

SCRUM i ett Systemteoretiskt perspektiv

Inledning och syfte.....	2
Vägen till SCRUM	2
(Mycket) kort introduktion till Scrum	3
Fördjupad analys.....	5
"Comittment" v.s. "Involvement"	5
<i>Regelverket i Scrum</i>	5
Processöversiktsbilden.....	6
Definitionsproblem av Product Backlog Items.....	9
Selforganizing teams.....	9
Komplexitet.....	10
Systemteoretiska betraktelser som kan förbättra chanserna för Scrum-projektet att lyckas	11

Inledning och syfte

Jag har personligen sett hur scrum-team insprängda i en traditionell organisation får stora problem, och tron att det är "ingen dokumentation" som gäller har orsakat nedläggning av flera scrum-projekt.

Hur kan systemteori förbättra SCRUM-konceptet? Vilka artefakter bör vara med ur ett systemteoretiskt synsätt för att scrum skall få ökade chanser att lyckas som metodologi?

Jag avser i denna rapport (mycket) kort introducera Scrum som metod och därefter diskutera ett antal Scrum-detaljer utifrån systemteoretiskt perspektiv och hur Scrum tar till vara och representerar de grundläggande idéerna från Takeuchi & Nonaka, Senge och det Agila Manifestet.

Slutligen avser jag sammanfatta några övergripande rekommendationer.

Notera att detta inte på något sätt är en fullständig redogörelse för alla aspekter av Scrums funktion, regler och koncept.

Vägen till SCRUM

Scrum attribueras ofta som en Agil utvecklingsprocess utvecklad av Jeff Sutherland när han jobbade på avdelningen för Object Technology på Easel Corporation in 1993, men faktum är att idéerna formulerades långt tidigare. 1986 publicerade Takeuchi och Nonaka en artikel i Harvard Business Review, där de detaljstuderat multinationella Amerikanska och Japanska företag som Honda, Hewlett-Packard, Canon, Epson, 3M, Fuji-Xerox, Brother och NEC. Gemensam för dessa företag var att de aktivt jobbade för att skapa nya mer effektiva sätt att utveckla produkter. De båda professorerna vid Hitotsubashi universitetet i Japan gjorde sina fallstudier inom avancerad produktutveckling av datorer, skrivare, kopieringsmaskiner och bilar.

Jag kommer att återkomma till denna välskrivna och tankeväckande artikel, som starkt påverkat både utvecklingen av SCRUM och andra agila metoder, och som har flera paralleller med Peter Senges "Den femte disciplinen".

Ytterligare två viktiga datum i denna korta historik är 1995 och 2001.

Jeff Sutherland samarbetade med Ken Schwaber under 1995 och formaliserade en enkel process för Scrum, som de presenterade på den internationella konferensen kring Object-Oriented Programming, Systems, Languages and Applications (OOPSLA'95). För alla skidåkare så appellerar the Agile Manifesto¹ från Februari 2001, mest till minnet, eftersom personerna lade grunden för manifestet vid skidorten Snowbird i Wasatch bergen i Utah. De utvalda 17 representanterna för en rad olika metoder och processinriktningar² enades om vissa grundläggande värderingar, i en gemensam reaktion mot sekventiella och dokumentdrivna "tung" systemutvecklingsmetodiker (som exempelvis RUP). Man betonade behovet av ett nytt sätt att tänka i en alltmer komplex omvärld och det nödvändiga i att lämna det linjära tänkandet och vattenfallsutvecklingen bakom sig.

¹ <http://agilemanifesto.org/>

² SCRUM, XP (Extreme Programming), Feature-Driven Development, DSDM, Pragmatic Programming, Adaptive Software Development etc...



Fig. 1. Förenklat relationsdiagram med "inspirationspilar" (författarens egen framställning)³

Efter 2001 har det Agila manifestet blivit en av hörnstenarna för implementation av Scrum, eftersom alla principerna i manifestet blivit grundläggande koncept i Scrum.

(Mycket) kort introduktion till Scrum

Scrum är en minimalistisk processbeskrivning med endast 4 roller, 3 artefakter, 4 aktiviteter (olika typer av möten), en processöversiktsbild, ett antal grundläggande koncept och regler samt en tankeväckande (rolig) historia. Givetvis existerar varianter och utökningar av dessa grundstenar, men här begränsar jag mig till Jeff Sutherland och Ken Schwabers klassiska avskalade framställning, och kommer inte gå in i detalj på alla regler och beskrivningar inom metoden.

