



**TITEL:**

Anvendelse Af Burndown Chart i  
Softwareudviklings Scrum Of Scrum  
Teams

**PROJEKTPERIODE:**

Sommer 2017

**PROJEKTGRUPPE:**

Master

**GRUPPEMEDLEMMER:**

[Navn]: James Collin Hansen  
[Studie nr]: 20151214

**SEMESTER:**

10. Semester

**VEJLEDER:**

John Stouby Persson

**ANTAL KOPIER:**

[x]

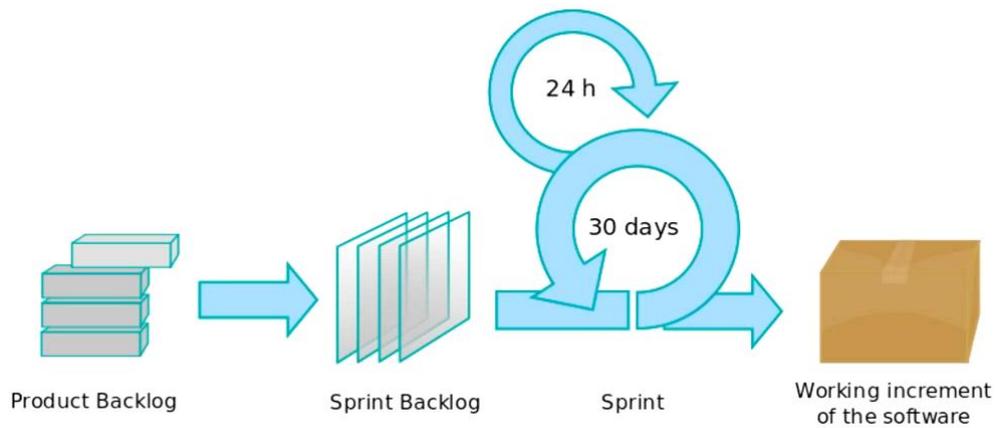
**ANTAL SIDER:**

[54]

**SYNOPSIS**

Scrum of Scrum (SOS) teams in distributed system development need agility and structure: for planning, for people and tasks: Where clarification of practices that contribute to agility is sought after and therefore there is an increasing need for reconsideration of what we understand as agility and the contributed of agility in transition from distributed software development Scrum teams to Scrum of Scrum teams: Without undermining agility, while simultaneously structuring the workflow process. Using a taxonomy, Design science research methodology (DSRM) and a theoretical evaluation, I will try to create a correlation between the contribution of agility in the transition from Scrum to Scrum of Scrum. thereby inspire the thought of rethinking the way of addressing agility in Scrum of Scrum distributed software development projects.

# Anvendelse Af Burndown Chart I Softwareudviklings Scrum Of Scrum Teams



<b>1 INDLEDNING</b>	4
<b>2 TEORETISK BAGGRUND</b>	5
2.1 Backlog, Scrum Board & Burndown Chart	5
2.2 Distribueret Softwareudviklings Scrum Teams	7
2.3 Scrum of Scrum fremdrift timer i story point	11
<b>3 FORSKNINGSMETODE</b>	12
3.1 DSRM Aktiviteter	13
3.2 Software innovation metode og værktøj	15
3.2.1 Configuration Tables, Tactics, Strategy og Rationale	21
3.3 Low tech prototyping	24
3.3.1 Demonstration af Low tech prototyping	24
3.3.2 Eksperimenter med prototyper	25
<b>4 DSRM AKTIVITETER</b>	26
4.1 Problem-Centered Approach	26
4.2 Aktivitet 1 Problem identification and motivation	26
4.3 Aktivitet 2: Define the objectives for a solution	26
4.4 Aktivitet 3: Design and Development	28
4.4.1 Scrum of Scrum Master rolle	28
4.4.2 Product Backlog hierarki struktur for SOS Masteren og prototypen	33
4.4.3 System arkitekturen af Low tech prototypen	33
4.4.4 System anvendeligheden af applikations domænet af Low tech prototypen	34
4.4.5 Use Case diagram for Low tech prototypen	36
4.4.6 Beskrivelse af Actor Specifikationer for Low tech prototypen	38
4.4.7 Løsning af Scrum of Scrum Master rollen og Low tech prototypen	40
4.5 Aktivitet 4: Demonstration	41
4.6 Aktivitet 5: Evaluation	41
4.6.1 Demonstrations evaluering af prototypen	41
4.6.2 Teoretisk evaluering af prototypen og SOS Master	44
4.7 Aktivitet 6: communication	47
<b>5 AKTIVITET 7: CONTRIBUTION &amp; DISKUSSION</b>	47
<b>6 KONKLUSION</b>	49
<b>7 LITTERATURLISTE</b>	50

**Forfatter: James Collin Hansen**  
**Supervisor: John Stouby Persson**

## **Resume**

Scrum of Scrum (SOS) teams i distribueret systemudvikling har behov for agilitet og struktur: for planlægning, for mennesker og opgaver. Hvor afklaring af praksisser som bidrager til agilitet er eftertragtet og derfor er der et stigende behov for gentænkning af hvad vi forstå ved agilitet og bidraget af agilitet i overgangen fra distribueret softwareudviklings Scrum teams til Scrum of Scrum teams: Uden at underminere agilitet, men samtidigt strukturere arbejdsprocessen. Ved hjælp af en taxonomi, Design science research methodology (DSRM) og en teoretisk evaluering vil jeg skabe en sammenhæng mellem bidraget af agilitet i overgangen fra Scrum til Scrum of Scrum med fokus på Burndown Charts. Dermed inspirere til gentænkning af adressering af agilitet, værktøjer der til og inspirere til nye studier i overgangen fra agilitet i Scrum teams til agilitet i SOS distribuerede softwareudviklingsprojekter.

## **Forfatter Keywords**

**DSRM, Agile ISD, Scrum of Scrum, Scrum board, Burndown Chart, Team, Teams og Prototyping.**

## 1 INDLEDNING

Scrum's tilgang og rødder stammer fra forventningen om: At analyse, design og udviklingsprocesser er uforudsigelige. Derfor komprimere Scrum en række faser i mindre enheder kaldet Sprints, som er karakteriseret ved en ikke lineær og fleksible process.

Scrum tilfalder agile metoder under det Agile Manifesto: Individuer og interaktioner frem for processer og redskaber, fungerende software over dokumentation, kundesamarbejde over kontrakter og forhandlinger, og agere på forandring - frem for at følge en plan (*Fowler and Highsmith, 2001 p. 7,1-7 : se eventuelt Agile Manifesto under på engelsk*) og (*Beck et al. 2001*).

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

Under udviklingsprocessen af det samlede projekt er projektet åbent overfor ændringer i sprint planlægningsfasen i projektet. Hvor et projekt bliver ved med at være åbent overfor tids komplekse -, konkurrence -, tids -og kvalitetspræget ændringer. Hvor sprints er anvendt til at udvikle det færdige produkt fra det første sprint, til det sidste sprint i et projekts levetid (*Schwaber Ken and Beedle Mike. 2002*) og (*Fitzgerald, Hartnett, Conboy 2006*).

Dog viser empirisk arbejde at metodeanvendelse i praksis variere i høj grad med varierende kombinationer: Hvorved der foretages til og fravalg i de enkelte anvendte metoder og der derved opstår skræddersyet metode-varianter eller in-house kreationer (*Conboy 2009*), (*Fitzgerald, Hartnett, Conboy 2006*), (*Solingen, Sutherland, Denny. 2011*), (*Kniberg, Henrik. 2007: se Figur 2*) og (*Kniberg, Henrik. 2015*).

Værktøjer som Scrum board'et var forudsat at øge et teams produktivitet og koordinering af aktiviteterne af backlog opgaverne (*Schwaber Ken and Beedle Mike. 2002*), hvor artefaktet (Burndown-chart) har vist sig at være et nyttigt referencepunkt eller værktøj for koordineringen af aktiviteterne i software udviklingsprocessen (*Ken Schwaber 2004*), i informations systemudvikling hvor et Scrum team har mulighed for at præsentere udviklingen af resultaterne af aktiviteterne, have et overblik over udviklingsprocessen og stadig bevare et tidsestimat af hver opgave i Scrum boardet (*Solingen, Sutherland, Denny. 2011*), (*Fitzgerald, Hartnett, Conboy 2006*) og (*Conboy 2009*).

Scrum transformerede nogle vilkårlige software virksomheders udviklingsprocesser, til kontrollerede processer der kunne forudsiges og styres i korte iterationer. Scrum skabte derfor et stærkt værktøj for softwareudvikling, der strategisk kunne kontrollere en proces der tidligere var ustruktureret og ukontrolleret. Scrum gav software organisationer muligheden for at undgå ulemperne af den manglende agilitet og stadig høste fordelene af udvikling af kvalitets software i software organisationen (*Baskerville, Pries-Heje, Madsen, 2011*).

- *"...De har udarbejdet IT-strategier og kravspecifikationer til nogle af de IT-projekter, som siden hen har fået alvorlige problemer. Konsulenternes rådgivning har i nogle tilfælde ikke været god nok. Det gælder blandt andet anbefalinger om at gennemføre projektet som ét stort projekt uden delleverancer..."(Petersen 2001).*

- “Agile information systems development (ISD) has received much attention from both the practitioner and forsker community over the last 10–15 years. However, it is still unclear what precisely constitutes agile ISD.”(Baskerville, Pries-Heje, Madsen, 2011).

Siden hen er software organisationer og softwareudviklingen blevet mere konkurrencepræget og distribueret (Persson et al. 2009): Dertil er flere og flere softwareudviklings organisationer begyndt anvende distribueret softwareudvikling Scrum teams, for at opnå mere agilitet (Conboy 2009), (Fitzgerald, Hartnett, Conboy 2006) og (Baskerville, Pries-Heje, Madsen, 2011). Dette som et led i at øge deres konkurrenceevne globalt og stadig fastholde agilitet i software organisation (Sutherland, Schoonheim, Guido. 2008), (Petersen 2001) og (Baskerville, Pries-Heje, Madsen, 2011).

Da et fysisk Scrum board med et Burndown-chart ikke kan flyttes til hvert enkelt distribueret lokation. Vil jeg i denne rapport ved hjælp af en taxonomi, Design science research methodology (DSRM) og en teoretisk evaluering: Adressere agilitet og skabe sammenhæng mellem bidraget af agilitet i overgangen fra Scrum til Scrum of Scrum (SOS). Derfor er mit forskningsproblem følgende:

*Hvordan anvendes et burndown-chart, i et Scrum board, i distribueret softwareudviklings Scrum teams?*

## 2 TEORETISK BAGGRUND

### 2.1 Backlog, Scrum Board & Burndown Chart

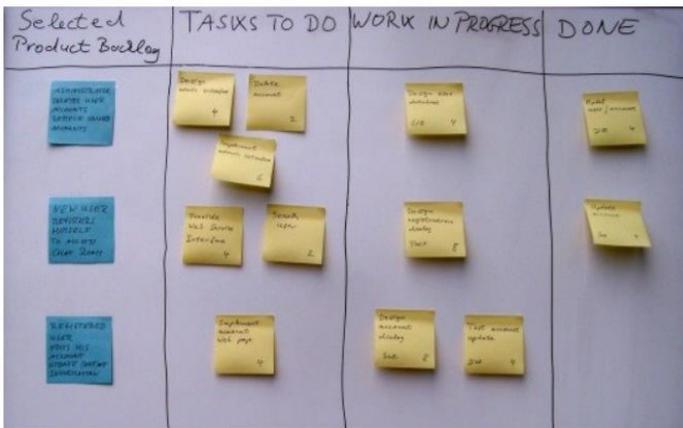
Agile softwareudvikling er en iterative tilgang frem for en sekventiel tilgang der adskiller sig fra vandfaldsmodellen (Fitzgerald, Hartnett, Conboy 2006): I at agile softwareudvikling er centraliseret omkring inkrementelle leveringer eller leveringer i iterationer, som et led i at skabe kvalitets software baseret på feedback se (Figur 13).

Hvor det forandrende miljø og de ukontrollerede specifikationer er en del af udviklingen i agile teams softwareprojekter. Hvor teamet er afhængigt af opgaver eller specifikationer med et estimeret tidsestimat: Hvor teamet effektivt kan reagere på forandringer i specifikationer og opgave specifikationen i det forandrende miljø (R. de Melo, A. Goldman, and C. Melo. 2013), (Fitzgerald, Hartnett, Conboy 2006) og (Schwaber Ken and Beedle Mike. 2002).

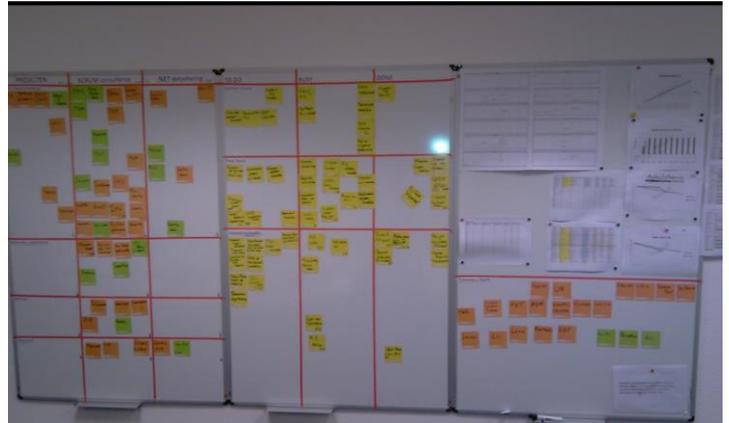
I agile teams anvendes Sprint Planning Meetings: Hvor Sprint Planning Meetings er den mest anvendte tilgang blandt agile praktikke og softwareudvikler hvor der er integreret selvorganiseret Daily Standups eller Daglige Scrum møde. Hvor teamet samles foran en tavle, et task board eller Scrum board se (Figur 1) og opdatere deres arbejdsstatus opgaver eller specifikationer.

Hvor et Task board eller Scrum board repræsenterer en Sprint backlog: Hvor Sprint Backloggen giver teamet muligheden for at tilføje nye opgaver, identificere nye opgaver, dele opgaver i mindre bidder, italesætte opgaver der ikke kan færdiggøres under et sprint, fjerne eksisterende opgaver og til sidst opdatere værktøjet (Burndown-Chart'et) som referencepunkt for team fremdriften se (Figur 4: Sprint Burndown-chart:),(Rubart. 2014).

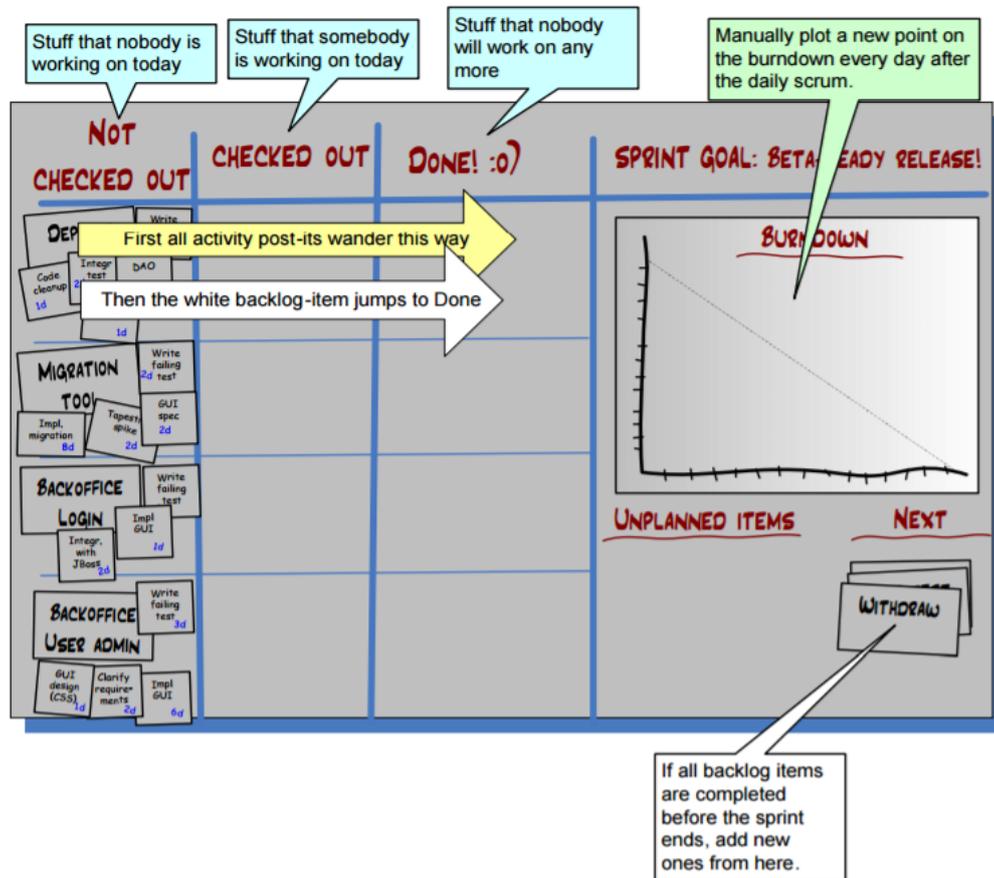
**Figur 1: Scrum Task Board med papir-baseret Backlog opgaver og specifikationer**  
(Rubart. 2014. p 2)



**Figur 2: Scrum board med burn-up chart**  
(Solingen, Sutherland, Denny. 2011)



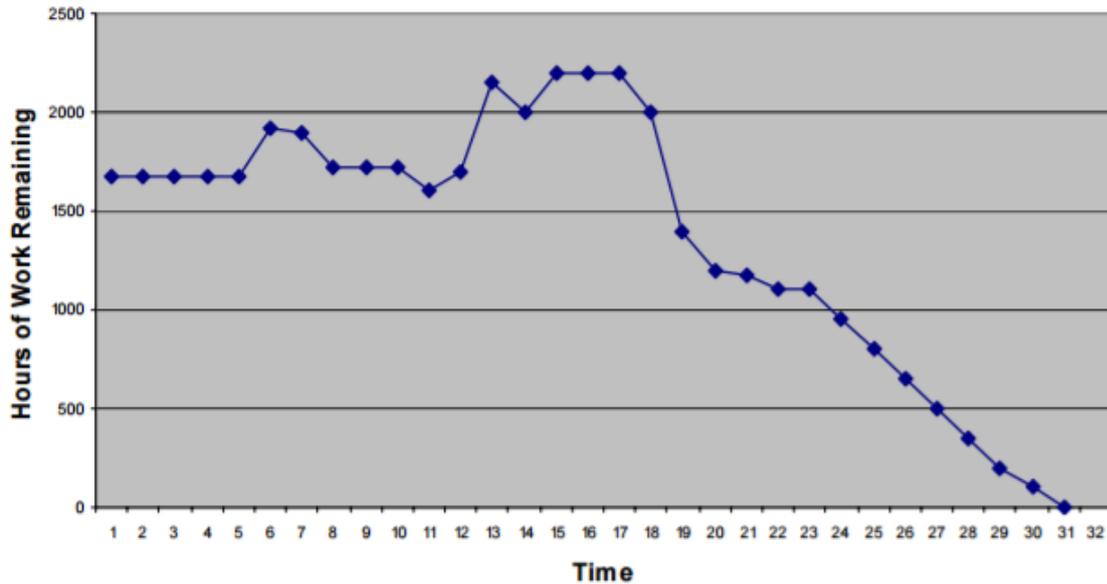
**Figur 3: Scrum board med burndown-chart**  
(Kniberg, Henrik. 2007) og (Kniberg, Henrik. 2015)



Selvom begge Scrum board er forskellige repræsenterer de stadig et Scrum board og værktøjet Scrum board: I det første Scrum board (Figur 1) er der repræsenteret et vilkårligt antal gule postits med opgaver. Hvor de er organiseret i en række kolonner og et burn-up chart. I Figur 3 repræsenteres Scrum boardet med en række hvide posted med opgaver og et Burndown-chart (Fitzgerald, Hartnett, Conboy 2006) og (Schwaber Ken and Beedle Mike. 2002).

**Figur 4: Burndown Chart**  
(Sutherland and Schwaber. 2007. s 15)

**Sprint Burndown Chart**

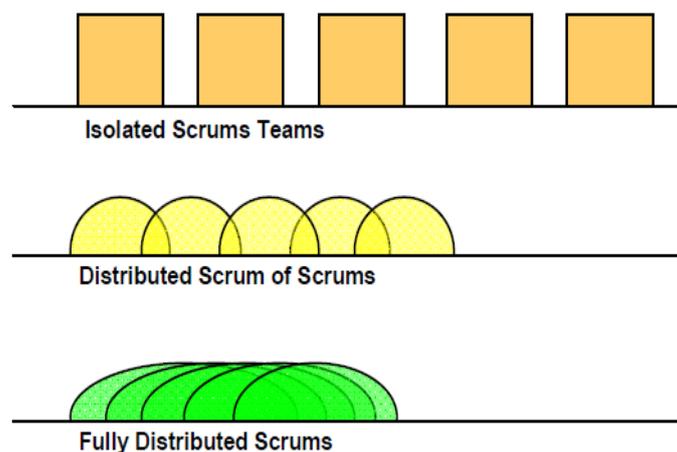


**2.2 Distribueret Softwareudviklings Scrum Teams**

I (Sutherland, Schoonheim, Guido. 2008) og (Sutherland og Schwaber. 2007) bliver distribueret softwareudviklings Scrum teams beskrevet som følgende 1. Isolated Scrums, 2. Distributed Scrum of Scrums og 3. Fully distributed Scrums: Hvor 1. Isolated Scrums: Der er isoleret geografisk Scrum teams, 2. Distributed Scrum of Scrums: Der er isoleret geografisk Scrum teams, som samles i Scrum of Scrum møder, og 3. Fully distributed Scrums: Hvor Scrum teamet er et selvorganiserende team med andre distribueret softwareudvikling Scrum teams teammedlemmer se (Figur 5), (Sutherland og Schwaber. 2007. s. 71-73) og (Mundra; Misra og Dhawale. 2013), (E. Hossain, A. B. Muhammad og y. P. Hye. 2009).

