



SCRUM

Februar 2010

Scrum: Udviklet og vedligeholdt af Ken Schwaber og Jeff Sutherland

INDLEDNING

GENERELT

SCRUM ER BASERET PÅ INDUSTRIANERKENDTE PRINCIPPER, DER GENNEM ÅRTIER HAR VÆRET ANVENDT OG VIST SIG NYTTIGE SOM BEDSTE PRAKSIS. DEREFTER BLEV DEN DEFINERET SOM EN EMPIRISK PROCES TEORI. SOM JIM COPLIEN EN GANG BEMÆRKEDE TIL JEFF: "ALLE VIL KUNNE LIDE SCRUM, DET ER HVAD VI ALLEREDE GØR, NÅR VI HAR RYGGEN MOD MUREN".

PERSONERNE BAG

Blandt de tusindvis af mennesker, der har bidraget til Scrum, bør vi fremhæve dem, der hovedsageligt bidrog i de første ti år. Først var der Jeff Sutherland, der arbejdede med Jeff McKenna og der var Ken Schwaber, der arbejdede med Mike Smith og Chris Martin. Scrum blev første gang formelt fremlagt og offentliggjort i forbindelse med OOPSLA 1995. I løbet af de efterfølgende fem år var det Mike Beadle og Martine Devos, som kom med betydelige bidrag. Og så alle de andre, uden hvis hjælp Scrum ikke ville have udviklet sig til hvad det er i dag.

HISTORIEN

Scrum kan allerede nu anses for at have en lang historie i softwareudviklingens verden. Vi påskønner de første steder, hvor Scrum blev afprøvet og forfinet, og vi nævner i denne sammenhæng: Individual Inc., Fidelity Investments og IDX (nu GE Medical).

OVERSÆTTELSE

Denne guide er en oversættelse af Ken Schwaber og Jeff Sutherlands oprindelige engelske version. Oversættelsen er udført af Certified Scrum Coach Bent Myllerup med bidrag fra Lektor Lone Borgersen, samt korrektur af Kirsten Myllerup Pallesen.

FORMÅL

Scrum er blevet anvendt til at udvikle komplekse produkter siden begyndelsen af 1990'erne. Dette dokument beskriver, hvordan man bruger Scrum til at bygge produkter. Scrum er ikke en proces eller en teknik til udvikling af produkter, men nærmere en ramme, inden for hvilken man kan anvende forskellige processer og teknikker. Scrums rolle er at gøre effektiviteten af din udviklingspraksis synlig, så du kan forbedre den og **sideløbende skabe en ramme for udviklingen af komplekse produkter.**

SCRUM TEORI

Scrum, som er funderet i teorien om empirisk proceskontrol, anvender en iterativ og inkrementel tilgang til at optimere forudsigelighed og kontrollere risici. Der er tre søjler, som hver støtter etablering af den empiriske proceskontrol.

DEN FØRSTE SØJLE ER TRANSPARENS

Transparens sikrer, at de aspekter af processen som påvirker resultatet, er synlige for dem, som er ansvarlige for produktet. Disse aspekter må ikke blot være gennemskuelige, det skal også være tydeligt, hvad der skal holdes for øje. Det betyder, at når nogen, der inspicerer en proces mener, at resultatet er færdiggjort, så skal det være i overensstemmelse med deres kriterier for færdiggørelse.

DEN ANDEN SØJLE ER INSPEKTION

De forskellige aspekter af processen skal tilses hyppigt nok til at uacceptable afvigelser i processen kan opdages. Hyppigheden af kontrollen må tage hensyn til, at alle processer bliver ændret som et resultat af inspektionen. Et problem opstår, når den krævede hyppighed af inspektioner overskrider tolerancen for hvor tit inspektionen kan foretages. Heldigvis synes dette ikke at være et problem i forbindelse med softwareudvikling. Den anden faktor er kompetencen hos de personer som inspicerer arbejdsresultaterne og den omhu hvormed de gør det.

DEN TREDJE SØJLE ER TILPASNING

Hvis man under inspektionen finder at ét eller flere aspekter i processen ligger udenfor de acceptable grænser – og det resulterende produkt derfor vil være uacceptabelt – skal processen eller arbejdsmaterialet tilpasses. Justeringerne må ske så hurtigt som muligt for at minimere yderligere afvigelser.

Der er tre punkter, hvor inspektion og tilpasning kan foretages i Scrum. Daily Scrum mødet bruges til at kontrollere fremdriften mod sprintets mål og til at foretage justeringer der optimerer værdien af den næste arbejdsdag. Derudover benyttes møderne Sprint Review og Sprint Planning til at kontrollere fremdriften mod målet for næste produktfrigivelse, samt til at foretage tilpasninger der optimerer værdien af de kommende sprints. Endelig benyttes mødet Sprint Retrospective til at evaluere det seneste sprint og til at bestemme hvilke tilpasninger, der vil gøre det næste sprint mere produktivt, berigende og fornøjeligt.

INDHOLDET AF SCRUM

De grundlæggende elementer i Scrum er **Scrum teams** og deres tilknyttede Roller, **Timeboxe**, Artefakter og **Regler**.

