens kultur och anda.

Utvecklingsgruppen

Flera projektdeltagare pekar på aspekten att den tidigare organisationen innan har agerat agilt till en viss grad innan tillämpningen av Scrum. Detta menar respondenterna har varit till en stor hjälp vid införandet av Scrum då förändringarna inte blivit så omvälvande.

Vissa utvecklare menar dock på att de skulle kunna vara ännu mer självgående under sprintarna. Under de dagliga mötena har projektledaren i det aktuella projektet varit förhållandevis aktiv.

Vissa utvecklare menar också att det ibland kan vara positivt att ha en projektledare som faktiskt pressar utvecklingsgruppen framåt.

Artefakter

Produktspecifikation

Eftersom att projektet tidigare har haft en relativt bra kravspecifikation där framtida funktionalitet och aktiviteter beskrivits i detalj har respondenterna inte mycket att säga om detta dokument. Dokumentet har under projektet underhållits av U2 och PL1.

Sprintspekifikation

Sprint specifikationen har under projektet visualiserats med hjälp av de tekniker som finns beskrivna i Scrum and XP from the trenches av Kniberg (2004).

Just denna teknik har under projektet blivit synonymt med Scrum bland projektdeltagarna och är en utav de delar som respondenterna anser vara den viktigaste. Tekniken bygger på att ett papper med fyra kolumner sätts upp på en vägg. I kolumn ett (ej färdigt) klistras de tidsuppskattade, tekniskt beskriva uppgifter som skall slutföras under sprinten upp. När en utvecklare börjar arbeta med en uppgift flyttar denne uppgiften från ”Ej färdigt” till ”Utcheckad”. Detta för att undvika att flera utvecklare arbetar på samma uppgift och för att få ett naturligt ansvarstagande för uppgiften. När utvecklaren anser sig vara färdig med uppgiften flyttar denne den aktuella lappen till ”Färdigt”. Allt eftersom projektet framskrider kan projektdeltagare och utomstående se hur lapparna rör sig från höger till vänster på pappret.

Den fjärde kolumnen används till burndown-grafen och oplanerade uppgifter som dyker upp under sprinten. Flera respondenter beskriver hur viktigt det är för motivationen att under sprinten se hur lapparna under ”Ej färdigt” hela tiden minskar och hur det samtidigt infinner sig ”en skön känsla” när man faktiskt kan flytta en lapp från ”Utcheckad” till ”Färdig”.

Vad som dock är viktigt och vad som har varit tvetydigt under projektet är faktiskt vad som menas med att en uppgift är färdig. Är koden incheckad? Är den testad? Har testspekifikationen uppdaterats?

En outalad definition har under projektet varit att koden är testad av utvecklaren och iallafall incheckad. Problemet med denna ganska vaga definition av ”klar” är att problem och buggar kan finnas vid senare, mer omfattande, tester av funktionalitet som betyder att projektet blir försenat. Olika utvecklare kan också ha olika uppfattningar om vad en ”klar” uppgift innebär vilket kan leda till oklarheter inom gruppen. Det är alltså viktigt att enas, innan projektet, om vad som menas med att en uppgift är färdig.

U1 anser att projektdeltagarna har varit för frikostiga att lägga till nya saker till sprintspekifikationen och dessa delar sedan har varit svåra att spåra. Resultatet enligt U1 blir då att burndown-grafen visar felaktig hastighet på projektet istället för att visa på att flera saker faktiskt har lagts till under sprinten.

I slutet av sprinten började dock utvecklingsgruppen dokumentera, dels med annan färg på lapparna på de uppgifter som lagts till under sprinten, men också genom en punkt på burndown grafen som visar antal timmar på uppgifter som lagts till den aktuella dagen.

Burndowngraf

Det konstiga är att jag faktiskt tycker att det är roligt att uppdatera burndown-grafen (U2)

Burndown-grafen har enligt alla respondenter varit mycket viktig under projektet. Det är det dokumentet som på ett enkelt sätt, kvantitativt, beskrivit för projektdeltagarna hur de legat till. U1 beskriver hur han tidigt kan se effekterna av att denne ibland varit tvungen att tillfälligt behövt hoppa in i ett annat projekt på burndown-grafen. Den dagliga

uppdateringen av burndowngrafen har alltid följts av stort intresse av både projektdeltagare och ibland också andra medarbetare.