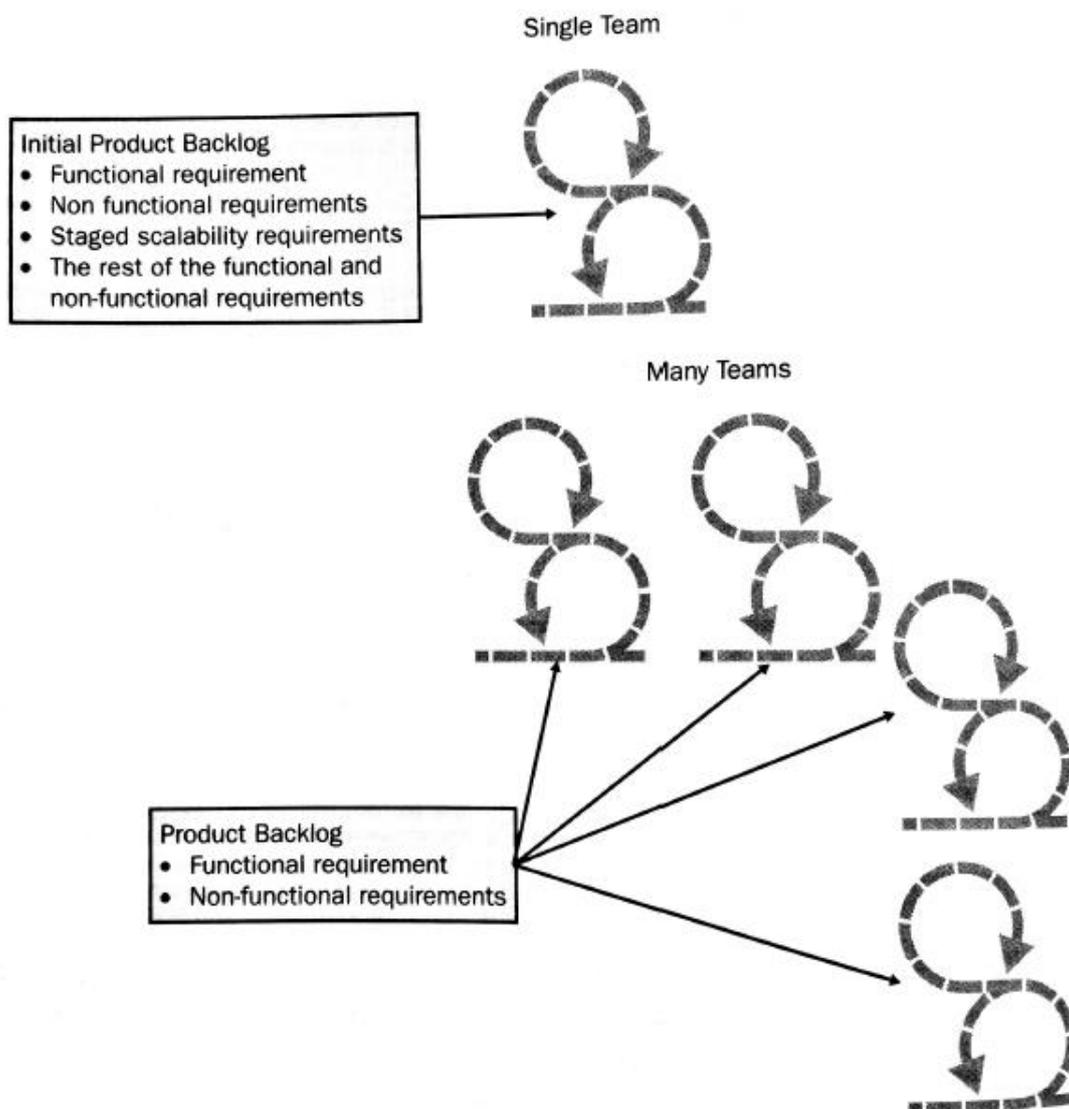
**Figur 5: Isolated Scrums Teams, Distributed Scrum of Scrums and Fully Distributed Scrums**

(Sutherland og Schwaber. 2007 s. 72) og (Sutherland, Schoonheim, Rijk. 2009. s. 2)



Scrum of Scrum (SOS) er flere Scrum teams der koordinere og arbejder sammen om et fælles produkt med et fælles mål: Hvis der derimod fokuseres på 2 forskellige produkter, er det to forskellige Scrum teams med hver deres mål og hver deres produkt. Scrum of Scrum er placeret på flere lokations og hvert team har et Scrum team og en Scrum Master. Hvor dette tager udgangspunkt i aktiviteterne af koordineringen af software produktudviklingen i flere lokation, hvor koordineringen af et fælles produkt er centraliseret i et Scrum of Scrum afhængig af flere distribueret teams. Hvor teamene koordinere via Scrum of Scrum teams møder og en eller flere repræsentanter fra hvert distribueret team deltager Se (Tabel 1), (Mundra; Misra og Dhawale. 2013), (E. Hossain, A. B. Muhammad og y. P. Hye. 2009), (A. Lamersdorf, M. Jürgen and R. Dieter. 2009), (Sutherland and Schwaber. 2007) og (Maria, Casper og Ville. 2012).

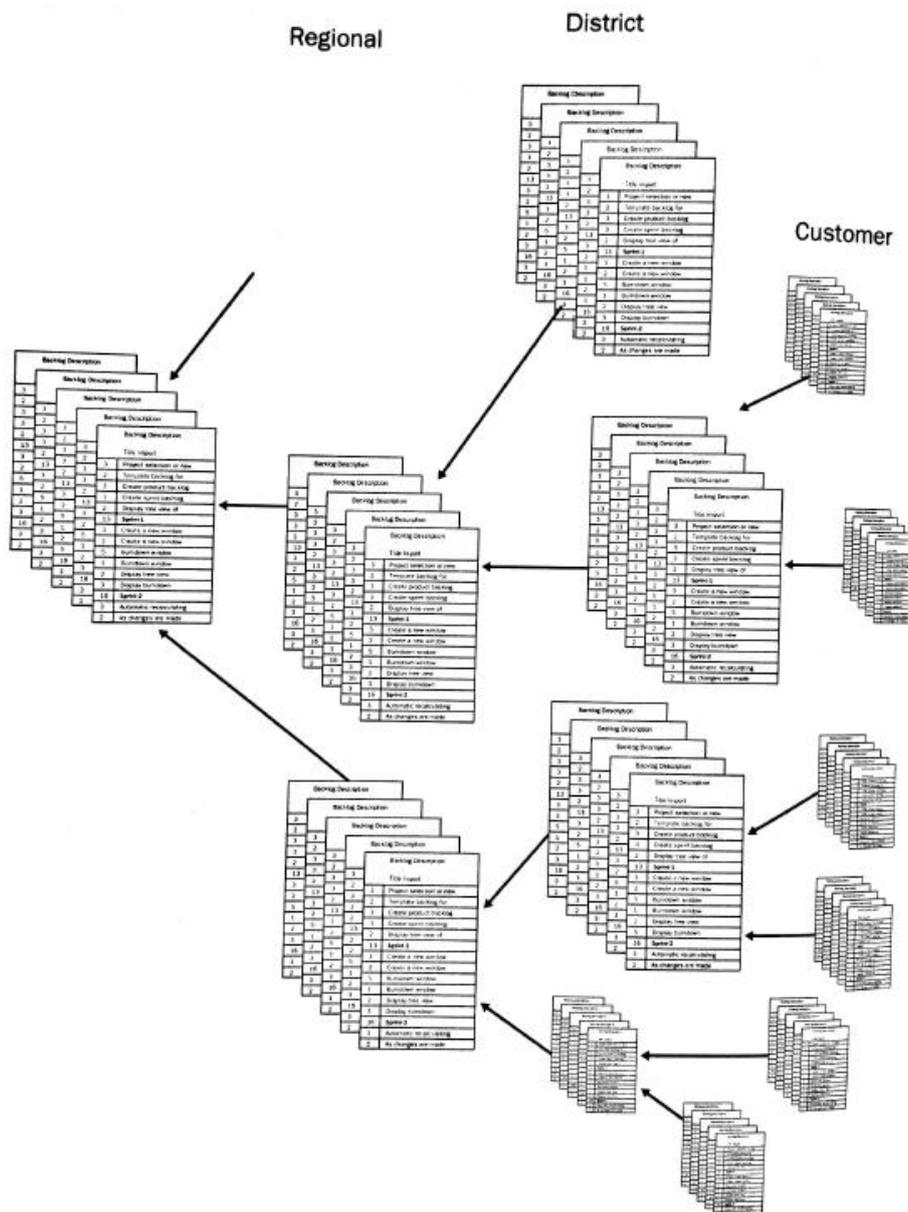
**Figur 6: Skaleret scrum, et scrum team og flere Scrum teams**  
(Ken Schwaber 2004)



Ifølge (Tabel 1: Scrum of Scrum team struktur, punkt 1) bør Product Owner i Scrum of Scrum dele den samme eller de samme Product Owner og koordinere at repræsentere et ansigt for teamet udadtil også til andre teams. Product Ownerne varetager derfor samme Product Backlog hvor Product backlogen altid repræsenterer opgaver af højeste prioritering og funktionalitet koordineret af Product Ownerne fra aktiviteterne af koordineringen af Backloggen. Men Product Ownerne varetager samme Product backlog, hvor i Product backlogen altid repræsenterer opgaver af højeste prioritering og funktionalitet koordineret af Product Ownerne eller af 1 Product Owner.

Men at en Scrum Master rolle er repræsenteret i hvert enkelt team og hvert team har deres egen Scrum Master. Hvor i Scrum Masteren deltager i alle møderne også i Scrum of Scrum møderne hvor i koordineringen af aktiviteterne foregår af den fælles Product Backlog af Scrum of Scrum Product Backlogsne se (Figur 7) og (Ken Schwaber 2004).

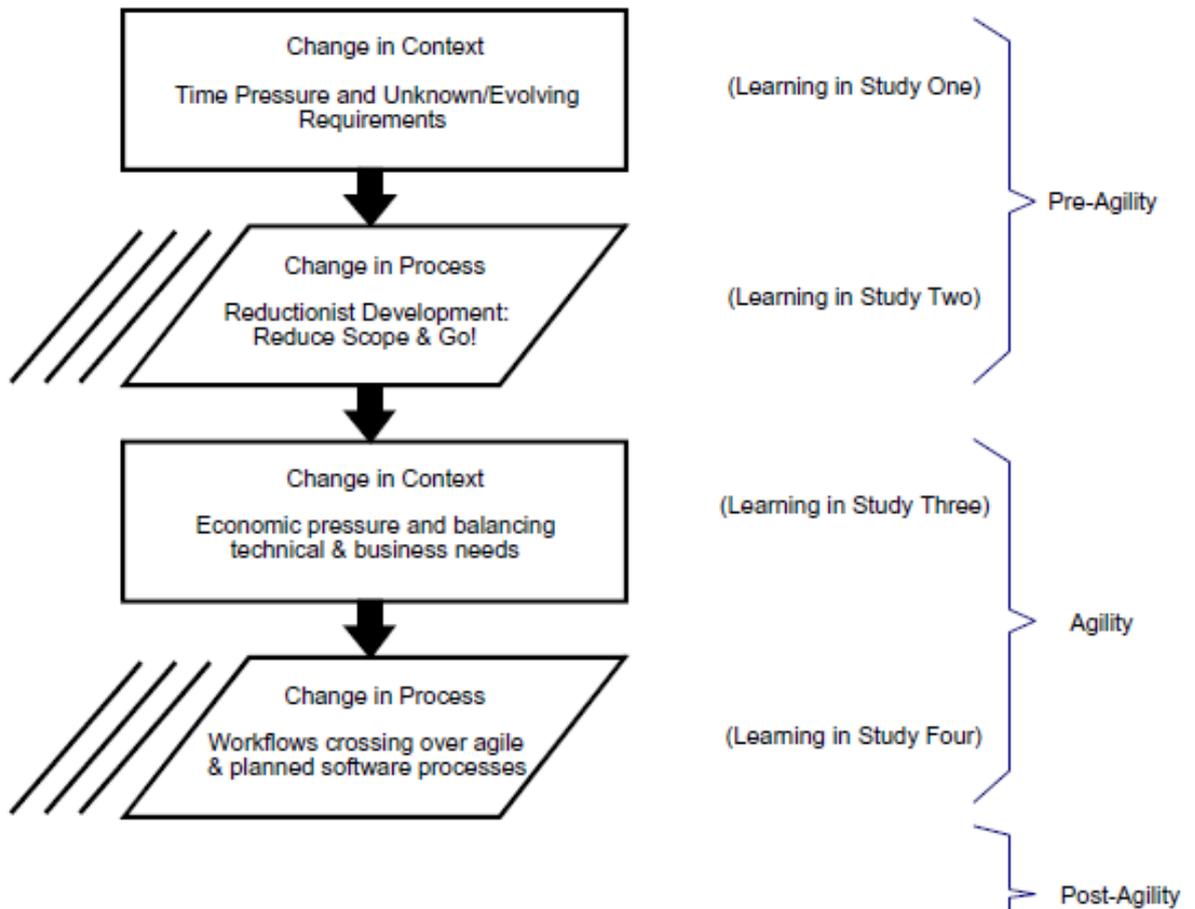
**Figur 7: Product backlog hierarki struktur**  
(Ken Schwaber 2004)



**Tabel 1: Scrum of Scrum i distribueret softwareudvikling**  
 (Mundra; Misra og Dhawale. 2013. s 3-4), (Sutherland and Schwaber. 2007. s 11-12)  
 (Sutherland and Schwaber. 2007) og (Ken Schwaber 2004)

<b>Guidelines for distribueret produktudvikling mellem Scrum of Scrum teams:</b>	1. Intet team bør arbejde på samme user story, 2. Man bør stræbe efter at fordele user stories efter funktioner og iterationer, prøve at fordele arbejdet på funktioner til samme team, være opmærksom på mere tid til planlægning i distribueret Scrum teams og de individuelle opgaver er stadig bestemt af de enkelte teams, 3. For planlægningens skyld er det nødvendigt ved nogle user stories at alle teams er repræsenteret, 4. Release planlægningen bliver udført som ved normalt Scrum, hvorved Scrum prioriteringen af user stories bliver opdelt i det samlet antal iterationer med den kombineret fremdrift af alle Scrum teams, 5. Iteration planlægningen bliver opdelt i user stories imellem Scrum teams proportionalt med teamenes fremdrift og teamenes individuelle fremdrift, 6. iterationerne af hvert enkelt team bør være den samme equivalent med andre teams, 7. For afhjælpning af koordinering og afhængighedsforhold bør teamet anvende Scrum of Scrum meetings til at dele fremdrift.
<b>Scrum of Scrum meetings</b>	1. Foregår efter et Daily stand-up meeting hvor andre Scrum of Scrum teams deltager skal opdatere omkring fremdriften fra sidste møde, deres nuværende fremdrift, deres planlagte fremdrift til næste møde: Hvor interne afhængighedsforhold også bliver diskuteret sammen med stridigheder og løsninger der bliver implementeret efter mødet, 2. Foregår efter release plannings mødet med målet om at identificere afhængighedsforhold og user stories der skal nås via koordinerende Scrum teams, 3. Foregår efter Sprint planning mødet med målet om at opmærksom gøre alle teams omkring Sprint mål produkt levering og 4. At der kan optræde 1 person alene til at repræsentere Scrum teamet i form af Scrum Masteren til Scrum of Scrum mødet, men 2 personer til stede til mødet i form af Product Owner, Scrum Master eller en tredje fra teamet som repræsentant: Men Scrum Masteren skal være til stede uanset omstændighederne; Hvorvidt om Scrum Masteren agere som Product Owner for Scrum teamet i Scrum of Scrum mødet eller som Scrum Master til mødet se eventuelt (2.6 Scrum of Scrum fremdrift i timer i story point per iteration og Figur 13: En Sprint iteration og Tabel 1).
<b>Scrum of Scrum team struktur</b>	1. Product Owner: Scrum of Scrum bør dele den samme eller de samme Product Owner og koordinere at repræsentere et ansigt for teamet udadtil også til andre teams. Product Ownerne varetager derfor samme Product Backlog hvor Product backlogen altid repræsenterer opgaver af højeste prioritering og funktionalitet koordineret af Product Ownerne fra aktiviteterne af koordineringen af Backlogen, 2. Hvert team har deres egen Scrum Master hvor Scrum Masteren deltager i alle møder også i Scrum of Scrum møderne og 3. Udviklere og testere: Er ikke delt i mellem teamene da hvert team har deres egne udviklere og testere.
<b>Scrum 3 Roller</b>	Product Owner, ScrumMaster, Team.
<b>Scrum 3 Ceremonier</b>	Sprint Planning, Sprint Review og Daily Scrum Meeting
<b>Scrum 3 Artefakter</b>	Product Backlog, Sprint Backlog og Burndown Chart
<b>Scrum Product Owner</b>	Product Owner har følgende ansvar og prioritering af Backlogen: 1. Definere funktionaliteterne af produktet, Release date og indhold, 2. Definere værdien af produktet (ROI), 3. Prioritere funktionaliteten i forhold til mest mulig markedsværdi, 4. Ændre Prioriteringen af funktionalitet hver 30 dag og 5. acceptere eller afvis opgavens færdiggørelse.
<b>Scrum Scrum Master</b>	Scrum Masteren er facilitator og teamleder med følgende ansvar: 1. At sikre at teamet er funktionelt og mest muligt produktivt, 2. At sikre mest muligt samarbejde på tværs af alle roller, funktioner og fjerne indbyrdes arbejds barriere mellem teamet, 3. Skærme teamet fra eksterne forstyrrelser og Product Owner eller Product Ownerne 4. Sikre at proceduren er fulgt, invitere til Daily Scrum, Sprint Review og Sprint Planning meetings møder.
<b>Scrum Team</b>	Teamet er: 1. Bestående af 7 plus minus 2 medlemmer, 2. Teamet vælger Sprint målene og specificere opgave resultater (Story Points etc.), 3. Er selvorganiserende inden for rammerne af proceduren og projektets retningslinjer for at nå Sprint målene, 4. Sikre at teamet er selvorganiserende, selv organisere deres arbejde og strukturere deres arbejde og 5. Demonstere deres arbejde til Product Owner.

**Figur 8: Overgangen fra Scrum team til distribueret softwareudviklings Scrum teams til Scrum of Scrum (SOS) teams** (Baskerville, Pries-Heje, Madsen, 2011), se eventuelt (Conboy 2009), (Ken Schwaber 2004) og (Petersen 2001).



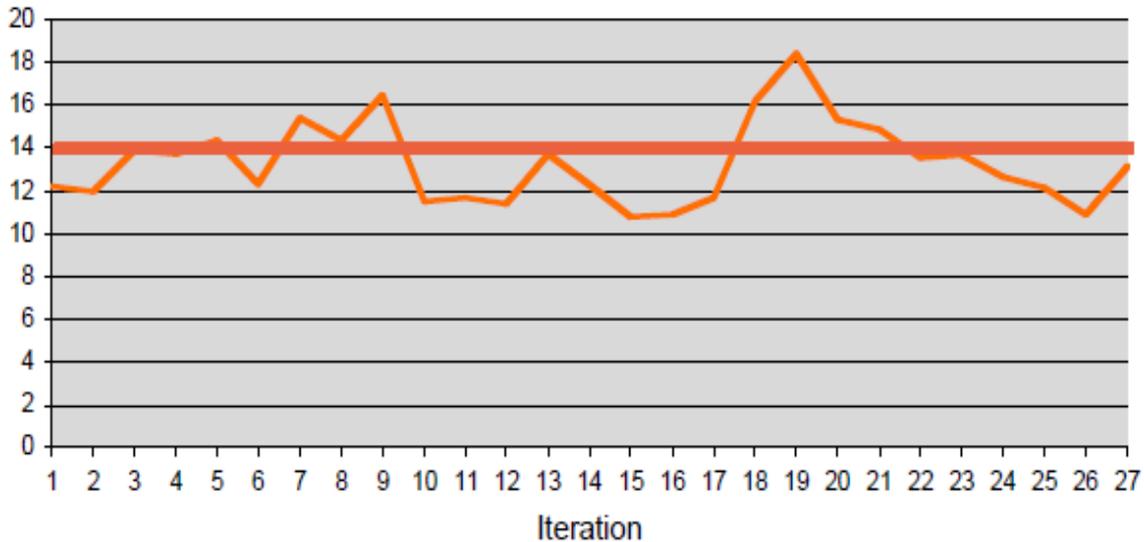
### 2.3 Scrum of Scrum fremdrift timer i story point

Da det kan være svært at estimere fremdrift i Scrum of Scrum bliver timer vurderet i timepris pr. story point se (Figur 9), hvor i at grafen viser at gennemsnit for 1 time er 14 story point per iteration i Scrum of Scrum teams. Hvor den vertikale akse viser story point pr. time og den horisontale story point pr. iterationer. Hvor i at alle opgaver i starten bliver vurderet i story point pr. opgave og løbende opgaver. Hvis man kigger på 6 iteration (Figur 9) kan man se en graf, hvor i at et lokaliseret team skifter til at være et distribueret softwareudviklings team, hvor i at deres story point pr. time ikke var påvirket (Sutherland, Schoonheim, Guido. 2008).

Bemærk at jeg ikke har specificeret hvor mange story points der skal til en time i denne opgave, da det er op til hvert enkelt selvorganiserende scrum team at vurdere story points pr. opgave og ikke Scrum of Scrum teamet for de enkelte teams. Dermed er ansvaret for gennemsnittet af teamenes timer vurderet i timepris pr. story point op til de enkelte Scrum of Scrum teams se (Tabel 1: Guidelines for distribueret produktudvikling mellem Scrum of Scrum teams og Scrum of Scrum meetings) og (Sutherland, Schoonheim, Guido. 2008).

I denne opgave vil jeg tage højde for hvert enkelt teams behov for at være et selvorganiserende team og dermed ikke estimere for teamene, men efterlade dette til en Scrum of Scrum Master rolle i Scrum of Scrum teamet. Jeg vil dog udvikle en Low tech prototype til at understøtte denne aktivitet i et Scrum of Scrum.

**Figur 9: Timer i Story points per iteration**  
(Sutherland, Schoonheim, Guido. 2008. s. 4)



### 3 FORSKNINGSMETODE

Design science research (DSR) i information systems (IS) er karakteriseret ved: Udviklingen af et eller flere IT artefakter som evalueres tiltænkt at skulle løse eller identificere organisatoriske IT problemer. Dette gøres ved en struktureret design process hvorved problemet observeres og løses som et led af et forsknings bidrag til at evaluere designet af forholdet mellem problem og løsning (Peppers et al. 2007).

Ifølge guidelines fra (Peppers et al. 2007) udføres DSR i stadier eller aktiviteter som har til formål at skabe rammerne for en valid model for Research science design: Hvor et af de vigtigste punkter eller stadier er at skabe et artefakt der italesætter et problem og en tilhørende løsning. Hvorved problemet ikke må være løst før og samtidigt er et relevant forskningsproblem. Hvor forskningen bidrager med et artefakt der metodisk kan valideres og verificeres understøttet af eksisterende Research (Persson, Nørbjerg og Nielsen 2016), (Conboy 2009), (Baskerville, Pries-Heje, Madsen, 2011) og (Petersen 2001).

I dette paper vil jeg tage udgangspunkt i forskningen fra (Peppers et al. 2007): Understøttet af DSRM Process Model modellen med aktiviteterne beskrevet i (Peppers et al. 2007. s. 52-56), min problemformulering og eksisterende agilitets evalueringsforskning fra (Conboy 2009) og (Persson, Nørbjerg og Nielsen 2016) til at bidrage til fastholdelse af agilitet i distribueret software udviklings Scrum teams i informations systemudvikling.

### 3.1 DSRM Aktiviteter

**1. Problem identification and motivation:** Scrum er et simpelt inspektions og tilpasnings framework der har 3 roller, 3 ceremonier og 3 artefakter designet til at levere software i Sprints (*Sutherland and Schwaber. 2007. s 11-12*) designet til fysiske lokaliseret teams og ikke distribueret softwareudviklings Scrum teams også kaldet Scrum of Scrum teams: Da et Scrum team er fysisk samlet omkring et Scrum board for at udnytte fordelene af Scrum framework aktiviteterne af et Scrum board. Er et distribueret softwareudviklings Scrum team afhængigt af et artefakt værktøj (Prototype) for at understøtte den distribueret softwareudvikling i distribuerede softwareprojekter. Værktøjet Scrum board er blevet udsat for en række studier hvor rækken af studier har haft mindre fokus på at udvikle nye værktøjer eller artefakter til Scrum of Scrum teams (*Esbensen et al. 2015*). For at understøtte forskning af distribueret softwareudviklings Scrum teams i informations softwareudviklingen, vil jeg bidrage denne forskning med at adressere agilitet i distribueret softwareudviklings Scrum teams med et artefakt værktøj understøttet af artefaktet (Burndown Chart) og tilhørende teori (*Persson, Nørbjerg og Nielsen 2016*), (*Conboy 2009*), (*Petersen 2001*) og (*Baskerville, Pries-Heje, Madsen, 2011*). Ved at jeg definere et specifikt research problem, retfærdiggøre jeg værdien af løsningen af dette problem. Hvor problemet bliver anvendt til at definere udviklingen af artefaktet, der kommer til at definere løsningen. Dermed ved at definere værdien af løsningen opnår man 2 ting: At motivere forskeren og publikummet i at jage en løsning og acceptere denne løsnings resultat eller resultater og hjælper med at forstå forskerens rationaliseringer associeret med problemet.

**2. Define the objectives for a solution:** Scrum board er som tidligere nævnt blevet udsat for en række studier hvor rækken af studierne har haft mindre fokus på at udvikle nye værktøjer eller artefakter. Studierne er gået på at anvende interactive bæredygtige løsninger med et Scrum board som redskab for aktiviteterne i et Scrum of Scrum udviklingsteam. Hvor andre har demonstreret at det er muligt at anvende Scrum board i distribueret softwareudvikling teams (*Esbensen et al. 2015*). Til den forskning vil jeg designe et artefakt værktøj (Prototype) til et distribueret softwareudviklings Scrum teams, hvor et artefakt værktøj skal illustrere aktiviteterne af en Backlog og et Burndown-chart (Multiple Dashboard) i et Scrum of Scrum. Artefakt værktøjet skal illustrere et distribueret softwareudviklings Scrum team der har det fulde overblik over udviklingsprocessen kan danne mening ud af denne udviklingsproces og derefter kan koordinere aktiviteterne af en Backlog til et Scrum of Scrum team. Hvorefter artefakt værktøjet skal vurderes gennem en demonstration af Ivan Aaen derefter itereres efterfølgende af den givne data fra et eller flere semistruktureret interviews understøttet af teori fra (*Rose. 2010*) og (*Aaen. 2017*). Jeg vil dertil udlede målene af en løsning fra problemformuleringen og viden omkring hvad der er muligt og kan gennemføres (*Conboy 2009*) og (*Persson, Nørbjerg og Nielsen 2016*). Med målet om en ønsket løsning kan opnås i større skala som kan kvantificeres, har en kvalitativ værdi eller en deskriptiv løsning af et nyt artefakt. Der er forventet at assistere i en eksisterende løsning eller at adressere en eksisterende løsning med et defineret problem (*Baskerville, Pries-Heje, Madsen, 2011*) og (*Petersen 2001*).