Scrum teams er konstrueret til at have fokus på at optimere fleksibilitet og produktivitet. Dette opnås via selvorganisering, tværfaglighed, og ved at arbejdet udføres i iterationer. I hvert Scrum team er der tre roller: 1) **ScrumMaster**, som er ansvarlig for at sikre, at processen er forstået og fulgt 2) **Product Owner**, der er ansvarlig for at maksimere værdien af Scrum teams arbejde og 3) **Teamet** der udfører arbejdet. Et team består af udviklere som samlet set har alle de færdigheder, der skal til for at omsætte Product Owners krav til ny funktionalitet i produktet, som potentielt kan frigives ved udgangen af et sprint.

Scrum anvender timeboxe til at skabe regularitet. De elementer i Scrum der bliver planlagt som timeboxe omfatter **Sprint** forløbet, samt møderne: **Release Planning**, **Sprint Planning**, **Daily Scrum**, **Sprint Review** og **Sprint Retrospective**. Hjertet af Scrum er et sprint, der er en iteration af en måneds varighed eller kortere. Sprints har en ensartet, fastsat længde og anvender de

samme Scrum-principper gennem hele udviklingsforløbet. Alle sprints resulterer i en funktionel udvidelse af produktet, som potentielt er klar til frigivelse. Et sprint starter umiddelbart efter at det forrige er afsluttet.

Scrum anvender fire primære artefakter: **Product Backlog** der er en prioriteret liste over alt, hvad der kan være behov for i produktet. **Sprint Backlog** der er en liste over opgaver der skal gennemføres i et sprint for at realisere dele af Product Backlog til en forøgelse af funktionaliteten i produktet der kan frigives. Et burndown-diagram anvendes til at illustrere den resterende backlog over tid. Et **Release Burndown** diagram illustrerer den resterende Product Backlog over tid af en release plan. Et **Sprint Burndown** diagram illustrerer de resterende opgaver i **Sprint Backlog** over tid af et sprint.

Scrum's timeboxe, roller og artefakter bindes sammen af nogle få **Regler**. Disse regler er beskrevet i resten af dette dokument. Det er for eksempel en Scrum-regel, at kun teammedlemmer – det vil sige, de mennesker der har forpligtet sig til at omsætte kravene i Product Backlog til en ny version af produktet - kan tale under et Daily Scrum møde.

Måder at implementere Scrum på, der ikke er regler, men blot forslag, er beskrevet i "Tip"-kasser.

TIP

Når der ikke er angivet regler, forventes det at brugerne af Scrum selv finder ud af, hvad de skal gøre. Forsøg ikke at finde den perfekte løsning, da problemet oftest ændrer sig hurtigt. Afprøv i stedet en mulig løsning og se, hvordan det virker. Denne inspicer- og-tilpas mekanisme, som er en del af Scrum's empiriske natur, vil hjælpe dig.

SCRUM ROLLER

I Scrum teamet indgår rollerne ScrumMaster, Product Owner og Team. Medlemmer af et Scrum team kaldes for "grise". Product Owner er "grisen", når det drejer sig om Product Backlog. Teamet er "grisen", når det drejer sig om arbejde, som udføres i sprintet. ScrumMaster er "grisen", når det drejer sig om Scrum processen. Alle andre er "kyllinger". Kyllinger kan ikke fortælle "grise", hvordan

de skal gøre deres arbejde. Analogien med kyllinger og grise kommer fra følgende historie:

En dag da en kylling og en gris er sammen, siger kyllingen: "Lad os starte en restaurant! "

Grisen tænker over det og siger: "Hvad skal vi kalde denne restaurant?"

Kyllingen siger: "Skinke og Æg!"

Grisen siger: "Nej tak, jeg vil være temmelig forpligtet, mens du blot vil være involveret!"

SCRUMMASTER

ScrumMaster er ansvarlig for at sikre, at Scrum teamet overholder Scrum's værdier, praksis og regler. ScrumMaster hjælper Scrum teamet og organisationen med at indføre Scrum. ScrumMaster uddanner Scrum teamet gennem coaching og ved at lede det til at være mere produktivt og fremstille produkter af højere kvalitet. ScrumMaster hjælper Scrum teamet med at forstå og anvende selvorganisering og tværfaglighed i dets arbejde. ScrumMaster hjælper desuden Scrum teamet til at gøre sit bedste i et organisatorisk miljø, der måske endnu ikke er optimeret til udvikling af komplekse produkter. Når ScrumMaster hjælper med at gennemføre disse ændringer, kaldes det "at fjerne

Tip

ScrumMaster arbejder sammen med kunder og ledelse om at finde en Product Owner og hjælper denne i gang med sit arbejde. ScrumMaster lærer Product Owner, hvordan hans eller hendes job skal udføres. Produkt Owner forventes at vide, hvordan man formår at optimere værdien af projektet ved hjælp af Scrum. Hvis de ikke gør det, vil vi holde ScrumMaster ansvarlig.

Tip

ScrumMaster kan være et medlem af teamet, f.eks. en udvikler, der udfører opgaver i Sprintet. Dette kan dog ofte føre til konflikter, når ScrumMaster skal vælge mellem at fjerne hindringer eller udføre egne opgaver. ScrumMaster bør aldrig samtidig være Product Owner.

forhindringer". ScrumMasterens rolle er at være den tjenende leder for Scrum teamet.

PRODUCT OWNER

Product Owner er den eneste person, der er ansvarlig for forvaltningen af Product Backlog og sikring af værdien af det arbejde teamet udfører. Product Owner vedligeholder Product Backlog og sikrer, at den er synlig for alle. Alle ved hvilke elementer, der har den højeste prioritet, så alle ved, hvad der vil blive arbejdet på.

