

Scrum

Af Poul Staal Vinje
Certified Scrum Master
Certified Scrum Practitioner

poul.vinje@gmail.com
www.agile-metoder.dk

Scrum	1
1. Konceptet	2
2. Navnet	3
3. ProductOwner.....	3
4. Produktets backlog	4
5. SprintPlanningWorkshop	5
6. Sprinten	6
7. SprintBacklog.....	6
8. DailyScrum	7
9. BurnDownChart	8
10. ScrumMaster	8
11. Teamet	9
12. SprintReviewMeeting.....	10
13. Retrospective.....	10
14. Skalering af Scrum	10
15. Opsummering: Fordele ved at bruge Scrum	11
16. Opsummering: Ulemper ved at bruge Scrum.....	11
17. Scrum er til nødlidende projekter	11

1. Konceptet

Scrum,

- bruges til at styre og overvåge agile udviklingsforløb
- er en iterativ proces der leverer et potentielt brugbart produkt i hver iteration
- bruger forretningsværdi som mål for produktivitet
- fungerer ved hjælp af Sprints i faste intervaller
- fungerer ved hjælp af selvorganiserende teams
- bruges til at afgøre konfliktende interesser
- bruges til at afgøre konfliktende behov
- giver maksimal kommunikation i teamet
- giver maksimal kommunikation mellem team og opdragsgiver
- fjerner forhindringer systematisk
- giver førstehåndsindtryk af status
- skalerer ved at bruge parallelle og overlappende Sprints og teams
- giver arbejdsglæde
- giver en oplevet følelse af at have ydet sit bedste

Scrum bruger tre roller, tre ceremonier og tre værktøjer. Desuden bruges en eller flere Retrospectives til evalueringer.

Rollerne er ProductOwner, Team og ScrumMaster. Ceremonierne er SprintPlanning-Workshop, Daily Scrum og SprintReviewMeeting, Værktøjerne er ProductBacklog, Sprint-Backlog og BurnDownChart.

De enkelte forløb kaldes 'Sprints'. Det er i denne forbindelse ligegyldigt, om der er tale om ny- eller videreudvikling. Til nyudvikling bruges Sprints til at dele projekter op i mindre dele. Til videreudvikling og vedligeholdelse kan det være omvendt. At et antal mindre opgaver samles til en Sprint.

Scrum bruges kun til at lede og styre forløbet. Scrum indeholder ikke teknikker til de konkrete udviklingsaktiviteter. Scrum suppleres derfor med andre metoder, for eksempel Extreme Programming (XP). Scrum er meget velegnet til at indpakke XP forløb.

Scrum skalerer ved at lade mange Sprints blive afviklet parallelt og overlappende.

Teamet organiserer selv arbejdet, efter at indholdet af den kommende Sprint er aftalt. Man kan kalde det autonomi, selvforvaltning eller selvorganisering.

Teams har deres fulde frihed, men inden for rammerne af organisationens standarder, vedtagne arkitekturer og koncepter.

Scrum bruger ikke projektplaner med ressourcer, aktiviteter og afhængigheder. Der følges op via daglige møder, der fungerer som kommunikationscentral. Ledelsen kan deltage, når den ønsker det, og få et førstehåndsindtryk af, hvordan det går. Teamet kan selvfølgelig anvende projektplaner og projektstyringsværktøjer, hvis de beslutter det.

Hvorfor fungerer Scrum så godt? Fordi Scrum baseres på:

- en systematisk og tydelig prioritering

- en systematisk og detaljeret gennemgang af de prioriterede opgaver
- teams der yder deres absolut bedste
- en regelmæssig og tæt opfølgning
- førstehåndsindtryk hos både ProductOwner og ScrumMaster
- sund fornuft

Forretningsværdien bestemmer prioritering og dermed leveringsrækkefølgen. Hvorfor detaljere, analysere og designe de 90 % af kravene? Hellere levere de vigtigste 10 % så hurtigt det kan lade sig gøre.

2. Navnet

Scrum er et begreb fra rugby, hvor en "Scrum" eller "Scrummage" er, når holdene står tæt sammen i en cirkel eller klump, og forsøger at erobre bolden.

Man skal i øvrigt ikke lægge mere i det, end at det er et godt navn. Der er ingen grund til at studere rugby for at bruge Scrum i sin organisation.