**3. Design and development:** Jeg har valgt at udvikle et artefakt værktøj bestående af nye tekniske egenskaber og informative egenskaber. Hvor et konceptuelt design artefakt værktøj bliver objektet for aktiviteten af koordineringen af Backloggen hvor i at et forskningsbidrag er indlejret i designet af en Low tech prototype og aktiviteterne der til: Hvor jeg vil definere artefakt værktøjets ønsket funktionalitet og derefter udvikle værktøjet (Prototypen). Hvor resurserne påkrævet fra at flytte fra objekt til design og udvikling er inkluderet i teorien som derefter kan bibringe et artefakt bidrag og efterfølgende en løsning til distribueret softwareudviklings Scrum teams. Jeg vil skabe et artefakt bestående af nye tekniske egenskaber, sociale eller informative egenskaber. Hvor et konceptuelt design artefakt kan være et i et værktøj hvor et forskningsbidrag er indlejret i designet (*Conboy*

2009). Denne aktivitet indeholder at definere artefaktets ønsket funktionalitet, ønsket arkitektur og skabe et artefakt. Hvor i resurser påkrævet fra at flytte fra objekt til design og udvikling er inkluderet i teorien som kan bibringe en løsning.

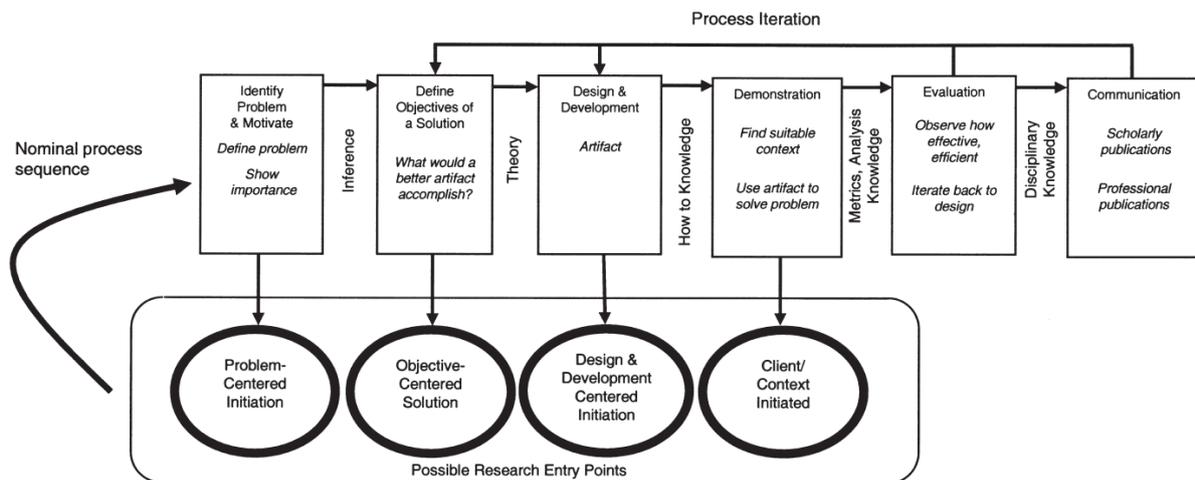
**4. Demonstration:** Her vil jeg demonstrerer brugen af artefaktet til at løse et eller flere problemer i flere forekomster af problemet: Dette kunne involvere artefaktet i anvendelse i en demonstration til en ekspert, et experiment (Wizard of Oz), i et casestudie (Baskerville, Pries-Heje, Madsen, 2011), proof of concept (*Conboy 200*) eller andre aktiviteter. Hvor resurserne påkrævet til demonstrationen inkludere viden omkring anvendelse af artefaktet til at løse problemet og anvendelsen i at løse problemet (*Persson, Nørbjerg og Nielsen 2016*) og (*Conboy 2009*). Jeg demonstrere brugen af artefaktet til software innovation eksperten Ivan Aaen, hvor artefaktet værktøjet skal vurderes i gennem en Low tech prototype demonstration, efterfulgt af et semistruktureret interview og derefter itereres efterfølgende af supplerende empirisk data understøttet af teori fra (*Conboy 2009*) og tilhørende teori (*Baskerville, Pries-Heje, Madsen, 2011*) og (*Persson, Nørbjerg og Nielsen 2016*).

**5. Evaluation:** Her vil jeg anvende artefakt værktøjet, et semistruktureret interview og tilhørende teori fra (*Conboy 2009*) og (*Baskerville, Pries-Heje, Madsen, 2011*). Dertil vil jeg sammenligne et semistruktureret interview med den anvendte litteratur og teori fra (*Conboy 2009*), observationerne fra empirien og dataen fra et eller flere semistruktureret interview og løsningen fra aktivitet 2. Derefter i slutningen af aktivitet 5 enten iterere tilbage til aktivitet 3 en eller flere gange indtil aktivitet 5 er fyldestgørende og derefter fortsætte til aktivitet 6. Da jeg i denne opgave gerne ville bidrage med at strukturere en arbejdsproces i distribueret softwareudviklings Scrum teams og fastholde agilitet i Scrum of Scrum med udgangspunkt i aktiviteterne af artefaktet (Burndown Chart) i Scrum of Scrum med en tilhørende Scrum of Scrum Master rolle og et understøttende Low tech prototype i Scrum of Scrum. Vil jeg anvende taxonomien fra (*Conboy 2009*), da den i tråd med (*Baskerville, Pries-Heje, Madsen, 2011*) opsummere agilitet, udfordringerne ved at adressere agilitet og efterspørgslen af værktøjer til at adressere agilitet i agile software organisationer. Jeg anvende taxonomien fra (*Conboy 2009. s 14*), ved at tage udgangspunkt i taxonomiens evalueringskriterier for anvendelsen af taxonomien til at identificere eksisterende og nye agile praksisser. Hvor Conboy definere følgende agilitets vurderingskrav for agile praksisser i 3 spørgsmål, som jeg efterfølgende anvende på Scrum, Scrum of Scrum Master rollen og en Low tech prototypen se punkt 1,2 og 3: Dermed simplificere ledelses, afhjælpe koordineringen og fremdriften i distribueret softwareudviklings Scrum teams også kaldet SOS understøttet af (*Persson, Nørbjerg og Nielsen 2016*). Hvor jeg observere og måler hvordan artefaktet understøtter løsningen til problemet: Denne aktivitet indeholder at sammenligne målene af en løsning med et eller flere aktuelle observeret resultater fra artefaktet i brug i en demonstration. Dette kræver viden omkring relevante måleenheder eller analyseteknikker der er afhænger af naturen af problemet i dets omgivelser i sammenhæng med artefaktet. Dette kunne indeholde sammenlignings opgaver fra (*Conboy 2009*) og (*Persson, Nørbjerg og Nielsen 2016*) af artefaktets funktionalitet med en løsning (*Baskerville, Pries-Heje, Madsen, 2011*) fra aktivitet 2, kunde feedback eller simuleringer (Wizard of Oz). Konceptuelt kunne denne aktivitet inkludere alle former for relevant empirisk evidence eller logisk bevis (*Baskerville, Pries-Heje, Madsen, 2011*). I slutningen af denne aktivitet kan en forsker vælge at iterere tilbage til aktivitet 3 eller fortsætte til aktivitet 6.

**6. Communication:** Her vil jeg kommunikere problemformuleringen, dens relevans, artefaktet, anvendelsen og bidraget. Formidlingen vil foregå igennem en rapport afhandling bestående af en normativ research proces efterfulgt af DSRM modellen: En problemformulering, et litteraturreview, metodeanvendelse, empirisk dataindsamling og analyse af dataen, repræsentation af resultater, en diskussion og konklusion. Processen vil blive beskrevet i en normativ sekventiel orden: Hvor i processen vil foregå i iterative stadier alt efter behov, hvor i sammenhængen mellem problem og løsning vil fremtræde, hvori de sekventielle processer og iterationerne er svaret på kommunikationen af artefaktet og løsningen indlejret i problemet af artefaktet. Dette vil tage udgangspunkt i en problemorienteret tilgang hvor aktiviteten er startet med et forskningsproblem i form af en

problemformulering. Jeg vil kommunikere et problem, problemets relevans, anvendelsen af artefakt værktøjet og artefakt bidraget indlejret. I formidlingen af en rapport, en videnskabelig udgivelse eller publikation hvor forsker anvender en normativ struktureret proces til at strukturere artiklen bestående af en normativ research proces i form af: En problemformulering, litteraturreview, metodeanvendelse, empirisk dataindsamling, analyse af data, repræsentation af resultater, en diskussion og konklusion (Peppers et al. 2007. s. 52-56). Processen er beskrevet i en normativ sekventiel orden hvori processen foregår i iterative stadier, se følgende (Figur 10). Hvor i processen viser sammenhængen mellem problem og løsning hvori de sekventielle processer og iterationerne er svaret på kommunikationen af artefaktet og løsningen indlejret i problemet af artefaktet. Hvor problemet er i gang sat af: 1. En problemorienteret tilgang med en basis for en nominal sekvens: Hvor aktiviteten starter med en forsker der starter med en ide for et research resultat observeret fra et problem eller et forslag til fremtidig forskning i et tidligere projekt (Baskerville, Pries-Heje, Madsen, 2011). 2. Hvor en målorienteret løsning starter i aktivitet 2 (Conboy 2009) dette kunne være igangsat af en industri eller research der mangler at blive adresseret af udviklingen af et artefakt eller værktøj (Baskerville, Pries-Heje, Madsen, 2011), 3. En design og udviklingscentret tilgang startet med aktivitet 3 (Aaen 2017) dette ville være et resultat af et tidligere artefakt der ikke har været formelt tiltænkt som en løsning (Burndown Chart) der kunne blive anvendt i problemdomænet (Rose. 2010) og (Persson et al. 2009). Dette artefakt kunne have været anvendt i et andet research domæne eller allerede anvendt i at løse et andet problem (Fitzgerald, Hartnett, Conboy 2006) og (Baskerville, Pries-Heje, Madsen, 2011). 4. En klient kontekst igangsat løsning hvori løsningen er igangsat af en observering af en praktisk løsning der allerede fungerer. Hvore processen starter i aktivitet 4, resulterende i en Design science (DS) løsning (Peppers et al. 2007).

**Figur 10: DSRM Process Model (Peppers et al. 2007. s. 52-56) 10**



### 3.2 Software innovation metode og værktøj

I (Aaen. 2017. chapter 6) beskrives der 4 typer initieringer af problemer User need, Technological opportunity, Solution reuse, Competitive stress hvor de er beskrevet således: 1. User need: Er når projektet er igangsat af eksterne brugere eller kunder der præsenterer en udfordring, 2. Technological opportunity: Er hvor en software-intensiv løsning udnytter en teknologisk løsning i et marked, 3. Solution reuse: Hvor en software-intensiv løsning udviklet for en løsning anvendes i et andet domæne og 4. Competitive stress: Hvor et organisations team har et behov for at udvikle konkurrenceevne se (Figur 11). I mit tilfælde er der tale om Competitive stress se (Figur 11), hvor problemet var igangsat af et behov for manglende agilitet og konkurrenceevne i nykredits distribueret softwareudviklings Scrum

teams i organisationen. Hvor Scrum teams var afhængig af koordinering af aktiviteterne af Product Backloggen da fordelene af en Backlog og et Burndown-Chart var stærkt afhængigt af at Scrum teamet var fysisk samlet omkring et Scrum board: Men hvordan var rollefordelingen i sådan en struktur, hvordan organiseres Product Backloggen og hvordan estimeres fremdrift når der tilføjes flere teams i distribueret softwareudvikling også kaldet Scrum of Scrum?

**Figur 11: Four types of initiating problems (11)**



Et af disse Scrum board se (Figur 12): Er ved hjælp af videokonference teknologier, understøttet af scrum og Scrum of Scrum se (Tabel 1). Ifølge (Sutherland og Schwaber. 2007) er Scrum et simpelt inspektions og tilpasnings framework, der har 3 roller, 3 ceremonier og 3 artefakter designet til at levere software i Sprints (Sutherland and Schwaber. 2007. s 11-12) designet til fysiske lokaliseret teams og ikke et Scrum of Scrum udviklingsteam se (Tabel 2).

**Tabel 2: Scrum framework Jeff Sutherland lokaliseret Scrum team**  
(Sutherland og Schwaber. 2007)

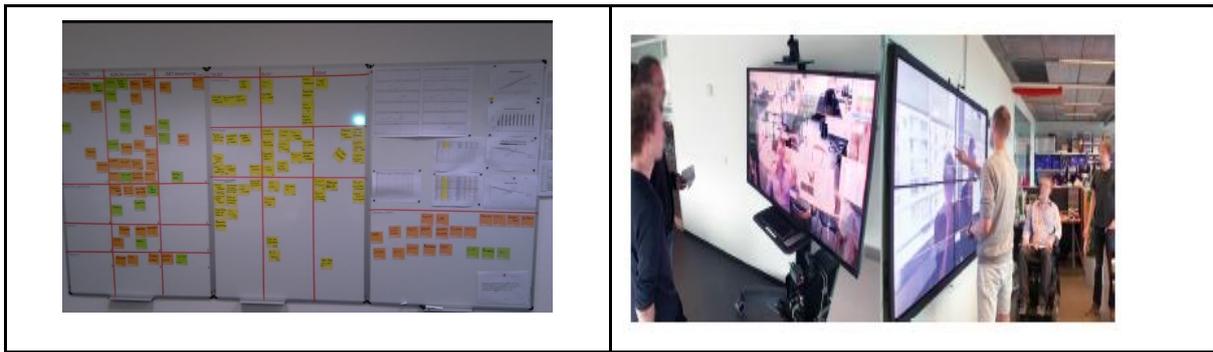
<b>3 Roller</b>	Product Owner, ScrumMaster, Team.
<b>3 Ceremonier</b>	Sprint Planning, Sprint Review, og Daily Scrum Meeting
<b>3 Artefakter</b>	Product Backlog, Sprint Backlog, og Burndown Chart

Hvor i SOS er der flere Scrum teams der koordinere og arbejder sammen, på det samme produkt med et fælles mål. Hvor Scrum of Scrum team er placeret ud på flere lokations, hvor hvert Scrum team er et Scrum team ud af et samlet Scrum of Scrum team. Hvor der tages udgangspunkt i en fælles koordineringen af software produktudvikling i flere lokations, hvor koordineringen af det fælles produkt er afhængig af flere distribueret Scrum teams.

Hvor i teamene koordinere via. Scrum of Scrum teams møder, hvor i en repræsentant fra hvert distribueret team deltager og en Scrum Master fra hvert scrum team. Hvor mødet foregår, afhængigt af Scrum of Scrum teamenes behov og afhængighed (Dependencies) ellers en gang dagligt (Se Tabel 1), (*Mundra; Misra og Dhawale. 2013*), (*E. Hossain, A. B. Muhammad og y. P. Hye. 2009*) og (*A. Lamersdorf, M. Jürgen and R. Dieter. 2009*).

### Figur 12: Fysisk og digitalt Scrum board (12)

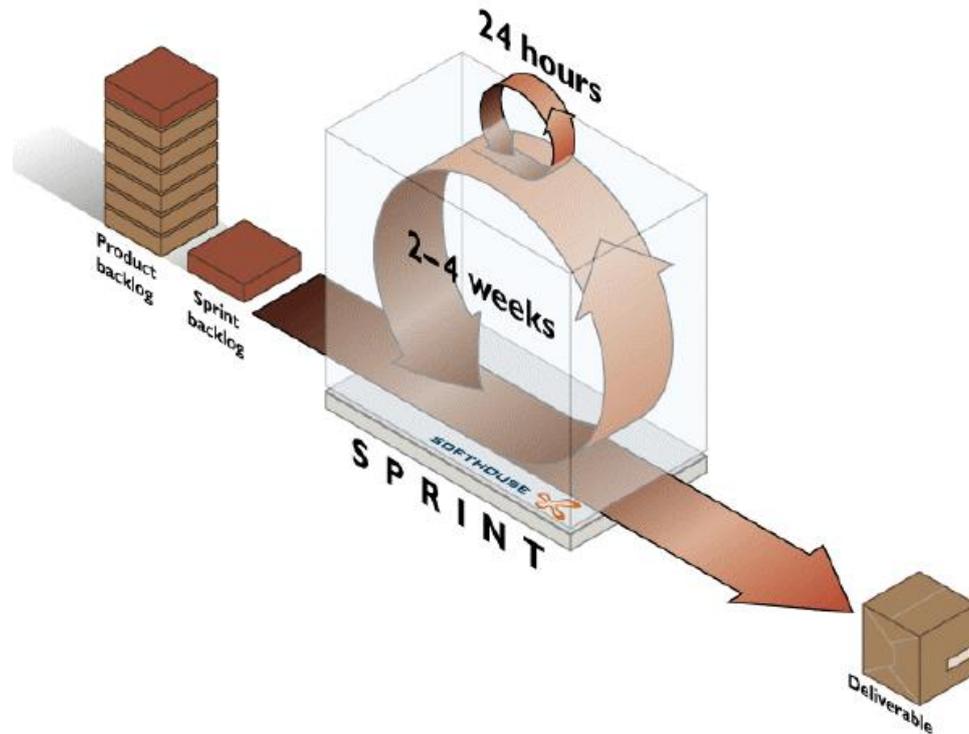
(*Solingen, Sutherland, Denny. 2011 og Esbensen et al. 2015. s 7*)



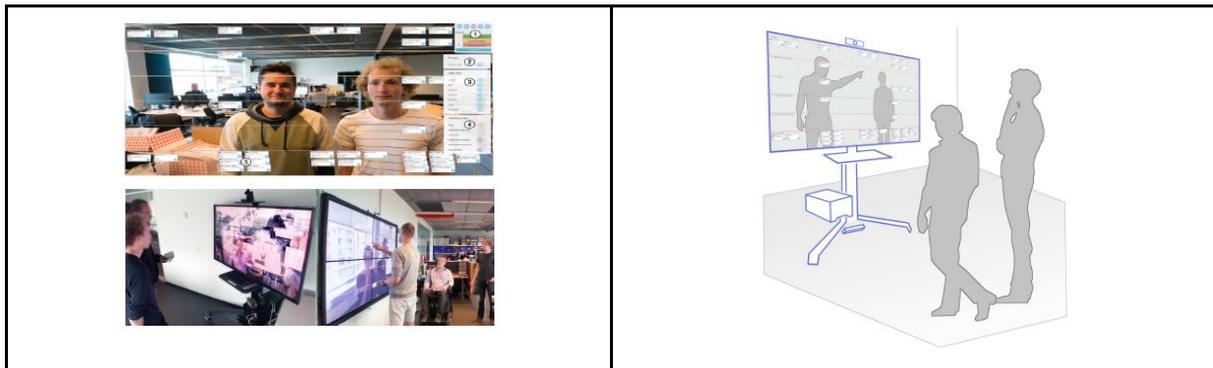
En agile softwareudviklingsproces er centraliseret omkring inkrementelle leveringer, som er et led i at skabe kvalitets software baseret på feedback. Hvor det forandrende miljø, er en del af udviklingen af software projektet med forandrende specifikationer eller ukontrollerede specifikationer. Hvor udviklingsprocessen af projektet handler om at opdage opgaver og håndtere dem undervejs. I denne proces er det vigtigt at teamet har det fulde overblik over udviklingsprocessen og kan danne mening ud af denne udviklingsproces se (Figur 13). I kernen af agil udviklingsprocesser bliver der anvendt et Scrum board, hvor et task board eller tavle bliver placeret i et softwareudvikling teams kontor. Hvor tavlen agere som bindende led mellem udviklere og opgaver for at afhjælpe koordineringen af aktiviteter og fremdriften af det daglige Sprint møde: Dermed fungere som koordinerings objekt, for at hjælpe de daglige møder i koordinering af diskussioner, koordinering af opgave fremdrift og som passiv informations facilitator af koordinering af opgave ansvarsfordeling og status update på eksisterende opgaver se (Tabel 1 og Figur 13: En Sprint iteration).

Siden hen er Scrum board se (Tabel 1) og blevet udsat for en række studier, hvor i rækken af studier har haft mindre fokus på at udvikle nye redskaber. Nogle af studierne er gået på at anvende interactive bærebær løsninger med et Scrumboard som redskab, for et Scrum of Scrum udviklingsteam. Hvor andre har demonstreret at det er muligt at anvende Scrum board i distribueret softwareudvikling teams, selvom researchen gældende anvendelsen af digital Scrum boards og hvordan sådan et artefakt ser ud er sparsom (*Esbensen et al. 2015*) og se (Figur 14).

**Figur 13: En Sprint iteration (13)**  
(Sutherland and Schwaber. 2007)



**Figur 14: Scrum board (14)**  
(Esbensen et al. 2015)



Første skridt i at skabe et artefakt hvor et konceptuelt design bliver objektet hvor et forskningsbidrag er indlejret i designet, bliver ved at jeg definere et artefaktets ønsket funktionalitet. Hvor resurser påkrævet fra at flytte fra objekt til design og udvikling er inkluderet i teorien som kan bibringe en løsning og til slut et artefakt eller prototype. I starten af skabelsen af mit artefakt vil jeg anvende redskaberne fra bøgerne (Aaen. 2017) og (Benyon David. 2014). Hvor jeg tager forudsætningen at jeg er et projektteam, med opgaven følgende problemstilling:

- I distribuerede softwareudviklingsprojekter er Scrum teams afhængig af koordinering af aktiviteterne af Product Backloggen; Da fordelene af en Backlog og et Burndown-Chart, er stærkt afhængigt af at Scrum teamet er fysisk samlet omkring et Scrum board.

- Videokonference teknologi kan hjælpe distribueret softwareudviklings scrum teams i deres scrum metode med at afhjælpe koordineringen af aktiviteterne af Product Backloggen og fremdriften af det af et Sprint møde i Scrum of Scrum.

Jeg ønskede at lave en software innovativ løsning med et Multiple dashboard (Scrum of Scrum Board), hvor jeg ville beskrive hvilken type innovation min løsning var for at kunne forstå sammenhængen mellem problem og løsning. Kreativitet er en del af software innovation, dermed er Software innovation styret af kreativitet og det at skabe nye og anvendelige ideer og produkter i form af brugbar relevante skalere bare løsninger. Innovation i sig selv som definition, implicere forandring af den virkelige verden. Hvor software innovation i sig selv omhandler at skabe systemer, hvor software spiller en vital rolle. For at igangsætte denne proces anvendte jeg 4 typer software innovation Product innovation, Process innovation, Project innovation og Paradigme innovation. Hvor Product innovation er styret af at skabe nye eller radikalt ændret software-intensive produkter. Hvor process innovation omhandler nye software intensive løsninger eller måder at tilbyde en radikalt forbedret måde at producere et produkt eller service. Project innovation er når eksisterende software-intensive løsninger, er tilpasset ind i nye applikations domæner. Paradigm innovation er når software-intensive løsninger er bundet med forandringer i den mentale model af en organisation. I min løsning ønskede jeg at tilbyde en radikalt forbedret måde at tilbyde en service på og derfor tilfalder min præ løsning en Proces innovation indtil yderligere iterationer (Aaen. 2017. chapter 3).

Første skridt i projektet er at anvende Visions som et led i at finde og organisere objekter, events og kvaliteter for mit case projekt: Et Multiple Dashboard (Scrum of Scrum Board), hvor Jeg tager forudsætningen i at Visions kan hjælpe mig med at organisere objekter, events, kvaliteter igennem de 4 typer Visions (Icon, Metaphor, Prototype, Proposition),(Aaen. 2017. chapter 2) og (Benyon David. 2014).

