

AgileLeanBooks

Scrum i fugleperspektiv

Hvordan man leder produktudvikling, projekter og arbejde generelt i en kompleks verden

Kurt B. Nielsen & Anne Due Broberg
AgileLeanHouse A/S, info@AgileLeanHouse.com
Lysholt Allé 6, DK-7100 Vejle, Danmark



Indholdsfortegnelse

Introduktion	3
Grundlaget	3
Hvor kommer Scrum og Agile fra?	3
En Tidslinje	4
Cynefin Modellen	6
Og nu til Fugleperspektivet	7
Arbejdet opdeles i små bidder - Sprints	8
Alle ønsker og krav samles på en ordnet liste	9
Alle har entydige roller	9
Produkt Owneren	9
Development Teamet (Teamet)	10
Scrum Masteren	10
Resten af organisationen	11
Scrum aktiviteter, møder eller ceremoni	11
Sprint Planning 1	11
Sprint Planning 2	12
Daily Scrum	12
Sprint Review	13
Sprint Retrospective	13
Product Backlog Refinement	14
Artefakter og Informationsformidling	15
Product Backlog	15
Sprint Backlog	16
Improvement Backlog	16
Product Burndown	17
Sprint Burndown	17
At arbejde i Sprints	17
Definition af ”helt færdig”	19
Opsummering	19
Engelsk Definition of termer	20

Copyright © 2017, AgileLeanHouse A/S. All rights reserved.

The content of this document is subject to change without notice and does not represent a commitment on the part of AgileLeanHouse A/S. AgileLeanHouse A/S makes no warranty of any kind with regard to this material.

AgileLeanHouse A/S shall not be liable for any errors contained herein or damages, including any loss of profits, or other incidental or consequential damages, arising out of your use of this written material.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information recording and retrieval systems, for any purpose, without the express written permission of AgileLeanHouse A/S.

All registered and unregistered trademarks used in this document are the exclusive property of their respective owners.

Document ID	WP-Scrum-011	Filename	ScrumIFugleperspektiv32
Author	Kurt Nielsen	Revised	3.2 2017-10-19

Introduktion

Scrum er en måde at tænke og organisere arbejde på. Det er en måde at vride motivation og resultater ud af situationer, der er for komplekse til traditionel planlægning. Scrum hører under de metoder, der ofte betegnes "Agile", det begreb hører så igen under "Lean tænkning". Scrum er først og fremmest defineret igennem dokumentet "**The Scrum Guide**" ("SG"), der vedligeholdes af fædrene Jeff Sutherland and Ken Schwaber, læs mere [her...](#)

Scrum opstod indenfor IT udvikling i 1990'erne og er i det nye årtusinde vokset til at være den mest brugte af de Agile projektledelsesmetoder. Scrum er tilstrækkeligt bred i sit grundlag til at kunne bruges i alle slags projekter og initiativer¹, og bliver nu brugt i generel produktudvikling, salg- marketing og andre domæner. Steve Denning, en foregangsmand inden for ledelse og innovation udtalte for et par år siden, "*Hvis der var en Nobel pris for management, og hvis der var nogen form for retfærdighed i denne verden, ville jeg tro at prisen ville gå til bl.a. Jeff Sutherland, Ken Schwaber og Mike Cohn for deres bidrag til opfindelsen af Scrum*". Scrum er virkelig en ledelsesrevolution.

Ifølge en nyere undersøgelse² har firmaer og teams der bruger en eller anden form for Agil model konsistent højere successrate, langt de fleste bruger Scrum (85%). Denne nye tankegang kan dateres tilbage til 2001, hvor Ken Schwaber skrev sin bestseller bog "Agile Software Development with Scrum" og det Agile Manifest³ blev underskrevet af en gruppe indflydelsesrige professionelle indenfor software verdenen. Siden da har Scrum bredt sig som en løbeild. Der er nu over 430.000 certificerede Scrum Mastere, som er i stand til at køre Scrum projekter, registreret under Scrum Alliance⁴. For eksempel kræves der af mange projekter i den offentlige sektor i Norge, at der anvendes Scrum

Hvorfor bruge Scrum og agile metoder i i initiativer i organisationer? For det første findes der solid empirisk evidens for et større udbytte, lavere omkostninger, øget forudsigelighed og mere tilfredse interessenter. Vi tror desuden også, at det er ved at gå op for mange beslutningstagere, at det er umuligt at lave den perfekte plan fra starten af i nutidens komplekse initiativer. Det er ikke engang muligt for os at opnå perfekte specifikationer. Vi befinder os simpelthen ikke i et ordnet domæne i de fleste tilfælde - vi befinder os i det komplekse domæne, hvor der er brug for at tage hensyn til al den viden, der opstår undervejs.

Grundlaget

Hvor kommer Scrum og Agile fra?

Scrum bliver ofte kaldet agil projektledelse, og "Agile" betyder stort set alt arbejdsrelateret, som fokuserer på menneskelige interaktioner, resultater, samarbejde med kunder og konceptet om at tilpasse sig forandring. I et bredere perspektiv ses Scrum og Agile som del af en større familie, der ofte kaldes "Lean Thinking", der fokuserer på skabe balanceret værdi for alle, at reducere spild og at opbygge respekt for mennesker og deres arbejde.

Der er dog også mere, der lægger til grund for Scrum og Agile: Studier i kompleksitet søger at beskrive, hvordan en forståelse af komplekse, adaptive systemer kan opnås. Per definition er kompleksitetsforskning kompleks og svær at indkapsle med et par enkelte nøgleord.

¹ Vi vil her bruge ordet "initiativ" til at dække over projekter, udvikling af produkter og services samt arbejde i al almindelighed.

² 2016/17 research, Prof. Dr. Ayelt Komus, University of Applied Sciences, Koblenz

³ Se <http://agilemanifesto.org/>

⁴ Se <https://www.scrumalliance.org>

Cynefin - en forståelsesramme som Dave Snowden⁵ har udarbejdet - har vist sig meget brugbar til at give en forståelse af, hvordan forskellige situationer indenfor ledelse af projekter, initiativer og organisationer kan håndteres.

I dette perspektiv, kan vi begynde at forstå, *hvorfor* Scrum fungerer så godt.

En Tidslinje

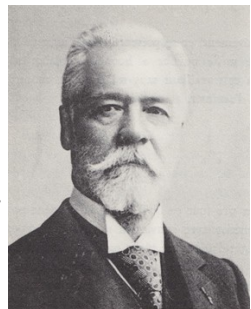
Fredric Winslow Taylor (USA 1856-1916) var den første, der forholdte sig til konceptet arbejde som et systematisk studie. Han blev stamfar til det såkaldte Scientific Management - Videnskabelig Ledelse. Hans bog fra 1911 om emnet (*Principles of Scientific Management*) markerede begyndelsen på en ny epoke med fokus på effektivitet, som sås mest tydelig hos Henry Ford og hans samlebånd for Model T, selvom Ford sandsynligvis aldrig mødte Taylor.



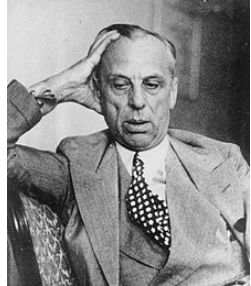
Taylors idéer affødte koncepter såsom Gant skemaet og budgettering. På grund af Taylors baggrund i industriel produktion var han dog dybt overbevist om, at alt arbejde kunne brydes ned af en ekspert til et sæt simple repeterbare opgaver, som kunne måles og studeres. Det ville altid være muligt for eksperten at finde den bedste måde at gøre ting på (best practice).

Taylors syn på det almindelige menneske var dog ikke særligt positivt. Han anså den gennemsnitlige arbejder for at være både for dum og for doven til at kunne have indflydelse på sit eget arbejde; eksperter måtte planlægge for ham og motivere ham med en blanding af gulerod og stok. Dette koncept findes stadig hos os i dag i det vi betegner Neo-Taylorisme. Både Lenin, Mussolini og Hitler var meget inspirerede af og begejstrede for Taylors idéer.

Henri Fayol. (Frankrig 1841-1925) var mineingenør, der endte med at blive chef for et stort mine foretagende. Her udviklede han over årene et sæt af 14 principper som han anså for at være helt grundlæggende for at styre et større foretagende. I 1916 udgav han bogen *Administration Industrielle et Générale*, som er et af de første arbejder med en decideret teori for management, og er stadig en af de mest omfattende.



Selvom Fayol ikke er meget kendt i offentligheden, så har hans tanker om administration, kommandostruktur osv. haft stor indflydelse management de sidste 100 år. Hans opfattelse af management som dem der fordeler, koordinerer, overvåger og kontrollerer er stadig hos os.



Alfred Pritchard Sloan, Jr. (USA 1875-1966) var øverste chef for General Motors Corporation i en menneskealder. Han reorganiserede firmaet på en måde, der blev brugt som model for næsten alle firmaer i det 20. århundrede. Han opdelte General Motors i separate selvstyrende divisioner, underlagt økonomisk, budgetmæssig og proceduremæssig kontrol fra en lille central stab. Sloan introducerede - som den første - også systematisk og detaljerede strategiske planlægningsprocedurer for firmaets divisioner. Hans fokus på budgetter, måltal og detaljeret up-front planlægning er stadig hos os i dag.