Scrum består i ett systemteoretiskt helhetsperspektiv av följande komponenter:

Roller: (Scrum) Team (med 5-9 Team Members), ProductOwner, Scrum-Master, Stakeholders (dvs. kunder, intressenter och användare).

Artefakter: Sprint Backlog(with Sprint Backlog Tasks), Product backlog (med Product Backlog Items),Burndown chart

Aktiviteter: Daily scrum, Sprint Demo(Sprint Review Meeting), Sprint Retrospective, Sprint Planning Meeting, sprint etc..

Det uttalade målet för ett Scrum system är att under en Sprint (15-30 arbetsdagar) skapa fungerande funktionalitet som är av intresse för Product Owner och de Stakeholders som han/hon representerar (Potentially Shipable/Usable Functionality)

Product Owner (PO) ansvarar för att det finns en Product Backlog med prioriterade Backlog Items som representerar den gällande visionen över den produkt som skall skapas. PO representerar alla existerande stakeholders i projektet ("one source of information" mot teamet) och skall alltid vara tillgänglig för frågor från teamet.

Scrum-master (SM) guidar och coachar teamet till effektiv agil utveckling och ansvarar för att implementera Scrum processen så att den fungerar i den existerande organisationen. SM har kontroll på alla Scrum regler och koncept och fungerar som en "fårarherde som skyddar sin flock"; dvs. gör allt för att inte störa teamets kreativa och produktiva flöde under en sprint.

Scrum Teamet består av 5-9 personer med ansvar att leverera det som man gemensamt accepterat som Sprint Backlog vid Sprint Planning Meeting. Sammansättningen är "cross-functional" och medlemmarna är både analytiker, designers, utvecklare och testare.

³ Jag skulle vilja återkomma och fördjupa analysen av hur idéerna i Takeuchi & Nonakas artikel harmoniserar med Senges tankar och vad som senare valdes ut till att inkorporeras i det Agila manifestet. Kanske i en kommande rapport.

Visualize Understand Communicate TM



Daily Scrum är ett dagligt stand-up möte med absoluta regler som leds av SM: Endast 15 minuter långt, alltid en person i taget som pratar, närvaroplikt och svaren på de standardiserade 3 frågorna: Vad man gjort sen förra Daily Scrum, Vad man tänker göra fram till nästa Daily Scrum och Vad som eventuellt hindrar mig från att göra det jag tänkt göra.

Fördjupad analys

"Comittment" v.s. "Involvement".

Den tankeväckande historien handlar om grisen och hönan som funderar på att öppna en restaurang ihop. Hönan föreslår namnet "ham n' egg". Grisen svarar direkt genom att göra den viktiga distinktionen mellan begreppen "Comittment" och "Involvement". Mike Cohn påpekar att denna historia på ett skämtsamt sätt tydliggör vilka personer i ett projekt som "Having their Bacon on the line"⁴.



Fig.2 Chicken & Pig.

Skämtsamt betonas således skillnaden mellan de som "verkligen utför arbetet" och de som "tjänar på" att något utförs men själva inte riskerar något, även om de bidrar till resultatet. Detta ligger helt i linje med Peter Checklands CATWOE (Checkland & Poulter 2006, p. 41 figure 2.9), där separationen mellan "Customers" (C) och "Actors" (A), kunderna/mottagarna/de som "tjänar på" systemet och de som utför/driver själva aktiviteterna i systemet, är tydlig. Och som Schwaber framlägger det: *"..a clear distinction between these two groups ...ensures that those who are responsible for the project have the authority to do what is necessary for its success and that those who aren't responsible can't interfere unnecessarily"* (Schwaber 2004, p. 7).

Men tyvärr så tappar Scrum bort fokus på både "View of the world (Weltanschauung)", "Transformation" och "Environmental constraints" i sin förenklingsiver. Utan fokus på en fullständig CATWOE, blir det upp till varje enskild teamkonstellation att tvinga fram denna helhetssyn.