**Figur 15: Four Types of Vision Representations (15)**

(Aaen. 2017. chapter 2)

	Simple	Complex
Concrete	<i>Icon</i> <ul style="list-style-type: none"> <li>• <i>Principal</i></li> <li>• <i>Disembodied</i></li> </ul>	<i>Prototype</i> <ul style="list-style-type: none"> <li>• <i>Functional</i></li> <li>• <i>Outward</i></li> </ul>
Abstract	<i>Metaphor</i> <ul style="list-style-type: none"> <li>• <i>Connotational</i></li> <li>• <i>Figurative</i></li> </ul>	<i>Proposition</i> <ul style="list-style-type: none"> <li>• <i>Denotational</i></li> <li>• <i>Defining</i></li> </ul>

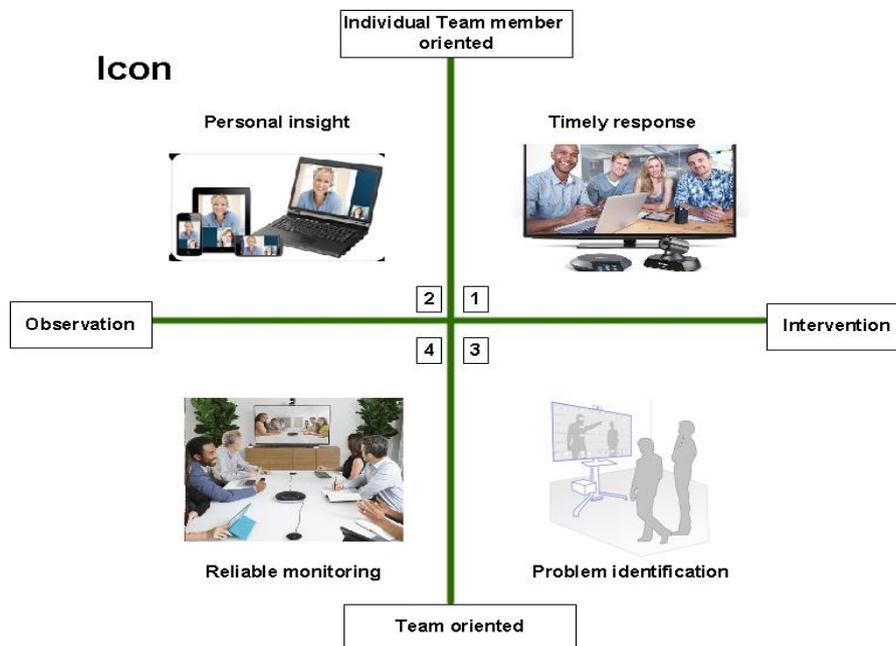
For at få indsigt i hvilke objekter, events og kvaliteter der kunne være i et muligt Multiple (Scrum of Scrum Board) anvender jeg Visions typen Icon.

Det icon'er særligt kan, er at de er simple og konkrete, hvorimod Metafor er simple og abstrakte. Hvor deres modpart prototyper er komplekse og konkrete, og forslag er komplekse og abstrakte. Icon: Ikoner kan repræsentere kvaliteter i et design, der er værdsat i en god løsning. De kan foreslå adskillige forslag, der er særligt ønsket, i en ønsket løsning. De kan pointere til adskillige ligheder i ikoniske løsninger. Metafor: Metafor kan repræsentere principper via analogier, ved hjælp af et figurativt sprog, symboler, slogans og ligheder. Ideen er at provokere ideer ved at kigge på principper indlejret i løsninger til

problemet, der på en eller flere måder har ligheder med problemet. Metafor kan dermed inspirere, til løsninger til et eksisterende problem.

Prototype: Prototyper repræsenterer design modeller som skitser, der dermed kan repræsenterer prototyper forslag til et design af et brugerinterface, dele af et muligt arkitektur og en måde at operere systemet. Proposition: Propositions er forslag til design ideer, hvori designs bliver repræsenteret litterært, forslagene definere er projekts fokus.

**Figur 16: Icon til Scrum of Scrum board (16)**



**Tabel 3: Icon kvadrant**

	Beskrivelser
<b>Kvadrant 1:</b>	<p>Her repræsenterer jeg hvilke kvaliteter i de devices, der kunne sikre kvaliteten i et multiple dashboards (Digitalt Scrums of Scrums board) med et Burndown-chart.</p> <ul style="list-style-type: none"> <li>- Softwaren kan køre på flere devices for tilgængelighed og anvendelighed</li> <li>- Softwaren tilpasser brugerens skærm behov og opløsning uanset device eller hardware.</li> </ul>
<b>Kvadrant 2:</b>	<p>Her repræsenterer jeg hvilke kvaliteter i de devices...</p> <ul style="list-style-type: none"> <li>- Et samlet Scrum of Scrum team kan fysisk være samlet omkring en fælles skærm og blive opdateret i fællesskab omkring et multiple dashboards Real-time.</li> </ul>
<b>Kvadrant 3:</b>	<p>Her repræsenterer jeg hvilke kvaliteter i de devices...</p> <ul style="list-style-type: none"> <li>- Flere Scrum Teams og et Scrum of Scrum team kan fysisk opdatere et fælles multiple dashboards og stadig være opdateret omkring ændringerne af Product Backloggen eller sprint opdateringer, da ændringerne bliver registreret i softwaresystemet.</li> </ul>
<b>Kvadrant 4:</b>	<p>Her repræsenterer jeg hvilke kvaliteter i de devices...</p> <ul style="list-style-type: none"> <li>- Flere Scrum Teams og et Scrum of Scrum team kan fysisk være samlet omkring at blive opdateret omkring et multiple dashboards, diskutere opgaverne i fællesskab og derefter planlægge opgaverne i et multiple dashboards der sikre høj transparens og visibilitet.</li> </ul>

### **3.2.1 Configuration Tables, Tactics, Strategy og Rationale**

Da jeg ønskede at beskrive mit Multiple dashboard (Scrum of Scrum Board) artefakt yderligere, valgt jeg at anvende Configuration Tables (Aaen. 2017. chapter 5). Da (Aaen. 2017) kræver en dialog mellem problemer og løsninger for at værdsætte behov og opdage nye løsninger. Først vil jeg beskrive et Configuration table og hvad det udvikles af Tactics, Strategier og Rationale.

Configuration tables er en kombination af interrelateret problemer, løsninger og rationale. Hvor Taktiske niveau er omhandlende handlinger der er designet til at tilstræbe specifikke mål: 1. Offer: nye tilstræbt løsninger til et problem, 2. Feature: på software attributter der plejede at give en bestemt løsning, 3. Scenario: En struktur eller sekvens af events der er anvendt som en del af løsningen.

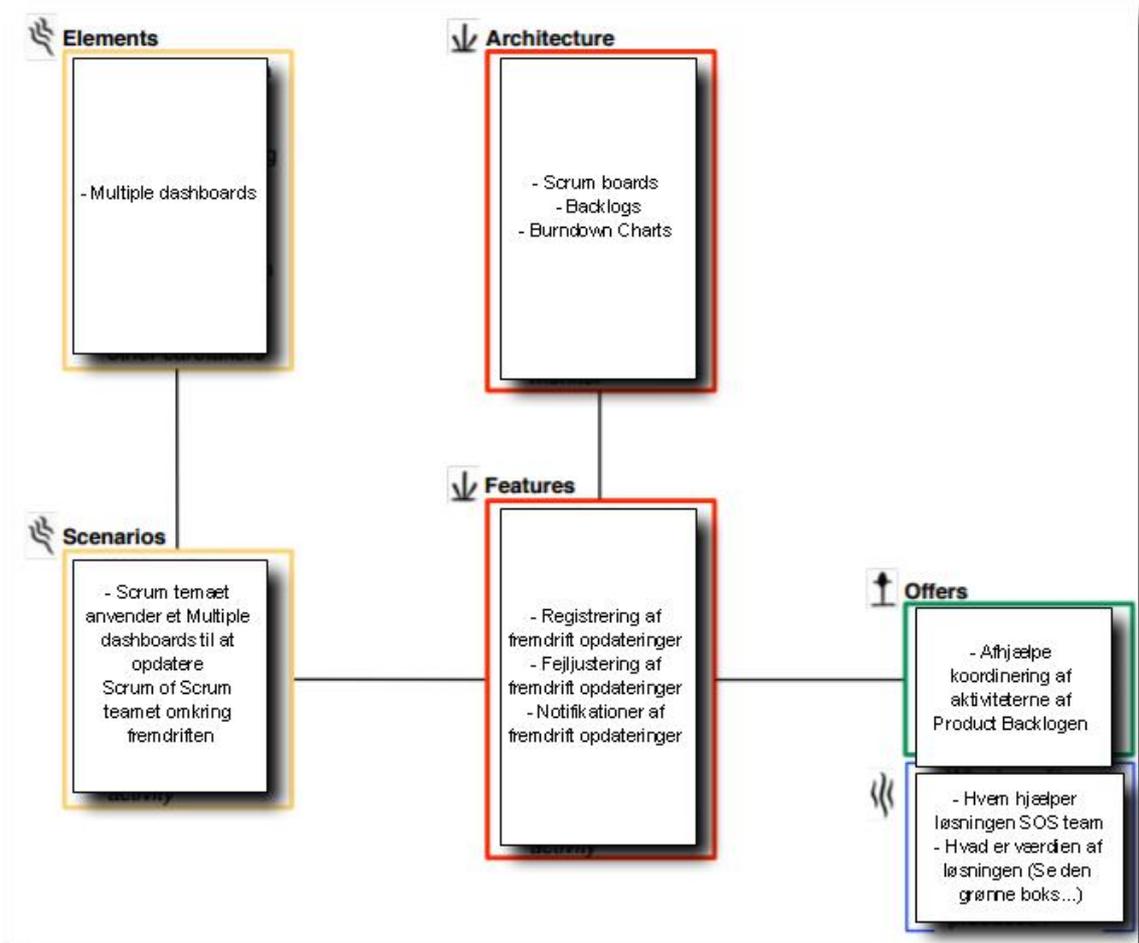
Det Strategiske niveau er omhandlende at opnå overall mål i gennem planlægning af handlinger: 1. Architecture: Den konceptuelle struktur, logisk organisering af software-intensive systemer under udvikling. Disse koncepter er anvendt som et led i overall handlinger og organisering af platforme der tilbyder at løse en problematisk situation selvom opfattelsen kan ændres i kursen af inquiry.

Det Rationelle niveau er omhandlende grunde eller logisk basis for handlinger for Vision for den nuværende end-in-view der gør mine handlinger meningsfyldte og hjælper mig med at rationalisere for strategier og taktikker: 1 Vision: End-in-view der gør det software-intensive system under designet en nøgle del af at løse den problematiske situation. 2. Warrant: Hvorfor at løse problemet er vigtigt, 3. Backing: Hvorfor Vision er attraktivt og anvendeligt for løsningen af problemet, 4. Problem: En forudindtagelse af problem situation med tilhørende komponenter af kontekst og kriterier for situationen, 5. Challenge: Det større problemdomæne af en problemsituation. Disse koncepter er anvendt til at opmærksom gøre hvorfor et problem skal løses og hvordan en software-intensive løsning hjælper.

#### **Scrum of Scrum Board tactic**

Her kigger jeg på hvad der specifikt tilbydes af systemet til at løse problemet ved features og korresponderende scenarier. I mit Scrum of Scrum Board tactic starter jeg med at arbejde med at bygge mit (15 Figur), hvor figuren relatere til løsningerne i scenarierne og features. Løsningerne er skabt gennem et sæt af features og disse features afkommer af scenarier. Første skridt i at bygge figuren er at forudindtage hvad sådan et system vil tilbyde: at Afhjælpe koordinering af aktiviteterne af Product Backloggen og fremdriften af et Scrum of Scrum sprint.

Figur 17: Tactic



De ovenstående (Figur 17): Tactics har jeg listet en række objekter, events som virker relevante for at identificere scenarier og features: 1. Hvor et scrum team, integrere med et interaktivt multiple dashboards (Digitalt Scrum board med et Burndown-chart), hvor systemet kunne være tilsluttet smartphones eller andre devices, 2. Brugere kunne personificere deres Scrum board med et eller flere Burndown-chart eller Backlog. Her kunne vi også kigge på tidligere anvendte videokonference systemer som kunne afhjælpe mig med at identificere tidligere anvendte teknologier og deres features. Sammen skaber disse 4 elementer selvindsigt i: Hvordan systemet kunne stimulere overvejelser og udkast fra denne refleksion over hvordan sådan et system kunne anvendes af et distribueret softwareudviklings Scrum team også kaldet Scrum of Scrum.

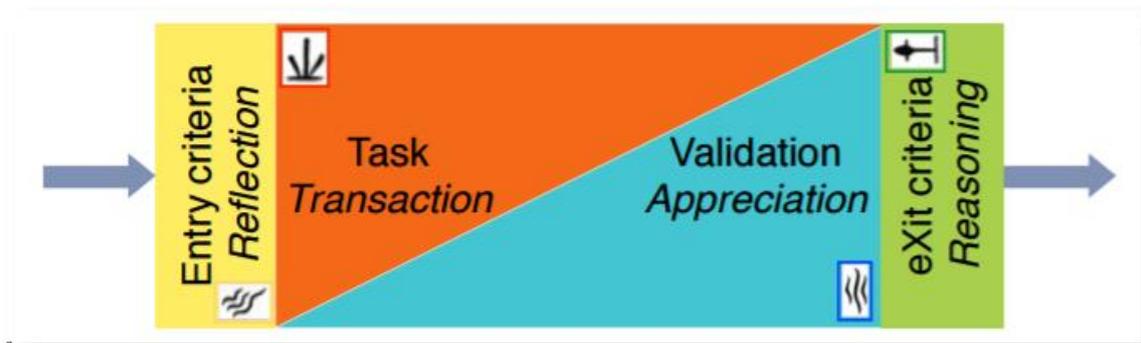
**Figur 18: Configuration Table**

View	 <b>Paradigm</b>	 <b>Product</b>	 <b>Project</b>	 <b>Process</b>
Value	REFLECTION	TRANSACTION	REASONING	APPRECIATION
Ratio- nale: Why?	<b>PROBLEMATIC</b> <ul style="list-style-type: none"> <li>- Challenge: Forbedring af at afhjælpe koordinering af aktiviteter i distributed software udviklings Scrum teams</li> <li>- Problem: Mangel på fysisk kommunikation i distributed software udvikling Scrum teams</li> </ul>	<b>TECHNOLOGIES</b> <ul style="list-style-type: none"> <li>- Multiple dashboards</li> </ul>	<b>RESOLUTION</b> <ul style="list-style-type: none"> <li>- Vison: Afhjælpe koordinering af aktiviteterne af Product Backloggen i Scrum of Scrum (SOS)</li> <li>- Warranty: afhjælpe fremdriften af Product Backloggen i SOS</li> <li>- Backing: Reduktion i manglende Sprint opdateringer, Ingen mulighed for manglende registrerede Sprint opdateringer og Transparens af Sprint opdateringen</li> </ul>	<b>CRITERIA FOR RESOLUTION</b> <ul style="list-style-type: none"> <li>Expectations: Problemet er veldefineret, Vision er praktisk og kvalificeret</li> <li>- Det er muligt med et multiple dashboards system</li> <li>Findings: - ineffektiv invention</li> <li>- Ikke anvendt sammensætning</li> </ul>
Strat- egy: What?	<b>ELEMENTS</b> <ul style="list-style-type: none"> <li>- Multiple dashboards</li> </ul>	<b>ARCHITECTURE</b> <ul style="list-style-type: none"> <li>- Scrum boards</li> <li>- Backlogs</li> <li>- Burndown Charts</li> </ul>	<b>QUALIFICATION</b> <ul style="list-style-type: none"> <li>- Qualiter: Kun for distribueret software udviklings Scrum teams</li> <li>Rebuttal: Mange teams kan få gavn af dette</li> </ul>	<b>CRITERIA FOR ARCHITECTURE</b> <ul style="list-style-type: none"> <li>Expectations: - Komponenterne er nødvendige og relevante i implementeringen i Vision</li> <li>Findings: - Interventionen skal tilpasses de distribueret software udviklings Scrum teams også kaldet Scrum of Scrum (SOS)</li> </ul>
Tac- tics: How?	<b>SCENARIOS</b> <ul style="list-style-type: none"> <li>- Scrum teamet anvender et Multiple dashboards til at opdatere Scrum of Scrum teamet omkring fremdriften</li> </ul>	<b>FEATURES</b> <ul style="list-style-type: none"> <li>- Registrering af fremdrift opdateringer</li> <li>- Fejljustering af fremdrift opdateringer</li> <li>- Notifikationer af fremdrift opdateringer</li> </ul>	<b>OFFERS</b> <ul style="list-style-type: none"> <li>- Afhjælpe koordinering af aktiviteterne af Product Backloggen</li> <li>- Hvem hjælper løsningen SOS team</li> <li>- Hvad er værdien af løsningen at afhjælpe koordinering af aktiviteterne af Product Backloggen)</li> </ul>	<b>CRITERIA FOR OFFER</b> <ul style="list-style-type: none"> <li>Expectations: - Afhjælpe koordinering af aktiviteterne af Product Backloggen</li> <li>Findings: - Mangelde rolle definition i Scrum of Scrum og Product Backlog struktur</li> <li>- Skyldes manglende Afhjælpning af koordinering af aktiviteterne af Product Backlog i Scrum of Scrum</li> </ul>

Da jeg nu har været igennem en Configuration table vil jeg anvende ETVX-modellen og dens aktiviteter som review for at afslutte en iteration af mit projekt (Aaen. 2017. chapter 8). ETVX-modellen er beskrevet således: 1. Entry Criteria: Kriterier der skal være opfyldt før projektet igangsættes, 2. Task: Beskrivelse af opgaven, der skal løses, 3. Validation: Validering af procedure for at sikre kvaliteten af opgaven der skal løses, 4. Exit: kriterierne der skal være opfyldt før at opgaven løses. Ifølge fortolkningen af ETVX modellen igennem (Aaen. 2017. chapter 5), relatere modellen til de 4 aktiviteter Reflection, Transaction, Appreciation og Reasoning: Hvor 1. Reflection: refleksion er forbundet med at sikre en solid forståelse af problemet der skal adresseres i aktiviteterne, 2. Transaction: Transaktion er forbundet med opgaven og drage dybere indsigt i problemet og måderne at løse opgaven med teknologiske løsninger, 3. Appreciation: Er forbundet med validering og skal tilgås så

objektivt som muligt til at vurdere bidraget om aktiviteterne stemmer overens med problemet, vision og Warrent, 4. Reasoning: Rationalisering er forbundet med Exit kriterierne, hvor Exit kriterierne skal sikre at aktiviteterne afsluttes med et rationale der stemmer overens med problem og løsning se (Figur 19), (Figur 18), eventuelt (Figur 27.1), (Figur 27.2), (Figur 27.3), (Tabel 7), (3.1 DSRM Aktiviteter) og (3.1.1 DSRM Projekt aktiviteter). Grundet at omdrejningspunktet er Low tech prototypen med udgangspunkt i værktøj artefaktet (Burndown chart) aktiviteterne se (Tabel 7) til at assistere koordinering af aktiviteterne af Backloggen i Scrum of Scrum vil evalueringen foregå som beskrevet i (4.6 Aktivitet 5: Evaluation), derefter diskuteret i (5 Aktivitet 7: Contribution & diskussion) og (6 Konklusion) som et bidrag i hvordan anvendes et burndown-chart, i et Scrum board, i distribueret softwareudviklings Scrum teams.

**Figur 19: Entry Task Validation og Exit**



### 3.3 Low tech prototyping

Som tidligere nævnt er Low tech prototypen ikke et svar på en ny løsning men en forlængelse af en eksisterende løsning med et særligt fokus i at understøtte Scrum of Scrum teams (Figur: 6), (Tabel: 2). Dermed er Low tech prototypen også særligt understøttet af software innovation (Rose. 2010). Ifølge (Rose. 2010) bliver Low tech prototyping anvendt i systemudvikling hvor prototyping bliver repræsenteret: I form af en horizontal prototyping, der er repræsenteret i prototyping niveauer, der går fra en lavteknologisk prototyping til en højteknologisk prototyping se (Figur 20). Ifølge (Figur 20) starter lavteknologisk prototyping med 1. funktionalitets liste og billeder (arkitektur og Rich picture), 2. Papirprototype (Wizard-of oz), 3. En low-fi skitse (PowerPoint), 4, eksisterende kode kombinationer, 5. En kode simulation og 6. En kode prototype.

#### 3.3.1 Demonstration af Low tech prototyping

I software innovation er der ikke en specifik normative process eller strategi for at vurdere anvendelsen af innovation i en Low tech prototype. Men der eksisterer en række metoder og proces frameworks for at adressere innovationen i softwareudviklingen af en Low tech prototype og et af disse frameworks er Low tech prototyping se (Figur 20). Ifølge Low tech prototyping i aktivitet 3 Design og udvikling skal jeg gennemgå 1-6 trin ud af de 6 trin ifølge (Figur 20) som et led af et samlet softwaresystem.

**Figur 20: Low tech prototyping table**  
(Rose. 2010 s. 70)

<b>low tech</b>		
	lists and pictures	feature list, simple architecture or component diagram, rich picture
	paper prototypes	sketches, wireframe, storyboard, card, wizard-of oz
	low-fi mockups	foam and cardboard models, drawing tool, video, PowerPoint, html
	existing code re-combination	mash-up, patchwork
	code simulation	screen generators, application definition or simulation software, component reuse, open source components
	code prototypes	rapid application generators, visual basic
<b>high tech</b>		

### 3.3.2 Eksperimenter med prototyper

Prototyper ifølge (Mathiassen et. al 2000. s 34) er defineret ved mindre komplekse systemer end et færdigt system, hvor prototype eksperimenter hjælper udviklere med at evaluere et system før et endeligt færdigt system. Hvor prototypen kan hjælpe udviklere med at evaluere system funktionalitet og før et færdigt system. I (Mathiassen et. al 2000. 34) opdeles prototyper i 2 prototype eksperimenter, en Eksplorative prototyper tilgang og en Evaluerings prototype tilgang. Hvor den Eksplorative tilgang fokusere på nye design og en Evalueringstilgang fokusere på at vælge et specifikt design.

Begge prototype typer variere fra kompleksitet til komponenter: Interface komponent, funktion komponent og model komponent. Oftest repræsenterer prototyper sjældent et helt færdigt komponent system og dermed kun repræsenterer billeder, vinduer, skitser, enkelte funktioner, få input af data og en model, der ikke opdatere med problem domæne forandringer. Dermed kan prototyper agere med begrænsede tekniske færdigheder, der ikke håndtere fejl i systemet, laver backups og databaser osv., med et eller flere evalueringsmål. Ifølge (Mathiassen et. al 2000) uanset tilgangen, er målet med prototype eksperimenter, at lære noget nyt: Der kan bidrage med noget nyt gøre os klogere på systemet eller gøre os bedre til at udvikle systemer (Mathiassen et. al 2000).

Grundet at distribueret softwareudvikling stadig er relativt nyt og derfor stadig mangler udforskning: I Hvordan der udvikles systemer til distribueret softwareudviklings teams og hvordan disse roller skal defineres og hvad der definerer sådan en rolle. Vil jeg i dette projekt udforske rollefordelingen i Scrum of Scrum med udgangspunkt i Burndown Chart, koordinering af aktiviteterne af Backloggen, timerne og Backlog strukturen.

## 4 DSRM AKTIVITETER

### 4.1 Problem-Centered Approach

I distribuerede softwareudviklingsprojekter er Scrum teams afhængig af koordinering af aktiviteterne af Product Backloggen: Da fordelene af en Backlog og et Burndown-Chart er stærkt afhængigt af at Scrum teamet er fysisk samlet omkring et Scrum board se (Figur 4: Burndown Chart, Figur 13: En Sprint iteration og Tabel 1), (*Esbensen et al. 2015.*), (*Sutherland and Schwaber. 2007*) og (*Sutherland, Schoonheim, Rijk. 2009*).

### 4.2 Aktivitet 1 Problem identification and motivation

Grundet at software projektudvikling er en avanceret proces til udvikling af teknologier, der skal løse eller understøtte forretningsproblemer og oftest foregår i teams: Er kompleksiteten høj grundet at kompleksiteten af et enkelt system er indlejret i et andet system og derfor for at håndtere kompleksiteten i software udviklingsfasen sammensættes et differentieret team med faste roller bestående af forskellige fagfolk og en Scrum Master som koordinator af et selvorganiserende Scrum team se (Tabel 1: Scrum of Scrum team struktur og Figur 8).