Product Owner er én person, ikke et udvalg. Der kan dog eksistere udvalg, som rådgiver eller påvirker denne person, men folk, der ønsker at ændre prioriteten af et krav, må først overbevise Product Owner. Virksomheder som indfører Scrum, kan komme til at opleve, at det over tid vil påvirke deres metoder for fastsættelse af prioriteringer og krav.

For at Product Owner skal kunne lykkes må alle i organisationen respektere hans eller hendes beslutninger. Ingen har lov til at fortælle teamet, at det skal arbejde ud fra et andet sæt prioriteter og teamet må ej heller lytte til nogen, som giver dem anderledes instruktioner. Product Owners beslutninger er synlige gennem indhold og prioritering af Product Backlog. Denne synlighed kræver, at Product Owner til stadighed gør sit bedste og det betyder at rollen som Product Owner både er krævende og givende.

TIP

I forbindelse med kommerciel udvikling, kan Product Owner være en produktchef. For intern udvikling, kan Product Owner være leder af den funktion i virksomheden, der er ved at blive automatiseret.

TIP

Product Owner kan være et medlem af teamet som også udføre udviklingsopgaver. Dette øgede ansvar kan besværliggøre Product Owners muligheder for at samarbejde med interessenter. Product Owner kan derimod aldrig samtidig være ScrumMaster.

TEAMET

Gennem hvert sprint omformer teams af udviklere Product Backlog til en tilvækst af funktionalitet, et inkrement, der potentielt er klar til ibrugtagning. Teams er tværfaglige. Teammedlemmer har tilsammen de færdigheder, der er nødvendige for at lave et inkrement. Teammedlemmer har ofte specialiserede færdigheder, såsom programmering, kvalitetskontrol, forretningsanalyse, arkitektur, design af brugergrænseflader eller databasedesign. Men de færdigheder, som teammedlemmerne skal være fælles om – dvs. færdigheder til at håndtere krav og omforme dem til et brugbart produkt – plejer at være vigtigere. Mennesker der nægter at kode, fordi de er arkitekter eller designere, passer ikke godt ind i teams. Alle bidrager, også selv om det kan kræve at nye færdigheder tilegnes eller at gamle genopfriskes. Der er ingen titler i et team, og der er ingen undtagelser fra denne regel. Teams bliver ikke inddelt i undergrupper med bestemte ansvarsområder som eksempelvis test eller forretningsanalyse.

Teams er selvorganiserende. Ingen – ikke engang ScrumMaster – fortæller teamet hvordan man omformer Product Backlog til en inkrementer af ny funktionalitet klar til ibrugtagning. Teamet finder ud af det på egen hånd. Hvert medlem af teamet anvender hans eller hendes ekspertise på hvert eneste problem. Den synergi som opstår, forbedrer hele teamets samlede produktivitet og effektivitet.

Den optimale størrelse for et team er syv mennesker, plus eller minus to. Når der er færre end fem teammedlemmer, er der mindre interaktion og som følge heraf mindre produktivitetsgevinst. Hertil kommer at teams i dele af et sprint kan opleve at de er begrænsede fordi de ikke har de nødvendige færdigheder til at levere et produkt, som kan bruges. Hvis der er flere end ni medlemmer, så opstår der omvendt behov for alt for meget koordination. Store hold skaber mere kompleksitet end en empirisk proces kan styre. Vi er dog stødt på succesfulde teams, der er større eller mindre end den anbefalede størrelse. Product Owner og ScrumMaster rollerne er ikke medregnet i dette antal, medmindre de også er grise som løser opgaver i Product Backlog.

Teamsammensætningen kan ændre sig ved afslutningen af et sprint. Hver gang teammedlemskabet ændres, vil produktiviteten opnået via selvorganisering blive mindsket. Der bør derfor udvises forsigtighed når teamsammensætning ændres.

TIMEBOXE

Scrums timeboxe er **Release Planning** mødet, **Sprintet**, **Sprint Planning** mødet, **Sprint Review**, **Sprint Retrospective** og mødet **Daily Scrum**.

RELEASE PLANNING MØDET

Formålet med Release-planlægning er at udarbejde en plan og etablere nogle mål, som Scrum teams og resten af organisationen kan forstå og kommunikere. Release-planlægning besvarer spørgsmålene: "Hvordan kan vi på den bedst mulige måde omforme vores vision om til et vindende produkt?" og "Hvordan kan vi opfylde eller overstige den ønskede kundetilfredshed og rentabilitet (Return on Investment)?" Release-planen etablerer målet for produktets frigivelse, den højeste prioriterede Product Backlog, de største risici, samt de overordnede funktioner og funktionalitet som produktet skal indeholde ved frigivelsen. Den etablerer også en sandsynlig frigivelsesdato og et budget for omkostningerne, der skal holde, såfremt der ikke kommer ændringer. Organisationens kan derefter kontrollere fremskridt og foretage ændringer til denne release-plan sprint for sprint.

Det er valgfrit om man vil anvende Release-planlægning. Hvis Scrum teams begynder at arbejde, uden at dette møde er gennemført, vil dets manglende artefakter vise sig som en forhindring der skal løses. Det arbejde som skal til for at løse den hindring, vil blive en opgave i Product Backlog.