Regelverket i Scrum

Relationerna mellan komponenter sammanfattas i ett löst sammansatt regelverk som exempelvis beskriver: vem som ansvarar för vilken artefakt, hur långa möten får vara, vilka som skall vara med på vilka möten och att funktionalitet som inte uppfyller den gemensamt överenskomna definitionen av begreppet "klar, inte får visas på en sprint-demo (antingen är funktionen "klar" eller så är den det inte!). Dessa regler är baserade på pragmatiska och funktionella idéer från Takeuchi & Nonaka, Senge och flera andra. Scrum-reglerna kalibreras av det Agila Manifestets deklARATIONER och inspirerande principer, vilka är starkt influerade av den systemteoretiska tanken att informationen är drivande vid förändring: *"Systemteorin har lämnat energitänkandet och arbetar istället med information som grundmetafor för förändring"* (Öquist 2008, p. 19). Även kontrollen av informationen anses av största vikt, att för mycket information kan vara förvirrande och göra att drivkraften och att "momentet" i projektet försvinner. Exempelvis så lyfter Schwaber fram följande regel för Sprint Review Meetings (Sprint Demo): *"Artifacts cannot be shown as work products, and their use*

⁴ Mike Cohn: <http://www.mountaingoatsoftware.com/?implementingscrum>

⁵ Skapad av Michael Vizdos och konstnären Tony Clark (<http://www.implementingscrum.com/>)

must be minimized to avoid confusing stakeholders or requiring them to understand how system development works” (Schwaber 2004, p.137)

Processöversikt bilden

Processöversikt bilden eller ”modellen” är en viktig komponent i Scrum-systemet och behovet av översikt bilder betonas av Peter Checkland: *“the complexity of human situation is always one of multiple interacting relationships. A picture is a good way to show relationships, in fact it is much better medium for that purpose than linear prose”* (Checkland & Poulter 2006, p.25).

Men frågan är om den aktuella Scrum bilden verkligen räcker till. Checklands Rich Pictures har som syfte att *“capture, informally, the main entities, structures and viewpoints in the situation, the process going on, the current recognized issues and potential ones”* (Checkland & Poulter 2006, p.25).

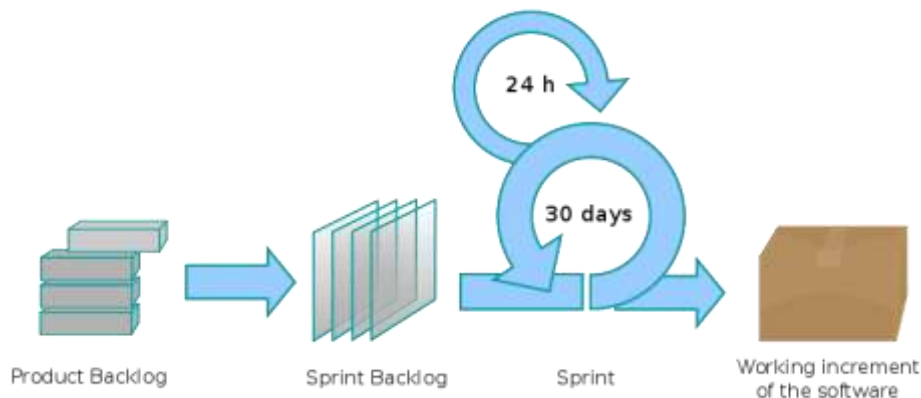


Fig.3 Klassisk Scrum Processöversikt

Processöversikt bilden visar endast hur delar av Product Backlog väljs ut och ger upphov till en Sprint Backlog som i sin tur utgör input till scrumsystemet under en Sprint (som i bilden är fixerad till 30 dagar). Output från systemet är *“Working increment of the software”* (Potentially shippable Functionality).

En av de 12 principerna i det Agila Manifestet är *“Simplicity--the art of maximizing the amount of work not done--is essential”*⁶.

Men det finns en uppenbar risk med att förenkla processöversikt bilden så att den inte ens innehåller alla ingående komponenter. Att varken visa roller eller en systemomgivning, är utifrån ett systemteoretiskt synsätt inte alls *“good enough”*!

Scrum-förespråkare menar att dessa brister fångas upp av de grundläggande koncepten och reglerna, men dessa utgörs av textuell logik som varje användare måste *“addera”* till bilden på sitt eget sätt.

Än värre är att fokus ligger helt på produkten och inte på en av de viktigaste delarna i all agil utveckling: människan! Ytterligare en princip från det Agila manifestet är: *“At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.”* Detta motsvarar Senges princip om Teamlärande (Senge 1990, p.22) och återfinns också hos Takeuchi & Nonaka, som *“Multilearning”*. Genom att lyfta fram teamets ökade förmåga på samma sätt som den utvecklade produkten, tydliggör Scrum-systemets egenskap av *“change agent”* för företaget som helhet och möjligheten till en lärande organisation genom förbättrade/uppdaterade *“tankemodeller”* (Mental Models).