Det är en gemensam glädjegrej när man hamnar under linjen, det märkte vi ju. (U1)

Som U1 beskriver ovan blev det, under projektet, ett mål i sig att försöka hålla burndowngrafen under den normativa linjen. Problemet som uppstår här är att utvecklingsgruppens enda verktyg för att visa på hur projektet framskrider är burndowngrafen, dvs. att saker har blivit gjorda. Detta är, enligt respondenterna, en av de mest uppenbara riskerna med Scrum i det aktuella projektet. Ingenstans finns det för utvecklarna ett sätt att visa på *hur* någonting faktiskt har implementerats. Det finns ingen naturlig del i Scrum som visualiserar kvalitet i det man gjort. Även PL1 pekar på samma problem och menar att Scrum premierar kvantitet över kvalitet och att detta är en risk vid långsiktig systemutveckling. U1 menar på att ”man har en yrkesstolthet att faktiskt leverera något bra” räcker för detta. Detta kan säkert stämma överens med vissa personer, men som generell kvalitetsäkring måste det sägas att det är undermåligt. Vad som bör nämnas i detta sammanhang är att Scrum faktiskt inte utger sig för att vara en ingenjörsciens (hur system implementeras) utan en projektstyrningsmetod. Därav den stora populariteten att använda Scrum med t.ex. Extreme Programming eller liknande metoder.

Flera av utvecklarna nämner också vissa svårigheter att visualisera uppgifter man har adderat till sprintspecifikationen under sprinten. Detta löstes dock senare genom att varje dag dokumentera hur mycket arbete man har lagt till.

Sammanfattningsvis kan burndowngrafen under projektet beskrivas som ett väldigt bra verktyg som ger en tydlig bild av hur mycket som faktiskt återstår. Dock skall, som nämnts, ett varningens finger höjas. Risken att utvecklingsgruppen blir alltför fokuserad på att leverera kvantitet och att då de kvalitativa aspekterna av systemet går förlorade.

Risker vid införandet av Scrum

Beroende mellan delar

De olika delarna i Scrum är relativt beroende av varandra och påverkar självfallet ett inkrementellt införande av Scrum. Jag har då, med fallstudien, intervjuerna och observationerna som bakgrund, valt att dela in Scrum i 4 självständiga delar där varje del självständigt går att införa i en projektgrupp. Dessa delar är:

- Artefakter (burndowngraf, sprintspecifikation och produktspecifikation)
- Dagligt möte
- Flöde (det iterativa momenten)
- Roller (produktägare, scrummaster och utvecklingsgruppen)

Artefakter

Flera respondenter pekade på burndowngraf, sprintspecifikationen och produktspecifikationen som vitala och relativt självständiga delar i Scrum. Sprintspecifikationen antyder dock på ett iterativt arbetssätt och är på så sätt inte helt självständig från Scrums övriga principer.

Dessa delar ger en god inblick i utvecklingen av projektet och omedelbar återkoppling till utfört arbete, men också till deras tidsuppskattningar. Tidsuppskattningarna och eventuella avvikelser från dessa blir påtagligt visuella med dessa verktyg vilket inte alltid behöver uppskattas bland utvecklarna. Enligt Sutherland (2007) är detta dock en nödvändighet i ett steg att bli bättre på att tidsuppskatta. Vad som också är viktigt att beakta med just dessa tre verktyg är de inte nämnvärt påverkade projektgruppens rutiner. Självklart måste dokumenten uppdateras och hållas levande, men oftast är det här, enligt PL1, dokument som finns i varje projekt i någon variant (dock vanligtvis hos projektledaren och inte för allmän beskådan). Med utgångspunkt i de positiva reaktionerna och den relativt mindre påverkan dessa verktyg har på projektgruppen i termer av organisation och rutiner kan risken anses vara liten att införa dessa i en projektgrupp.

Dagliga möten

Den andra delen som respondenterna ansåg vara en relativt fristående del i Scrum som levererar många positiva effekter på projektgruppen var de dagliga mötena. Rent praktiskt så förutsätter de dagliga mötena med de tre frågorna: ”Vad gjorde du igår?”, ”Vad skall du göra idag?” och ”Finns det några hinder som hindrar dig från att göra det du skall?” ingenting av de övriga delarna i Scrum.