Se eventuelt artiklen i *Harward Business Review*: "The New New Product Development Game", Hirotaka Takeuchi og Ikujiro Nonaka.

Apropos navne veksles der i Danmark frit mellem dansk og engelsk. For eksempel ProductOwner og produktejer. Og ProductBacklog og produktbacklog.

3. ProductOwner

Scrum er udvikling drevet af forretningsværdi. ProductOwner er dermed uundværlig.

ProductOwner er en af de tre roller.

ProductOwner er en produktansvarlig. Enten i udviklings- eller brugerorganisationen. Det er vigtigt at vælge en beslutningsdygtig og handlekraftig person. Det er ProductOwner der styrer produktbackloggen. Og det er ProductOwner, der er den drivende kraft i prioritering af indholdet, og i processen med at vælge indholdet til en Sprint.

En af de mest betydningsfulde faktorer i succesfulde projekter, er nøgtern kravstyring. ProductOwner står for denne kravstyring i Agile projektforløb.

ProductOwner er med til at reducere risikoen for fejlagtige investeringer.

ProductOwner kan behøve hjælp med det rent praktiske i at vedligeholde ProductBacklog. Og til at få skrevet nogle gode Userstories.

Lidt detaljer på ProductOwners aktiviteter:

Indsamler alle ønsker til funktionalitet, således at der ikke er andre kanaler ind til teamet.

Vurderer forretningsværdien af funktionalitet.

Indsamler alle udestående fejl og mangler.

Indsamler ønsker til non-funktionelle egenskaber, jævnfør ISO 9126.

Prioriterer, enten 1,1,1,2,2,2,3,3,3 eller 1,2,3,4,5,6,7,8,9. Træffer selv valget mellem de to former.

Grovestimerer. Henter hjælp til det fra udviklere når det er nødvendigt.

Vælger indhold til næste Sprint.

Forandrer retning ved at ændre prioritering.

Afgør konfliktende interesser.

Afgør konfliktende behov.

Reagerer på synligheden der er til stede i Sprinten under vejs.

Reagerer på synligheden der er til stede ved afslutningen af en Sprint.

Evaluerer resultatet efter en Sprint. Reel vurdering, gerne med tydelig plus/minus.

Vurderer resultatet under vejs. Hvorfor vente med accepttest.

Deltager på SprintPlanningWorkshop.

Deltager på SprintReviewMeeting.

Deltager efter ønske og behov på Daily Scrum. Dog kun ved at lytte.

4. Produktets backlog

Eller på engelsk, ProductBacklog.

ProductBacklog er kravspecifikationen.

Produktejeren behøver ikke at afvise et ønske. Alt kan registreres på produktets backlog. Men alt bliver også prioriteret, og det betyder, at ønsker med laveste prioritet har begrænsede chancer for at blive en del af en Sprint. Det er alligevel vigtigt, at det optræder synligt på backloggen. Vise at ønsket er kendt og med hvilken prioritet, det optræder. Hvis nogen ønsker prioriteringen ændret, kan man henvende sig til ProductOwner.

Det er kun produktejeren, der kan beslutte nye indgange i backloggen. Og ændringer til de bestående, herunder ændringer til prioriteringen.

Produktejeren skal aldrig præsentere en afsluttet backlog, hvor der er taget stilling til et endeligt produkt. Backloggen lever et aktivt liv, så længe produktet er levende.

Backloggen skal være synlig. Udgiv den på et intranet, og forsyn den gerne med notifikationer i form af automatiske e-mail til dem, der måtte ønske at vide, når der sker ændringer.

Indsamlingen og registreringen kan automatiseres ved at lade interessenterne selv oprette nye indgange direkte. De kan oprette forslag, der dog kun kan få status af at være ubehandlede og uprioriterede.

Alt kan komme på listen: ønsker, krav, egenskaber, teknologi, arkitektur og fejlrettelser. Desuden kan den indeholde generelle problemer, der endnu ikke er oversat til krav. Den kan for eksempel beskrive, at sikkerhedsniveauet er for lavt, og robustheden er mangelfuld, uden at der er taget stilling til, hvad der skal gøres konkret for at øge sikkerheden og forbedre robustheden. Problemer optræder på backloggen med tydelig identifikation, så man kan se, at produktejeren skal omsætte problemet til krav, før det bliver en del af en Sprint.