“These examples show the important role that multilearning plays in company’s overall human resource management program. It fosters initiative and learning by doing on the part of the employees and helps keep them up to date with the latest developments. It also serves as a basis for creating a climate that can bring about organizational transition” (Takeuchi & Nonaka 1986, p.7)

Att ständigt matas med liknande felaktiga processbilder, riskerar att hämma förmågan att se helheten och snarare främja ett minimalistiskt tänkande (suboptimering) där det egna teamet är det enda som är av

⁶ Se <http://www.agilemanifesto.org/principles.html>

intresse. Senges Personal Mastery, med den individuella motivationen och passionen, och Team Learning skulle kunna motverka helhetstänkandet och snarare skapa en "vi och dom" känsla i företaget; vi som kör Scrum och de andra som jobbar traditionellt.

En ytterligare faktor i detta spel, är att Scrum mastern ersätter den traditionella projektledare och jobbar mer coachande än styrande och ansvarar för teamets kunskapsutveckling (increased team capability), vilket många projektledare inte vill göra. Inom många företag är dessutom ofta projektledarrollen ett steg på karriärstegen mot högre befattningar. Det är inte säkert att Scrum Masterns tekniknära och fascilerande roll har samma karriärfunktion.

"I worry about what happens when we surround ourselves with process pictures which (1) don't include people, and (2) only tell half the story... With no people, and only half the double-feedback loop, it's silently reinforcing the notion that it's all about the product, and that we humans are mere cogs in the machine..."⁷

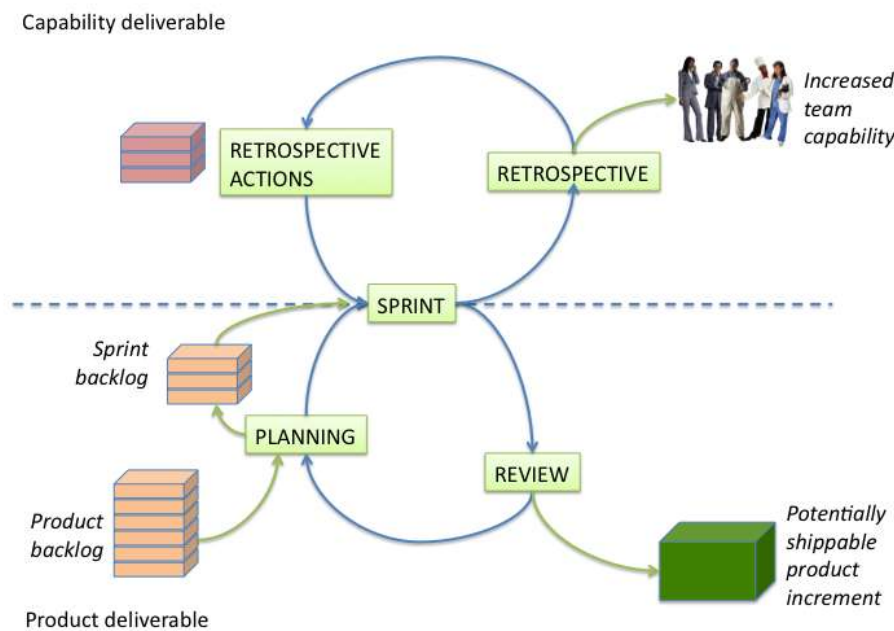


Fig. 4 Uppdaterad Processöversikt bild med både Produkt och Team-fokus.

Genom att utöka översikt bilden till en mer fullständig systembild med tydliga mål, systemgränser, komponenter och input och output flöden, blir den mer komplex och svårare att komma ihåg, men samtidigt mer fullständig och genererar troligtvis mer intressanta frågeställningar som kan få positiva effekter för projektets utveckling.