Dock uttrycker flera av respondenterna en initial skepsis och känsla av att det blir för mycket möten och att mötena var ineffektiva. Detta är en vanlig första reaktion mot de dagliga mötena, men brukar efter tid försvinna (Deemer & Benefield, 2006), vilket de gjorde även i fallstudien.

I förhållande till de kvantitativa verktygen innebär de dagliga mötena ett förhållandevis stort ingrepp i projektgruppens och de individuella projektmedlemmarnas rutiner. Det tar

också enligt observationerna och intervjuerna en tid innan man får dessa möten effektiva. Detta innebär en större risk att inte lyckas i införandet.

Flödet

Iterativ utveckling i sprintar, planeringsmöten, granskningsmöten och retrospektiva möten är enligt respondenterna beroende av varandra och bör införas som en helhet.

Det iterativa momentet i utvecklingsarbetet, med färdig programvara vid slutet av sprinten, innebär en relativt stor förändring för utvecklingsgruppen och projektledningen. Detta arbetssätt påverkade även tekniken inom projektet då iterativ utveckling kräver mindre ansats till driftsatt programvara då driftsättning sker oftare än utan iterativ utveckling.

Det iterativa arbetet synliggör också projektet. Vid slutet av en iteration blir det ibland smärtsamt uppenbart vad som har gjorts och vad som inte har gjorts. Precis som med artefakterna där projektets framgångs synliggörs ökar också det iterativa arbetet transparensen i projektet.

Scrumms flöde och iterativa angreppssätt ställer också högre krav på projektplaneringen. Inför varje iteration hålls ett planeringsmöte och i början var det självklart svårt att få rätt mängd arbete inom en iteration. Precis som Scwhaber (2001) säger så synliggör Scrumms iterativa egenskaper svårigheter och hinder på ett rättfram sätt. Problemet med detta var att jag under studien observerade finns en risk att projektgruppen istället för att uppfatta denna återkoppling som ett problem som går att lösa, uppfattas den istället som ett betyg på deras förmåga att planera eller utföra ett systemutvecklingsprojekt. Självklart innebär där här en stor risk för mottagandet av Scrum i en utvecklingsgrupp och i förlängningen införandet av Scrum i en organisation.

Roller

Eftersom Scrum definition av roller tillämpats i ganska låg grad inom det aktuella projektet är det svårt att presentera vilka risker och motstånd som kan uppstå med dem. Dock pekar ju det faktum att projektgruppen och Stendahls.net inte har använt sig utav att de är svåra att tillämpa. I vårt fall var det svårt att hitta en helt given produktägare med ensamrätt i prioriteringen över vad som skall implementeras. Också Scrummaster rollen kändes ganska svårt att tillämpa på ett bra sätt. Scwhaber (2001) och Sutherland (2006) pekar ju dock på att de här två rollerna är viktiga för Scrum som metod, och vilka effekter projektgruppen gick miste om genom att inte tillämpa dessa i särskilt hög grad kan jag bara spekulera om.

Utvecklingsgruppens roll var däremot lite enklare att ta till sig även om det i vårt fall inte innebär någon större förändring då utvecklingsgrupperna tidigare har varit relativt självständiga i sitt arbete och tagit stort ansvar för den slutgiltiga produkten.

Sammanställning av ramverk

Med utgångspunkt i de fyra delarna av Scrum som anses kunna införas självständigt erhålls nedan det slutgiltiga ramverket i tabellen (Tabell 2: Slutgiltigt ramverk) nedan.

| | Positiva effekter | Motstånd/risker |
|---------------|---|---|
| Artefakter | Projekt transparens Direkt återkoppling | Tidsuppskattningar blir visuella. Risk för fokus på kvantitet istället för kvalitet. |
| Dagliga möten | Vardagsstruktur Socialisering Kommunikation | Ineffektiva möten. Stor ”overhead” i början. |
| Flöde | Mindre stress i projektgruppen Kontinuerlig förbättring av både process och programvara | Större krav på planering. Scrum's metoder för återkoppling kan uppfattas som kritik av utvecklarna |
| Roller | Scrummaster kan balansera krav från affärssidan och teknisksidan. Scrummaster ser till att metoden efterföljs. | Nya ansvarsområden Svårt att hitta givna personer till dessa roller |