Alternativt kan produktejeren vælge at holde problemer ude af backloggen. Men det er i mange tilfælde en fordel for produktejeren selv, at det er tydeligt og offentligt, hvilke problemer der er registreret om produktet. I de tilfælde, hvor et problem allerede er registreret i en fejlrapport, skal man selvfølgelig overveje, om det er fornuftigt at skabe redundant information. Det samme gælder i forhold til et eventuelt call-centers database og andre supportfunktioners registreringer. Links kan løse problemet.

Et problem kan komme med som opgave i en Sprint, med en afklaring af det som resultat.

Krav i backloggen kan beskrives med de oplysninger, man anser for nyttige. Det kan for eksempel være:

- kilden til kravet
- link til en mere detaljeret beskrivelse
- referencer og links der er nyttige når kravet skal implementeres

Det er tilladt at have en blanding af overordnede krav og detaljerede funktionsbeskrivelser. I takt med, at et overordnet ønske prioriteres højt nok til at skulle med i en Sprint, kan indholdet detaljeres. Estimatet på udviklingstiden kan som en konsekvens ændre sig - i op eller nedadgående retning. Det kan give grund til at ændre prioriteringen - igen i den ene eller den anden retning. Produktets backlog er dynamisk, fordi den afspejler produktejeren viden og ønsker på ethvert givent tidspunkt.

Der skal være estimer på alt i backloggen. Estimerne kan beregnes med forskellige estimeringsteknikker, herunder Storypoints, Function Points, Use Case Points, Class-Method Points og successiv kalkulation.

Indholdet til produktets backlog kan komme fra mange kilder. Brugere og kunder kan foreslå ændringer. Interne dele af organisationen kan stille forslag til sikkerhed og til andre af de non-funktionelle egenskaber. Marketing kan stille forslag baseret på konkurrerende produkter. Fejlrapporter kan komme mange steder fra. Ønsker om integration til andre systemer kan komme fra andre produktejere.

Produktbackloggen består af opgavebeskrivelser. I modsætning til sprintbackloggen der består af aktiviteter.

Userstories er perfekt som beskrivelseskabelon til indholdet i produktbackloggen.

Produktbackloggen lever så længe systemet lever. Med de på ethvert givent tidspunkt udestående krav og ønsker.

5. SprintPlanningWorkshop

Sprintens indhold defineres på denne workshop. Deltagerne er produktejeren, Scrum-Master og teamet. Formålet er at vælge det udsnit af backloggen, der ønskes gennemført af produktejeren, og som svarer til teamets formåen.

Workshoppen giver teamet en fælles forståelse for indholdet i den kommende Sprint.

Den typiske fremgangsmåde er at vælge fra højeste prioritet - for derefter at supplere med de dele med lavere prioritet, som det vil være naturligt, logisk eller bare let at tage med i samme Sprint.

Teamet nedbryder hver Userstory i de aktiviteter der skal udføres af teamet for at levere en implementering af Userstorien.

Teamet estimerer varigheden af hver aktivitet ved hjælp af Planning Poker.

Teamet afgør hvor meget de kan levere i Sprinten. Timerne, der er resultatet af estimeringen, akkumuleres. På et tidspunkt er der udsolgt, i forhold til timerne der er til rådighed for teamet som helhed.

Aktiviteterne udgør herefter SprintBacklog. Teamet vælger om de vil sætte teamdeltageres navne på aktiviteterne. I så fald gøres det sjældent for mere end en uge frem. Senere i Sprinten bliver det tilsvarende gjort for et kommende stykke tid, som teamet løbende selv vælger varigheden af.

Erfarne Scrum brugere foretrækker simple værktøjer. For eksempel at hænge de udvalgte Userstories op på væggen i teamets lokale.

6. SprintBacklog

Teamet opretter og vedligeholder en liste over de aktiviteter, der skal udføres i den givne Sprint. Den kaldes for en SprintBacklog. Det er aktiviteter som kodning, test og design. I modsætning til ProductBacklog der indeholder opgavebeskrivelser i forretningsprog.

SprintBacklog erstatter projektplaner.

Backloggen oprettes på SprintPlanningWorkshop. Hvis et team senere kommer foran, kan der tilføjes. Modsat hvis det senere kommer bagud. Ændringerne der måtte besluttes af produktejeren under vejs i Sprinten, vil normalt også have konsekvenser for aktiviteterne i Sprintbacklog.