⁷ Se <http://www.teamsandtechnology.com/>

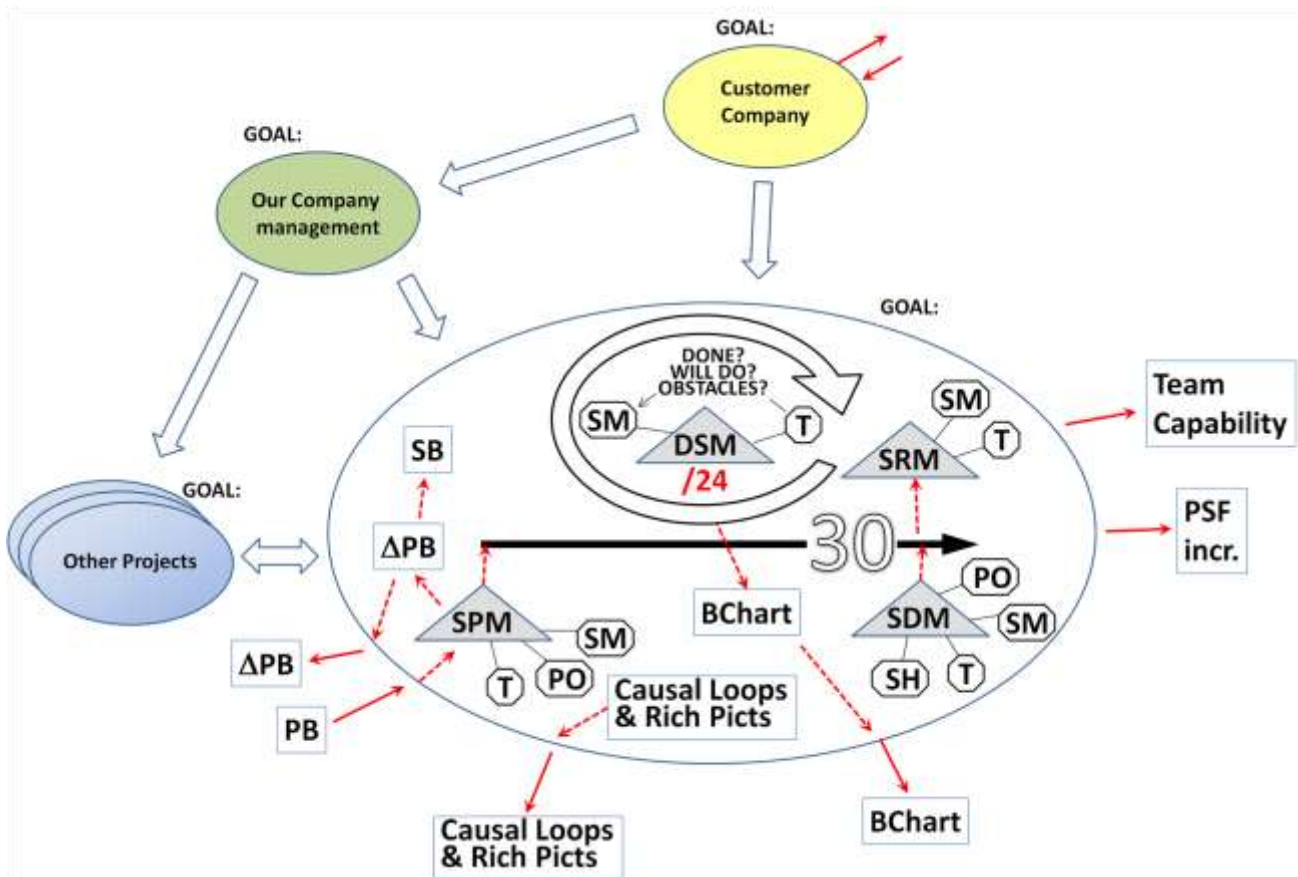


Fig. 5 Förslag på en systemeteoretiskt utvecklad Processöversiktbild för ett SCRUM-projekt (författarens egen framställning).

Artifacts: PB= Product Backlog, SB= Sprint Backlog, Bchart= Burndown Chart, Rich Picts= Rich Pictures, PSF incr.= Potentially Shipable Functionality increment.

Roller: T= Team (består av ett antal Team members), PO= Product Owner, SM= Scrum Master, SH= Stakeholders

Möten: SPM= Sprint Planning Meeting, SDM= Sprint Demo Meeting, SRM= Sprint Retrospective Meeting, DSM= Daily Scrum Meeting

Det i bilden uppförstorade SCRUMprojekt-systemet samspelar enligt lagen om koordination med alla andra eventuella projekt inom företaget (Scrum projekt eller traditionellt drivna projekt). Företagsledningen kalibrerar alla projekten inom företaget genom att tilldela resurser och tidplaner etc.enligt principen om integration. Intressant i sammanhanget är att det externa företag (kunden) som köpt tjänsten av det aktuella företaget kalibrerar både företaget (genom att ställa krav och definiera vad som skall ingå i själva affären) men också direkt mot den/de projekt som ansvarar för den aktuella utvecklingen (exempelvis genom närvaro vid Sprint Demo och vid alla kontakter med Product Owner). Alla system är per definition målsökande, och dessa mål bör vara allerstädes närvarande i "de dagliga visuella planeringsartifakterna", exempelvis inskrivna i processöversiktsskissen. Detta gör att Översiktsskissen anpassas och modifieras till den aktuella organisationen och utvecklingsmiljön. På den generella översiktsskissen kan därmed alla System Goals vara markerade för ifyllnad, vilket ger upphov till nödvändig medvetenhet och fokus på systemmålen. I bilden ovan (bild 5) är målformuleringen för det egna företaget ("Our Company management") exempelvis: "Sälja in och realisera utvecklingsprojekt som genererar vinst för företaget". Kundens företagsmål kan formuleras som: "Göra en investering som löser ett identifierat problem och levererar ett mätbart slutresultat", medans scrum-systemet definitionsmässigt styr mot att "leverera Potentially Shippable Functionality som accepteras av Product Owner och samtidigt öka teamets gemensamma förmåga till iterativ utveckling".