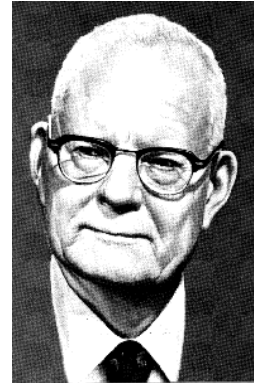


Walter A. Shewhart (USA 1891-1967) var fysiker, ingeniør og statistiker. Han beskrev som den første mulighederne for statistisk kvalitetskontrol under sit arbejde i Western Electric i 1920'erne. Shewharts arbejde påviste nødvendigheden af at reducere variation i produktionsprocesser og arbejde i det hele taget. På den måde kom de første idéer

⁵ For en introduktion se denne video <http://www.youtube.com/watch?v=N7oz366X0-8>

frem om konstant forbedring. Shewhart skabte de første versioner af control charts som sidenhen er blevet brugt til at monitorere alle former for processer for at kunne afgøre om det der sker er et resultat af indbyggede tilfældigheder i systemerne, eller om der er noget specielt på færde.

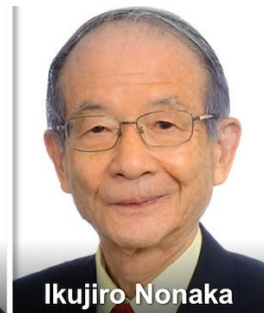
W. Edwards Deming (USA 1900-1993) har bidraget utroligt meget med en fokus på kvalitet og det at involvere hele mennesket i arbejdsprocessen. Han udgjorde en opposition til Taylor, Ford og Sloan. Deming bidrog til den amerikanske krigsindsats under Anden Verdenskrig ved at hjælpe med at skabe og undervise om den kvalitetsstandard, der var altafgørende for fremstillingen af USA's krigsudstyr. En kultur af konstant forbedring opstod. Senere blev Deming medvirkende til at Japan blev løftet op til en første klasses industrination efter 2. verdenskrig. Deming lærte meget fra sin mentor Shewhart i 1930'erne og er stamfaderen til det, vi i dag kender som "Lean". I 80'erne blev Toyotas måde at drive produktion, produktudvikling og supply chain management kendt i den vestlige verden, og dette medførte en stor ændring i hvordan arbejde blev organiseret i årene, der fulgte. Vi kender det som "Lean", men ideerne kommer faktisk fra Deming.



Tom Gilb, en amerikansk software konsulent bosiddende i Norge, tog i 1980'erne skridtet ind i ledelsesverdenen blandt andet med bogen "Principles of Software Engineering Management"⁶ fra 1988. I bogen beskriver Gilb blandt andet en iterativ udviklings- og leveranceprincip kaldet Evo, som stadig danner skole. Hvis Deming er oldefar til det, vi kender som Scrum i dag, så kan man roligt sige at Gilb er bedstefaderen. Han er stadig aktiv som underviser og konsulent.



Hiroataka Takeuchi og Ikujiro Nonaka udgav i 1986 en rapport i Harvard Business Review, den hed "The new, new product development game" ("Det nye, nye produktudviklingspil"). Den beskrev en ny parallel måde at finde løsninger på, som forfatterne sammenlignede med Rugby: De beskrev en proces, som var karakteriseret af selvorganiserende projektteams, overlappende udvikling, implementering og test, konstant og mangesidet læring og "subtil kontrol". De introducerede ordet Scrum, som netop kommer fra Rugby.



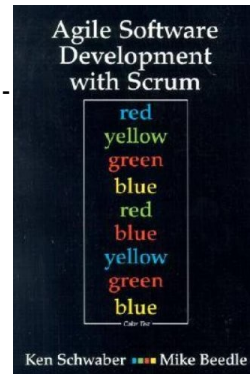
Jeff Sutherland og Ken Schwaber tog denne tankegang op sammen med en påvirkning fra procesteorier og deres "we are on a mission" tænkning fra deres militære baggrund. De, "fædrene", udviklede og formulerede Scrum op gennem 90'erne. Navnet 'Scrum' blev valgt ud fra det Rugbyordvalg som Takeuchi og Nonaka havde introduceret. Schwaber og Sutherland sammenkædede deres forståelse af Scrum med empirisk proceskontrol, hvilket ofte anvendes, når den underliggende mekanisme ikke kan forstås tilstrækkeligt til, at der kan arbejdes efter en opskrift eller en checkliste. Andre strømninger påvir-



⁶ Principles of Software Engineering Management, 1988. ISBN 0-201-19246-2 (19th printing).

kede også konceptet (særligt Extreme Programming, som blev formuleret af Ken Beck i 1997).

Ken Schwaber skrev i 2001 bogen: "Agile software development with Scrum" ("Agil softwareudvikling med Scrum"). Denne bog var tilpas enkel og gjorde for alvor Scrum kendt, den var med til at udbrede konceptet og give den brede accept, Scrum har fået, indenfor IT og software industrien. Siden 2001 er Scrum og andre agile metoder blevet accepteret og anvendt i stigende grad i mange andre situationer. Flere ting er blevet tilpasset i konceptet og nye elementer er også blevet introduceret. Alt dette er dog sket med en yderst pragmatisk tilgang, hvilket er godt; der er noget i standard Scrum, man kan bruge umiddelbart med gode resultater. Scrum er blevet defineret som "*sund fornuft gjort synlig og anvendt systematisk*".



De senere år er **Dave Snowden** og hans mønster for forståelse af systemer og situationer: Cynefin, blevet kendt og har gjort det muligt for os, at give en meget bedre forklaring på, hvorfor vi gør, som vi gør, i Scrum. Scrum har over de sidste år udviklet sig til at være et godt match for arbejde, der befinder sig i det komplekse og det komplicerede domæne. Cynefin forståelsesrammen giver os både redskaber at bruge, i de arbejdssituationer vi befinder os i, og desuden et teoretisk fundament til at forstå det med.

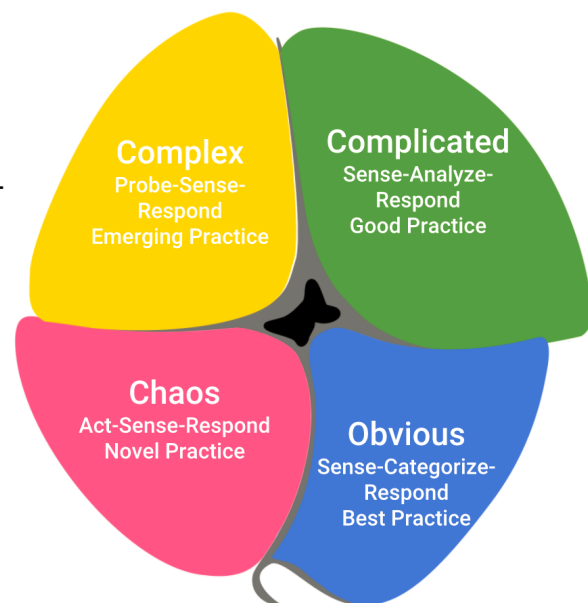


Cynefin Modellen

Cynefin modellen blev udviklet af Dave Snowden & Cynthia Kurtz fra 1999 til 2005. Den bruges til at beskrive problemer, situationer, systemer og ledelse. I Cynefin får man en måde at tænke på, en måde at erkende sammenhænge på ("sense-making"), modellen behandler, hvordan vi angriber forskellige typer af udfordringer.

I Cynefin taler man om fire sammenhænge (domains) og "Disorientering" i midten:

- **Indlysende/Simple sammenhænge (Obvious)**, hvor der er en ordnet og tydelig sammenhæng imellem årsag og virkning. Enhver nogenlunde fornuftig person kan se, hvad der skal gøres. Man "*Fornemmer - Kategoriserer - Reagerer*". Man kan helt fint have en "bedste måde" at løse opgaverne på (Best Practice). Dette er administratørens domæne.
- **Komplicerede sammenhænge (Complicated)**, hvor der er ordnet sammenhæng imellem årsag og virkning, men det er svært. Man må analysere, undersøge eller have ekspertviden før man kan reagere. Man "*Fornemmer - Analyserer - Reagerer*". Der kan være flere gode løsninger afhængig af den ekspertise, man har til rådighed (Good Practice). Dette er ekspertens domæne.
- **Komplekse sammenhænge (Complex)**, hvor sammenhæng imellem årsag og virkning, kun kan forstås delvist eller i tilbageblik, men ikke fuldstændigt på forhånd, viden er ty-



⁷ Extreme Programming Explained, Ken Beck 1999

pisk fragmenteret og ufuldstændig. Mange spillere er involverede og påvirker hele tiden hinanden, systemet og grundlaget. Man “Forsøger - Fornemmer - Reagerer”, man må prøve noget og se resultatet for at få hoved og hale i tingene. Man udfører kontrollerede eksperimenter for at se, hvad der virker, og hvad der ikke gør. Dette kan ikke udtænkes alene ved et skrivebord. Ny kombinationer af løsninger spirer frem (Emerging Practice).

- **Kaotiske sammenhænge (Chaos)**, hvor der ikke kan erkendes nogen overordnet sammenhæng imellem årsag og virkning. Man “Handler - Fornemmer - Reagerer”. Man må gøre noget rimeligt drastisk for at se, om man kan stabilisere noget og få bare ét fast holdpunkt. Man kan opdage helt nye løsninger (Novel Practice).
- **Disorientering (Disorder)**, i midten, er mere en tilstand, hvor man ikke har erkendt, hvor man er - hvilket domæne man er i. I denne tilstand vil folk typisk antage, at de er i det domæne, som de er mest komfortable med og falde tilbage på at træffe beslutninger baseret på fortiden, og hvad man var komfortabel med dengang. Det er selvsagt ret farligt.