Aktiviteterne brydes ned til det niveau, som udviklerne finder, er tilstrækkeligt til at blive estimeret pålideligt. Det vil for de fleste sige mellem en halv og en hel arbejdsdag pr. aktivitet.

Nogle arbejder med længere aktiviteter - afhængigt af deres fortrolighed med indholdet af opgaverne. Men brug 16 timer som maksimum for en aktivitet. Men der er forskel på om længden skyldes at det samme arbejde udføres mange gange, eller om den skyldes en høj kompleksitet.

Status på aktiviteterne i Sprintbackloggen holdes løbende ajour. Ændringer føres ind, så snart de kendes, således at backloggen er levende og ajourført i Sprintens levetid.

SprintBacklog revideres kun af teamet. Den må gerne offentliggøres og gerne på et intranet, som interessenterne har adgang til.

Estimaterne revideres ligeledes løbende - både på eksisterende aktiviteter og hvis en aktivitet nedbrydes yderligere. Estimaterne føres ajour dagligt for at have den bedst mulige status under vejs.

Alle estimater er restestimater. Altså udestående timer der skal bruges til færdiggørelse af en aktivitet.

SprintBacklog lever kun i den givne Sprint.

Erfarne Scrum brugere foretrækker simple værktøjer. For eksempel at hænge de aktiviteter der hører til Userstories op på væggen. Det er let at flytte dem mellem "kolonner" på væggen, i forhold til om de er "udestående", "igangværende" eller "afsluttede".

7. Sprinten

En sprint er en periode på en måned, hvor teamet arbejder på det valgte udsnit af produktbackloggen. Der er oversat af teamet til en SprintBacklog.

Teamet leverer produktet efter de 30/31 dage

En måned er den mest almindelige iterationslængde. Eller 4 uger rent, hvis det foretrækkes. Andre iterationslængder er 2 og 1 uge(r).

Det er en god idé at teste og integrere løbende. Dagligt. Hvis den underliggende produktionsmetode er XP, giver det sig selv. Fordelen ved at teste og integrere produktet på det tidligst mulige tidspunkt er, at det er muligt at bruge de daglige møder til at fjerne problemer og dermed øge produktiviteten. Samtidig reduceres risikoen for at beregne en forkert fremdrift.

Teamet beregner løbende, hvor mange timers arbejde der udestår på Sprintbacklog, sammenholder det med de timer, der resterer i Sprinten, og arbejder sammen med ScrumMaster, hvis der opstår uoverensstemmelser. Der er i så fald brug for at reducere eller øge mængden af det arbejde, der skal præsteres i Sprinten.

Efter en eller flere Sprint er der et produkt, som produktejeren kan vælge at release.

8. DailyScrum

ScrumMaster afholder og leder den daglige Scrum. Hele teamet deltager hver dag - helst personligt, men ellers via webcam, Video Conferencing, eller telefon.

DailyScrum har tre faste punkter på dagsordenen, hvor hver deltager i teamet efter tur fortæller:

1. hvad vedkommende har arbejdet med siden sidste møde
2. hvad vedkommende vil arbejde med frem til næste møde
3. om eventuelle hindringer for arbejdet

Mødet varer ca. et kvarter.

Der udføres ikke arbejde eller løses problemer på selve mødet.

Hindringer for arbejdet gentages hver dag, indtil de er fjernet.

Mødet holdes samme sted og tid og med de samme deltagere - nemlig teamet og ScrumMaster.

Det bliver altså cirka 22 møder i en Sprint. Mødet er et genialt styringsredskab - ikke mindst fordi det er så enkelt.

Alle andre er velkomne til at deltage, kun begrænset af den plads, der er til rådighed. Men det er kun teamet og ScrumMaster, der taler. Alle andre er tavse og befinder sig i baggrunden af lokalet - langs væggen eller på anden måde, så de ikke blander sig med teamet. De er Chickens. Deltagerne i teamet er Pigs.

DailyScrum giver et førstehåndsindtryk for enhver der måtte ønske det.

9. BurnDownChart

Opfølgningen vises som et BurnDownChart. Det viser den resterende mængde timer i Sprinten dag for dag. Det er basis for at reducere eller øge indholdet i Sprinten.

Der startes med den sum teamet sluttede med på SprintPlanningWorkshop. Og sluttes med nul på den sidste dag i Sprinten.

Virkelighedens fremdrift er ikke altid jævn og perfekt. Der kan forekomme udsving i BurnDown.