Dertil er det Scrum Masterens opgave at reducere kompleksiteten i teamets softwareudviklings fase (Et sprint) og dermed sørge for at teamet er så effektivt som muligt med mindst mulige forstyrrelser og den bedst mulig fremdrift af Product Backloggen.

Da et fysisk Scrum board med et Burndown-chart ikke kan flyttes til hvert enkelt distribueret lokation eller projekt. Har dette skabt en udfordring for distribueret softwareudviklings Scrum teams hvilket har skabt en efterspørgsel på en manglende løsning til koordinering af aktiviteterne af Product Backloggen i deres SOS teams se (Figur 21 og Figur 8), (*Esbensen et al. 2015.*), (*Sutherland and Schwaber. 2007*), (*Sutherland, Schoonheim, Rijk. 2009*), (*Baskerville, Pries-Heje, Madsen, 2011*), (*Conboy 2009*) og (*Persson, Nørbjerg og Nielsen 2016*).

### 4.3 Aktivitet 2: Define the objectives for a solution

I denne opgave vil jeg bidrage med at strukturere en arbejdsproces i distribueret softwareudviklings Scrum teams også kaldet SOS og fastholde agilitet. Det gør jeg med udgangspunkt i Burndown Chart i Scrum of Scrum med en tilhørende Scrum of Scrum Master rolle og et understøttende Low tech system.

Jeg anvender taxonomien fra (*Conboy 2009*) understøttet af (*Persson, Nørbjerg og Nielsen 2016*), da de i tråd med (*Baskerville, Pries-Heje, Madsen, 2011*) opsummerer agilitet, udfordringerne ved at adressere agilitet og efterspørgslen af værktøjer til at adressere agilitet i agile software organisationer. I denne opgave vil jeg anvende taxonomien fra (*Conboy 2009. s 14*), ved at tage udgangspunkt i taxonomiens evalueringskriterier. Hvor Conboy definerer følgende agilitets vurderingskrav i 3 spørgsmål og anvende dem efterfølgende på Scrum of Scrum Master rollen og Low tech prototypen se punkt 1,2 og 3: Dermed simplificere ledelsen, afhjælpe koordineringen og fremdriften i distribueret softwareudviklings Scrum teams.

## Agilitet i metoder for systemudvikling:

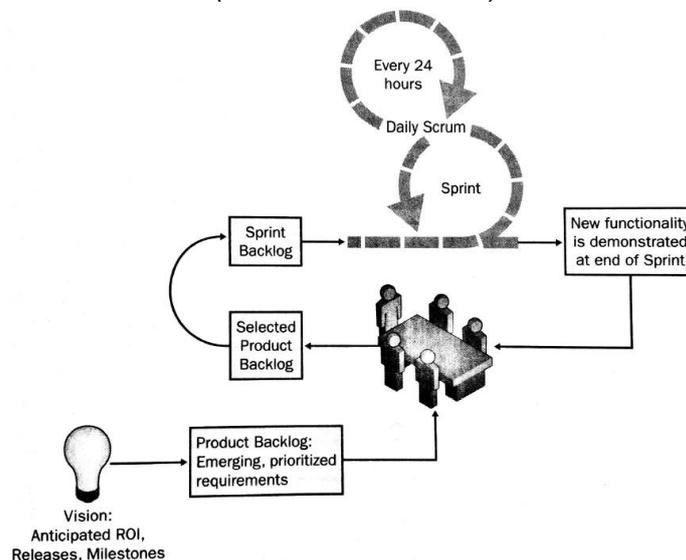
- |  |  |
|--|--|
| <ul style="list-style-type: none"><li>1. For at være agil, skal et metodeelement bidrage til en eller flere af de følgende:</li></ul>  | <ul style="list-style-type: none"><li>a. (i) Skabe forandring</li><li>b. (ii) Proaktivitet før forandring</li><li>c. (iii) Reaktion til forandring</li><li>d. (iv) Læring fra forandring</li></ul> |
| <ul style="list-style-type: none"><li>2. For at være agil, skal et metodeelement bidrage til en eller flere af de følgende og må ikke fratække fra nogen:</li></ul>                | <ul style="list-style-type: none"><li>a. (i) Økonomi</li><li>b. (ii) Kvalitet</li><li>c. (iii) Simplicitet</li></ul>   |
| <ul style="list-style-type: none"><li>3. For at være agil, skal et metodeelement løbende være klar, dvs. minimal tid og omkostninger til at forberede elementet til brug</li></ul> |  |

(Conboy 2009, s 14)

Da det kan være svært at estimere fremdrift for timerne i Scrum of Scrum, bliver timer vurderet i timepris pr. story point se (Figur 9), I denne opgave vil jeg tage højde for hvert enkelt teams behov, for at være et selvorganiserende team og dermed ikke estimere et gennemsnits story points pr. time, men efterlade dette til en Scrum of Scrum Master rolle i et Scrum of Scrum team og en Low tech prototype til en Scrum of Scrum Master rolle og et Scrum of Scrum team (Persson, Nørbjerg og Nielsen 2016), (Conboy 2009) og (Figur 8). Dertil er rollerne en vital del af Scrum of Scrum møderne for koordinering af aktiviteterne af Product Backlogen og derfor en del af kernen af et Scrum of Scrum team. Forudsat af en veldefineret Scrum of Scrum team struktur se (Tabel 1: Scrum of Scrum team struktur) og deres afhængig ved andre Scrum of Scrum team medlemmers evne til at kunne identificere vigtig information: Forstå den information og derved forstå hvordan de kan understøtte den information og det eksisterende arbejde i mellem teamet se (Figur 8) og (Tabel 1: Guidelines for distribueret produktudvikling mellem Scrum of Scrum teams, Scrum of Scrum meetings og Scrum of Scrum team struktur), (Ken Schwaber 2004). (Sutherland, Schoonheim, Guido. 2008) og (Conboy 2009).

**Figur 21: Lokaliseret Scrum proces**

(Ken Schwaber 2004)



## 4.4 Aktivitet 3: Design and Development

Da design and development tager udgangspunkt i at skabe et artefakt, har jeg valgt at udvikle en Low tech prototype bestående af nye tekniske egenskaber og informative egenskaber med udgangspunkt i artefaktet Burndown Chart. Her vil jeg definere Low tech prototypens funktionalitet og arkitektur. Hvor i resurser påkrævet fra at flytte fra objekt til design og udvikling, er inkluderet i teorien, som derefter kan bibringe en løsning. Da distribueret softwareudviklingsprojekter er afhængig af koordinering af aktiviteterne af Product Backlogen, (Figur 4: Burndown Chart, Figur 13: En Sprint iteration og Tabel 1), (Esbensen et al. 2015.), (Sutherland and Schwaber. 2007) og (Sutherland, Schoonheim, Rijk. 2009).

Første skridt i at starte et projekt er afhængigt af et problem. Dette problem leder en mod en løsning, i mit tilfælde var problemet igangsat af et arrangement med Nykredit omhandlende deres distribueret softwareudviklings Scrum teams se (4.1 Problem-Centered Approach).

### 4.4.1 Scrum of Scrum Master rolle

Hvad er Scrum og hvad har Scrum at gøre med koordinering af aktiviteterne af Backlogen og timerne at gøre i distribueret softwareudviklingsprojekter? Scrum er et simpelt inspektions og tilpasnings framework der har 3 roller, 3 ceremonier og 3 artefakter designet til at levere kvalitets software i Sprints (Sutherland and Schwaber. 2007. s 11-12) designet til fysiske lokaliseret teams og ikke et Scrum of Scrum Team se (Tabel 2).

Hvor et Scrum team har et Task board eller Scrum board der repræsenterer en Sprint backlog: Hvor Sprint Backlogen giver teamet muligheden for at tilføje nye opgaver, identificere nye opgaver, dele opgaver i mindre bidder, italesætte opgaver der ikke kan færdiggøres under et sprint, fjerne eksisterende opgaver og til sidst opdatere værktøjet (Burndown-Chart'et) som referencepunkt for team fremdriften se (Figur 4: Sprint Burndown-chart:) Hvor sprints er anvendt til at udvikle det færdige produkt fra det første sprint, til det sidste sprint i et projekts levetid og under udviklingsprocessen af det samlede projekt: Er projektet åbent overfor ændringer i sprint planlægningsfasen i projektet. Derfor bliver projektet ved med at være åbent overfor tids komplekse -, konkurrence -, tids -og kvalitetspræget ændringer (Schwaber Ken and Beedle Mike. 2002) og (Fitzgerald, Hartnett, Conboy 2006)

Hvor Scrum's tilgang og rødder stammer fra forventningen om: At analyse, design og udviklingsprocesser er uforudsigelige. Derfor komprimere Scrum en række faser i mindre enheder kaldet Sprints, som er karakteriseret ved en ikke lineær og fleksible process. Hvor Scrum tilfalder agile metoder under det Agile Manifesto: Individuer og interaktioner frem for processer og redskaber, fungerende software over dokumentation, kundesamarbejde over kontrakter og forhandlinger og agere på forandring - frem for at følge en plan (Fowler and Highsmith, 2001 p. 7,1-7 : se eventuelt Agile Manifesto under på engelsk) og (Beck et al. 2001).

Hvor imod SOS er flere Scrum teams der koordinere og arbejder sammen om et fælles produkt med et fælles mål: Hvis der derimod fokuseres på 2 forskellige produkter er det to forskellige Scrum of Scrum med hver deres fælles mål og hver deres produkt. Scrum of Scrum er placeret ud på flere lokations og hvor hvert team har et Scrum team og en Scrum Master. Hvor dette tager udgangspunkt i aktiviteterne af koordineringen af software

produktudviklingen i flere lokation, hvor aktiviteterne af koordineringen af et fælles produkt er centraliseret i et Scrum of Scrum afhængig af flere distribueret teams. Hvor teamene koordinere via Scrum of Scrum teams møder og en eller flere repræsentanter fra hvert distribueret team deltager Se (Tabel 1), (*Mundra; Misra og Dhawale. 2013*), (*E. Hossain, A. B. Muhammad og y. P. Hye. 2009*), (*A. Lamersdorf, M. Jürgen and R. Dieter. 2009*), (*Maria, Casper og Ville. 2012*).

Da der i kernen af et Scrum team og den agile software produkt udviklingsproces bliver anvendt et Scrum board hvor et task board eller tavle bliver placeret i et softwareudvikling teams kontor. Hvor tavlen agere som bindende led mellem udviklere og opgaver for at afhjælpe aktiviteterne af koordineringen og fremdriften af det daglige Scrum of Scrum møde.

Hvor er visualiseringen af en effektiv repræsentation se (Tabel 2, Tabel 1 og Figur 13: En Sprint iteration) er afhængig af Scrum of Scrum teamets evne til at kunne identificere vigtig information: Forstå den information og derved forstå hvordan de kan understøtte den information og det eksisterende arbejde i mellem teamet se (Tabel 1: Guidelines for distribueret produktudvikling mellem Scrum of Scrum teams, Scrum of Scrum meetings og Scrum of Scrum team struktur). Dermed er visualisering af softwareudviklingen i softwareprojekter betragtet som en vital del og nødvendig del af optimering af software kvaliteten i et softwareprojekt og hvordan vi forstår bidraget af kvaliteten i et softwareprojekt.

Derfor er artefakt værktøjer (Burndown Chart) og tilhørende aktiviteter vigtige for software udviklingsprocessen til at opmærksom gøre teamet omkring udviklingen af projektet og koordinering af projektet se (Tabel 1: Scrum artefakter), (*J. Paredes, C. Anslow, and F. Maurer. 2014*).

Videokonference teknologier har været anvendt i sammenhæng med et fysisk Scrum board, hvor Scrum boardet har ageret som redskab i videokonferencen til at afhjælpe koordineringen og fremdriften af det daglige Sprint møde. (*Esbensen et al. 2015.*), (*Sutherland and Schwaber. 2007*) og (*Sutherland, Schoonheim, Rijk. 2009*).

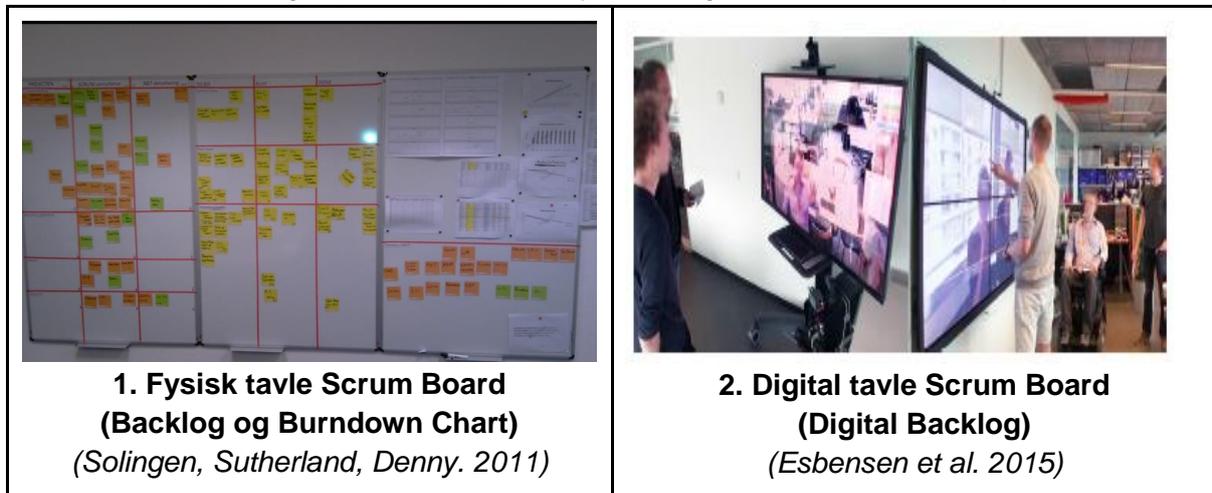
Hvor et Task board eller Scrum board har repræsenteret en Product backlog og Sprint Backlog og dermed gav muligheden for aktiviteterne at tilføje nye opgaver, identificere nye opgaver, dele opgaver i mindre bidde, fjerne eksisterende opgaver, italesætte opgaver der ikke kunne færdiggøres under et sprint og til sidst opdatere værktøjet (Burndown-Chart) som visualiseringsværktøj for den samlet team fremdrift se (Tabel 1) og (Figur 4: Sprint Burndown-chart:),(*Rubart. 2014*).

Da opdateringen omkring fremdriften foregår efter et Daily stand-up meeting hvor i Scrum of Scrum teams deltager skal opdatere omkring: fremdriften fra sidste møde, deres nuværende fremdrift og deres planlagte fremdrift til næste møde se (Tabel 1). Er aktiviteterne af opdateringen af fremdriften i artefakt værktøjet (Burndown Chart) i distribueret softwareudvikling Scrum teams uafhængigt af værktøj artefakterne (Product Backlog og Sprint Backlog) Se (Tabel 1: Scrum of Scrum meetings, Scrum 3 Artefakter og Guidelines for distribueret produktudvikling mellem Scrum of Scrum teams).

Dertil påvirker artefaktet Burndown Chart hele teamets koordinering af aktiviteterne af Backloggen og timerne se (Figur 4: Burndown Chart, Figur 13: En Sprint iteration og Tabel 1) i de enkelte Scrum teams i et Scrum of Scrum, dermed også Scrum of Scrum mødet se (Tabel 1: Scrum of Scrum meetings), strukturen for Scrum of Scrum mødet (Tabel 1: Scrum of Scrum team struktur) og ceremonierne for Scrum teams efterfølgende (Tabel 1: Scrum 3 Ceremonier) (*Mundra; Misra og Dhawale. 2013*), (*E. Hossain, A. B. Muhammad og y. P. Hye. 2009*), (*A. Lamersdorf, M. Jürgen and R. Dieter. 2009*), (*Maria, Casper og Ville. 2012*). Da fremdriften aktiviteterne af distribueret softwareudviklings Scrum teams er indlejret i

værktøjet (Burndown Chart) se (Tabel 1: Scrum of Scrum meetings og Scrum 3 Artefakter) og (Tabel 1: Scrum 3 Roller), har jeg valgt at videreudvikle på aktiviteterne for Burndown Chart for at strukturere aktiviteterne af arbejdsprocessen i distribueret softwareudviklings Scrum teams og fastholde agiliteten i Scrum of Scrum (*Baskerville, Pries-Heje, Madsen, 2011*), se (Figur 21 og Figur 8), (*Esbensen et al. 2015.*), (*Sutherland and Schwaber. 2007*), (*Sutherland, Schoonheim, Rijk. 2009*), (*Persson, Nørbjerg og Nielsen 2016*) og (Figur 22).

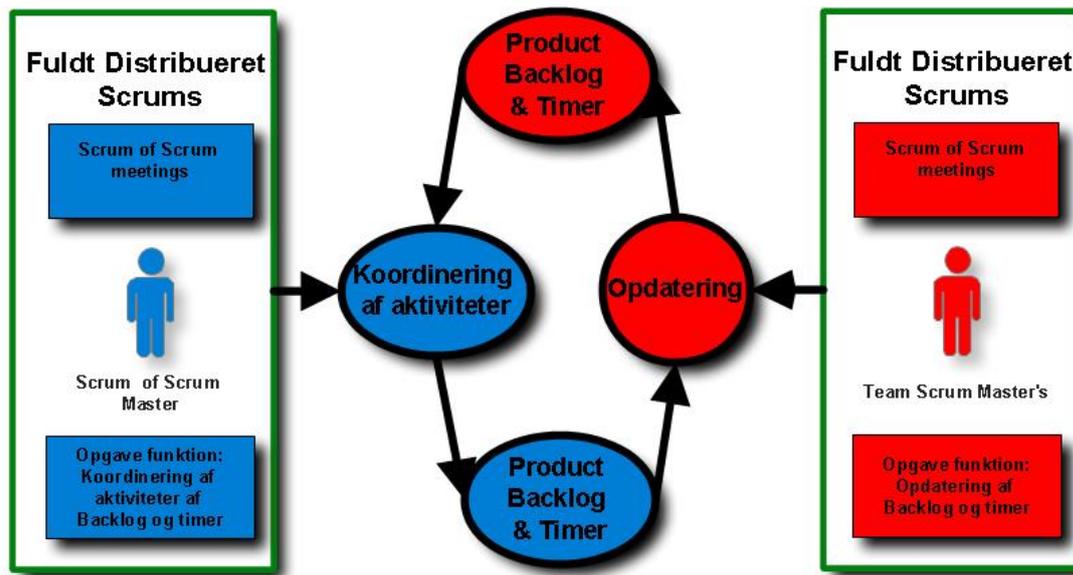
**Figur 22: 1. Fysisk tavle Scrum Board, 2. Digital tavle Scrum Board,**  
(*Solingen, Sutherland, Denny. 2011 og Esbensen et al. 2015*)



I lokaliseret Scrum teams tiltræder Produkt Backlog opgaver i Sprintet som faste opgaver i en periode af sprintet, men stadig kan ændres i Backloggen af følgende grunde: 1. Udviklings teamet får en bedre forståelse af opgaven og i finder sig nødsaget til at tilføje eller fjerne en opgave til Sprint Backloggen for at nå i mål med de givne Product Backlog opgaver, 2. Fejl bliver identificeret og gemt som ekstra opgaver. Hvor ekstra opgaverne bliver defineret som udefineret opgaver og markeret som ufærdig opgaver, 3. Product Owner arbejder med teamet i løbet af Sprintet, får indsigt i opgaven og derfor ændre Sprint målene: Scrum Masteren og teamet vælger at ændre eller forkorte eller forlænge Sprintet for at optimere kundeværdien af en eller flere opgaver se (Figur 4: Burndown Chart, Figur 13: En Sprint iteration og Tabel 1).

Hvor i distribueret softwareudviklings Scrum teams eller Scrum of Scrum teams foregår opdateringen af opgaver efter et Daily stand-up meeting se (Figur 23). Hvor Scrum of Scrum teams deltager opdatere omkring: fremdriften fra sidste møde, deres nuværende fremdrift og deres planlagte fremdrift til næste møde, dertil koordinere derefter og løser stridighederne i deres Scrum teams se (Tabel 1), (Figur 24) og (Figur 4: Burndown Chart, Figur 13: En Sprint iteration og Tabel 1), (*Mundra; Misra og Dhawale. 2013*), (*E. Hossain, A. B. Muhammad og y. P. Hye. 2009*), (*A. Lamersdorf, M. Jürgen and R. Dieter. 2009*), (*Maria, Casper og Ville. 2012*), (*Sutherland og Schwaber. 2007*) og (Figur 23).

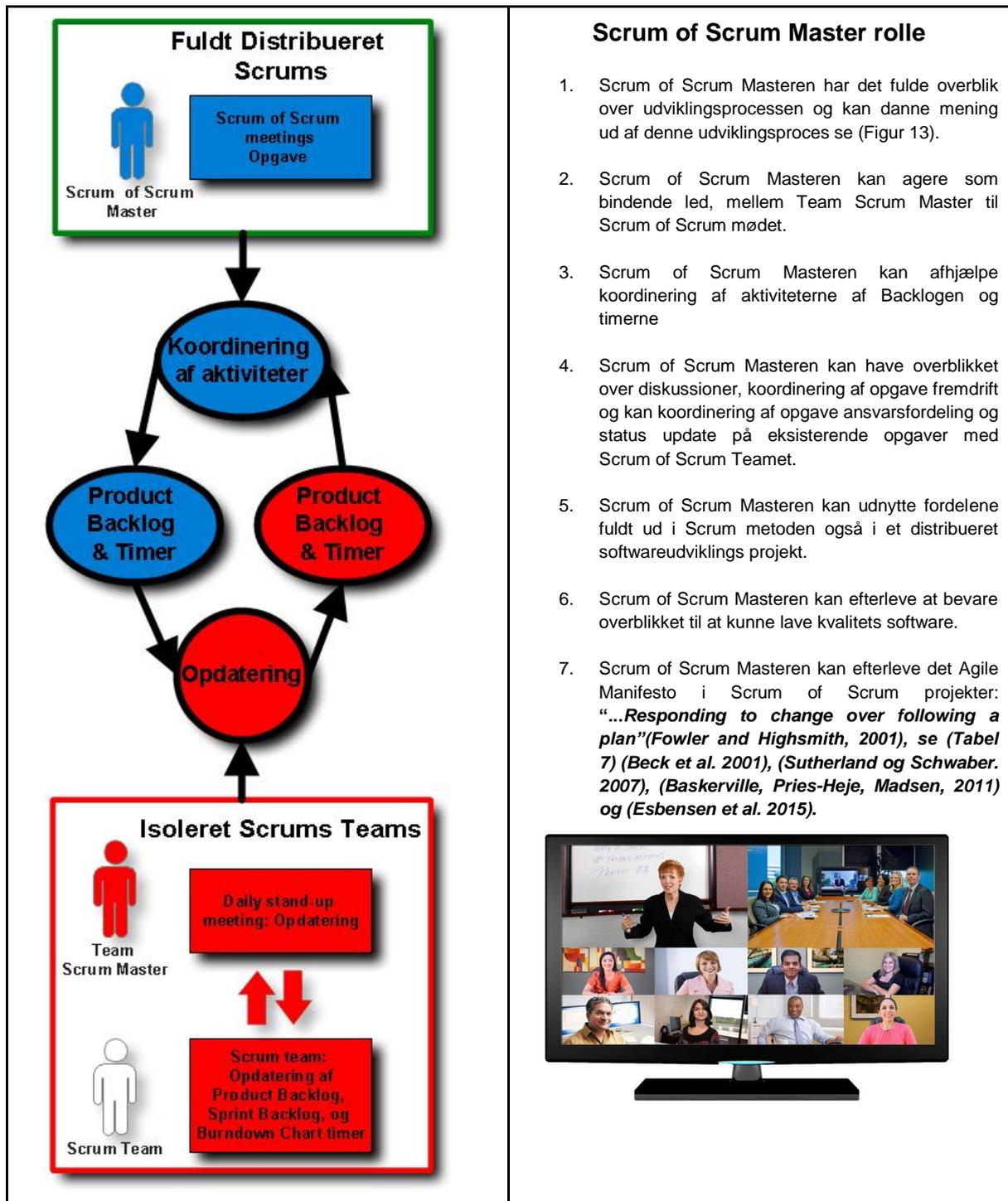
Figur 23: Distribueret Scrum of Scrum team



Da agil softwareudviklingsprocesser er centraliseret omkring inkrementelle leveringer se (Figur 13: En Sprint iteration) som et led i at skabe kvalitets software, hvor (Figur 23) illustreres afhængighedsforholdet af aktiviteterne mellem Scrum of Scrum Masteren og Team Scrum Master: Hvor udviklingsprocessen af projektet handler om at opdage opgaver, håndtere opgaver undervejs og koordinere dem efterfølgende se (Tabel 1: Guidelines for distribueret produktudvikling mellem Scrum of Scrum teams) og (Figur 4: Burndown Chart, Figur 13: En Sprint iteration og Tabel 1), (Esbensen et al. 2015.), (Sutherland and Schwaber. 2007) og (Sutherland, Schoonheim, Rijk. 2009).