Grænsen imellem det Indlysende og det Kaotiske domæne er forskellig fra de andre grænser i modellen, denne grænse beskriver en risiko for katastrofal kollaps: Hvis man oversimplificerer situationen, stoler på sine indøvede procedurer og ”best practices”, stoler på at fortidens succes gør os immune overfor fremtidig fiasko. Så er man rykket ind i den ”**Selv-fede zone**” (Zone of Complacency) og risikerer at styrte ned i kaos i en krise, i situationer med accelereret forandring eller i situationer med stor usikkerhed.

I mange projekter og initiativer har man et stort indhold af opgaver i det Komplekse domæne og burde handle ud fra det, men har ofte ikke erkendt det. Som ledere i dette domæne skal man (ifølge Snowden):

- Forsøge, fornemme, reagere. Man udfører en serie af små eksperimentelle tiltag, hvoraf nogle fejler (det er OK, man lærer af dem), man går ikke efter skud-sikre løsninger.
- Skabe omgivelser og eksperimenter, der afslører sammenhænge og mønstre, genererer ny viden.
- Forøge niveauet af samspil og kommunikation mellem parterne. Det er nødvendigt, at mennesker arbejder sammen på en åben og tillidsfuld måde, de kan ikke tvinges til at dele viden.
- Brug metoder, der hjælper til at generere ideer:
 - Åben diskussionen op (“Large Group Methods”).
 - Sæt grænser og rammer (“Constraints”), for at stimulere ideer.
 - Stimuler de tiltag, der giver ønskværdige resultater, og dæmp de tiltag, der giver uønskede resultater.
 - Giv opmuntring til alternative synspunkter og forskellighed.
 - Styr startbetingelser og holde øje med det, der pibler frem (emergence).

Og nu til Fugleperspektivet

Selvom der er mange forskellige agile fremgangsmåder, vil vi her fokusere på Scrum, da det er den, der primært beskæftiger sig med de overordnede principper på tværs af fagområder. Vi vil parkere diskussionen om de professionelle færdigheder, der er brug for, til det konkrete fagområde. For eksempel kan Teamet i et IT projekt forventes at bruge principperne fra Extreme Programming til at opnå hastighed og kvalitet, i medicalsektoren

har man et sæt andre faglige rammer, der sikrer kvalitet. Vi vil antage at sådanne gode faglige kompetencer er på plads, Scrum forudsætter dette.

Scrum er en veldokumenteret metode, den handler om de rammer og begrænsninger, vi lægger ind over initiativet; den giver os ikke en detaljeret opskrift. I dette kapitel tager vi en tur hen over landskabet for at få et overblik over de vigtigste dele af Scrum metoden.

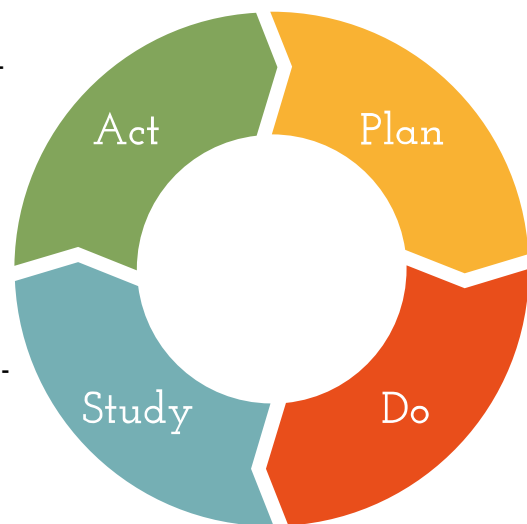
Arbejdet opdeles i små bidder - Sprints

I Scrum erkender vi, at vi ikke ved alt, når vi begynder på et projekt: Prioriteter kan skifte; brugere kan ombestemme sig, teknologi kan opføre sig anderledes, end vi gik ud fra, leverandører kan skuffe os osv. Alt i alt bærer initiativer og projekter alle de karakteristika, der hører til det komplekse domæne.

For at få noget fra hånden deler vi hele projektet op i små bidder af samme tidsmæssige størrelse, disse kalder vi **Sprints** eller iterationer. Normalt er Sprints fra to til fire uger lange⁸. Før hvert Sprint tager vi stilling til: "Hvad er det klogeste vi nu kan bruge vores næste Sprint på?". Det kalder vi Sprint Planning, det mål fokuserer vi så benhårdt på. Så længe Sprintet er i gang, har vi taget noget af kompleksiteten ud af spørgsmålet: "Hvad skal vi gøre?".

Dette er en implementering af PDSA cirklen⁹, som blev introduceret af W. Edwards Deming. Den bruges ofte til kvalitetssikring, men den ligeså gyldig her.

En anden måde at forklare dette er: Vi bruger en lineær tilnærmelse til en ikke-lineær proces. I det komplekse domæne er der ikke-lineært årsags- og virkningssammenhænge, men vi beslutter os for at anse det som lineært i en kort periode. Vi gør også alt, hvad vi kan, for at beskytte Teamet imod ændringer under Sprintet, så vi med rimelig præcision kan forudsige udfaldet.



Princippet med Sprints bliver ind imellem illustreret med det at sejle en ubåd igennem farvand fyldt med forhindringer. "Periskop op", vi tager bestik af situationen, i hvilken retning skal vi sejle i det næste dyk? Læg kursen, periskop ned og så af sted. Efter det fastlagte tidsinterval, så hæver vi periskopet igen og lægger en ny kurs, og sådan bliver vi ved.

Sprintet skaber en ramme i det komplekse rum; indenfor denne selvskabte arena kan Teamet selvorganisere, finde løsninger og komme i flow¹⁰. Baseret på erfaring bør Sprintet være mellem to og fire uger langt. Kortere Sprints tillader ikke rigtigt Teamet at finde løsninger og komme i flow, og der opstår en følelse af, at skulle være færdig inden arbejdet rigtigt er begyndt, det føles som produktion. Længere Sprints har en tendens til at blive lidt for afslappede i forhold til målet i starten; det er let at få en følelse af at Sprint slutningen er meget langt væk, og så vi får en meget uens arbejdshastighed i Sprintet med ringe forudsigelighed, stress og sikkert også ringe kvalitet som resultat.

⁸ Den officielle *Scrum Guide* siger nu at det er OK med Sprints af én uges længde, vi anbefaler det dog ikke.

⁹ Se her: <https://www.deming.org/theman/theories/pdsacycle>

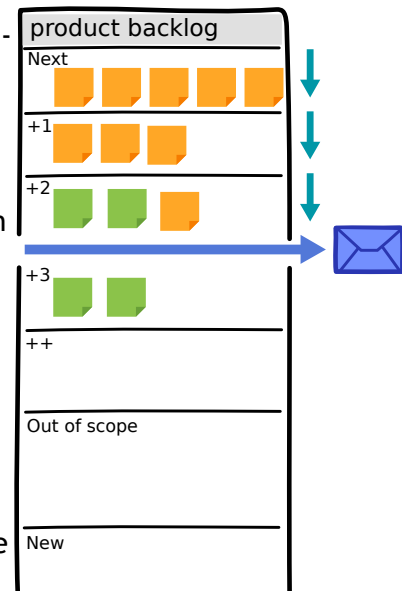
¹⁰ Flow, en psykologisk tilstand, læs mere her: [https://en.wikipedia.org/wiki/Flow_\(psychology\)](https://en.wikipedia.org/wiki/Flow_(psychology))

Alle ønsker og krav samles på en ordnet liste

Alt det, interessenterne gerne vil have leveret i projektet, har brug for en beskrivelse. Detaljer kan ændre sig og nye ønsker kan endda komme til overfladen under projektet.

Vi samler alle disse krav og ønsker i en sorteret liste med de emner, der skal arbejdes på først, øverst. Denne liste kaldes **Product Backlog**. Netop én person har ansvaret for denne liste, han kaldes **Product Owner**. Krav og ønsker beskrives i en tilpas ensartet form, så de kan sammenlignes, holdes op mod hinanden, prioriteres og derved placeres på det rigtige sted på Product Backloggen. De kaldes **Product Backlog Items**.

Product Backlog Items er ofte formuleret som **User Stories** (Bruger Fortællinger)¹¹, narrative af lignende type som de, der bruges i Cynefin. Teamet vurderer selv *arbejds mængden* (effort) for hvert Product Backlog Item, andre udenfor teamet (typisk interessenter eller brugere) vil vurdere *værdien* af hvert Item. Nogle af disse Items vil være komplekse, nogle komplicerede og andre indlysende, forhåbentlig ingen kaotiske.



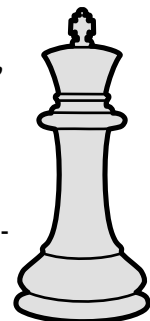
Product Backloggen er også en bevidst valgt ramme i projektet. Alt, hvad Teamet skal bruge tid på, skal ind over denne liste, nogle emner vil være eksperimenter, nogle vil være undersøgelser, nogle vil producere brugbare leverancer, og der vil sikkert også være nogle relativt uvedkommende ting, som bare skal gøres. Listen er offentlig tilgængelig, alle interessenter kan se den og ved, hvor de skal henvende sig, hvis de mener, den burde prioriteres anderledes eller indeholde mere eller mindre, nemlig til Product Owneren.