Med Scrum, bliver produkter bygget iterativt, og hvert sprint skaber et produktinkrement, dvs. en forøgelse af funktionaliteten, startende med de mest værdifulde og risikofyldte områder. Sprint efter sprint skabes der yderligere produktinkremerter og hvert inkrement udgør en bid af det endelige produkt som potentielt set er klar til ibrugtagning. Produktet bliver frigivet, så snart

funktionaliteten har nået et stadie, som giver brugsværdi for investorerne.

De fleste organisationer har allerede en release-planlægningsproces, og i de fleste af disse processer sker planlægningen i begyndelsen af projektet og bliver ikke ændret efterhånden som tiden går. I Scrum består release-planlægning i fastlæggelse af en overordnet målsætning og af de sandsynlige resultater. Denne release-planlægning kræver som regel ikke mere end 15-20 % af den tid som en organisation ville bruge på at udarbejde en traditionel release-plan. Under en Scrum release vil der dog jævnligt udføres just-in-time planlægning. Det sker ved hvert Sprint Review og Sprint Planning møde, samt dagligt på alle Daily Scrum møderne. Samlet set bruges der lidt mere tid på release-planlægning i Scrum end ved traditionelle projekter.

Release-planlægning kræver estimering og prioritering af Product Backlog for releasen. Der er mange teknikker, hvormed dette kan udføres. Disse teknikker ligger udenfor rammerne af Scrum, skønt de er ganske nyttige at anvende i tilknytning hertil.

SPRINTET

Et sprint er en iteration. Sprint bliver planlagt i timeboxe. Gennem sprintet sikrer ScrumMaster, at der ikke foretages ændringer, som vil påvirke sprintets mål. Både teamets sammensætning og kvalitetsmål forbliver konstante gennem hele sprintet. Sprints indeholder - og består af - Sprint Planning mødet, selve udviklingsarbejdet, Sprint Review mødet og Sprint Retrospective mødet. Det ene sprint efterfølges af det næste uden nogen mellemliggende tid.

TIP

Hvis teamet føler at det har lovet mere end det kan holde, mødes det med Product Owner for at fjerne opgaver fra eller reducere omfanget af de opgaver der var udvalgt til Sprintet fra Product Backlog. Hvis teamet føler at det har ekstra tid, kan det sammen med Product Owner vælge yderligere opgaver fra Product Backlog.

Et projekt har til formål at udrette noget. I software-udvikling er formålet at bygge et produkt eller et system. Ethvert projekt består

af en definition af, hvad der skal bygges, en plan for at bygge det, arbejdet der udføres i henhold til planen, samt det færdige produkt. Ethvert projekt har en horisont, en tidsramme, indenfor hvilken planen er god. Hvis horisonten bliver for lang, kan definitionen have ændret sig, for mange variabler kan være kommet til, risikoen kan være for stor, osv. Scrum er en ramme for et projekt, hvis horisont ikke er mere end en måned lang ad gangen og hvor der er nok kompleksitet til at en længere horisont vil være for risikabel. Projektets forudsigelighed skal kontrolleres mindst hver måned og risikoen for, at projektet kan komme udenfor kontrol eller blive uforudsigeligt, bliver håndteret mindst én gang om måneden.

TIP

Når et team begynder at anvende Scrum, kan det anvende Sprints af to ugers varighed, således at usikkerheden minimeres under læringsprocessen. Sprints af denne længde kan synkroniseres med andre hold ved at lægge to forløb sammen.

Et sprint kan afbrydes før timeboxen er afsluttet. Det er kun Product Owner, som har den fornødne myndighed til at afbryde sprintet, skønt han eller hun kan gøre det efter påvirkning fra interessenter, team eller ScrumMaster. Under hvilke omstændigheder kan det være nødvendigt at afbryde et sprint? For ledelsen kan det være nødvendigt at afbryde et sprint, hvis dets mål er blevet uaktuelle. Det kan forekomme, hvis virksomheden skifter retning, eller hvis markedet eller teknologien ændrer sig. Generelt bør et sprint annulleres, hvis det ikke længere giver mening omstændighederne taget i betragtning. Men på grund af den korte varighed af sprints, giver det sjældent mening at gøre det.

Hvis et sprint bliver afbrudt, bliver ethvert afsluttet og færdiggjort punkt fra Product Backlog gennemgået. Disse bliver accepteret, hvis de udgør en tilvækst i produktets funktionalitet, som potentielt kan frigives. Alle andre punkter fra Product Backlog bliver lagt tilbage med deres oprindelige estimat. Arbejdet der er udført på disse antages at være tabt. Sprint afbrydelser er ressourcekrævende, pga. omgrupperinger og fordi alle skal deltage i et nyt Sprint

Planning møde for at starte et nyt sprint. Sprint afbrydelser er ofte traumatiske for teamet og de er meget usædvanlige.

SPRINT PLANNING MØDET

Sprint Planning mødet er det sted, hvor iterationen bliver planlagt. Mødet er planlagt som en timebox på op til otte timer når sprintet har en måneds varighed. For kortere sprints, afkortes dette møde tilsvarende, så det forholdsmeæssigt har den rette længde (f.eks. vil Sprint Planning mødet i et to ugers sprint være planlagt som en timebox på op til fire timer). Et Sprint Planning møde består af to dele. Den første del er aktiviteten, hvor der tages beslutning om, hvad der skal ske i sprintet. Den anden del (en fire timers Timeboxe, når sprintet er af én måneds længde) er aktiviteten, hvor teamet finder ud af, hvordan det gennem sprintet kan udvide produktet med den ønskede funktionalitet.