For at definere Scrum of Scrum Master rollen yderligere og koordineringsopgaverne yderligere har jeg defineret dem således se (Figur 24: Scrum of Scrum Master rolle) punkt 1-7: 1. Scrum of Scrum Masteren har det fulde overblik over udviklingsprocessen og kan danne mening ud af denne udviklingsproces se (Figur 13), 2. Scrum of Scrum Masteren kan agere som bindende led mellem Team Scrum Master til Scrum of Scrum mødet, 3. Scrum of Scrum Masteren kan afhjælpe koordinering af aktiviteterne af Backloggen og timerne, 4. Scrum of Scrum Masteren kan have overblikket over diskussioner, koordinering af opgave fremdrift og kan koordinering af opgave ansvarsfordeling og status update på eksisterende opgaver med Scrum of Scrum Teamet, 5. Scrum of Scrum Masteren kan udnytte fordelene fuldt ud i Scrum metoden også i et distribueret softwareudviklings projekt med eller uden et system, 6. Scrum of Scrum Masteren kan efterleve at bevare overblikket til at kunne lave kvalitets software, 7. Scrum of Scrum Masteren kan efterleve det Agile Manifesto i Scrum of Scrum projekter: "...Responding to change over following a plan"(Fowler and Highsmith, 2001), se (Tabel 7), (Beck et al. 2001), (Sutherland og Schwaber. 2007), (Baskerville, Pries-Heje, Madsen, 2011), (Esbensen et al. 2015) og (Figur 6).

Figur 24: Scrum of Scrum Master rolle (24)



### Scrum of Scrum Master rolle

1. Scrum of Scrum Masteren har det fulde overblik over udviklingsprocessen og kan danne mening ud af denne udviklingsproces se (Figur 13).
2. Scrum of Scrum Masteren kan agere som bindende led, mellem Team Scrum Master til Scrum of Scrum mødet.
3. Scrum of Scrum Masteren kan afhjælpe koordinering af aktiviteterne af Backloggen og timerne
4. Scrum of Scrum Masteren kan have overblikket over diskussioner, koordinering af opgave fremdrift og kan koordinering af opgave ansvarsfordeling og status update på eksisterende opgaver med Scrum of Scrum Teamet.
5. Scrum of Scrum Masteren kan udnytte fordelene fuldt ud i Scrum metoden også i et distribueret softwareudviklings projekt.
6. Scrum of Scrum Masteren kan efterleve at bevare overblikket til at kunne lave kvalitets software.
7. Scrum of Scrum Masteren kan efterleve det Agile Manifesto i Scrum of Scrum projekter: *"...Responding to change over following a plan"*(Fowler and Highsmith, 2001), se (Tabel 7) (Beck et al. 2001), (Sutherland og Schwaber. 2007), (Baskerville, Pries-Heje, Madsen, 2011) og (Esbensen et al. 2015).



#### **4.4.2 Product Backlog hierarki struktur for SOS Masteren og prototypen**

Ifølge eksemplet i (*Ken Schwaber 2004. s. 128*) se (Figur 7) varetog en Product Owner den fælles Scrum of Scrum Backlog bestående af flere samlet Backlogs og varetog rollen som Product Owner og Scrum of Scrum Master.

Hvor i Product Owneren også havde ansvaret for at spørge Scrum teamene i et Scrum of Scrum meeting: Hvad kunderne ønskede implementerede fra Backlogen, Hvilken funktionalitet de ønskede implementeret først fra Backlogen og der igennem prioriterede den fælles Product Backlog efter hvert teams sprint kapacitet i det fælles Scrum of Scrum. Dermed tog hensyn til hvert enkelt teams varierende sprint kapacitet i de enkelte Scrum teams Backlogs og planlagde de enkelte teams Sprint efterfølgende.

I dette projekt vil jeg anvende den ovenstående Product Backlog hierarki struktur for Low tech prototypen og derfor tage udgangspunkt i 1 Product Owner der varetager prioriteringen af Scrum of Scrum Product Backlogen og en Scrum of Scrum Master rolle til koordinering af aktiviteterne af Product Backlogen i Scrum of Scrum teamet.

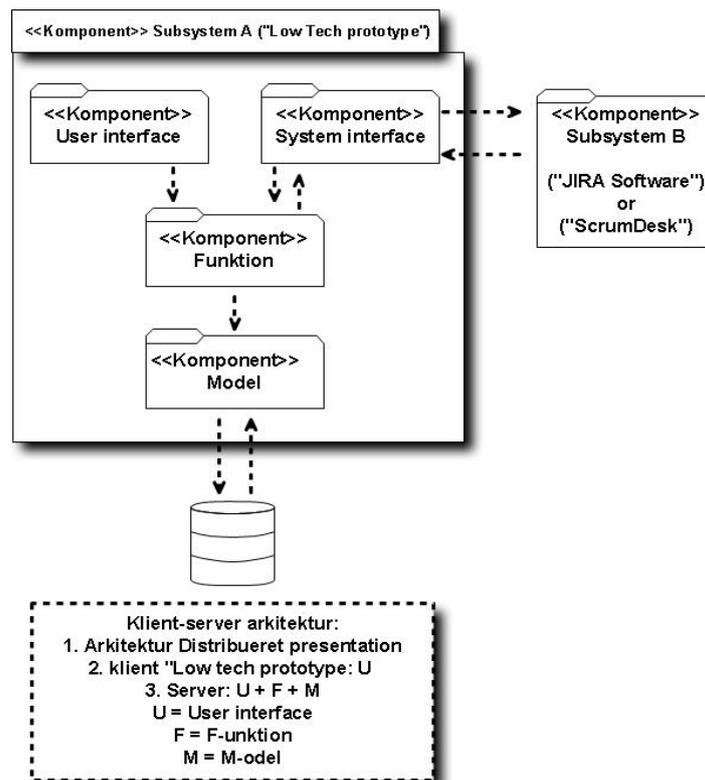
#### **4.4.3 System arkitekturen af Low tech prototypen**

I skaleringen af ethvert projekt skal der defineres en tilhørende arkitektur: Et system for at synkronisere deres arbejde, dele deres arbejde og koordinere arbejdet. Også en mere detaljeret Produkt Backlog og teknisk arkitektur så opgaverne kan fordeles ligeligt mellem teamene. Hvis teamene er geografisk distribueret, skal der anvendes høj båndbredde, for at kunne koordinere opgaver og timer (*Ken Schwaber 2004*).

I dette projekt vil der blive fokuseret på arkitekturen og hvordan den kan være med til at understøtte Scrum of Scrum master rollen hvor koordinering består i, at koordinere aktiviteterne af Product Backlogen. For at mindske forvirringen vil fremdriften blive omtalt i timer i arkitekturen og ikke i Story points se (Figur 4: Burndown Chart, Figur 13: En Sprint iteration og Tabel 1), (*Esbensen et al. 2015.*), (*Sutherland and Schwaber. 2007*), (*Sutherland, Schoonheim, Rijk. 2009*) og (*Ken Schwaber 2004*).

I (*Mathiassen et. al 2000*) defineres et subsystem ved at interface system komponent arkitekturen giver andre subsystemer adgang til et transparent subsystems funktionalitet. Hvor System interface og User interface komponenter stadig har den samme funktionalitet. Som agere funktions komponentet afhængigt af system interface komponentet, dermed indgår anmodninger igennem user interface komponentet til funktions komponentet. Dermed skal funktions komponentet vide hvilke funktioner der er blevet anmodet i dets eget system komponent Model og er anmodet igennem andre subsystemer. Hvorved hvis der kommer en anmodning fra et andet subsystem, går de direkte igennem system interface komponentet og hvorefter de anmodninger bliver dirigeret til de relevante subsystemers interface komponenter og dermed stadig fuld transparens i systemet se (Figur 25: Relaxed arkitektur af subsystemer), (*Mathiassen et. al 2000*), (*Atlassian 2017: JIRA*) og (*ScrumDesk 2017*).

**Figur 25: Relaxed arkitektur af subsystemer**



#### 4.4.4 System anvendeligheden af applikations domænet af Low tech prototypen

Når det handler om applikations domænet, handler det om at vurdere hvordan aktørerne interagerer med et system. Her anvendes Uses Cases som kan analysere et eksisterende applikations domæne eller et nyt uden at skulle lave den mængde detaljeret information om et gammelt system der ingen værdi eller nytteværdi har for udviklingen af et nyt system. For at give et hurtigt overblik over Applikations domæne begreber se (Tabel 4).

**Tabel 4: Applikations domænes begrebsdefinition**

Begrebsdefinition	Funktionsbeskrivelse
1. Actors	En abstraction for brugere eller andre systemer der interagerer med Low tech prototypen.
2. Use cases	Mønstre for interaktionen mellem systemet og actors i applikations domænet.
3. Princip	1. Princip At definere applikations domænet med Use Cases, Princip 2. Evaluere Use Cases med brugerne af systemet og 3. Vurdere forandringerne i applikations domænet.
4. Resultatet	En beskrivelse af alle Use Cases med og Actors

Nu hvor jeg har skabt et overblik over applikations domæne begreberne vil jeg fortsætte med en Actor table og beskrive Actors for Low tech prototypen og Use Cases i en tabel se (Tabel 5). I Tabel 5 ses der i første række Actors, hvorefter første rækken er efterfulgt af 4 Actors der interagerer med systemet bestående af: En Scrum of Scrum Master, Team Scrum Master, En administrator og en time Monitor og Burndown Chart. Hvor første Kolonne ses Use Cases: Sidste Sprint Time Opdatering, Nuværende Sprint Time Opdatering, Næste Sprint møde Time Opdatering, Samlet Time opdatering, Samlet Time information, Backlog Opdatering, Backlog information, Registrering, Monitoring og Fejljustering.

I Tabel 5 ses hvordan Scrum of Scrum Masteren interagerer med systemet for at understøtte koordineringen af aktiviteterne af Product Backlog opgaverne, understøttet af Team Scrum Masteren der allerede har opdateret den Sidste Sprint Time Opdatering, Nuværende Sprint Time Opdatering og Næste Sprint møde Time Opdatering (Tabel 1: Scrum of Scrum meetings). Dermed reduceres simplificeres eller kompleksiteten af aktiviteterne af opgave koordinering af Backlog opgaverne se (Figur 8) og hvor han derefter kan se den Samlet Time information i Time Monitor Burndown Chart'et. Hvorefter systemet eller Scrum Masteren kan regne gennemsnittet af teamenes timer vurderet i timepris pr. story point for de enkelte Scrum of Scrum teams ud, hvor der stadig er taget højde for hvert enkelt teams behov for at være et selvorganiserende team og selv estimere et gennemsnits story points pr. time for opgaverne i de selvorganiserende Scrum teams.

Dermed bevares praksis i Scrum teamet i Scrum of Scrum teamet i opgave estimering pr. story point pr. time og Scrum of Scrum Masteren bevare overblikket over Scrum of Scrum teamenes gennemsnits estimering for et fælles Scrum of Scrum team, uden at forstyrre Scrum of Scrum team eller underminere Scrum teams. Dermed fastholdes fleksibiliteten i at estimere et tidsestimat afhængigt af opgaven eller specifikationerne af opgaven for teamet og teamet kan reagere på forandringerne i specifikationer og stadig været fokuseret omkring den enkelte opgave og dens tidsestimat. Teamet bevare overblikket over det samlede projekt i Scrum teamet, uden at være påvirket af en Scrum of Scrum Masters beslutninger eller aktiviteter (Tabel 1: Scrum of Scrum team struktur og Figur 8), (*Baskerville, Pries-Heje, Madsen, 2011*) og (*Conboy 2009*). Samtidigt med at Scrum of Scrum Masteren kan bevare overblikket, bliver han understøttet af Administrator der underretter Scrum of Scrum Masteren hver gang der foregår en Registrering af en Team Scrum Master eller en Fejljustering opstår i Lowtech prototypen se (Figur 4: Burndown Chart, Figur 13: En Sprint iteration, Tabel 1 og (Figur 8).

**Tabel 5: Actor tabel for Low tech prototypen**

	<b>Actors</b>			
<b>Use Cases</b>	Scrum of Scrum Master	Team Scrum Master	Administrator	Time Monitor (Burndown Chart)
Sidste Sprint Time Opdatering	x	x		
Nuværende Sprint Time Opdatering	x	x		
Næste Sprint møde Time Opdatering	x	x		
Samlet Time opdatering	x		x	
Samlet Time information	x		x	x
Backlog Opdatering	x	x		
Backlog information	x	x	x	
Registrering			x	
Monitoring			x	
Fejljustering			x	

#### **4.4.5 Use Case diagram for Low tech prototypen**

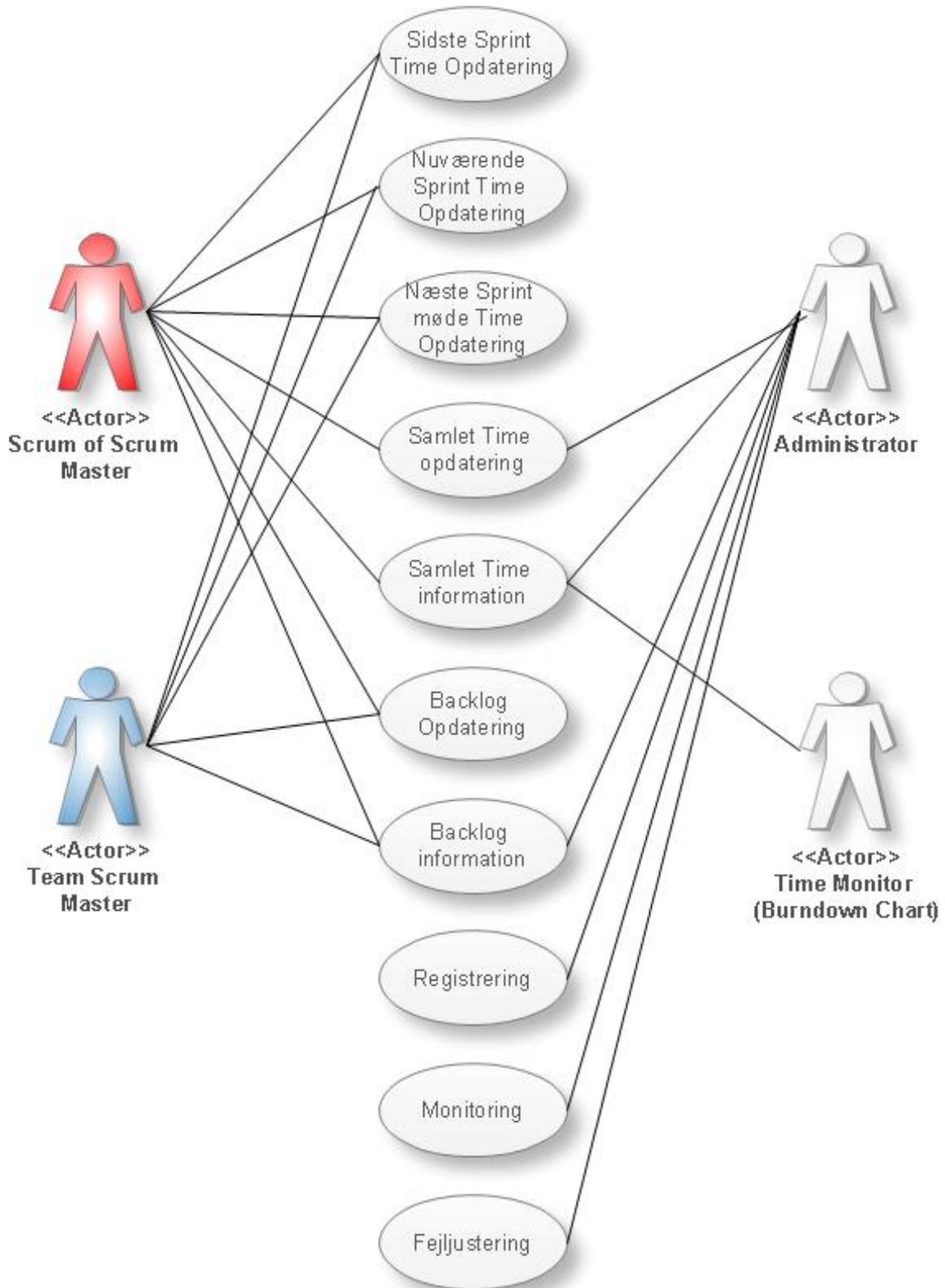
I Use case diagrammet kan man se Actor 1 Scrum of Scrum Master interagere med 7 Use Cases bestående af 1. Sidste Sprint Time Opdatering, 2. Nuværende Sprint Time Opdatering, 3. Næste Sprint møde Time Opdatering, 4. Samlet Time opdatering, 5. Samlet Time information, 6. Backlog Opdatering og 7. Backlog information.

Hvor Actor 2 Team Scrum Master interagere med 5 Use Cases bestående af 1. Sidste Sprint Time Opdatering, 2. Nuværende Sprint Time Opdatering, 3. Næste Sprint møde Time Opdatering, 4. Backlog Opdatering og 5. Backlog information.

Actor 3 Administrator interagere med 6 Use Cases bestående af 1. Samlet Time opdatering, 2. Samlet Time information, 3. Backlog information, 4. Registrering, 5. Monitoring og 6. Fejljustering.

Og Actor 4 Time Monitor Burndown Chart interagere med 1 Use Cases bestående af Samlet Time information.

**Figur 26:Use Case diagram for Low tech prototype**



#### 4.4.6 Beskrivelse af Actor Specifikationer for Low tech prototypen

Actor 1 Kravspecifikationer	Scrum of Scrum Master
<b>Mål:</b>	En person som står for hele Scrum of Scrum teamet. Hans primær behov er koordinering af aktiviteterne af Backlog opgaverne og timerne i Scrum of Scrum meetings
<b>Karaktertræk:</b>	1. Føler sig i stand til at bevare overblikket over Scrum of Scrum teamets opgaver og timer, 2. Lynhurtigt kan reagere på forandringerne i opgaver, timer, story points, 3. Stadig være fokuseret på de enkelte Scrum of Scrum team opgaver, de individuelle teams tidsestimater og det samlet Scrum of Scrum tidsestimat og 4. Sørge for at alle Scrum teams er mest muligt produktive i forhold til deres individuelle fremdrift efter deres Scrum team se (Figur 27.1).
<b>Eksempler:</b>	1. En team Scrum Master estimere gentagende gange forkert og systemet underretter Scrum of Scrum Masteren. Hvorefter Scrum Masteren underretter Team Scrum Masteren og han underretter Scrum of Scrum Masteren at fejlen skyldes manglende Team medlemmer og 2. Et team har mistet et vigtigt team medlem og derfor ikke kan levere den ønskede fremdrift, men et andet team levere for meget fremdrift og derfor flyttes et teammedlem til at understøtte den fremdrift se (Figur 27.3)

Actor 2 Kravspecifikationer	Team Scrum Master
<b>Mål:</b>	En person som står for at opdatere, hvordan det går på hans Scrum teams fremdriften fra sidste møde, deres nuværende fremdrift, deres planlagte fremdrift til næste møde for Scrum of Scrum teamet.
<b>Karaktertræk:</b>	1. Føler sig i stand til at bevare overblikket over sit Scrum teams opgaver og timer i forhold til Scrum of Scrum teamets fælles mål, 2. Lynhurtigt kan reagere på forandringerne i opgaver, timer, story points i det samlet Scrum of Scrum team, 3. Stadig være fokuseret på de enkelte opgaver i Scrum teamet, deres timer, story point, de individuelle Scrum teams medlemmers produktivitet og det samlet Scrum teams produktivitet og 4. Sørge for at alle Scrum team medlemmer er mest muligt produktive i forhold til deres individuelle produktivitet og velfærd efter deres Scrum team uafhængigt af Scrum of Scrum teamet se (Figur 27.2).
<b>Eksempler:</b>	1. Teamet rammer ikke gentagende gange deres estimeret fremdrift og Scrum of Scrum Masteren spørg hvad årsagen er, hvor Team Scrum Masteren svare, at alle er så effektive som de kan være. Jeg tror at det skyldes opgave kompleksiteten af opgaven og derfor ønsker jeg et teammedlem mere, for at nå den ønsket Scrum of Scrum fremdrift se (Figur 27.3).

Actor 3 Kravspecifikationer	Administrator
<b>Mål:</b>	En person eller system som står den samlet time opdatering, samlet time information, Backlog information, registrering, monitoring og fejljustering.
<b>Karaktertræk:</b>	1. Et system eller en person der gør at Scrum of Scrum Masteren og Team Scrum Mastere føler sig i stand til at bevare overblikket over deres Scrum of Scrums opgaver og timer i forhold til Scrum of Scrum teamets fælles mål og produkt, 2. De lynhurtigt kan reagere på forandringerne i opgaver, timer, story points i det samlet Scrum of Scrum team, 3. de Stadig kan være fokuseret på de individuelle opgaver i Scrum of Scrum teamet og være mest muligt produktiv i det samlet Scrum of Scrum teams produkt og 4. Sørge for at alle Scrum of Scrum team medlemmer er mest muligt produktive i forhold til deres individuelle produktivitet og velfærd i Scrum of Scrum teamet.
<b>Eksempler:</b>	Kunden vælger at forlænge projektets samlet levetid på grund af projektets øget kompleksitet og mangel på Scrum Team medlemmer i de enkelte teams. Dermed efterlader Scrum of Scrum Masteren og Team Scrum Masteren om at fordele fremdriften i mellem sig og få mest muligt udbytte af fremdriften fordelt på de forskellige Scrum teams i det fælles Scrum of Scrum team, for at lette fremdrifts presset på den Samlet Time Monitor Burndown Chart se (Tabel 7)

Actor 4 Kravspecifikationer	Time Monitor (Burndown Chart)
<b>Mål:</b>	Et system der bevare overblikket over den samlet Time information
<b>Karaktertræk:</b>	Sørger for at Scrum of Scrum Masteren og administratoren hele tiden er opdateret omkring den samlet time information for det samlet Scrum of Scrum team og de individuelle Scrum teams i det samlet Scrum of Scrum team se (Tabel 7)
<b>Eksempler:</b>	2 ud af 4 Teams over performer hele tiden i deres fremdrift og 1 ud af 4 teams underperformer i Scrum of Scrum teamet. På baggrund af den samlet time information fra Burndown Chart'et, vælger Scrum of Scrum Masteren at spørge de 2 ud af 4 Team Scrum master, om de kan undvære nogle teammedlemmer til det 1 ud af 4 teams for Scrum of Scrum teamets fælles mål. Derefter rammer alle 4 ud af 4 Scrum teams deres fremdrift og det samlet Scrum of Scrum team er mest muligt produktivt se (Figur 23.1), (Figur 23.2), (Figur 23.3), (Tabel 7), (Conboy 2009) og (Persson, Nørbjerg og Nielsen 2016).

Nu hvor jeg har skabt Actors, Use Cases og forholdet imellem dem se (Figur 26, Tabel 5 og 4.5.2 Beskrivelse af Actor Specifikationer). Er det nu vigtigt at evaluere dem, ved at gennemgå dem systematisk og derefter teste Use Casene og se om de fungerer i praksis (Conboy 2009), (Persson, Nørbjerg og Nielsen 2016) og (*Baskerville, Pries-Heje, Madsen, 2011*). Grundet at omdrejningspunktet er Low tech prototypen med udgangspunkt i værktøj artefaktet (Burndown chart) aktiviteterne til at assistere koordinering af aktiviteterne af Backloggen i Scrum of Scrum vil evalueringen foregå som beskrevet i (4.6 Aktivitet 5: Evaluation), derefter diskuteret i (5 Aktivitet 7: Contribution & diskussion) og (6 Konklusion).