Alle har entydige roller

Den næste afgrænsning, der er defineret i Scrum, er de roller, som deltagerne i projektet har. De, der har påtaget sig en forpligtigelse for at skabe resultater gennem projektet, kaldes samlet **Hele Scrum Teamet**. De består af **Product Owneren**, **Development teamet** (Teamet) og **Scrum Masteren**.

Produkt Owneren

Product Owneren (ofte betegnet Produktejeren i Skandinavien) har ansvaret for, at der kommer det bedst mulige resultat ud af projektet. Han har derfor fokus på den værdi, der skabes. For hvert Produktemne estimeres denne værdi (Business Value). Han har også ansvaret for investeringsafkastet (Return on Investment - ROI) dvs. i hvilken rækkefølge skal Product Backlog Items leveres, så der skabes værdi tidligt. Han udfører dette hverv ved at prioritere, hvad der skal arbejdes på, hvornår.



Product Owneren forstår kundens og brugernes domæner og tager også ansvar for at forklare og facilitere dette domænekendskab til Teamet, der arbejder på leverancerne, på Product Backlog Items. Teamet har lige præcis én person at spørge om, "hvad" der skal skabes i produktet og "hvornår" - nemlig Product Owneren.

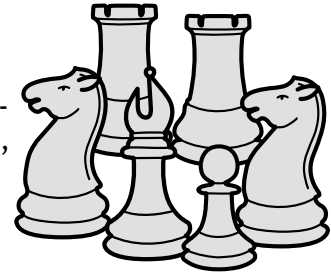
Product Owneren opererer indenfor det **strategiske** område og navigerer domænet med komplekse og komplicerede Product Backlog Items og han planlægger de **taktiske** Sprints sammen med Teamet. Product Owneren arbejder ustandseligt med Product Backloggen,

¹¹ Selvom Scrum ikke forlanger dette format, er det en god praksis.

forbedrer beskrivelser og acceptanskriterier, indsamler nye idéer, lytter til signaler fra brugere, re-prioriterer osv.

Development Teamet (Teamet)

Et lille Team (5-9 personer¹²) tager ansvar for at udføre arbejdet i projektet; vi kalder dette Udviklingsteamet eller forkortet - Teamet. Teamet er selvorganiserende, tværfunktionelt (Cross Functional) og har ideelt alle de nødvendige færdigheder og kompetencer, der er brug for, for at producere de ting, der er beskrevet i Product Backloggen.



Ideelt skal der ikke være nogen afhængigheder til nogen eller noget uden for teamet. Ideelt har Teamet ikke andre arbejdsopgaver end de, der fremgår af Product Backloggen og senere Sprint Backloggen.

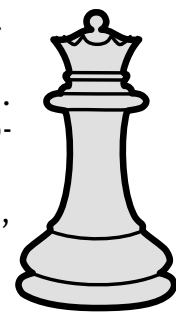
Teamet har autoriteten til at finde den rigtige løsning på, hvordan Product Backlog Items skal laves, bygges eller konstrueres. De har den nødvendige autoritet og ressourcerne til at tage dette ansvar. Teamet vælger i dialog med og baseret på Product Ownerens prioritering, det sæt Items, de kan forpligte sig på, i det kommende Sprint. Derefter blander ingen sig principielt i Teamets arbejde i Sprintet, men Teamet har lov og pligt til at søge hjælp under Sprintet. De har også pligt til at sørge for synlighed af deres fremdrift og de forhindringer, de støder på.

Teamet arbejder sammen med Product Owneren om at få den bedst mulige forståelse af Product Backlog Items, såsom at finde acceptanskriterier. Teamet estimerer arbejdsmængden for hvert Product Backlog Item. Teamet hjælper også Produktejeren med at bryde Produktemnerne ned i passende, håndterbare størrelser, der kan løses i et Sprint.

Scrum Masteren

Der er netop én person, der tager ansvaret for at alle følger Scrum principperne, han kaldes **Scrum Masteren**. Han er også ansvarlig for at implementere forbedringer og for at fjerne forhindringer for både Team og Product Owneren. Han er slet ikke den gængse projektleder eller værkfører, der deler arbejdsopgaver ud; i stedet er han en tjenende leder (Servant Leader).

Scrum Masteren er fokuseret på at nå frem til en kultur af konstant forbedring, så han er nødt til at være dybt overbevist om værdien af Scrum. Ofte er det ham, der arbejder med den omgivende organisation og forklarer hvorfor og hvordan, Scrum bruges, og hvad værdien for alle er.



Scrum Masteren skaber værdi i projektet ved altid at observere og lytte efter svage signaler, der enten indikerer muligheder eller potentiel fare. Han er på en måde den, der er mest udsat for det komplekse domæne, da han lever og ånder i det og sørger for at hele Scrum Teamet ikke glider over i en misforstået opfattelse af at verden er enkel, hvad den absolut sjældent er. Han forsøger at skabe plads i Sprintet - i tid og rum - for at tillade Teamet at finde og skabe løsninger; dette gøres delvist ved at beskytte Teamet fra konstante ændringer og afbrydelser fra omgivelserne.

Scrum Masteren er som træneren for et elite sportshold, dirigenten for et orkester eller fårehunden, der beskytter Teamet. Altid tjener og leder han Teamet og Produktejeren.

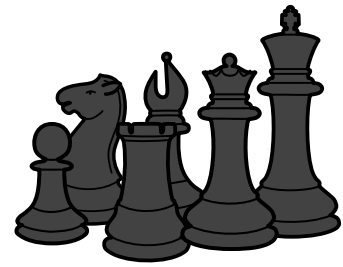
¹² The Scrum Guide skriver nu at man kan gå ned på 3 personer, det er OK, men man mister noget dynamik.

Resten af organisationen

Hele Scrum Teamet spiller sammen med interessenter og med ledelsen. Interessenter vil typisk inkludere kunder, som på en eller anden måde betaler for leverancerne, og brugere, som er forbrugere af det, der bliver produceret.

Ledelsen er involveret i at stille de nødvendige redskaber og ressourcer til rådighed til Scrum Teamet og i at fjerne forhindringer, ofte på Scrum Masterens initiativ.

Alle har lov til sætte nye Produktemner på Produktlisten, se på de offentlige tavler og deltage i de offentlige Scrum aktiviteter.



Scrum aktiviteter, møder eller ceremoni

Scrum definerer en række aktiviteter eller møder, der hver tjener sit specifikke formål og igen tilsammen udgør nogle af rammerne for, hvordan projektet afvikles, så alle er tro mod processen, holdes årvågne og holder udvig efter alle indikationer af forandringer, der måtte være på vej. På denne måde er der en fast rytme i et Scrum projekt, dette hjælper alle med at finde ind i nogle gode vaner og derigennem udvikle en kultur, der passer til det komplekse domæne, hvor hyppige inspektioner er nødvendige.

Der er et par grunde til at møder i Scrum er låste i tid, **Tidssatte** (time-boxed). Uden en tidsafgrænsning har møder en tendens til kunne fortsætte i det uendelige og blive meget uproduktive. Dette lette tidspres med en fast tid kan hjælpe med at få resultater frem og tillader folk hurtigt at kunne komme videre med deres egentlige arbejde. Alle deltagere ved så, hvad de kan forvente og kan planlægge andre aktiviteter efter det. Der skal være denne underliggende følelse af, at dette haster lidt.

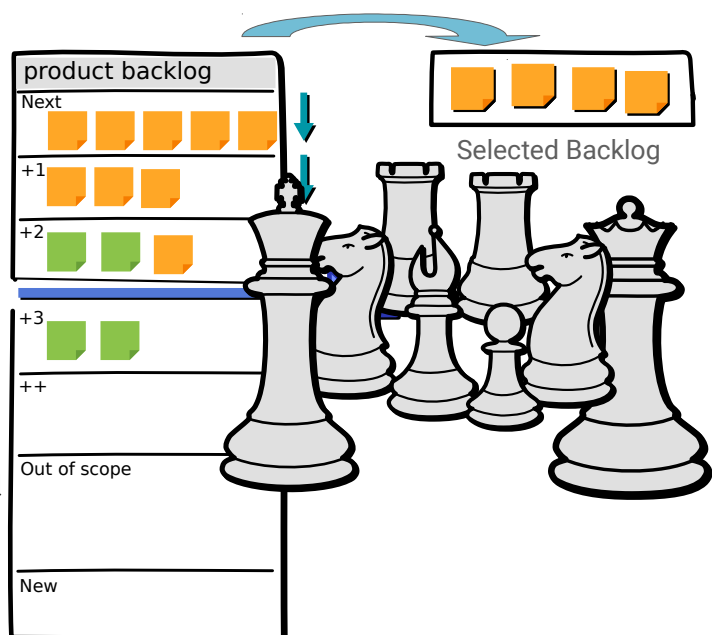
Scrum Masteren er tilstede ved alle møder og faciliterer disse.

Sprint Planning 1

Ved begyndelsen af hvert Sprint mødes Product Owneren og Teamet for at udvælge det optimale sæt af Produktemner, der skal arbejdes på i Sprintet. Dette er et Tidssat møde: Ca. En time for hver uge i Sprintet og kaldes **Sprint Planning 1**.

Før mødet har Product Owneren ordnet og prioriteret Produkt Backloggen, og de relevante emner er blevet estimeret af Teamet. Han har også formuleret et forslag til et **Sprint mål** - en enkel sætning, der beskriver, resultat af Sprintet. Teamet har inden mødet forholdt sig til deres egen kapacitet og har et bud på deres **Arbejdshastighed** (Velocity). Er der f.eks. nogen på ferie, eller er der andre grunde til, at Teamets Arbejdshastighed skulle være anderledes end forventet. De har altså gjort sig klart, hvad de regner med at kunne yde.