Tabell 2: Slutgiltigt ramverk

Vad som går att utläsa från tabellen (Tabell 2: Slutgiltigt ramverk) ovan är att de enklaste delarna i termer av positiva effekter och eventuella risker är artefakter. En burndowngraf och en produktspecifikation utgör inga större risker att möta på motstånd bland utvecklarna. Dock skall man vara uppmärksam på attityden mot tidsuppskattningar och ha insikten att detta är inget man lär sig på en dag.

Dagliga möten innebär ett större ingrepp i projektgruppens rutiner och utlöser initialt ett större motstånd då dessa möten kan anses vara ineffektiva och vara för mycket ”overhead” i projektet.

De iterativa momenten (flödet) är det som i den här fallstudien anses vara den största förändringen vid införandet av Scrum. Detta är en risk att beakta och kan möta motstånd bland projektdeltagarna.

Rollerna kan efter den här studien fortfarande beaktas som ett frågetecken i fråga om risker. Men, som nämnts tidigare, innebär dessa roller förändrade ansvarsområden vilket är en känslig del att ändra inom en organisation. Vad studien dock har visat på är att det är svårt att tillsätta dessa roller med personer från en mer traditionell projektgrupp. Valet faller sig föll sig inte naturligt i fallstudien.

Slutsats

Syftet med den här studien var att undersöka vilka positiva effekter och vilka risker de olika delarna i Scrum innebär. Som presenterat i tabellen ovan (Tabell 2: Slutgiltigt ramverk) visar jag på vilka positiva effekter och vilka risker de olika delarna medför. Scrums iterativa moment och rollbegreppen visar sig medföra en högre risk vid införandet än artefakterna och det dagliga mötet.

Det går inte utifrån studien att föreslå en faktisk ordning i vilka dessa delar skall införas. Alla organisationer har olika förutsättningar inför en Scrum implementering. Dock genom att tydliggöra riskerna med de olika delarna fungerar ramverket som ett diskussionsunderlag.

Hos Stendahsl.net AB har införandet av Scrum enligt respondenterna avlöp mycket bra. Projektets omfattning och utvecklingsgruppen erfarenhet och intresse av Scrum är självklart en bidragande orsak till att Scrum har fungerat så bra i fallstudien. Genom att utelämna delar av Scrums rollbegrepp upplever också respondenterna att Scrum varit väldigt smidigt att implementera. Flera av respondenterna menar också att i ett projekt av denna omfattning inte behöver Scrums rollbegrepp utan fungerar väldigt väl med redan invanda roller och ansvarsområden.

6. Referenser

Backman J. (1998). Rapporter och uppsatser, Studentlitteratur, Lund.

Banks, Richard Organizational change and Scrum

<http://richardsbraindump.blogspot.com/2006/02/organizational-change-and-scrum.html>
(2007-12-16)

Benefield G. (2007) Rolling out Agile at a large Enterprise, Yahoo Inc.

Brady, Kevin, 2006

AGILE /SCRUM Fails to get to grips with Human Psychology.

<http://www.claretyconsulting.com/it/comments/agile-scrum-fails-to-get-to-grips-with-human-psychology/2006-08-17/> (2007-12-11)

Cornford T, Smithson S. (2006). Project Research In Information Systems, Palgrave Macmillian

Engwall B, Emma Jacobsen (2007) Agil systemutveckling – en jämförelse mellan den agila och traditionella projektledaren

Elsameadisy, A. (2007). Is Scrum Atomic? Tillgänglig:

http://www.infoq.com/news/2007/05/Is_Scrum_Atomic (2008-02-11)

Fowler, M. (2005) The New Methodology. Tillgänglig:

<http://www.martinfowler.com/articles/newMethodology.html> (2007-11-21)

Gunnarsson R, (2006) Forskningsmetodik - Kvantitativa (statistiska) och kvalitativa ansatser. Tillgänglig: <http://infovoice.se/fou/bok/10000033.htm> (2008-02-11)

Hallgren, Magnusson (2007) Tillämpning av lättroliga systemutvecklingsmetoder, Institutionen för tillämpad informationsteknologi, Göteborg Universitet.