BurnDown en afspejling af SprintBacklog. Det er en sum, beregnet dagligt på grundlag af Sprintbackloggens restestimer.

BurnDownChart vedligeholdes som udgangspunkt af teamet, og det er alene teamets ansvar. Det er dog almindeligt at teamet beder ScrumMaster om at producere og publicere den dagligt. Indholdet er stadig teamets ansvar.

10. ScrumMaster

ScrumMaster er en af de tre roller.

ScrumMaster fungerer som coach i Sprinten. Både for teamet og for produktejeren.

ScrumMaster kan være den 'normale' daglige leder, for eksempel gruppe- eller projektlederen. Men der er intet i vejen for at have ScrumMastere, der udpeges specifikt til jobbet.

Lidt detaljer på ScrumMasters aktiviteter:

Arbejder sammen med produktejeren på alle ledelsesmæssige områder, herunder etablering af grundlaget for en Sprint og evalueringen efter en Sprint.

Faciliterer SprintPlanningWorkshop.

Leder den daglige Scrum.
Reviderer BurnDownChart hvis teamet beder om det.
Løser de problemer og fjerner de forhindringer, der kommer frem på DailyScrum.
Vedligeholder listen over udestående forhindringer (Impediments).
Fungerer som eneste kanal ind og ud af teamet.
Fungerer som coach for teamet mellem DailyScrums.
Samarbejder med teamet om at ændre på SprintBacklog når og hvis det er nødvendigt.
Faciliterer Retrospectives.

11. Teamet

Teamet er en af de tre roller.

Teamet påtager sig arbejde til den kommende Sprint på SprintPlanningWorkshop.
Teamet afgør selv, hvor meget det kan påtage sig til en Sprint. Det er en skidt begyndelse, hvis der trækkes urealistiske - eller for den sags skyld blot 'fremmede' - estimater ned over hovedet på det.

Teamets størrelse er identisk med de teamstørrelser, der arbejdes med i XP. Det vil sige 7-9 personer som foreslået af Kent Beck med flere, selvom nogle organisationer med held eksperimenterer med teamstørrelser på op til 12 personer.

Et team er ansvarlig for alt det arbejde, der skal udføres. Og det er en fordel, hvis teamet selv kan udføre det hele. Hvis en Sprint er afhængig af mange specialister eller konsulenter, der skal med i mindre dele af Sprinten, mistes der noget gruppedynamik. Derfor skal teamet omfatte udviklere, testere, designere og andre, der måtte være brug for. Det er det samme forhold, der gør sig gældende for projekter i almindelighed. Det er en fordel at have den viden og den ledelsesmæssige kompetence inden for projektet, der skal til for at udføre opgaven.

Teamet kan bruge roller og rollebeskrivelser som de måtte ønske. De kan låne fra RUP, hvis der er brug for inspiration til at forskellige testerroller. Og har brug for at arbejde med testdesigner, testanalytiker, tester eller testleder. Men roller i Scrum er ikke så formelle, som i RUP. Rollerne bruges primært for at skabe en fælles forståelse for, hvad man laver, når man har en rolle. Det vil sige, at når man påtager sig testerrollen i et tidsrum eller for nogle opgaver, så ved teamet, hvad der udføres af arbejde i den forbindelse.

Der etableres parallelle og/eller overlappende teams, når behovet for udvikling overstiger, hvad et team kan levere inden for de ønskede tidsfrister. Se skalering.

Teamet estimerer hver aktivitet, og det er teamets ret at ændre på det eksisterende estimat. Og pligt. Det er derfor en fordel, hvis teamet har haft indflydelse på estimatet, mens det optrådte på produktets backlog. Det giver desuden et bedre grundlag for produktjevers prioritering.

Teamet opretter selv SprintBacklog med de aktiviteter og opgaver, der skal udgøre den kommende Sprint. Teamet organiserer derefter selv arbejdet og sørger selv for at overholde organisationens standarder og generelle krav til systemudvikling.

Deltagerne i teamet arbejder Cross-functional i det videst mulige omfang. Det vil sige at de påtager sig alle de aktiviteter der er mulige. Hjælper hinanden, teamet, og dermed sig selv.

Lidt detaljer på teamets aktiviteter:

Planlægger aktiviteter og opretter den i sprintbackloggen.

Estimerer omfanget i timer.