Customer Company-systemet har givetvis både input och output. Viktig input kan exempelvis vara trender och användarundersökningar som kan påverka inriktningen, behovet och nyttan av den påbörjade satsningen. Notera här att Scrum-systemet tillåter att kundsystemet påverkas av sin omgivning och ändrar i det pågående projektet. På samma sätt kommer systemet att påverka sin omgivning genom att information

från de olika Sprint Review Meetings ("Sprint Demo") kommer att påverka företagets relation med omgivningen (om de inte väljer att hålla hela projektet hemligt för konkurrenter etc...)

Även Scrum-systemet har ett antal input och output vid systemgränsen. Viktiga artefakter som Product Backlog. Jag väljer här att skilja på Product Backlog och Product Backlog Delta- PB', där det senare är en delmängd av den förstnämnda; delmängden skapas vid urvalet av PB-items under Sprint Planning Meeting i början av den aktuella sprinten. Ytterligare Output i förslaget är Burndown Chart och "Causal loops & Rich Pictures".

I bilden särskiljs även Artefakter, Möten och Roller.

I ett scrum-system så väljer man att inte mäta de enskilda teammedlemmarnas kompetensutveckling och förmåga, utan enbart HELA teamets förändring och utveckling, exempelvis "leveransförmåga", "Sprint velocity" etc... Orsaken är att det är teamet som skall fungera tillsammans och uppdelning av teammedlemmarna skapar oftast onödig friktion inom systemet, vilket kan påverka teamets leveransförmåga.

Bilden innehåller dock svår att memorera. Detta visar på trade-off effekterna mellan ökad detaljeringsgrad och minskad läsbarhet och reducerade minnesmöjlighet av översiktsskissen.

Definitionsproblem av Product Backlog Items

Det agila Manifestets andra princip är: *"Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage"*⁸

Det är Product Owner som ansvarar för att det existerar en Product Backlog, med prioriterade Backlog Items. Problemet är att ofta saknas kunskap om hur detta Item skall utformas och på vilket sätt kraven skall brytas ned och förfinas. Scrum menar att problemen inte kan till fullo förstås eller definieras, och istället för att fastna i definitioner och traditionellt kravarbete, vill man sätta igång teamet direkt när de har en aning om hur de skall ta sina första steg, och sen se till att PO helat tiden finns tillgänglig under resans gång.

Detta sätt att tänka stöds delvis av Öquist: *"vi måste överge den mekanistiska uppfattningen om krafter som verkar på ting... Kommunikation är tvärtom ett resonansfenomen. Det viktiga som sker, sker alltid hos mottagaren. Sändarens uppgift är att vara KATALYSATOR snarare än MOTOR för förändring"* (Öquist 2008, p. 72)

PO har som uppgift att förklara vad han vill ha ut av teamet under den kommande sprinten (definiera inkrementet), I Langefors infologiska ekvation⁹ kan tiden (T) säkras genom Scrumprocessens regler om den obligatoriska PO-närvaron vid behov från teamet och de inplanerade Sprint Planning mötena.

Det saknas en grunddefinition av ett antal enkla och vägledande artefakter som kan underlätta och förbättra insikten om alla stakeholders önskemål och behov.

Selforganizing teams

Takeuchi och Nonaka valde metaforen rugby, för att lyfta fram både den individuella tuffhet och den taktiskt fulländade lagandan som krävs för att vinna. Snabbhet, flexibilitet och samarbete mot ett gemensamt, enkelt och tydligt mål, där alla spelarna deltar och kan växla roller utefter hur det aktuella läget ser ut:

"Companies are increasingly realizing that the old sequential approach to developing new products simply won't get the job done. Instead, companies in Japan and the United States are using a Holistic method – as in rugby, the ball gets passed within the team as it moves as a unit up the field" (Takeuchi & Nonaka 1986, p.2)

⁸ Se <http://www.agilemanifesto.org/principles.html>

⁹ $I = i(D, S, t)$

Scrum är en term inom rugby som betecknar ett sätt att återstarta en match efter en foul, ungefär som en tekning inom ishockey. Och som metafor är begreppet scrum attraktivt, eftersom hela laget samlas i en "scrum-circle" och kämpar tillsammans för sin överlevnad.