Der er to dele af Sprint Planning mødet: Det er "Hvad?"-delen, samt "Hvordan?"-delen. Nogle Scrum teams kombinerer disse to. I den første del behandler Scrum team spørgsmålet "Hvad?". Her præsenterer Product Owner den højest prioriterede del af Product Backlog til teamet. De arbejder sammen for at finde ud af, hvilken funktionalitet der skal udvikles i løbet af det kommende sprint. Input til dette møde er Product Backlog, den seneste frembragte version af produktet, teamets kapacitet, samt teamets tidligere formåen. Mængden af opgaver som teamet påtager sig, er udelukkende op til teamet selv. Det er kun teamet, der kan vurdere, hvad det kan udrette i løbet af det kommende sprint.

Når elementerne fra Product Backlog er udvalgt, bliver sprintets mål afpasset. Sprint-målet er det mål, der vil blive opfyldt ved udarbejdelse af elementerne i Product Backlog. Dette er en erklæring, der giver teamet vejledning om, hvorfor den kommende funktionalitet i produktet udvides. Sprint-målet er en delmængde af hele produkt-releasens mål.

Formålet med at have et sprint-mål er at give teamet et råderum i forhold til funktionaliteten. For eksempel kunne målet for ovenstående sprint også være: "Automatiser funktionalitet vedr. kundens kontoændringer ved hjælp af sikker, genskabende

transaktionsmiddleware". Mens teamet arbejder, holder de sig dette mål for øje. For at opfylde dette mål, implementeres den fornødne funktionalitet og teknologi. Hvis det viser sig at arbejdet er sværere at gennemføre end teamet havde forventet, samarbejder teamet med Product Owner og implementerer kun en delmængde af funktionaliteten.

I den anden del af Sprint Planning mødet behandler teamet spørgsmålet "Hvordan?". Under Sprint Planning mødets anden del (en fire-timers timebox, når sprintet er af én måneds længde), fastlægger teamet hvordan det vil omforme den del af Product Backlog, som er udvalgt under første del af Sprint Planning mødet ("Hvad" delen) til en færdig leverance. Normalt vil teamet starte med at tilrettelægge arbejdet. Mens tilrettelæggelsen sker, vil teamet identificere opgaver. Disse opgaver er de detaljerede arbejdsopgaver som er nødvendige for at omforme elementerne i Product Backlog til funktionsdygtig software. Opgaverne bør brydes ned i mindre dele, så de kan gennemføres på mindre end én arbejdsdag. Listen af opgaver kaldes samlet for Sprint Backlog. Teamet bruger selvorganisering til at udføre det arbejde, som Sprint Backlog beskriver. Dette sker enten i løbet af Sprint Planning mødet eller rettidigt i løbet af sprintet.

Product Owner er til stede under den anden del af Sprint Planning mødet for at afklare spørgsmål om elementerne i Product Backlog og hjælpe teamet med afvejninger. Hvis teamet skønner, at der er for meget eller for lidt arbejde, kan det genforhandle omfanget af elementer fra Product Backlog med Product Owner. Teamet kan også invitere andre personer til at deltage med henblik på at yde teknisk eller domænemæssig rådgivning. Det er oftest først under dette møde, at et nyt team vil finde ud af om det som team, ikke som enkeltpersoner, vil synke eller flyde ovenpå. Teamet vil indse, at det først og fremmest må stole på sig selv. Når dette faktum går op

TIP

Normalt vil kun 60-70 % af den samlede Sprint Backlog blive udfærdiget under Sprint Planning mødet. Resten er henlagt til senere detaljering eller givet grove estimater som nedbrydes senere i sprintet.

for teamet, begynder det at blive selvorganiserende og påtage sig de egenskaber og den adfærd, som et rigtigt team må have.

SPRINT REVIEW

Ved afslutningen af sprintet afholdes der et Sprint Review møde. Dette møde er en timebox på fire timer, når sprintet er af en måneds længde. For kortere sprints, afkortes dette møde tilsvarende, så det forholdsæssigt har den rette længde (f.eks. vil Sprint Review i et to ugers sprint have en tidsramme på to timer). Under Sprint Review arbejder Scrum teamet og de berørte parter sammen om at vurdere det opnåede arbejdsresultat. Baseret på denne vurdering, samt de ændringer som kan være tilføjet Product Backlog i løbet af sprintet, samarbejdes der om at finde frem til de næste ting som kan blive gennemført. Dette er et uformelt møde, hvor præsentation af den udviklede funktionalitet har til formål at fremme samarbejdet om, hvad der videre skal ske.

Mødet omfatter mindst følgende elementer. Product Owner identificerer, hvad der er blevet færdiggjort, og hvad der ikke er. Teamet drøfter, hvad der gik godt under sprintet, hvilke problemer de løb ind i, og hvordan disse blev løst. Teamet demonstrerer herefter det arbejde som er færdiggjort og besvarer spørgsmål. Product Owner diskuterer dernæst den aktuelle status for Product Backlog. Baseret på forskellige antagelser om teamets hastighed, fastslår han eller hun den forventede slutdato for releasen. Hele gruppen evaluerer herefter, hvad man har set og hvilken indflydelse dette har i forhold til den videre planlægning. Sprint Review giver værdifulde input til det efterfølgende Sprint Planning møde.