Arbejder sammen, typisk i skiftende par.

Beregner restestimater dagligt på aktiviteter der har været i spil.

Beslutter under vejs om der er plads til nye opgaver.

- eller modsat om der skal reduceres i Sprintens indhold.

Fordeler opgaver og aktiviteter mellem sig.

12. SprintReviewMeeting

Sprinten afsluttes med et review af det arbejde, der er blevet præsteret. Produktet skal svare til det, der blev aftalt på SprintPlanningWorkshop. Korrigeret med de ændringer der er besluttet under vejs.

Produktejer, team og ScrumMaster deltager. Reviewet faciliteres af ScrumMaster.

Nøgleinteressenterne er velkomne, og i realiteten alle der ønsker det.

Reviewet varer typisk 2-3 timer og gennemgår det fysiske, færdige produkt.

ProductOwner vurderer resultatet. En ligefrem nøgtern vurdering.

13. Retrospective

Der er som regel to Retrospectives. En ekstern der handler om produktet. Og en intern der handler om processen.

I den interne Retrospective støtter man sig til en tidslinie over Sprinten. Hvornår var produktiviteten høj/lav? Hvornår var humøret højt/lavt?

Tre spørgsmål der ofte stilles i en intern Retrospective:

1. Hvad gik godt?
2. Hvad kunne være gået bedre?
3. Hvad forbavsede os?

14. Skalering af Scrum

Skalering sker ved at sætte mange Sprints i gang. De koordineres ved hjælp af særlige Daily Scrums til netop dette formål. Plus at der findes en række teknikker til at koordinere indholdet i de parallelle og overlappende Sprints.

Skalering af Scrum er nødvendig når et stort projekt skal gennemføres på kort tid.

15. Opsummering: Fordele ved at bruge Scrum

Trækket på ledelsessystemet lettes, uden at der mistes noget. Faktisk er det enklere at overvåge alle kørende projekter via sine ScrumMastere. Og set fra teamet er ledelsen dagligt tilgængeligt via ScrumMaster.

Scrum fungerer godt med hensyn til de grundlæggende ledelsesmæssige forhold. Det er tydeligt, hvad der foregår, og det er let at gribe ind.

Scrum er 'Agile'. Altså adræt, eller smidig, og arbejder let sammen med andre metoder. Især Extreme Programming.

På dokumentationssiden kan man benytte sig af UML som standard. Se mere om modellering i 'agile' sammenhæng på www.agilemodeling.com (Scott W. Ambler).

Scrum er velegnet til udvikling, der baseres på en ufuldstændig kravspecifikation eller en kravspecifikation, der ændrer sig. For en god ordens skyld skal det bemærkes at Scrum også er velegnet til projekter med en komplet, korrekt og stabil kravspecifikation. For i den situation blot at nyde godt af den høje produktivitet i Scrum.

I et projektforsløb er styring af risici noget nær alfa og omega. Risikostyring er ofte en forudsætning for succes. Scrum handler i høj grad om at fjerne risici.

Alle dele af Scrum bidrager til at reducere risici

- Produktejerens direkte prioritering
- Den daglige opfølgning
- Synliggørelsen af fremdrift
- Synliggørelsen af problemer
- Synliggørelsen af projektets aktiviteter på Sprintbacklog

Scrum kan iværksættes på et enkelt projekt - man kan få erfaringer og udvide brugen over tid. Scrum er let at implementere, da det er et enkelt koncept. Det hele kan prøves af med en begrænset investering.

16. Opsummering: Ulemper ved at bruge Scrum

De samme ulemper som i iterativ systemudvikling generelt.

Det vil sige, at produktet udvikles i en række konkrete leveringer. Der hver især først planlægges i detaljer umiddelbart før de gennemføres.

De store, sammenhængende dokumentationer produceres ikke. Man ender ikke med Use Cases, sekvensdiagrammer, klassemodeller og så videre, der alle hænger sammen og dokumenterer systemet. Selvfølgelig kan de produceres. Ved at oprette dem som aftalte produkter i ProductBacklog. No problem. De opstår blot ikke af sig selv i Scrum processen.

17. Scrum er til nødlidende projekter

Men før de bliver det. Nødlidende.

Meget af det der gøres i Scrum er det vi gør i den sidste måned af et langt projekt. For eksempel klar prioritering og daglig opfølgning. Det gøres blot i den første måned. Og alle efterfølgende.