#### **4.4.7 Løsning af Scrum of Scrum Master rollen og Low tech prototypen**

Måden jeg har løst problemet i aktivitet 1 og 2: Er ved at definere en ny rolle i distribueret softwareudviklingsprojekter og et tilhørende artefakt bestående af et Low tech system, i forlængelse af værktøj aktiviteterne af Burndown Chart og rollen af en Scrum of Scrum Master (Tabel 2, 2, Figur 22 ,Figur 26, Tabel 5 og 4.5.2 Beskrivelse af Actor Specifikationer).

Hvor Scrum of Scrum Master rollen foregår i forlængelse af den almindelige Scrum Master rolle, hvor der tidligere blev udnævnt en Scrum Master i et almindeligt Scrum team bliver der udnævnt en Scrum of Scrum Master til Scrum of Scrum teamet, der er agere som en fuldbyrdet Scrum Master se (Tabel 1) med ansvaret for Scrum of Scrum teamet. Dermed er opsummeringen af hvad løser rollen og systemet: At afhjælpe koordinering af aktiviteterne af Backloggen og timerne i et distribueret softwareudviklingsprojekt ved at tage udgangspunkt i Burndown Chart se (Tabel 1: Scrum 3 Artefakter) og (Tabel: 2).

Dermed bliver koordinering af aktiviteterne af Backloggen og timerne understøttet af rollefordelingen i distribuerede softwareudviklingsprojekter se (Tabel 1: Guidelines for distribueret produktudvikling mellem Scrum of Scrum teams): Ved at koordinering af aktiviteterne af Backloggen og timerne bliver understøttet af rollen Scrum of Scrum Master og Low tech prototype (Figur 22 ,Figur 26, Tabel 5 og 4.5.2 Beskrivelse af Actor Specifikationer).

Hvorvidt om der skal udnævnes en Scrum of Scrum Product Owner der skal assistere Scrum of Scrum Masteren, vil jeg overlade til andre forskere: Da der ikke under et Scrum of Scrum møde løses interne stridigheder på team niveau da det ansvare tilhøre en team Scrum Master. Hvor Scrum Masteren allerede er tilstede under Scrum of Scrum mødet se (Tabel 1: Scrum of Scrum team struktur) og et Scrum of Scrum team: Allerede har som fælles mål at færdiggøre et fælles software produkt, koordinere det i fællesskab, arbejde sammen i fællesskab, være et fælles team og agere som et samlet team (*Mundra; Misra og Dhawale. 2013*), (*E. Hossain, A. B. Muhammad og y. P. Hye. 2009*), (*A. Lamersdorf, M. Jürgen and R. Dieter. 2009*), (*Sutherland and Schwaber. 2007*), (*Maria, Casper og Ville. 2012*) og (*J. Paredes, C. Anslow, and F. Maurer. 2014*).

## 4.5 Aktivitet 4: Demonstration

Ifølge Low tech prototyping metoden i systemudvikling aktivitet 3 kan jeg præsentere og demonstrere artefaktet som: En horizontal prototype af en højteknologisk prototype der repræsenterer en vinkel af et færdigt system, der kunne repræsenterer et helt færdigt funktionelt system til distribueret softwareudviklings Scrum teams også kaldet Scrum of Scrum (Aaen. 2017) og (Rose. 2010).

## 4.6 Aktivitet 5: Evaluation

I aktivitet 5 vil jeg anvende artefaktet værktøjet, et semistruktureret interview og tilhørende teori fra (Conboy 2009) og (Baskerville, Pries-Heje, Madsen, 2011). Dertil vil jeg sammenligne et semistruktureret interview med den anvendte litteratur og teori, observationerne fra empirien og dataen fra et eller flere semistruktureret interview og løsningen fra aktivitet 2. Derefter i slutningen af aktivitet 5 enten iterere tilbage til aktivitet 3 en eller flere gange indtil aktivitet 5 er fyldestgørende og derefter fortsætte til aktivitet 6.

Da jeg i denne opgave gerne ville bidrage med at strukturere en arbejdsproces i distribueret softwareudviklings Scrum teams og fastholde agilitet i Scrum of Scrum med udgangspunkt i aktiviteterne af artefaktet (Burndown Chart) i Scrum of Scrum med en tilhørende Scrum of Scrum Master rolle og et understøttende Low tech prototype i Scrum of Scrum.

Jeg vil anvende taxonomien fra (Conboy 2009) understøttet af (Persson, Nørbjerg og Nielsen 2016), da de i tråd med (Baskerville, Pries-Heje, Madsen, 2011) opsummere agilitet, udfordringerne ved at adressere agilitet og efterspørgslen af værktøjer til at adressere agilitet i agile software organisationer. I denne opgave ville jeg anvende taxonomien fra (Conboy 2009. s 14), ved at tage udgangspunkt i taxonomiens evalueringskriterier. Hvor Conboy definerer følgende agilitets vurderingskrav i 3 spørgsmål og anvende dem efterfølgende på Scrum of Scrum Master rollen og Low tech prototypen se punkt 1,2 og 3: Dermed simplificere ledelsen, afhjælpe koordineringen og fremdriften i distribueret softwareudviklings Scrum teams også kaldet SOS.

### 4.6.1 Demonstrations evaluering af prototypen

Første inspektionen af Low tech prototypen blev foretaget af innovations eksperten Ivan Aaen og forfatter af (Aaen 2017).

Hvor et semistruktureret interview blev anvendt, optaget som empirisk data og 3 spørgsmål til at understøtte aktivitet 5 og iterere tilbage til aktivitet 3 hør eventuelt bilag lydfilen (Tabel 6).

**Tabel 6: Data fra interviewet og første Iterations hovedpointer**

Data fra interviewet omkring Low tech prototypen	Iterations hovedpointer til at understøtte aktivitet 5 og iterere tilbage til aktivitet 3
<p><b>Spørgsmål A,B,C og Ivan Svar:</b></p>	<p><b>Aktivitet 5 iterations hovedpointer:</b></p>
<p><i>A. Understøtter Low tech prototypen del 1. ud af 6. distribueret softwareudviklings Scrum teams, i deres geografisk distribueret softwareprojekter (Hvis ja nej, hvorfor)? se (Figur 20)</i></p> <p>Ivan: "Det kan jeg ikke svare på...", "Men jeg har nogle spørgsmål til det... Hvad er det for en Low tech prototype du taler om", "Hvad antager du om mig nu, antager du at jeg er en praktiker eller en designer, der skal forholde mig til et klassediagram selve UML", "Jeg havde en fornemmelse af, at dette var en general prøve. Inden at du skulle ud og interviewe nogle ude i virkeligheden", "Du giver 2 forskellige roller... den ene giver man en general prøve og den anden... en ekspertise du skal snakke med", "Hvor kan jeg se hvad prototypen kan gøre for mig, hvis at jeg er en praktiker", "... prototype skulle gerne fortælle en funktionalitet til en praktiker, en kvalitet han kan bruge", "Hvor kan jeg se den henne", "Lav et slideshow og drop de 10 komplekse sider og sæt dig i tilhørernes sted", "...Respondentens sted..."</p>	<p>1. Der skal være en klar definition af praksissen, rollen tiltænkt Low tech prototypen og hvem der skal anvende Low tech prototypen se (Tabel 7) og (2.1).</p>
<p><i>B. Understøtter Low tech prototypen del 1. ud af 6. Scrum frameworkets artefakt Burndown Chart til distribueret softwareudviklings Scrum teams (Hvis ja nej, hvorfor)? (Figur 20)</i></p> <p>"Simplere spørgsmål", "Hvis du ikke kan forstå funktionaliteten, hvad er der så svært at forstå ved funktionaliteten", "Hvis du får denne funktionalitet, er det så en funktionalitet du kan bruge og hvad kan du bruge den til, er der nogle steder, hvor du mener, at du ikke kan anvende den, er der nogle ting under distribueret softwareudvikling der ikke understøttes", "Og prøv at brug de ord, at en praktiker bruger", "... Har du nogle før spørgsmål, hvem er du... din funktion, er du Scrum Master eller Product Owner eller udvikler"</p>	<p>2. Manglende tiltænkt respondent og bruger af Low tech prototypen se (4.4.1).</p> <p>3. Der skal være en klar definition på en Product Backlog struktur, Scrum Master og Product owner se (Figur 7) og (4.4.2).</p>
<p><i>C. Bidrager Low tech prototypen del 1. ud af 6. til distribueret softwareudviklings Scrum teams Scrum board teknologier (Hvis ja nej, hvorfor)? (Figur 20)</i></p> <p>Ivan "...Du har spørgsmål som helhed...", "Så har du et omkring Burndown Chart og noget om Scrum Boards", "Og i og med at Scrum Boards og Burndown Charts, givet vis er nogle af de funktionaliteter du skal understøtte, for at man kan lave projektstyring i distribueret projekter, så er det fint... at du har fokus på de konkrete... detaljer... om du så skal have de overordnet spørgsmål først eller til sidst, er et spørgsmål om smag.. eller kig på hvilken rejse, at du gerne vil have at respondenter kommer igennem, når du stiller spørgsmålene", "Så at du har en slide præsentation af din prototype... som den skal anvendes af en praktiker... Så at han har rejse over hvordan funktionaliteten er opbygget og at han har fået indblik i to konkrete værktøjer", "Det kan være meget godt, at have respondenter igennem, før at du ber om en overordnet mening"</p>	<p>4. Burndown-chart og scrum Board er den rigtige retning se (4.4.1 - 4.4.7).</p>

**Figur 27.1: Multiple dashboards Low tech prototype overview**  
Wizard of Oz Slide 1

SOS Teams (SOS Master Peter)	Samlet Story Point I Timer per SOS Sprint	Scrum Master	Opdateret Team Fremdrift
Team 1 Funktioner	200	Kasper	<input checked="" type="checkbox"/>
Team 2 Model	300	Jesper	<input checked="" type="checkbox"/>
Team 3 Database	600	Jonathan	<input type="checkbox"/>
Team 4 Interface	300	Thorbjørn	<input checked="" type="checkbox"/>
-Delete Team	x	x	<a href="#">Y/N</a>
+Add new team	x	x	<a href="#">Y/N</a>

- Et Multiple dashboards Low tech Prototype som repræsenterer estimeringen fra flere Team Scrum Master i timer i Burndown Chart og en fælles Product Backlog fra flere distribueret softwareudviklings teams også kaldet SOS se (Figur 27.1).

- Prototypen tillader en fælles template som referencepunkt for et SOS team

- Templates og checklister gav tidlig identifikation af potentielle estimerings udfordringer og estimerings konflikter i SOS temaet.

- Tidsbesparende i opsætning og koordinering af aktiviteterne af Product Backloggen og estimeringen.

**Figur 27.2: Multiple dashboards Low tech prototype Burndown Chart**  
Wizard of Oz Slide 2

- Transparens, identifikation og visibilitet lærte de andre Scrum team omkring de andre Scrum Teams Backlog, timer og estimering.

- Opsætning af management værktøjer krævede mindre ressourcer og hjalp Team Scrum Masteren med overblik over koordinering af aktiviteterne af Product Backloggen og estimering af Scrum teamet.

- Opsætning af management værktøj templates, checklister, opdateringer, Backlogs og teams krævede færre ressourcer og var nemmere at vedligeholde.

**Figur 27.3: Multiple dashboards Low tech prototype Video Call Wizard of Oz Slide 3**

- Hvis en Team Scrum master har glemt at opdatere alle 3 opdateringer pr. Scrum of Scrum Sprint til Scrum of Scrum Masteren registreres dette af systemet og en notifikation meddeles til Scrum of Scrum Masteren..

- Et fælles referencepunkt for koordinering af aktiviteterne af Backloggen og timerne for SOS teamet.

- Metode frameworket tillod større flexibilitet til koordinering af aktiviteterne af Product Backloggen og estimeringen i et selvorganiserende Scrum team.

#### 4.6.2 Teoretisk evaluering af prototypen og SOS Master

Ud fra følgende ovenstående empiri fra (Tabel 6) og understøttende teori: Blev der itereret tilbage fra aktivitet 5 til aktivitet 3 flere gange. Indtil aktivitet 5 igen kunne foretages med udgangspunkt i de 4 punkter fra interviewet med Ivan Aaen (Tabel 6) og Taxonomi af Agilitet (Conboy 2009). Hvor jeg kom frem til følgende resultat se (Tabel 7) understøttet af et teoretisk agilitets evaluering framework fra (Persson, Nørbjerg og Nielsen 2016) som tager udgangspunkt i (Conboy 2009) og effekterne af agilitet i systemudvikling.

I min teoretiske evaluering var det vigtigt, at der blev taget højde for at understøtte agilitet og et framework der tidligere havde adresseret agilitet: Derfor validerede selve tilgangen til evalueringen af agilitet i Scrum of Scrum teams og et understøttende værktøj prototypen (Multiple Dashboard) se (Figur 27.1, Figur 27.2 og Figur 27.3).

I (Tabel 7) defineres ISD management af team og teams (SOS Master) for eksempel ved at bidrage til simplicitet af management af SOS, ved at give et template framework som fælles referencepunkt for at afhjælpe koordinering af aktiviteterne af Product Backloggen og estimeringen hvor forandringen bidrager til simplicitet af Scrum teams ved at afhjælpe selvorganiserende Scrum team til at afhjælpe koordinering af aktiviteterne af Product Backloggen og estimeringen i de enkelte teams.

Hvor begge forandringer bidrager til læring fra forandring i Scrum og SOS, ved at understøtte til læring fra forandring i koordinering af aktiviteterne af Backloggen og timerne. Hvor i Scrum teamet understøtter læring fra forandring selvorganisering i Scrum teamet og i SOS som et fælles referencepunkt for SOS teamet bestående af flere Scrum teams.

Hvor denne forandring integreret i et management værktøj (Tabel 7: Integrere ISD management med et værktøj (Prototypen)) i mit tilfælde prototypen, som tillod Scrum teams og SOS at være proaktive på forandringer. Hvor ved at have et fælles overblik over udviklingsprocessen, opgaverne og anvendte ressourcer hjalp med at håndtere forandringer på forhånd i SOS teamet. Hvor i Scrum teamet hjalp et Fælles overblik referencepunkt over udviklingsprocessen, opgaverne og anvendt ressourcer med at håndtere forandringer i Scrum teamet på forhånd. Men også understøttede læring fra forandring i Scrum og SOS hvor integration af et værktøj (Prototypen) gav Scrum team transparens, identifikation og visibilitet af Backloggen, timerne af andre Scrum Teams.

Hvor i SOS gav integration af et værktøj (Prototypen) templates og checklister gav tidlig identifikation af potentielle estimerings udfordringer og estimerings konflikter i SOS temaet. Hvor disse forandringer er skabt af ændringerne fra værktøjet (Prototypen Multiple dashboard), men har sine rødder i: et fysisk Scrum board med et Burndown-chart ikke kan flyttes til hvert enkelt distribueret lokation eller projekt og hvordan et burndown-chart anvendes, i et Scrum board, i distribueret softwareudviklings Scrum teams.

Begge forandringer har referencer til Scrum agilitet og SOS agilitet i forandrings sammenhæng mellem forandring og agilitet. Hvor sammenhængen er bindeleddet og hvordan den forandring bidrager til agilitet på tværs af Scrum og SOS. For eksempel i (Tabel 7:(2) Integrere ISD management med et værktøj (Prototypen)) bidrager simplicitet ved at opsætning var tidsbesparende og derfor simplificerede koordinering af aktiviteterne af Product Backloggen og estimeringen for Scrum teamet, hvor i SOS bidrager simplicitet til et framework referencepunkt var tidsbesparende og derfor simplificerede koordinering af aktiviteterne af Product Backloggen og estimeringen for SOS teamet.

Selvom bidraget af simplicitet i (Tabel 7:(2) Integrere ISD management med et værktøj (Prototypen)) ligner hinanden og har en sammenhæng ved at bidrage til simplicitet. Skal simpliciteten stadig adskilles i bidraget til simplicitet på SOS team niveau ved at et framework referencepunkt var tidsbesparende, hvor dette adskilles i Scrum teamet ved at opsætning var tidsbesparende.

Dermed selvom at der er lighed mellem to bidrag og de afhænger af hinanden. Er det ikke nødvendigvis ensbetydende med at bidraget til agilitet, er den samme på SOS som ved Scrum og derfor er det vigtigt at forstå adskillelsen af bidraget til agilitet og forventningen om forbedringen af den ønsket agilitet.

Da den ønsket agilitet på SOS team niveau ikke nødvendigvis forbedre agiliteten på Scrum team niveau og kan underminere agilitet på Scrum team niveau og omvendt, hvorvidt begge former for agilitet understøtter: Simplicitet, læring fra forandring og proaktivitet før forandring se (Tabel 7), (*Conboy 2009*) og (*Persson, Nørbjerg og Nielsen 2016*).

**Tabel 7: Evaluering af forandrings bidrag til ISD Agilitet**

		(1) Definere ISD management af team og teams (SOS Master)		(2) Integrere ISD management med et værktøj (Prototypen)	
Taxonomi af Agilitet (Conboy 2009)		Scrum	Scrum of Scrum (SOS)	Scrum	Scrum of Scrum (SOS)
1. For at være agil, skal et metodeelement bidrage til en eller flere af de følgende:	a. Skabe forandring	b. Tillader teamet at være selvorganiserende på team niveau.  d. Afhjælper selvorganisering i Scrum teamet med koordinering af aktiviteterne af Backloggen og timerne.	b. Tillader en fælles template som referencepunkt for et SOS team  d. Et fælles referencepunkt for koordinering af aktiviteterne af Backloggen og timerne for SOS teamet.	b. Et Fælles overblik referencepunkt over udviklingsprocessen, opgaverne og anvendt ressourcer hjælp med at håndtere forandringer i Scrum teamet på forhånd.  d. Transparens, identifikation og visibilitet lærte de andre Scrum team omkring de andre Scrum Teams Backlog, timer og estimering.	b. Et fælles overblik over udviklingsprocessen, opgaverne og anvendte ressourcer hjælp med at håndtere forandringer på forhånd i SOS teamet.  d. Templates og checklister gav tidlig identifikation af potentielle estimerings udfordringer og estimerings konflikter i SOS temaet.
	b. Proaktivitet før forandring				
	c. Reaktion til forandring				
	d. Læring fra forandring				
2. For at være agil, skal et metodeelement bidrage til en eller flere af de følgende og må ikke fratække fra nogen:	a. Økonomi	c. At afhjælpe selvorganiserende Scrum team til at afhjælpe koordinering af aktiviteterne af Product Backloggen og estimeringen i det enkelte teams.	b. Transparens af Sprint opdateringen introducerede Team Scrum Master'ne i at motivere hinanden i at hjælpe andre Teams i SOS teamet.  c. Et framework template tillod Scrum teams at have et fælles referencepunkt for at afhjælpe koordinering.	c. Opsætning var tidsbesparende og derfor simplificerede koordinering af aktiviteterne af Product Backloggen og estimeringen for Scrum teamet.	c. Et framework referencepunkt var tidsbesparende og derfor simplificerede koordinering af aktiviteterne af Product Backloggen og estimeringen for SOS teamet.
	b. Kvalitet				
	c. Smplicitet				
3. For at være agil, skal et metodeelement løbende være klar, dvs. minimal tid og		Metode frameworket tillod større flexibilitet til koordinering af aktiviteterne af Product Backloggen og estimeringen i et selvorganiserende Scrum team.	Metode frameworket var lettere at anvende end en ad hoc metode sammensat af flere selvorganiserende Scrum teams til koordinering af aktiviteterne af Product Backloggen og estimeringen.	Opsætning af management værktøjer krævede mindre ressourcer og hjælp Team Scrum Masteren med overblik over koordinering af aktiviteterne af Product Backloggen og estimering af Scrum teamet.	Opsætning af management værktøj templates, checklister, opdateringer, Backlogs og teams krævede færre ressourcer og var nemmere at vedligeholde.

## 4.7 Aktivitet 6: communication

6. Communication: Her vil jeg kommunikere problemformuleringen og dens relevans, artefaktet, anvendelsen og bidraget. Formidlingen vil foregå igennem en rapport afhandling, bestående af en normativ research proces efterfulgt af DSRM modellen: En problemformulering, et litteraturreview, metodeanvendelse, empirisk dataindsamling og analyse af dataen, repræsentation af resultater, en diskussion og konklusion.

Processen vil blive beskrevet i en normativ sekventiel orden: Hvor i processen vil foregå i iterative stadier alt efter behov, hvor i sammenhængen mellem problem og løsning vil fremtræde, hvori de sekventielle processer og iterationerne er svaret på kommunikationen af artefaktet og løsningen indlejret i problemet af artefaktet. Dette vil tage udgangspunkt i en problemorienteret tilgang, hvor aktiviteten er startet med et forskningsproblem i form af en problemformulering.

## 5 AKTIVITET 7: CONTRIBUTION & DISKUSSION

Frameworket adressere ISD agilitet i distribueret softwareudviklings teams også kaldet Scrum of Scrum teams: Hvor frameworket adressere agilitet, hvordan agiliteten adskilles fra Scrum team til distribuerede softwareudviklings Scrum teams og hvordan agiliteten individuelt bidrager til Scrum og distribueret softwareudviklings Scrum teams. I mit tilfælde ville jeg bidrage med at strukturere en arbejdsproces i distribueret softwareudviklings Scrum teams og fastholde agilitet. Hvor jeg anvendte taxonomien fra (*Conboy 2009*) og (*Persson, Nørbjerg og Nielsen 2016*), hvor jeg definere ISD management for Scrum og SOS. Hvor agilitets bidraget blev opdelt og defineret i gennem nogle ISD typer og guidelines for at håndtere disse agilitets typer se (Tabel 7 og 4.6.2).

Som resultat af dette kunne jeg adskille nogle agilitets bidrag, der understøtter en sammenhængen mellem overgangen fra Scrum teams til distribueret softwareudviklings teams og tilhørende agilitets udfordring for distribueret softwareudviklings teams med koordinering af aktiviteterne af Product Backlog se (Tabel 7: d. Læring fra forandring, b. Proaktivitet før forandring og c. SimPLICITET).

I min teoretisk evaluering af prototypen og SOS Master bidrager evalueringen til konfigurerings ISD metoder og redskaber til at forstå agilitet eller adressere agilitet. Hvor en af de mest opsigtsvækkende resultater: Er hvorvidt man forstå agilitet i distribueret softwareudviklings Scrum teams er det ikke ensbetydende med at man adressere agiliteten i Scrum teams eller Scrum of Scrum teams. Som relatere til diskussionen af post agilitet for ISD (*Baskerville, Pries-Heje, Madsen, 2011*) hvordan agilitet kan erhvervs af software organisationer: Hvor frameworket fremhæver viden omkring praksis af et dualitets mål af et behov for agilitet på den ene side og den anden et behov for struktur (Alignment: for planlægning, for mennesker og opgaver) og erhvervelsen i sammenhængen af forholdet mellem softwareudviklingen.

I min teoretiske evaluering mellem distribueret softwareudviklings Scrum teams også kaldet Scrum of Scrum teams og Scrum teams: Demonstreres der tilhørende agilitets udfordring for begge teams med koordinering af aktiviteterne af Product Backlog se (Tabel 7: d. Læring fra forandring, b. Proaktivitet før forandring og c. SimPLICITET). Hvor disse udfordringer yderligere kan adskilles i agilitet for SOS teamet eller Scrum teamet. Hvor ved i Scrum teamet (Tabel 7: d. Læring fra forandring, b. Proaktivitet før forandring og c. SimPLICITET) er fokus rettet på at erhverve teamet struktur (Alignment: for planlægning, for mennesker og opgaver) og agiliteten i Scrum teamet. Hvor i SOS teamet (Tabel 7: d. Læring

fra forandring, b. Proaktivitet før forandring og c. Simplicitet) er fokus rettet på at erhverve agilitet i det overordnede SOS team og dermed kunne håndtere forandringer igennem opsætning af fælles referencepunkter, templates, checklister som ikke nødvendigvis understøtter et Scrum team i at erhverve agilitet eller struktur (Alignment: for planlægning, for mennesker og opgaver).

Dermed er den teoretiske evaluering i sammenhæng med DSRM af ISD agilitet (Conboy 2009) en måde at adressere management af agilitet på, især når der skal integreres et nyt værktøj eller at forstå at strukturere en arbejdsproces i distribueret softwareudviklings Scrum teams med målet om at fastholde agilitet eller en forbedring af en ønsket agilitet. For ikke at underminere agilitet på Scrum team niveau eller SOS team niveau, hvorvidt om den ønsket agilitet er en samlet løsning for forbedringen af agiliteten på SOS team niveau.