Product Owneren gennemgår mundtligt de enkelte relevante Items for Teamet. Om nødvendigt har Product Owneren medbragt andre folk, der kan svare på uddybende spørgsmål.



Teamet stiller afklarende spørgsmål og viden opbygges, afhængigheder imellem Items eller Team medlemmers kompetencer afdækkes. Måske ændrer estimer og prioritering sig, måske må man kontakte andre for at få svar.

Til sidst er der fundet et sæt Product Backlog Items, som alle kan acceptere som det bedste, der kan opnås i det kommende Sprint. Teamet er overbeviste om, at de kan klare det, der er aftalt. Ved afslutningen af dette møde krydses der over fra det strategiske til det taktiske område - fra "hvad der skal laves" til "hvordan skal det laves".

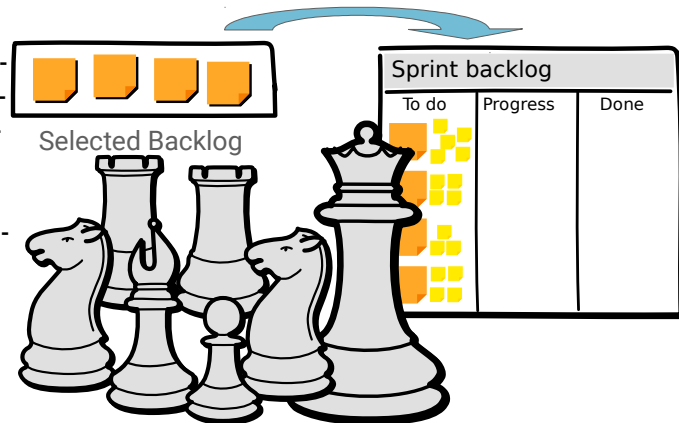
Sprint Planning 2

Direkte derefter fortsætter Teamet alene med anden halvdel af Sprint Planningen - **Sprint Planning 2**. Product Owneren kan tilkaldes, hvis der skulle dukke spørgsmål op. Nogle Teams vælger at have Product Owneren med også til dette møde. Mødet er Tidssat til omtrent samme længde som Sprint Planning 1.

Teamet færdiggør nu analyse og design af hvert enkelt Item, så de ved i nødvendige og tilstrækkelige detaljer, hvordan de vil gennemføre leverancen. Typisk resulterer dette i, at de nedbryder hver enkelt Item i en række **Tasks** (Arbejdsopgaver). Princippet er, at en Task kun skal være i gang én dag - så skal den være færdig, flere må godt arbejde på den.

Hvis nye spørgsmål dukker op, tilkaldes Product Owneren, som så uddyber eller finder svar. Teamet har jo set på tingene før, både under **Product Backlog Refinement** og under Sprint Planning 1. Hvis ny viden dukker op, kan sættet af udvalgte Items selvfølgelig ændre sig.

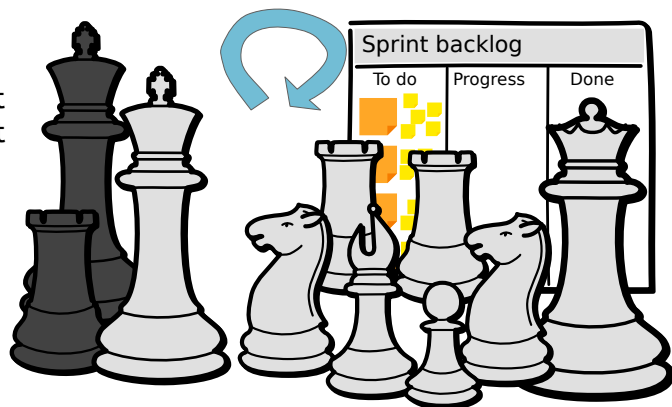
Til slut har Teamet en plan, ikke en perfekt plan, men nok til at komme i gang. Denne plan kaldes **Sprint Backloggen** (Sprint planen). Teamet offentliggør denne, og det er deres bedste bud på, hvad de kan nå - de siger "Efter vores bedste overbevisning kan vi nå dette!", de siger også: "Vi vil gøre vores bedste for at nå det!". Vi kalder dette Teamets Sprint Mål eller Commitment¹³.



Daily Scrum

Hver dag i Sprintet mødes Teamet og Scrum Masteren samme tid og sted for at følge op på fremdriften frem imod Sprint målet, for at synkronisere de taktiske aktiviteter og planlægge næste dag. Dette kaldes **Daily Scrum**. Alle (inklusive Scrum Masteren) besvarer tre spørgsmål:

- Hvad har jeg opnået siden i går?
- Hvad regner jeg med at opnå til i morgen?
- Er der nogle forhindringer, der generer mit optimale arbejde?



¹³ I den nyeste version af *The Scrum Guide* anvendes udtrykket "forecast" istedet for "commitment", for at det ikke skal lyde for meget som et helt fast løft. Vi synes omvendt at "forecast" lyder lidt for meget som vejrudsigt: "Måske sker det, måske ikke, jeg kan ikke gøre noget ved det!", det er lidt for slapt.

Daily Scrum er tidssat til 15 minutter og er offentlig; alle må gerne komme, det er en del af synligheden, men de taler ikke. Hvis de har kommentarer eller spørgsmål, henvender de sig til Scrum Masteren efter Daily Scrum.

Under Daily Scrum kan det være, at Teamet og Scrum Masteren opdager et behov for yderligere diskussion; i det tilfælde bliver andre småmøder hurtigt aftalt. Der bliver ikke diskuteret detaljer eller løsninger på Daglig Scrum.

Nye forhindringer eller muligheder for forbedringer, der dukker op, bliver noteret på Scrum Masterens arbejdsplan: Improvement Backloggen. Umiddelbart efter Daily Scrum tager Scrum Masteren fat på tingene.

Scrum Alliance anbefaler at Product Owneren deltager i Daglig Scrum og taler, nogle Teams finder dette at være en hjælp.

Sprint Review

Ved afslutningen af Sprintet afholdes der **Sprint Review**, hvor Teamet viser resultaterne - de afsluttede Product Backlog Items - frem. Kun de ting, der er helt afsluttede demonstreres. Alle med interesse er velkomne og opfordres til at komme med tilbagemeldinger på det, de ser.

Product Owneren er naturligvis til stede og godkender formelt efterfølgende de afsluttede Items.

Hvis det skulle ske, at noget ikke godkendes eller skal ændres, sættes et nyt Item på Product Backloggen. Product Owneren noterer sig alle nye ideer og kommentarer fra publikum.

Sprint Review er Tidssat ligesom alle andre Scrum møder, samme længde som Sprint Planning møderne; ofte holdes det dog lidt kort (f.eks. 90 min) for at tilpasse sig andre folk i organisationen og deres travle hverdag. Dette møde afslutter PDSA cirklen med hensyn til det **produkt**, der bliver leveret. Mødet giver også mening i lyset af Cynefin, da det er en invitation til mange med forskellig baggrund om at give feedback - ikke kun eksperterne. Det er vigtigt at holde sig "review" delen for øje, det er ikke kun en fremvisning, men en aktivitet, der genererer konstruktiv feedback.



Sprint Retrospective

Til allersidst i Sprintet mødes hele Scrum Teamet for at dele erfaringer fra det nyligt overståede Sprint og se, om der er nogle ideer til yderligere forbedringer af arbejdsmetoder og processer. Dette kaldes **Sprint Retrospective**.

Scrum Alliance og Scrum Guide anbefaler at Produktejeren deltager i dette møde. Nogle Teams finder at det kan være nødvendigt i starten bare at holde mødet med Teamet og Scrum Masteren til stede, hvis der endnu ikke er opnået den fornødne tillid og åbenhed til at kunne diskutere frit.

Scrum Alliance siger at mødet er Tidssat som de andre møder, Scrum Guide siger 3 timer for 4 ugers Sprints. Mange Teams finder det optimalt med et kortere møde - max. en time.



Der er mange måder hvorpå hele Scrum Teamet kan vælge at gennemføre Retrospektive på. Det, der er vigtigt at opnå, er, at:

- Alle deler de faktuelle erfaringer fra Sprintet. Især fokuseres der på det uventede, det der ikke gik som planlagt. Her er der ofte læring at hente.
- Alle reflekterer over betydningen af disse erfaringer og beslutter, hvad der er positivt, og hvad der kunne forbedres.
- Der prioriteres, hvad hele Teamet vil satse på at forbedre. Hvad kan Teamet gøre, hvad kan Product Owneren gøre og hvad Scrum Masteren må gøre. Der tages en beslutning om at gøre det.

Dette møde afslutter PDSA cirklen i forhold til, hvordan arbejdet bliver udført, *processen*. Mødet er også i overensstemmelse med måden at tænke på i Cynefin. Vi anerkender, at vores arbejde typisk er mere komplekst end et kompliceret ekspertdomæne. Ved at lade de involverede selv vurdere samspillet, opnås større indsigt.