SPRINT RETROSPECTIVE

Efter Sprint Review og forud for næste Sprint Planning møde, har Scrum teamet et Sprint Retrospective møde. Det er et tre-timers møde, når sprintet er på én måned (afpas længden tilsvarende hvis sprintets længde reduceres). På dette møde tilskynder ScrumMaster Scrum-teamet til, inden for Scrum-processens rammer og praksis, at revidere deres udviklingsproces med henblik på at gøre den mere effektiv og sjov i det kommende sprint. Mange bøger dokumenterer teknikker der er nyttige at anvende under retrospektives.

Formålet med Retrospectives er, at undersøge, hvorledes det seneste Sprint forløb med hensyn til mennesker, relationer, proces og værktøjer. Inspektionen bør identificere og prioritere de vigtigste elementer der gik godt, samt de områder som - hvis de blev udført anderledes - kunne gøre resultatet endnu bedre. Disse omfatter Scrum teamets sammensætning, mødearrangementer, redskaber, definition af "færdiggjort," metoder til kommunikation og processer til at forme elementerne i Product Backlog om til "færdiggjort" funktionalitet. Ved afslutningen af Sprint Retrospective bør Scrum-teamet have identificeret konkrete forbedringstiltag, som det iværksætter i det kommende Sprint. Disse ændringer afspejler tilpasningen til den empiriske inspektion.

DAILY SCRUM

Hvert team mødes dagligt i et 15-minutters "inspicer-og-tilpas" møde kaldet Daily Scrum. Daily Scrum afholdes på samme tid og samme sted gennem alle sprintene. Under mødet forklarer hvert teammedlem:

1. Hvad han eller hun har gennemført siden sidste møde
2. Hvad han eller hun vil gennemføre inden næste møde og
3. Hvilke hindringer der er i vejen for, at han eller hun kan gennemføre sit arbejde.

Daily Scrums forbedrer kommunikation, overflødigger øvrige møder, identificerer og fjerner forhindringer for udviklingen, fremhæver og fremmer en hurtig beslutningsproces, samt forbedrer samtlige deltageres kendskab til projektet.

Det er ScrumMaster som sikrer at teamet afholder mødet. Teamet er ansvarlig for gennemførelse af Daily Scrum mødet. ScrumMaster lærer teamet at holde Daily Scrum mødet kort, ved at håndhæve reglerne og sørge for at folk taler kortvarigt. ScrumMaster må også håndhæve reglen om, at kyllingerne ikke får lov til at tale eller på nogen måde forstyrre afholdelsen af Daily Scrum.

Daily Scrum er ikke et statusmøde. Det er kun for dem, som omformer elementerne i Product Backlog til en ny leverance af

produktet (teamet). Teamet har forpligtet sig til at opfylde sprintets mål og de berørte elementer i Product Backlog. Daily Scrum er en inspektion af fremdriften mod sprintets mål (de tre spørgsmål). Man afholder normalt efterfølgende møder hvor der følges op på det kommende arbejde i sprintet og nødvendige tilpasninger foretages. Hensigten med dette er, at optimere mulighederne for at teamet vil opfylde sine mål. Dette er et grundlæggende "inspicer-og-tilpas" møde i Scrums empiriske proces.

SCRUM ARTEFAKTER

Scrum artefakter omfatter Product Backlog, Release Burndown, Sprint Backlog samt Sprint Burndown.

PRODUCT BACKLOG OG RELEASE BURNDOWN

Kravene til det produkt som teamet/teamene er ved at udvikle, er opført som elementer i Product Backlog. Product Owner er ansvarlig for Product Backlog, dens indhold, dens tilgængelighed og dens prioritering. Product Backlog er aldrig komplet og viser i sin første version kun de krav der er kendte og forståede fra begyndelsen. Product Backlog udvikler sig efterhånden som produktet og det miljø hvori det vil blive anvendt udvikler sig. Backloggen er dynamisk, idet den hele tiden ændrer sig for at afspejle hvad produktet behøver for at være afpasset, konkurrencedygtig og anvendeligt. Så længe et produkt eksisterer, vil dets Product Backlog ligeledes eksistere.

Product Backlog repræsenterer alt det, der er nødvendigt for at udvikle og lancere et vellykket produkt. Det er en liste over features, funktioner, teknologier, forbedringer og fejlrettelser, der udgør de ændringer, som vil blive gjort på produktet i fremtidige udgivelser. Hvert Product Backlog element har

TIP

Product Backlog elementer er normalt angivet som User Stories. Use Cases er også velegnede, men de er bedre til brug i forbindelse med udvikling af livs- eller missionskritisk software.

attributterne beskrivelse, prioritet og estimat. Prioritet er bestemt af risiko, værdi og nødvendighed. Der er mange teknikker til vurdering af disse attributter.

Product Backlog er sorteret i prioriteret rækkefølge. De højest prioriterede elementer i Product Backlog styrer de nærmeste udviklingsaktiviteter. Jo højere prioritet, jo mere haster det, jo mere er der tænkt over det, og jo mere enighed er der om dets værdi. Elementer med højere prioritet er klarere og har mere detaljerede oplysninger end lavere prioriterede elementer, og de giver et bedre estimeringsgrundlag. Jo lavere prioritet, des mindre detalje - helt ned til meget vage kommentarer.