Dermed stadfæster DSRM i sammenhæng med ISD agilitet (Conboy 2009) understøttet af (Persson, Nørbjerg og Nielsen 2016) en måde at forske management ISD agilitet og stadig understøtte implementering af agilitet i en software organisation og stadig adskille agilitet på Scrum team niveau og SOS team niveau. Hvorvidt om bidraget tager udgangspunktet i at implementere nye praksisser for ISD eller systemer der danner grundlag for nye ISD praksisser og bidraget af den tilhørende agilitet til denne implementering.

Der hvor dette paper også gør sig enestående er sammenhængen mellem praksis der ikke er kommercielt etableret eller offentligt accepteret som agil "SOS": selvom praksissen udspringer af en agil tradition, Scrum eller flere Scrum teams. Hvor dette kan skyldes mangel på hvad Conboy beskriver som mangel på afklaring, mangel på teori, mangel på en akkumuleret tradition, mangel på sparsommelighed eller mangel på anvendelighed "Lack of clarity..." (Conboy 2009. s 16-17).

Hvor det kan diskuteres om den manglende afklaring etc. eller offentligt acceptering af distribueret softwareudviklings Scrum teams som SOS teams, skyldes manglende behov på struktur (Alignment: for planlægning, for mennesker og opgaver) eller manglende behov for evidence af agilitets praksisser for praktiker eller forskere: relatere dette til diskussionen af post agilitet for ISD (Baskerville, Pries-Heje, Madsen, 2011).

Hvor dette paper også lægger op til adresseringen af kompleksiteten af traditionen af agilitet og et teoretisk grundlag i at adressere eksisterende ISD agilitet Scrum i SOS med (DSRM) for at forstå agiliteten af distribueret softwareudviklings Scrum teams: Der kan bidrage til forskers og praktikers arbejde i at forstå agilitet og bidrage til agilitet og bidrage til agilitet som helhed: Hvorvidt om det er på Scrum team niveau eller distribueret softwareudviklings team niveau i distribuerede softwareudviklingsprojekter og adressere post agilitet (Baskerville, Pries-Heje, Madsen, 2011): 1. Hvorvidt man anvender agile metoder, 2. sammenligner agile metoder, 3. anvender in-house metode varianter, 4. adressere underminerende agilitets praksisser, 5. eller facilitere en overgang af agilitet (Conboy 2009).

Naturligvis kan dette paper ikke stadfæstes som en endegyldig konklusion på hvordan man adressere agilitet, men derimod skabe eller inspirere et grundlag for et framework for at adressere eksisterende agilitet i distribuerede softwareudviklingsprojekter. Eller som framework til at adressere agiliteten i in-house metode variationer der skal bidrage til agilitet eller praksisser der skal fremme agilitet i distribuerede softwareudviklingsprojekter se (Tabel 7: d. Læring fra forandring, b. Proaktivitet før forandring og c. Simplicitet), (Tabel 7), (Baskerville, Pries-Heje, Madsen, 2011), (Conboy 2009) og (Persson, Nørbjerg og Nielsen 2016).

## 6 KONKLUSION

Burndown-charts kan anvendes til at understøtte SOS teams praksisser se (Tabel 7) og (Mundra; Misra og Dhawale. 2013). Grunden kan være den manglende etablering af distribueret softwareudviklings Scrum teams som SOS teams eller agilitets bidraget af SOS (*Baskerville, Pries-Heje, Madsen, 2011*), mangel på afklaring af agilitets praksisser som bidrager til agilitet i SOS (*Conboy 2009*) og (*Persson, Nørbjerg og Nielsen 2016*).

Uanset hvad kan vi ikke komme udenom som forskers eller praktikker, at vi har brug for redskaber til at adressere traditionen af agilitet i distribueret softwareudvikling: Hvad vi forstå ved agilitet, hvad vi kalder agilitet og hvordan vi drager nytte af den agilitet og overgangen mellem Scrum til SOS. Hvordan et agilitets artefakt (Burndown Chart) som værktøj ikke nødvendigvis er agilt i distribuerede softwareudviklingsprojekter og derfor gentænke hvad vi forstå ved agilitet og bidraget af agilitet i overgangen fra distribueret softwareudviklings Scrum teams til SOS teams.

Hvorvidt om SOS og Burndown Chart går imod punkt 1,2,3 og 4 i det Agile Manifesto kan diskuteres da frameworket i (*Baskerville, Pries-Heje, Madsen, 2011*) fremhæver et dualitets mål af et behov for agilitet på den ene side og den anden side et behov for struktur (Alignment: for planlægning, for mennesker og opgaver). Kan det diskuteres igennem (*Conboy 2009*) hvorvidt om Burndown Chart er en passiv agilitet, men i sammenhæng med SOS praksisser bliver et led der understøtter agilitets praksisser for SOS se (Tabel 7: b. Proaktivitet før forandring og c. Simplicitet), ved at være et referencepunkt for agilitets praksisser og derfor understøtter agilitet i distribueret softwareudviklings Scrum teams.

Der hvor udfordringen med Burndown Chart bliver evident, er hvorvidt om den tiltaler det agile Manifesto og den første del af hver sætning i det agile manifesto: Individuer og interaktioner frem for processer og redskaber, fungerende software over dokumentation, kundesamarbejde over kontrakter og forhandlinger, og agere på forandring - frem for at følge en plan. Men hvad der bliver tydeligt igennem den teoretiske evaluering: Er dualitets målet af et behov for agilitet på den ene side og et behov for struktur (Alignment: for planlægning, for mennesker og opgaver) på den anden side.

Men hvorvidt om man skal udfylde den ene eller anden ende af sætningen af det agile Manifesto i distribueret softwareudviklings Scrum projekter, for at bidraget af et Burndown Chart artefakt til SOS: Kan vi ikke komme udenom at et fysisk Scrum board med et Burndown-chart ikke kan flyttes og behovet for koordinering af aktiviteterne af Backloggen, timerne eller struktur (Alignment: for planlægning, for mennesker og opgaver) er en udfordring for distribueret softwareudvikling Scrum teams i diskussionen relateret til post agilitet (*Baskerville, Pries-Heje, Madsen, 2011*).

Men at udfordringen ikke nødvendigvis kan forstås i den ønsket agilitet på team niveau eller SOS team niveau, men i en sammenhæng af at adressere bidraget af agiliteten se (Tabel 7) og (*Conboy 2009*. s. 22) i praksissen af et Burndown Chart eller Low tech prototypen (*Conboy 2009*) for Scrum of Scrum og der igennem kan agiliteten adresseres eller forstås (*Persson, Nørbjerg og Nielsen 2016*), (*Baskerville, Pries-Heje, Madsen, 2011*) og den ønsket agilitet adresseres på team niveau fastholdes og samtidigt udvikles til et distribueret softwareudviklings Scrum team eller SOS team niveau uden at underminere agiliteten (*Conboy 2009*), (*Persson, Nørbjerg og Nielsen 2016*) og (*Baskerville, Pries-Heje, Madsen, 2011*). Dermed er frameworket af hvordan vi forstår agilitet en vigtig del af at adressere agilitet eller at forbedre agilitet i ISD (*Conboy 2009*) og distribuerede softwareprojekter.

## 7 LITTERATURLISTE

A. Lamersdorf, M. Jürgen and R. Dieter, "A decision model for supporting task allocation processes in global software development," in *Product-Focused Software Process Improvement*, 2009. Available 22-02-2017: [https://link.springer.com/chapter/10.1007/978-3-642-02152-7\\_25](https://link.springer.com/chapter/10.1007/978-3-642-02152-7_25)

Aaen, I. (2017) *Essence – Pragmatic Software Innovation* (pdf draft version)

Atlassian 2017. Available 22-02-2017: <https://www.atlassian.com/software>

Austin D. Robert and Devin Lee. *Research Commentary: Weighing the Benefits and Costs of Flexibility in Making Software: Toward a Contingency Theory of the Determinants of Development Process Design*. *Information Systems Research* Vol. 20, No. 3, *Flexible and Distributed Information Systems Development* (September 2009), pp. 462-477 Published by: INFORMS Stable URL: <http://www.jstor.org/stable/23015475> Page Count: 16: Available 22-02-2017: [http://www.jstor.org/stable/23015475?seq=1#page\\_scan\\_tab\\_contents](http://www.jstor.org/stable/23015475?seq=1#page_scan_tab_contents)

Benyon David (2014) *Designing Interactive Systems - A comprehensive guide to HCI, UX and interaction design*. Edition 3rd ISBN 9781447920113. ISBN 10 1447920112 Published 13/09/2013. Published by Pearson United Kingdom Pages 640 Available 13-09-2016: <http://www.pearson.com.au/products/A-C-Benyon-David/Designing-Interactive-Systems-A-Comprehensive-Guide-to-HCI-UX-and-Interaction-Design/9781447920113?R=9781447920113>

Barrett Michael, Oborn Eivor. *Boundary object use in cross-cultural software development teams*. *Human Relations*, vol. 63, 8: pp. 1199-1221. First Published May 19, 2010. Sage Journals: sagepub. Available 22-02-2017: <http://journals.sagepub.com/zorac.aub.aau.dk/doi/pdf/10.1177/0018726709355657>

Baskerville Richard, Pries-Heje Jan, Madsen Sabine. *Post-agility: What follows a decade of agility?* *Original Research Article Information and Software Technology*, Volume 53, Issue 5, May 2011, Pages 543-555. Available 22-02-2017: <http://www.sciencedirect.com/science/article/pii/S0950584910001953>

Beck, K., M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, and R. Jeffries. 2001, "Manifesto for Agile Software Development," <http://Agilemanifesto.Org/>

Bhatt Mehul, Schultz Carl og Huang Minqian (2012). *The shape of empty space: Human-centred cognitive foundations in computing for spatial design*. 2012 *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* Year: 2012 Pages: 33 - 40, DOI: 10.1109/VLHCC.2012.6344477 *IEEE Conference Publications*. Available 22-02-2017: [http://www.jstor.org/stable/23015468?seq=1#page\\_scan\\_tab\\_contents](http://www.jstor.org/stable/23015468?seq=1#page_scan_tab_contents)

Conboy Kieran. *Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development*. National University of Ireland, Galway, Ireland Published in: *The ACM Digital Library · Journal Information Systems Research archive* Volume 20 Issue 3, September 2009 Pages 329-354 *INFORMS Institute for Operations Research and the Management Sciences (INFORMS)*, Linthicum, Maryland, USA: 12: Available 22-02-2017: [https://aran.library.nuigalway.ie/xmlui/bitstream/handle/10379/1563/Conboy\\_K\\_ISR\\_2009.pdf?sequence=1](https://aran.library.nuigalway.ie/xmlui/bitstream/handle/10379/1563/Conboy_K_ISR_2009.pdf?sequence=1)

E. Hossain, A. B. Muhammad and y. P. Hye, "Using scrum in global software development: a systematic literature review," in *Fourth IEEE International Conference on Global Software Engineering*, 2009. Available 22-02-2017: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5196931>

Esbensen Morten, Tell Paolo, Cholewa Jacob B., K. Pedersen Mathias, Bardram Jakob. *The dBoard: A Digital Scrum Board for Distributed Software Development*, November 2015 *ITS '15: Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces* Publisher: ACM. Available 22-02-2017: <http://dl.acm.org/zorac.aub.aau.dk/citation.cfm?id=2817746&CFID=905893449&CFTOKEN=48006483>

livari Juhani, livari Netta (2011). *The relationship between organizational culture and the deployment of agile methods*. *Information and Software Technology Volume 53, Issue 5, May 2011, Pages 509–520 Special Section on Best Papers from XP2010*. Available 13-09-2016, URL:

<http://www.sciencedirect.com/science/article/pii/S095058491000193X>

J. Paredes, C. Anslow, and F. Maurer. 2014. *Information Visualization for Agile Software Development*. In *Software Visualization (VISSOFT), 2014 Second IEEE Working Conference on*. 157–166. Available 22-02-2017:

[http://vissoft.iro.umontreal.ca/docs/JuliaParedes\\_Paper8\\_VISSOFT.pdf](http://vissoft.iro.umontreal.ca/docs/JuliaParedes_Paper8_VISSOFT.pdf)

J. R. Evaristo, R. Scudder, K. C. Desouza, and O. Sato, "A dimensional analysis of geographically distributed project teams: A case study," *J. Eng. Technol. Manage.*, vol. 21, no. 3, pp. 175–189, 2004. Available 22-02-2017:

<http://www.sciencedirect.com/science/article/pii/S0923474804000293>

Kjeldskov J., Skov M. B. and Stage J. (2004) *Instant Data Analysis: Evaluating Usability in a Day*. *Proceedings of NordiCHI 2004*. Tampere, Finland. ACM, pp. 233-240.

Fitzgerald, Brian. Hartnett, Gerard and Conboy, Kieran(2006). *Customising agile methods to software practices at Intel Shannon*. *Journal European Journal of Information Systems - Including a special section on business agility and diffusion of information technology archive Volume 15 Issue 2, April 2006 Pages 200 - 213* . Available 13-09-2016:

<http://dl.acm.org/citation.cfm?id=1148997>

Fowler M and Highsmith J (2001) *The Agile Manifesto*. *Software Development, August. 2015 Agile Alliance*. Available 22-02-2017:

[http://andrey.hristov.com/fht-stuttgart/The\\_Agile\\_Manifesto\\_SDMagazine.pdf](http://andrey.hristov.com/fht-stuttgart/The_Agile_Manifesto_SDMagazine.pdf)

Ken Schwaber (2004) *Agile Project management with Scrum*. Published by Microsoft Press Redmond Washington 98052-6399

Kniberg, Henrik. 2007. *Scrum and XP from the Trenches*. C4Media, Publisher of InfoQ.com. Available 13-09-2016: <http://tschering.mono.net/upl/10004/110224ScrumAndXpFromTheTrenches.pdf>

Kniberg, Henrik. *Scrum and Xp from the Trenches 2nd Edition Paperback – 24 Nov 2015*. Format: Paperback ISBN-13: 9781329224278 Language: English C4Media, Publisher of InfoQ.com. Available 13-09-2016:

<https://www.amazon.co.uk/gp/search?index=books&linkCode=gs&keywords=9781329224278>

Petersen Frelle Lars (2001) *Teknologirådet: Erfaringer fra statslige IT-projekter? - Rapport og anbefalinger fra en arbejdsgruppe under teknologirådet Erik Bonnerup (formand) Annelone Jensen Birgitte Gregersen Erik Andreasen Hans Henrik Østergaard Inge Mærkedal Karsten Dybvad Kim Viborg Andersen Kim Østrup Martin Toft Hansen Nina Esmark*. *teknologirådets rapporter 2001/3 isbn 87-90221-56-7 issn 1395-7392 rapporten bestilles hos: Teknologirådet Antonigade 4 1106 København K Telefon 33 32 05 03 Telefax 33 91 05 03 [tekno@tekno.dk](mailto:tekno@tekno.dk)* : Available 22-02-2017: [http://www.tekno.dk/pdf/projekter/p01\\_Rapport\\_it\\_proj.pdf](http://www.tekno.dk/pdf/projekter/p01_Rapport_it_proj.pdf)

Maria Paasivaara; Casper Lassenius; Ville T. Heikkilä . *Inter-team coordination in large-scale globally distributed scrum: Do Scrum-of-Scrum really work?* *Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement Year: 2012 Pages: 235 - 238, DOI:*

10.1145/2372251.2372294 Cited by: *Papers (3) IEEE Conference Publications*. Available 22-02-2017:

<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6475422>

Mundra, Ashish; Misra, Sanjay and Dhawale A. Chitra. *Practical Scrum-Scrum Team: Way to Produce Successful and Quality Software*. 2013 *13th International Conference on Computational Science and Its Applications Year: 2013 Pages: 119 - 123, DOI: 10.1109/ICCSA.2013.25* Cited by: *Papers (3) IEEE Conference Publications*. Available 22-02-2017: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6681108>

Mathiassen L., Munk-Madsen A., Nielsen P.A., and Stage J., (2000), *Object-oriented analysis & design*, 1st. ed., Publisher: Marko. Aalborg Denmark.

Nykredit (2017) *Kalvebod Brygge 1-3, 1780 København V*. Available 22-02-2017: <https://www.nykredit.dk/>

O. Salo; P. Abrahamsson, *Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum*, IET Software Year: 2008, Volume: 2, Issue: 1 Pages: 58 - 64, DOI: 0.1049/iet-sen:20070038 Cited by: Papers (13) IET Journals & Magazines. Available 22-02-2017:

<http://agilesouthflorida.pbworks.com/f/Agile%2BMethods%2Bin%2BEuropean%2BEmbedded%2BSoftware%2BDevelopment.pdf>

Osterwalder, A., Pigneur, Y., Clark, T. (2010) *Business Model Generation: A handbook for visionaries, game changers, and challengers*. Wiley, Hoboken, NJ. ISBN 9780470876411 (Danish version available from Gyldendal ISBN 9788702145502). Available 22-02-2017: <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470876417.html>

Peffer Ken, Tuunanen Tuure, Rothenberger A Marcus. & Samir Chatterjee (2007) *A Design Science Research Methodology for Information Systems Research*, *Journal of Management Information Systems*, 24:3, 45-77. Available 22-02-2017: <http://dx.doi.org/10.2753/MIS0742-1222240302>

Petersen, Kai (2008) *Systematic mapping studies in software engineering*. *Proceeding EASE'08 Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering Pages 68-77 Italy — June 26 - 27, 2008 BCS Learning & Development Ltd. Swindon, UK*. Available 13-09-2016, URL: <http://dl.acm.org/citation.cfm?id=2227123>

Persson, J. S., Mathiassen, L., Boeg, J., Madsen, T. S., & Steinson, F. (2009). *Managing Risks in Distributed Software Projects: An Integrative Framework*. *IEEE Transactions on Engineering Management*, 56(3), 508-532. DOI: Located 22-02-2017:

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4815496&isnumber=5166534>

Persson, John S.; Nørbjerg, Jacob; and Nielsen, Peter Axel, "IMPROVING ISD AGILITY IN FAST-MOVING SOFTWARE ORGANIZATIONS" (2016). *Research Papers*. 96.

[http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1098&context=ecis2016\\_rp](http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1098&context=ecis2016_rp)

Powell, A.; Piccoli, G. and Ives, B. "Virtual teams: A review of current literature and directions for future research", *DATA BASE for Advances in Information Systems* (35:1) 2004 pp. 6-36..Available 13-09-2016, URL:

<https://pdfs.semanticscholar.org/1474/dd96d925e3db455d6c3084aab7676ab4a196.pdf>

Templier, Mathieu and Paré, Guy (2015) "A Framework for Guiding and Evaluating Literature Reviews," *Communications of the Association for Information Systems: Vol. 37, Article 6*. Available 13-09-2016, URL: [https://www.researchgate.net/profile/Guy\\_Pare/publication/281465222\\_A\\_Framework\\_for\\_Guiding\\_and\\_Evaluating\\_Literature\\_Reviews/links/55ea03ba08ae21d099c451b2.pdf](https://www.researchgate.net/profile/Guy_Pare/publication/281465222_A_Framework_for_Guiding_and_Evaluating_Literature_Reviews/links/55ea03ba08ae21d099c451b2.pdf)

R. de Melo Oliveira, A. Goldman, and C. Melo. *Designing and managing Agile informative workspaces: Discovering and exploring patterns*. In *Proc. of HICSS*, pages 4790–4799. IEEE, 2013. Available 22-02-2017:

<http://ieeexplore.ieee.org/abstract/document/6480421/>

Rittitum Patsakorn; Vatanawood Wiwat; Thongtak Arthit. *Digital Scrum board using leap motion 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS) Year: 2016 Pages: 1 - 4, DOI: 10.1109/ICIS.2016.7550836 IEEE Conference Publications*. Available 22-02-2017:

<http://ieeexplore.ieee.org.zorac.aub.aau.dk/stamp/stamp.jsp?arnumber=7550836>

Rose, J. (2010) *Software Innovation: eight work-style heuristics for creative software developers*, Aalborg, Software Innovation, Dept. of Computer Science, Aalborg University: paperback, ebook, pdf. Available 22-02-2017: [http://vbn.aau.dk/files/20010497/SI\\_book\\_beta\\_bw.pdf](http://vbn.aau.dk/files/20010497/SI_book_beta_bw.pdf)

Rubart Jessica. 2014. *A Cooperative Multitouch Scrum Task Board for Synchronous Face-to-Face Collaboration*. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces (ITS '14)*. ACM, New York, NY, USA, 387–392. Available 22-02-2017: <http://dl.acm.org/citation.cfm?id=2669551>

Rubin, J. and Chisnell, D. (2008) *Handbook of Usability Testing - How to Plan, Design, and Conduct Effective Tests*. Wiley Publishing, Inc. Chapters 3, 5 and 12

Sutherland J.; Schoonheim, G.; Rijk, M. *Fully Distributed Scrum: Replicating Local Productivity and Quality with Offshore Teams* 2009 42nd Hawaii International Conference on System Sciences Year: 2009 Pages: 1 - 8, DOI: 10.1109/HICSS.2009.225 Cited by: Papers (7) IEEE Conference Publications. Available 22-02-2017: <http://ieeexplore.ieee.org.zorac.aub.aau.dk/stamp/stamp.jsp?arnumber=4755769>

Sutherland, J., and Schwaber, K.: "The Scrum Papers: Nuts, Bolts, and Origins of an Agile Method", Scrum, Inc., Boston, 2007. Available 22-02-2017: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.108.814>

Sutherland, Jeff; Schoonheim, Guido; Rustenburg, Eelco; Rijk, Maurits. *Fully Distributed Scrum: The Secret Sauce for Hyperproductive Offshored Development Teams*, Agile 2008 Conference Year: 2008 Pages: 339 - 344, DOI: 10.1109/Agile.2008.92 Cited by: Papers (22) IEEE Conference Publications. Available 22-02-2017: <http://ieeexplore.ieee.org.zorac.aub.aau.dk/stamp/stamp.jsp?arnumber=4599502>

Solingen van Rini, Sutherland Jeff, Denny de Waard. *Scrum in Sales: How to Improve Account Management and Sales Processes*. 2011 Agile Conference Year: 2011 Pages: 284 - 288, DOI: 10.1109/AGILE.2011.12 Cited by: Papers (1) IEEE Conference Publications. Available 28-02-2017: <http://ieeexplore.ieee.org.zorac.aub.aau.dk/stamp/stamp.jsp?arnumber=6005833>

Star Leigh Susan Science. *This is Not a Boundary Object: Reflections on the Origin of a Concept*. Technology, & Human Values, vol. 35, 5: pp. 601-617. , First Published August 10, 2010. Sage Journals: sagepub. Available 22-02-2017: <http://journals.sagepub.com.zorac.aub.aau.dk/doi/pdf/10.1177/0162243910377624>

Schwaber Ken and Beedle Mike. (2002). *Agile Software Development with Scrum*. Paperback, 158 pages, published October 21st 2002 by Prentice Hall. ISBN 0130676349 (ISBN13: 9780130676344). Available 22-02-2017: <https://www.amazon.com/Agile-Software-Development-Scrum/dp/0130676349>

ScrumDesk 2017. Available 22-02-2017: <https://www.scrumdesk.com/start/manual-for-scrumdesk-start/start-about-scrumdesk-start/>

Ågerfalk J. Pär, Fitzgerald, Brian and Sandra A. *Introduction to the Special Issue: Flexible and Distributed Information Systems Development: State of the Art and Research Challenges*. Slaughter Information Systems Research Vol. 20, No. 3, Flexible and Distributed Information Systems Development (September 2009), pp. 317-328 Published by: INFORMS Stable URL: <http://www.jstor.org/stable/23015468> Page Count: 12: Available 22-02-2017: [http://www.jstor.org/stable/23015468?seq=1#page\\_scan\\_tab\\_contents](http://www.jstor.org/stable/23015468?seq=1#page_scan_tab_contents)

Wang, Xiaofeng and Conboy, Kieran (2009). *Understanding agility in software development from a complex adaptive systems perspective*. 17th European Conference on Information Systems. Available 13-09-2016, URL: [https://ulir.ul.ie/bitstream/handle/10344/2105/2009\\_Wang,X.pdf?sequence=2](https://ulir.ul.ie/bitstream/handle/10344/2105/2009_Wang,X.pdf?sequence=2)