Fig.6 Två lag som kämpar i en scrumcirkel (bilden från <http://www.riceryugbyclub.com>).

För att ett lag skall kunna prestera, måste alla spelarna vara på plan; samtidigt! Scrum kräver att alla medlemmarna jobbar på samma plats (co-location).

"No company finds it easy to mobilize itself for change, especially in noncrisis situations. But the self-transcendent nature of the project teams and the hectic pace at which the team members work help to trigger a sense of crisis or urgency throughout the organization. A development project of strategic importance to the company, therefore can create a wartime working environment even during times of peace" (Takeuchi & Nonaka 1986, p.10)

Komplexitet

Systemutveckling är en komplex aktivitet. Enligt Warren Weaver's klassiska komplexitetsuppdelning i 3 steg, så utgör, enligt Flood and Carson, "People Range" ytterligare en dimension: *"We agree that social science is beyond Weaver's three ranges of complexity"*(Flood & Carson 1993, p.35).

Även Ken Schwaber är inne på samma spår genom att identifiera 3 dimensioner för komplexiteten inom systemutveckling: Kravställning, Teknik och Människor.

"The materials that we use to create the end product are extremely volatile: user requirements for a program that the user have yet to see, the interoperation of other programs' signals with the program in question, and the interaction of the most complex organism on the planet – people" (Schwaber 2004, p.1) och *"the people developing the software ..all have different skills, intelligence levels, experience, viewpoints, attitudes and prejudices. Every morning, each wakes up in a different mood than the day before, depending on his or her sleep, health, weather, neighbors, and families. These people then start to work together, and the complexity level goes through the roof."* (Schwaber 2004, p. 5)

På ett sätt handlar utvecklingen av Scrum som metod om att försöka reducera de traditionella komplexitetsfaktorerna som gör att komplexiteten normalt ökar i ett system:

- reducera antalet komponenter genom att definiera ett fåtal roller och artefakter
- minska antalet inbördes relationer genom att "isolera" Scrum-teamet ("clean room development") och skapa "one source of information" medelst Product Owner och Scrum Master rollerna,
- se till att D:et i Langefors infologiska ekvation (informationsdatat) är gemensam för alla genom de planerade Sprint Planning, Sprint Demo och Sprint Retrospective mötena för att
- S:et i Langefors infologiska ekvation kontrolleras genom att alla är närvarande och även om man har olika förkunskaper som ger olika tolkningar, så finns det möjlighet till Team Learning.

- Nonholonomic Constraint minimeras genom Daily Scrums enkla, men tydliga 3-frågor: Antingen fogar "vildhjärnan" sig till teamet eller så får han/hon chansen att hjälpa teamet mot bättre effektivitet genom sina "vilda idéer" (så länge som de accepteras av SM)
- konfrontera olinjära samband och brist på symmetri, med tydliga iterationer och utpekade arbetspaket som inkrement till helheten. Man lägger inte tid på att planera det okända som ändå kommer att ändras.
- Även The Law of Ignorance motarbetas i och med att Scrum **kräver** engagerade deltagare och om inte en representativ PO är utpekad och tillgänglig och kan svara på frågor från teamet under sprintarna, då är det ingen Scrum!

I en komplex värld är Daily Scrum ett exempel på "en vägg av konstanter", som "står för det säkra och förutsägbara, behövs som regulatorer i ett system" (Öquist 2008, p. 34).

Dessa dagliga möten är det som håller ihop hela Scrum processen och bygger på Närvaro.

Systemteoretiska betraktelser som kan förbättra chanserna för Scrum-projektet att lyckas

En av anledningarna till framgången med Scrum är att processen så enkel att lära sig och att förstå. Scrum påstås vara metoden och arbetssättet för snabba, föränderliga och intensiva projekt, med ett arbetssätt som gör att teamen kan uppnå underverk på kort tid. Men Ken Schwaber påpekar själv att, "Scrum is hard. It's not hard because of the things you do; it's hard because of the things you don't do" (Schwaber 2004 p.ix)

Många systemutvecklingsgrupperingar har upplevt ett stort antal processer och metoder genom åren. Stora processer skapar ofta frustration hos användarna med ett överföd av mallar och dokument. Projekten har en tendens att bli dokument-drivna och frustrerade medarbetar upplever att de "dokumenterar mer än de jobbar"¹⁰. Scrum erbjuder således "en väg ut", men enligt min mening så har man drivit förenkling några steg för långt.