Efterhånden som et produkt anvendes, som dets værdi stiger og som markedet giver feedback, vil dets Product Backlog udmønte sig i en større og mere omfattende liste. Krav vil konstant ændre sig. En Product Backlog er et levende dokument. Ændringer i forretningskrav, markedsforhold, teknologi og personale forårsager ændringer i Product Backlog. For at minimere unødigt arbejde er det kun de højest prioriterede elementer, der skal være beskrevet i detaljer. De elementer i Product Backlog, som vil holde teamet beskæftiget de nærmeste sprints, er finkornede og er blevet nedbrudt således, at et hvilket som helst element kan blive færdiggjort inden for varigheden af et sprint.

TIP

Scrum teams bruger ofte 10 % af hvert sprints tid på at bearbejde Product Backlog så den lever op til ovenstående definition. Vi kalder dette for "Grooming af Backloggen". Når bearbejdningen er resulteret i det rette niveau af nøjagtighed, vil elementerne i toppen af Product Backlog (højeste prioritet, størst værdi) være nedbrudt i sådan en grad, at de passer ind i et sprint. De er blevet analyseret og tænkt igennem i løbet af bearbejdningsprocessen. Når Sprint Planning mødet finder sted, vil disse højest prioriterede Product Backlog elementer være velbeskrevet og lette at udvælge.

Flere Scrum teams arbejder ofte sammen om det samme produkt. Der bruges da stadig blot én Product Backlog til at beskrive det kommende arbejde på produktet. En ekstra attribut anvendes i så fald til gruppering af elementerne i Product Backlog. Gruppering kan

finde sted i henhold til relateret funktionalitet, teknologi eller arkitektur og bruges ofte til organisere arbejdet imellem de forskellige Scrum teams.

Release Burndown grafen angiver summen af den forventede arbejdsindsats for den resterende del af Product Backlog. Den anslåede indsats er angivet i den enhed for arbejdsindsats som Scrum teamet og organisation nu engang har besluttet sig for at anvende. Måleenheden for tid er normalt sprints.

Estimaterne for Product Backlog elementerne bliver udarbejdet første gang i forbindelse med Release Planning. Nye Product Backlog elementer bliver estimeret efterhånden som de bliver oprettet. Når Product Backlog "groomes" (på dansk: "trimmes") bliver estimerne gennemgået og revideret. Dog kan de blive opdateret til enhver tid. Teamet er ansvarlig for alle estimerne. Product Owner kan hjælpe teamet med en bedre forståelse og udvælgelse af kompromisser, men det endelige estimat foretages udelukkende af teamet. Product Owner sørger for, at Product Backlog og Release Burndown til enhver tid er ajourført og tilgængelig. En trendlinje kan tegnes på baggrund af ændringerne i den resterende arbejdsindsats.

TIP

Accepttest anvendes ligeledes ofte som en attribut på elementerne i Product Backlog. De kan ofte erstatte detaljerede beskrivelser med en testbar beskrivelse af, hvad Product Backlog elementet skal leve op til, når det er færdiggjort.

TIP

Med mindre teamene har arbejdet sammen før, kender produktet godt og forstår den bagvedliggende teknologi, kan trendlinjer være upålidelige i de første to til tre Sprints i et projekt.

TIP

I nogle organisationer tilføjes der mere arbejde til backloggen end der afsluttes. Dette kan skabe en trendlinje der er vandret eller endda stigende. For at kompensere for dette og bevare transparens, kan en ny baseline blive oprettet, når arbejde er tilføjet eller fjernet. Denne baseline bør kun reduceres eller forøges når der sker væsentlige ændringer, og bør være veldokumenterede.

SPRINT BACKLOG OG SPRINT BURNDOWN

Sprint Backlog består af de opgaver, som teamet skal udføre for at omforme Product Backlog elementerne til en færdiggjort udvidelse af produktet. Mange af opgaverne er oprettet i løbet af Sprint Planning mødet. Opgaverne beskriver alt det arbejde, som teamet identificerer som nødvendigt for at opfylde sprintets mål. Opgaverne i Sprint Backlog skal brydes ned i dele, som har en størrelse, der gør at det er muligt at følge fremdriften i forbindelse med Daily Scrum. Én dags arbejde eller mindre, er det almindelige omfang for en Sprint Backlog opgave.

Teamet ændrer på Sprint Backloggen gennem hele sprintet, ligesom Sprint Backloggen udvikler sig gennem sprintet. Efterhånden som der oprettes individuelle opgaver, vil teamet erkende, at der muligvis er behov for flere eller færre opgaver, eller at givne opgaver vil komme til at tage længere eller kortere tid end først antaget. Efterhånden som nyt arbejde findes nødvendigt, vil teamet tilføje det til Sprint Backlog. Når opgaverne bliver bearbejdet eller afsluttet, vil estimatet for det resterende arbejde for hver opgave blive opdateret. Opgaver bliver fjernet, hvis det viser sig, at de er unødvendige. Det er kun teamet som gennem sprintet kan ændre i Sprint Backloggen. Det er kun teamet, der kan ændre indholdet eller estimatet på opgaverne. Sprint Backlog er et meget synligt, tidstro billede af det arbejde, som teamet planlægger at gennemføre i løbet af sprintet og den tilhører udelukkende teamet.

Sprint Backlog Burndown er en graf, som over tid viser størrelsen af det arbejde, der er tilbage i et sprint. Grafen tegnes ved at man hver dag i sprintet opsummerer backlog-estimerne. Den mængde arbejde, der er tilbage i et sprint, er summen af det resterende arbejde for hele Sprint Backloggen. Der holdes rede på denne summering hver dag. Ved at tegne en linje

TIP

Når det er muligt, så håndtegn burndown diagrammet på et stort ark papir, der hænges op i teamets arbejdsområde.