Jacobsen D. I., Thornvik J. (2002). Hur moderna organisation fungerar

Jason Yip, It's Not Just Standing Up: Patterns of Daily Stand-up Meetings

<http://martinfowler.com/articles/itsNotJustStandingUp.html>

Juell-Skiels G. Irrlärar frälser oss inte. Tillgänglig:

<http://computersweden.idg.se/2.2683/1.128201> (2008-02-11)

Kniberg H. (2004). Scrum and XP from the trenches. Tillgänglig:

<http://www.crisp.se/henrik.kniberg/ScrumAndXpFromTheTrenches.pdf> (2008-02-11)

Leffingwell D., Smits H. (2005). CIO Playbook for adopting Scrum. Tillgänglig:

http://www.leffingwell.org/Document_Store/CIO_Playbook_For_Adopting_Scrum_080805.pdf (2008-02-11)

Lunell H. (2003) Fyra rundor med RUP,

Mathiassen L, Munk-Madsen A, Nielsen PA, Stage J, Objektorienterad analys och design,

Pete Deemer and Gabrielle Benefield (2006) Scrum Primer, Yahoo

Schwaber K. Beedle M. (2001). Agile Software Development with Scrum, Prentice-Hall

Schwaber K. (2003). Agile Project Management with Scrum, Microsoft Press,

Skowronski, V., 2004, Do Agile Methods Marginalize Problem Solvers?

Takeuchi H, Nonaka I (1986) The New New Product Development Game, Harvard Business Review

Scrum Scrum Scrum

<http://blogs.msdn.com/dtjones/archive/2006/01/11/511586.aspx>

Schwaber K (200), Scrum et al. Tillgänglig:

<http://video.google.com/videoplay?docid=-7230144396191025011> (2008-02-11)

Sutherland. J. (2006). Scrum tuning: Lessons Learned at Google, Google TechTalk, 2006. Tillgänglig:

<http://video.google.com/videoplay?docid=8795214308797356840> (2008-02-11)

Sutherland. J. (2007). On Scrum and Not-Scrum. Tillgänglig:

<http://www.infoq.com/interviews/jeff-sutherland-scrum-rules> (2008-02-11)

Striebeck M (2006) Ssh! We are adding a process... (at Google), Google Inc.

Spayd M., Little J. Little and Spayd on Agile and Organizational Change (2007)

Tillgängligt: <http://www.infoq.com/interviews/agile-organizational-change-little-spayd>

7. Bilagor

Intervjuunderlag

Presentation av uppsatsen
IT-universitetet
Magisteruppsats

Upplägg av intervjun

Formella frågor
Frågor angående projektet
Frågor angående Scrum
Frågor angående Scrum för din roll i projektet

Formella frågor

Hur länge har du verkat som systemutvecklare/systemarkitekt?
Hur länge har du arbetat inom de aktuella projektet?
Beskriv din roll i projektet.

Scrum

Nu har vi använt oss av Scrum i 2 månader, hur anser du att det har fungerat?
Vad är ”att lyckas i ett projekt” för dig?
Anser du att du har lyckats bättre i ditt arbete som systemutvecklare när vi har använt oss av Scrum?
Anser du att du har fått tillräckliga kunskaper om Scrum?
Vilka positiva följder anser du att Scrum har gett dig som utvecklare?
Vilka negativa följder anser du att Scrum har gett dig som utvecklare?
Vilka delar inom Scrum anser du vara kritiska framgångsfaktorer för att Scrum skall vara fungera för dig?
Varför tror du Scrum i det här projektet har lyckats/misslyckats?
Anser du det vara enkelt eller svårt att tillämpa Scrum?
Tror du att det skulle vara enklare att införa Scrum genom att endast börja med vissa utvalda delar av metoden?
Vilka delar skulle detta vara?
Varför skulle dessa delar vara enklare att börja med?

Scrum

Roller

Product owner rollen
Scrummaster rollen
The team rollen

Artefakter

Sprint backlog
Product backlog
Release backlog
Burndown graph

Sprintar

Daily meeting
Sprint planning
Sprint review