Den första viktiga distinktionen att göra vid en systemanalys, brukar vara Transformation (T) och View of the world (W): "many people find it useful, when model building, to start the process by defining first T and W, then the other CATWOE elements" (Checkland & Poulter 2006, p.42)

Schwaber och Sutherland är och var starkt påverkade av Takeuchi & Nonaka och Peter Senge (även om de inte säger det rent ut).

Det finns givetvis Scrum team som modellerar, visualiserar och tydliggör hela CATWOE, men det är inte en framlyft princip i Scrum och därför en onödig risk att helhetssyn och sammanhang ignoreras.

Jag anser att det är viktigt att skilja på olika typer av artefakter. Systemteoretiska causal loops och rich pictures etc. ökar förståelsen för alla, och designas så att förståelsen är "intuitiv" Dessa kan då användas och förädlas under Sprintarna (som en del av Daily Scrum) och självklart redovisas på Sprint Demo och eventuellt uppdateras och utökas under Sprint Planning Meeting. Det är viktigt att göra skillnad på dessa förståelseskäpande visualisering och mer utvecklingsnära avancerade UML diagram.

Processöversiktsskildern bör också utvecklas till att bli en fullständig systembild med ett tydligt mål, systemgräns, alla viktiga komponenter och input och output¹¹. Detta måste göras för att man skall kunna se Scrum-projektet som ett system som kan ha effekt på Hela företaget.: "...new product development also acts as a catalyst to bring about change in the organization" (Takeuchi & Nonaka 1986, p.10)

Detta pekar på en viktig aspekt som ofta glöms bort. Valet av SCRUM görs ofta av trend och mode. Detta är viktigt att dra nytta av vid införandet men bakgrunden är baserad på ideer inspirerade av systemteoretiska betraktelser av verkligheten och dessa bör lyftas fram för att ytterligare stärka metodens validitet i

¹⁰ Påståendet baseras på författarens interna processutredningar från bla. Ericsson AB och Volvo.

¹¹ Se författarens förslag: bild 6.

Visualize Understand Communicate

organisationen i jämförelse med alla andra metoder som florerar och som vill ta sig in i verksamheternas kontrollrum.

"...the new approach can act as a change agent: it is a vehicle for introducing creative, market –driven ideas and processes into an old, rigid organization" (Takeuchi & Nonaka 1986, p.2)

Istället för att ersätta "dokumentationen" med "verbal kommunikation", bara för att "verkligheten rör sig" och är komplex, anser jag istället kan man låta sig inspireras av systemteoretiska modeller och visuella analyser för att hitta ett batteri av flexibla visualiseringar som kan användas som kommunikationshjälpmedel, som "lagringsminne" för tidigare diskussioner och som referenspunkt vid förändringar, exempelvis Causal Loop Diagrams, som hela tiden diskuteras och uppdateras.

REFERENSER:

Böcker:

- (Flood & Carson 1993). Flood, R.L. & Carson, E.R. (1993). Dealing with Complexity: An Introduction to the Theory and Application of Systems Science. ISBN: 9780306442995
- (Gustafsson. et.al.1982) Gustafsson, L. et.al. (1982) System och Modell - en introduktion till Systemanalysen. ISBN: 978-91-44-18551-4
- (Schwaber 2004) Schwaber, K. (2004) Agile Project Management with SCRUM. ISBN: 0-7356-1993-X
- (Senge 1990) Senge, P.M. (1990) Den femte disciplinen - den lärande organisationens konst. (sv. övers. Tomas Cato) ISBN: 91-88384-15-2
- (Öquist 2008) Öquist, O. (2008): Systemteori i praktiken. ISBN: 978-91-7205-575-9

Artiklar:

- (Sutherland 2004). Sutherland, J. (2004). "Agile Development: Lessons learned from the first Scrum" (PDF:<http://jeffsutherland.com/scrum/FirstScrum2004.pdf>.)
- (Takeuchi & Nonaka 1986) Takeuchi, H. & Nonaka, I. (1986). "The New New Product Development Game". Harvard Business Review. (PDF: <http://apl-richmond.pbwiki.com/f/New%20New%20Prod%20Devel%20Game.pdf>.)

Websidor

- <http://www.agilemanifesto.org/principles.html>
- <http://www.teamsandtechnology.com/>
- <http://www.implementingscrum.com/>
- <http://www.mountaingoatsoftware.com/?implementingscrum>