Teams er mere tilbøjelige til at se et stort, synligt diagram, end de er til at se på et Sprint Burndown diagram i Excel eller et værktøj.

gennem punkterne på grafen, kan teamet styre fremdriften i sprint-arbejdet. Der bliver ikke set på det faktiske tidsforbrug i Scrum. Det er kun det resterende arbejde og datoen der har interesse.

En af Scrum's regler vedrører formålet med hvert sprint, nemlig det at levere en udvidelse af produktets funktionalitet, der potentielt er klar til frigivelse i henhold til en operationel definition af "færdiggjort".

FÆRDIGGJORT

Scrum kræver at teamene laver et produktinkrement i hvert sprint. Dette inkrement skal potentielt være klar til frigivelse, fordi Product Owner kan vælge at tage funktionaliteten i anvendelse straks. For at gøre det muligt, skal leverancen være en komplet del af produktet. Det skal være "færdiggjort". Hvert inkrement skal tilføje noget til alle foregående inkremitter og skal testes grundigt. Testen skal sikre at alle inkremitter kan arbejde sammen.

TIP

"Ufærdigt" arbejde bliver ofte samlet i et Product Backlog element, der kaldes "Ufærdigt arbejde" eller "Implementeringsarbejde." I takt med at dette arbejde akkumulerer, bliver Product Backlog burndown stadig mere præcis, end hvis det ikke var akkumuleret.

I forbindelse med produktudvikling, kan en påstand, om at funktionaliteten er færdiggjort, føre til at nogen antager, at den i det mindste er kodet, refaktoreret, unit-testet, bygget og accepttestet. Andre antager måske, at koden blot er blevet bygget. Hvis ikke alle ved, hvad definitionen på "færdiggjort" er, vil de to andre søjler af den empiriske proceskontrol ikke virke. Når en person beskriver noget som "færdiggjort", skal alle forstå, hvad "færdiggjort" betyder.

"Færdiggjort" definerer hvad teamet mener, når det forpligter sig til at "udføre" et Product Backlog element i et sprint. Nogle produkter indeholder ikke dokumentation, så i de tilfælde omfatter definitionen af "færdiggjort" ikke dokumentation. En helt

"færdiggjort" leverance omfatter al analyse, design, refaktorering, programmering, dokumentation og test for leverancen samt alle Product Backlog elementerne i leverancen. Testning omfatter unit-, system-, bruger- og regressionstest, ligesom ikke-funktionelle test som ydelses-, stabilitets-, sikkerheds- og integrationstest. "Færdiggjort" omfatter også al internationalisering. Nogle teams er endnu ikke i stand til at favne alt, hvad der kræves i henhold til deres definition af "færdiggjort". Dette må være klart for Product Owner. Det resterende arbejde, som mangler i forhold til "færdiggjort", skal gennemføres før produktet kan implementeres og tages i anvendelse.

AFSLUTTENDE TANKER

Nogle organisationer er ikke i stand til at bygge et komplet inkrement indenfor ét sprint. De har måske endnu ikke den fornødne infrastruktur til automatiserede test til at gennemføre alt testarbejdet. I dette tilfælde oprettes der to kategorier for hvert inkrement: Det "færdiggjorte" arbejde og det "ufærdige" arbejde. Det "ufærdige" arbejde er den del af hvert inkrement, der er nødt til at blive afsluttet på et senere tidspunkt. Product Owner ved nøjagtigt, hvad han eller hun inspicerer ved afslutningen af sprintet, fordi inkrementet opfylder definitionen på "færdiggjort" og Product Owner forstår denne definition. "Ufærdigt" arbejde bliver tilføjet et Product Backlog element med navnet "Ufærdigt arbejde", så det kan akkumuleres og afspejles korrekt på Release Burndown grafen. Denne teknik skaber synlighed vedr. fremdriften mod en frigivelse. "Inspicer og tilpas" delen i Sprint Review er lige så nøjagtig, som denne synlighed tillader.

Hvis et team eksempelvis ikke er i stand til at udføre ydelses-, regressions-, stabilitets-, sikkerheds- og integrationstest for hvert Product Backlog element, så må den del af arbejdet holdes op mod den del der faktisk kan udføres (analyse, design, refaktorering, programmering, dokumentation, unit- og brugertest). Lad os sige, at denne andel er seks dele "færdiggjort" og fire dele "ufærdigt". Hvis teamet afslutter et Product Backlog element af seks enheders arbejde (teamet har baseret sit estimat på de dele af opgaven, som

man kender løsningen på), skal der tilføjes fire enheder til "ufærdigt arbejde" i Product Backlog element, når de er færdige.

Sprint efter sprint akkumuleres det "ufærdige" arbejde for hvert inkrement og det skal håndteres inden produktet frigives. Arbejde bliver akkumuleret lineært, skønt det faktisk har en form for eksponentiel akkumulering, der er afhængig af den enkelte organisations karakteristika. Release Sprints føjes til slutningen af projektet, således at dette "ufærdige" arbejde kan afsluttes. Antallet af nødvendige sprints er uforudsigeligt i et omfang svarende til i hvor høj grad akkumuleringen af "ufærdigt" arbejde ikke er lineær.