Product Backlog Refinement

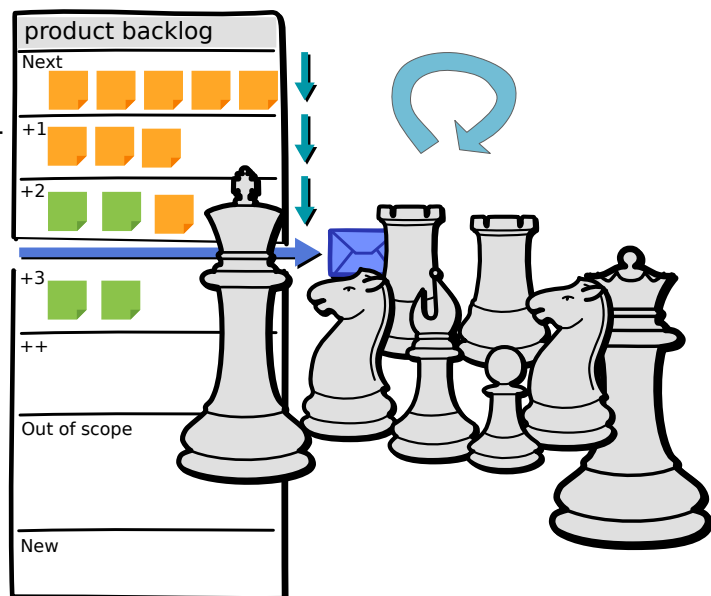
Der er endnu en vigtig aktivitet, den kaldes **Product Backlog Refinement**. Det er nødvendigt at Product Backloggen bliver gennemarbejdet, forstået og forberedt og at relevante Items bliver estimeret før Product Owneren kan færdiggøre sin prioritering og gå ind i Sprint Planningen. For at sikre sig at dette faktisk sker, afholdes der et eller flere møder imellem Product Owneren og Teamet, typisk i midten af det nuværende Sprint - med fokus på det næste Sprint.

Mødet er en forberedelse til Sprint Planning 1, så formen er næsten ens. Product Owneren og Teamet er til stede. Product Owneren gennemgår de Items, han allerede har udvalgt som forslag til næste Sprint og lidt mere, sammen med nye Items og Items, hvor omstændighederne eller kravene har ændret sig. Der er igen dialog og en fælles forståelse bygges op, afhængigheder og acceptanskriterier findes.

Deltagerne bryder ofte store Items (ofte kaldet **Epics**) ned i mere grydeklare størrelser, raffinerer beskrivelser, fanger acceptanskriterier og estimerer omkostninger og størrelsen af arbejdet.

Forskellen på et Product Backlog Refinement og Sprint Planlægning 1 er, at i stedet for at udvælge Items bliver disse blot raffineret og estimeret i forhold til omkostninger (typisk i form af arbejde). Det er vigtigt at estimeringen udføres af de folk, der i sidste ende skal levere - nemlig Teamet.

Product Backlog Refinement er til stor gavn for Productowneren, da det hjælper ham med at forbedre prioriteringen. Aktiviteten giver ham det bedst mulige overblik over sammenhængene og et omkostningsestimat. Bagefter har han stadig tid til at forberede prioriteringen af emner til næste Sprint frem til Sprint Planning 1.



Artefakter og Informationsformidling

Artefakter er "vidnesbyrd om menneskelige aktiviteter" og bruges som begreb i arkæologien. I mangel af et bedre ord bruger vi også dette i Scrum som betegnelse for de lister, tavler, grafer og andet, som vi bruger til at styre og synliggøre hele Scrum processen med, de er "Information Radiators".

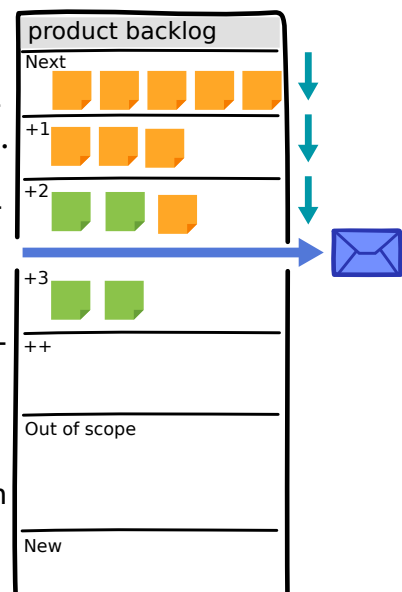
Når man bruger empirisk proceskontrol som i Scrum har man brug for tre søjler: **Synlighed, inspektion og korrektion**. Synligheden opnår vi ved at lade bestemte aktiviteter være offentlige og ved at bruge Artefakter til at "udstråle" information. Disse Artefakter skal være nemme at forstå og selvdokumenterende, så selv den, der kun ser på dem engang imellem, kan uddrage meningen.

I Scrum bruger vi flere af disse Artefakter. Med et Cynefin udtryk hjælper disse os med "forståelses-processen" (the "sense-making" process), de hjælper os med at se, hvad vores observationer betyder.

Product Backlog

Product Backloggen (ofte kaldes den bare Backloggen) er en sorteret og ordnet liste af **Product Backlog Items** - individuelle småleverancer, som ønskes, men som ikke er leveret endnu. De Items, der er nær toppen, er mere ønskværdige at få begyndt på, end de, der er længere nede, set fra Product Owners synspunkt. Alle må gerne bidrage med Items til Product Backloggen, men Product Owneren er den eneste, der flytter rundt på dem på Backloggen. På ethvert tidspunkt udtrykker Backloggen Product Ownerens aktuelle plan; han ejer Backloggen.

Backloggen afbilledes ofte med sektioner, der viser hvornår forskellige releases eller faser finder sted og de planlagte Sprints, der ligger indenfor disse. Der er typisk også en sektion til u-prioriterede Emner (de ting, som Product Owneren ikke tror Teamet klarer inden for den afsatte tids- og omkostningsramme for initiativet) og en til nye idéer eller krav.



Product Owneren arbejder konstant på at forbedre og prioritere Backloggen. Nogle gange får han hjælp fra Teamet til dette, bl.a. gennem aktiviteten Product Backlog Refinement, nogle gange har Product Owneren også andre i organisationen til at hjælpe sig, det er helt OK.

Hvert Product Backlog Item indeholder nok information til at gøre det muligt for Product Owneren at planlægge og for Teamet at implementere. Det er meningen at informationen skal være overskrifter og påmindelser om at huske at få snakket om detaljer. Product Backlog Items indeholder ofte information så som:

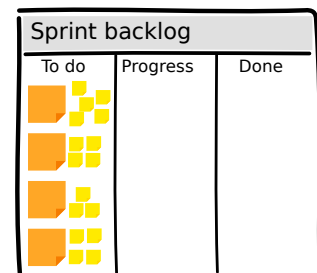
- Et ID, typisk alfanumerisk (nogen kan bestemt ikke lide dette).
- En titel eller navn
- En markering af hvilket hovedområde, ofte kaldet Tema (Theme), dette Item hører under.
- Et estimat af størrelsen af arbejdet, ofte med en usikkerhedsangivelse i form af et 50% og 90% estimat.
- Et estimat af værdien, ofte med en tilsvarende usikkerhedsangivelse.

- Et estimat af de direkte omkostninger.
- Hvor ønsket kom fra og hvornår.
- Reference til hvor evt. uddybende information kan findes.
- Eventuel estimat af Cynefin og Kano vurdering¹⁴.
- Beskrivelse af ønsket formuleret som Bruger Fortællinger (User Stories): *Som en sådan-og-sådan* bruger, vil jeg gerne kunne *gøre det-og-det*, for at kunne opnå *jada-jada-jada*.
- Ekstra information om detaljerede specifikationer, typisk formuleret som acceptanskriterier: *"I den-og-den-situation, hvis denne-hændelse opstår, så skal resultatet være sådan-og-sådan!"*

Product Backloggen viser den nuværende plan og prioritering for alle, der har interesse i projektet. Den kan være en fysisk tavle med post-it lapper eller kort med magneter på, et regneark eller et dertil dedikeret IT system.

Sprint Backlog

Når Teamet har detailplanlagt Sprintet under Sprintplanlægning 2, så bliver dette præsenteret på **Sprint Backloggen**, som viser de udvalgte Product Backlog Items og typisk de Tasks, de er nedbrudt i. Sprint Backloggen viser løbende en status for hver Task, så alle kan se fremdriften. Teamet forholder sig løbende til, om de kan nå at færdiggøre alt det, de forpligtede sig på i Sprintet.



Sprint Backloggen er typisk delt op i forskellige kolonner, som Arbejdsopgaverne kan flyttes rundt i. Typisk har man som minimum en kategori, der hedder "Ventende", en der hedder "Igangværende" en "Færdig" og en "Blokeret", som betyder at Teamet venter på noget udenfor Teamet. Arbejdet er typisk brudt ned til en sådan størrelse, at hver Task kun forventes at være "Igangværende" én dag.

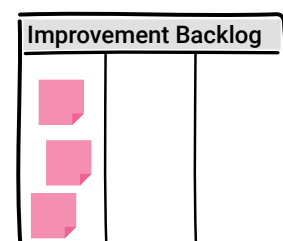
Det er god praksis også at registrere enhver "Uplanlagt" aktivitet, da disse aktiviteter ofte er den største forhindring for Teamets fremdrift og konstante forbedring. Blot det at gøre dem synlige, får ofte organisationen til at reagere på dem. Disse bliver også ofte diskuteret på Sprint Retrospective.

Sprint Backloggen kan være en fysisk tavle i Teamets lokale eller et regneark, en liste eller en dedikeret applikation i et IT system. Det sidste kan være nødvendigt, hvis Teamet ikke sidder sammen, men en simpel fysisk tavle foretrækkes af langt de fleste Teams, den giver konstant og øjeblikkelig synlighed.

Sprint Backloggen ejes og vedligeholdes af Teamet, og det er sammen med Daily Scrum den måde, Teamet viser den omgivende organisation deres fremdrift. Teamet opdaterer status på Sprint Backloggen mindst en gang om dagen sådan at status ved afslutning af Daily Scrum er korrekt. På denne måde kan alle kan følge med i fremdriften, og der skabes tillid til Teamet og projektet.

Improvement Backlog

Det er god praksis, men ikke et krav, at have en **Improvement Backlog**. Den minder om Sprint Backloggen, men i stedet for at vise det arbejde, der har med produktet at gøre, viser den de ting, Scrum Masteren arbejder på for at forbedre arbejdsbetingelserne for Teamet og Produktejeren, så de kan nå op på deres maksimale potentiale.



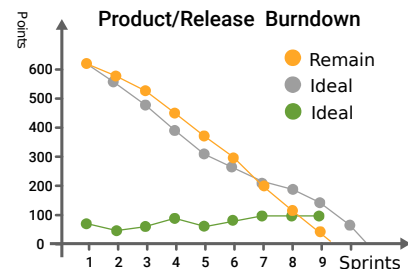
¹⁴ Se her: https://en.wikipedia.org/wiki/Kano_model

Improvement Backloggen er et følsomt redskab. Alene det, at den er offentlig, provokerer til handling: Det at gøre forhindringer tydelige begynder ofte processen med at få dem fjernet. Ingen ønsker at se deres navn som et problem på listen. Det kan dog også fremprovokere modstand og konflikt. Scrum Masteren er nødt til at have fingeren på pulsen, med hensyn til hvordan han bedst håndterer situationerne.

Product Burndown

Ledelsen og Product Owneren ønsker ofte at monitorere fremdriften af hele initiativet, en fase eller en release, for at kunne forudsige, hvornår det er færdigt, eller alternativt for at kunne se, hvor meget der kan nå indenfor en given tidsmæssig ramme.

I Scrum bruger man ofte (men det er ikke et krav) en graf, kaldet **Product Burndown**, den viser hvor meget arbejde, der er tilbage efter hvert Sprint, ud fra estimerne på Product Backloggen.



Ud fra disse registreringer kan Teamets **Velocity** (Arbejdshastighed) også udledes: Hvor meget kan Teamet normalt gennemføre på et Sprint? Hvis vi indfører ændringer til Teamets arbejdsmiljø, er det vigtigt også at monitorere om dette egentlig medfører en forøget Velocity eller ikke.

Ud fra et kvalificeret estimat af Teamets Velocity kan der laves forudsigelser om, hvornår produktet er færdigt, ofte angivet som som et interval.

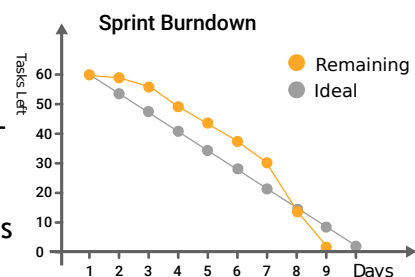
Det er også muligt på Product Burndown at registrere hvor meget Værdi, der er høstet. Hvor meget risiko og usikkerhed, der er på estimerne osv.

Det er meget almindeligt at plote nye Items ind, som er blevet lagt til Product Backloggen ("Scope creep"), eller Items, der er fjernet fra Backloggen på Product Burndown. På samme måde skal det noteres på Product Burndown, hvis større re-estimeringer har ændret perspektivet. Dette må aldrig blandes sammen med Teamets Velocity.

Nogen gange dækker denne graf kun et enkelt release eller en enkelt fase, så kaldes den **Release Burndown** eller **Fase Burndown**.

Sprint Burndown

Hvis der arbejdes med lange Sprints (såsom fire-ugers Sprints), hvis Teamet er stort eller den generelle kompleksitet i Items er stor, kan det ofte hjælpe på overblikket at have en graf over **Sprint Burndown**. Hvis Teamet ikke er tryk ved det overblik, som Sprint Backloggen giver, eller hvis de ofte fejler deres mål, så kan det være på sin plads at bruge tid på en Sprint Burndown, men det er ikke et krav at have en sådan.



Sprint Burndown viser simpelthen, hvor mange Arbejdsopgaver, der er tilbage (ikke afsluttede) efter hver dag i Sprintet. Sprint Burndown kan bruges til at sige noget om "forventet ankomsttid", og hvis dette er efter det tidspunkt, hvor Sprintet slutter, så må der gøres noget.

Det er almindeligt at registrere nye uventede opgaver, der bliver lagt til Sprintet eller bliver fjernet fra Sprintet, for at undgå at forplumre fremdriften med uplanlagte hændelser.

At arbejde i Sprints

Hvordan er det så at arbejde i Sprints? Hvad sker der på en normal dag?

Teamet selvorganiserer med fokus på at få flyttet så mange Items så hurtigt som muligt over til "Færdig". Når først der er begyndt på et Item, så er fokus på at få gjort det færdigt; igangværende arbejde bør begrænses. Teammedlemmerne hjælper hinanden, nogle bytter midlertidigt roller for at assistere andre, alle gør brug af sund fornuft. Sprint Backloggen bliver konstant opdateret, når Tasks eller Items skifter status. Daily Scrum afholdes for at synkronisere og holde sammen på Teamet. De fokuserer på, hvordan de skal få de Items, de har forpligtiget sig på, færdiggjort og nå Sprintmålet.

Scrum Masteren har travlt med at få fjernet forhindringer og at implementere de planlagte forbedringer i arbejdsforholdene. Scrum Masteren beskytter Teamet mod afbrydelser og fungerer som stødpude, mens Teamet er i Sprint. Al kontakt udefra går til ham, når Teamet er i Sprint. Han hjælper også Product Owneren og forsøger at fjerne ting, der sinker eller blokerer dennes vigtige arbejde med Product Backloggen.

Product Owneren assisterer Teamet i dets taktiske udførelse og besvarer spørgsmål under Sprintet, når Teamet spørger. Imens har han travlt med at forbedre Product Backloggen, prioritere og finde frem til Værdi af Backlog Items. Product Owneren arbejder også på ikke kun at præcisere specifikationer men også acceptens kriterier, risiko og andre forhold, der muligvis kan påvirke prioriteringen for det næste Sprint. Product Owneren planlægger frem imod det næste Sprint og gennemfører Product Backlog Refinement sammen med Teamet, det er også her han får estimater udført af Teamet. Han fokuserer på den langsigtede planlægning og overlader det nuværende taktiske Sprint til Teamet.

Alle andre med interesse for projektet følger fremgangen ved at holde sig opdateret på Artefakterne og ved at dukke op til Daily Scrum. Hvis de har nogle kommentarer til Product Backloggen, nye idéer eller input til værdifastsættelse, estimater eller specifikationer går de til Product Owneren med dette. Hvis de har kommentarer til udførelsen af det nuværende Sprint, går de til Scrum Masteren, som - hvis det er nødvendigt - tager det videre til Teamet eller arrangerer et møde.

Uplanlagt arbejde, som ikke kan udskydes til senere Sprints er af det onde, men findes i det virkelige liv. Dette bryder planen. Normalt forsøger Scrum Masteren at bremse uplanlagt arbejde, men noget må fra tid til anden accepteres. Teamet tager hånd om dette og registrerer det som uplanlagt arbejde på Sprint Backloggen, hvor alle kan se det.

Skulle Teamet nå til den konklusion under Sprintet, at de ikke kan færdiggøre alt det, de planlagde i Sprintet, undersøger de først, om der er alternative, lettere måder at opnå det, de har sat sig for. Hvis dette ikke er muligt, kontakter de Product Owneren med det samme for at diskutere hvilken af de tilbageværende Items, der muligvis kunne fjernes fra Sprint Backloggen, uden at det gør alt for ondt. Dette kan ske, hvis uplanlagt arbejde rammer Teamet, eller hvis det går op for dem, at nogle ting er sværere, end de regnede med under Sprint Planningen. Selvfølgelig bør Teamet også kontakte Product Owneren for at få flere Items, skulle den - mindre sandsynlige - situation opstå, hvor de finder ud af, at de bliver færdige med de planlagte Items før tid.

Hvis Sprintmålet skulle blive irrelevant kan Product Owneren afbryde Sprintet. Det kunne ske, hvis den kunde man arbejder for går konkurs eller Teamet bliver ramt af så mange forstyrrelser eller ændringer så det oprindelig Sprintmål ikke tilnærmelsesvis kan opfyldes. Dette sker meget sjældent og bør selvfølgelig undgås, da det som regel skaber en masse konflikter.

Til slut er der et resultat af Sprintet. Det skal være et ægte skridt på vejen eller et tillæg til det resultat, der ønskes for Initiativet. Kun "helt færdige" Product Backlog Items tæller med i resultatet.

Definition af "helt færdig"

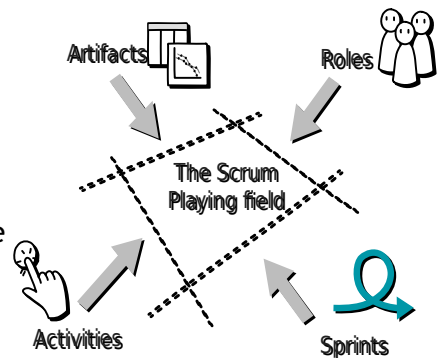
Når et Product Backlog Item erklæres "helt færdig" må alle involverede være enige om, hvad det betyder. Som et absolut minimum, skal det betyde at Teamet ikke sidder inde med viden om, at der er arbejdsopgaver skubbet under gulvtæppet til senere. Det er god praksis at være meget specifik om, hvilke kvaliteter, der må være i et "helt færdig" Product Backlog Item, dette inkluderer standarder for arbejdets udførelse, forskellige spilleregler, der skal overholdes og verifikation, der skal være gennemført.

Denne definition af helt færdig kan komme fra organisationen, der har sat forskellige standarder. Det er typisk at Teamet kommer med forskellige forbedringer og tillæg til denne definition for at blive bedre, dette bliver ofte diskuteret på Sprint Retrospective

Opsummering

Scrum metoden eller mønstret sætter nogle rammer i de komplekse omgivelser for et projekt. Dette hjælper os med at få hoved og hale på alle de faktorer, der spiller ind, få udført noget arbejde og opnå resultater.

- Vi arbejder i Sprints, observerer hele tiden for at blive på kursen og hele tiden forbedre produktet og processen.
- Vi leverer hyppige helt færdige småleverancer for at få tilbagemelding og være sikre på, at vi leverer de rigtige ting.
- Vi har en tydelig ansvarsfordeling, så enhver ved, hvad der forventes af dem, og enhver ved, hvor man skal gå hen med et givet problem.
- Vi fokuserer på synlighed, synlighed og atter synlighed. Dette øger ansvarligheden og giver hurtig indsigt, så nødvendige kursændringer kan gennemføres hurtigt.
- Vi er hele tiden fokuseret på at levere værdi, på de resultater, der gavner kunden, og vi er hele tiden fokuseret på konstant forbedring.



Appendices

Engelsk Definition of termer

Term-EN	Abbr.	Description
Acceptance Criteria		A description of the Done criteria for a User Story, for example using the following template: " Given a--certain-context and some-more-context. When an-event Then an-outcome and another-outcome."
Acceptance Test		The activity of verifying that a Product Backlog Item has been implemented correctly.
Artifacts		The tools used to visualize what to deliver, prioritization, tasks to do, impediments and progress. The term normally refers to the Product Backlog, the Sprint Backlog and the Impediment Backlog, but can also refer to the Product Burn-down and the Sprint Burn-down, if these are used.
Backlog Refinement		A meeting, typically in the middle of a Sprint, where the Team and the Product Owner get together in order to refine and estimate in detail relevant Product Backlog Items, which the Product Owner then in turn considers as candidates for the next Sprint.
Business Value		The quantified value that the Product Owner (on behalf of the Customer) estimates to reap from getting a particular Product Backlog Item.
Constraints		Conditions for the project that are defined by the Product Owner (on behalf of the Customer). Any solution that is developed during the course of the project must be within the definition of the collection of constraints. This often includes performance, platforms, usability, documentation etc.
Cross-functional Team		A team where the members have different skills and together possess all the necessary skills and competences to complete the Product Backlog Items in the Project
Customer		A secondary role in Scrum: the person who purchases the Project and pays for it. The Customer naturally wants maximum value and return on investment.
Daily Scrum		A daily, 15 minute, standup meeting held in the same place and on the same time everyday, where the Team members and the Scrum Master synchronize. Each person answers the three questions: what have you accomplished since yesterday? What do you plan to accomplish until tomorrow? Are there any impediments preventing you from working optimally?
Delivery-In		A milestone in the Time-line, specifying that at this time we plan to receive certain deliveries IN to the project, for example from other teams or suppliers.
Delivery-Out		A milestone in the Time-line, specifying that at this time we plan with certain deliveries OUT of the project, for example to other teams or to suppliers.
Development Team		The small cross-functional team that actually produces or delivers the Product Backlog Items. Also often called just the Team.
Done		Finished, finito, no-more-work. The state of a completed Task, piece of work or Product Backlog Item. The Team declares with confidence that there is no more work to be done according to Done-criteria, general quality levels and other constraints for the project.
Done Criteria		Those conditions that must exist for a certain Product Backlog Item for the Team to declare it Done. This includes user acceptance criteria.
Entire Scrum Team		All people having responsibilities and commitments in the Scrum process: the Product Owner, the Team and the Scrum Master.
Epic		An Epic is a large User Story, typically one that later on will be broken down into appropriately sized User Stories.

Estimate	EST	Estimate of the size of work of a Product Backlog Item. Estimates are produced by the Team and used by the Product Owner to order and prioritize the Product Backlog.
Impediment		Anything preventing the Team from performing at their very best. Impediments are recorded on the Impediment Backlog and dealt with by the Scrum Master. Also called obstacles.
Impediment Backlog	IB	A list of current Impediment also showing their progress. The Impediment Backlog is owned by the Scrum Master.
Item		Short for Product Backlog Item.
Management		A secondary role in Scrum whose objective is to secure the success of the organization in which we find the Product Owner, Team and Scrum Master. The Manager typically assists in setting the Teams and removing impediments.
Milestone		A general Milestone in the Time-line for the Project, typically specifying that certain conditions must be met by this time, a special case is a Release Milestone.
PDSA cycle	PDSA	The Plan-Do-Study-Act cycle described by W. Edwards Deming, the foundation for the work in Sprints with Sprint Review and Sprint Retrospectives to close the cycle with respect to the Product developed and the work processes.
Product		The result of the Project.
Product Backlog	PB	An ordered list of all requirements in the Project. General requirements are called Constraints and are kept outside the main Product Backlog. Typically the Product Backlog is sorted in a number of Releases of Sprints, indicating which Product Backlog Items are expected to be completed in which Sprints.
Product Backlog Item	PBI	A requirement or deliverable in the Product Backlog, typically formulated as a User Story with accompanying Acceptance Tests. Such an Item must as a minimum have a Business Value and an Estimate of cost associated to allow the Product Owner to order and prioritize the Product Backlog.
Product Burn-down		A graph showing how much work remains after each sprint. This allows forecasts of the expected finish time of the project to be made. It is typical also to include a graph showing the Team's Velocity in each Sprint, this again helps to forecast the next Sprint. Finally, it is also common to record the accumulation of Business Value, in order to allow an assessment of the prioritization done.
Product Owner	PO	A primary role in Scrum. This person assumes responsibility of Business Value and return of investment (ROI). The Product Owner owns the Product Backlog, anyone can add to it, but only the Product Owner moves items up and down in priority. The Product Owner is also the anchor person to turn to for explanation of Product Backlog Items.
Project		The complete undertaking consisting of Time-line, Constraints, Product Backlog, Product Owner, Team and Scrum Master. Sometimes called Initiative
Release		A collection of Product Backlog Items that together make up a portion of the Product in some well-defined form and completeness that makes it useable to the User and Customer. A Release typically consists of several Sprints and can be associated with a Release Milestone in the Time-Line. Also called a Project Phase.
Release Burn-down		The same as a Product Burn-down, but only with focus on a certain Release as opposed to the whole Product.
Release Milestone		A Milestone in the Time-Line associated with a Release
Release Sprint		The last Sprint before a Release, where certain different activities need to be performed (such as special deploy tests etc.). Some projects use this kind of Sprints. The Team must in these cases after each Sprint be prepared for that the Product Owner declares the next Sprint to be a Release Sprint.
Retrospective		Short for Sprint Retrospective
Return on investment	ROI	A measure of the rate in which the Project generates value. Earlier value rather than later is of course preferable. This is a primary concern of the Product Owner.

Scrum		The name of the most popular Agile process; it takes its name from an event in the game of Rugby, where all players gather in a so-called Scrum to get a ball back into play.
Scrum Master	SM	A primary role in Scrum. This person assumes responsibility of the Scrum Process and for constant optimization. The Scrum Master ensures that everyone follows the process and does what he or she has committed to doing. He has a constant focus on optimization and removal of impediments; he owns the Impediment Backlog.
Scrum Team		Another name for the Entire Scrum Team.
Selected Product Backlog		A collection of Product Backlog Items that the Team and the Product Owner have agreed to do in the upcoming Sprint. The selection is a result of Sprint Planning Meeting #1.
Sprint		A period of time in which the Team works on delivering a set of Product Backlog Items. A Sprint is sometimes also referred to as an iteration. A Sprint is normally between two and four weeks in time.
Sprint Backlog	SB	A list of Product Backlog Items selected for this Sprint together with the decomposed work (typically Tasks) that the Team has broken down the Product Backlog Items into. Each Product Backlog Item consists of one or more pieces of work. The Sprint Backlog shows the progress of these towards "Done".
Sprint Backlog Item	SBI	A piece of work belonging to a certain Product Backlog Item, a deliverable. These are pieces of work defined during Sprint Planning #2 (or during the Sprint). Normally work is broken down so that it only needs to be "In Progress" for one day, however, several Team members can work on it during that day.
Sprint Burn-down		A graph showing how much remaining work exists each day in the Sprint, this allows for forecasting if the Sprint is likely to end with all work completed, or if corrective action is called for.
Sprint Goal		A sentence describing the achievement of a Sprint; it is used to communicate to stakeholders and as a guideline if the Sprint plan has to be changed during the Sprint.
Sprint Planning		A term covering the two meetings Sprint Planning #1 and Sprint Planning #2
Sprint Planning #1		A planning meeting held at the very beginning of a Sprint. The Product Owner and the Team agree on a set of Product Backlog Items that the Team believes it can complete in this upcoming Sprint. Before entering the meeting the Product Owner has ordered the Product Backlog with the relevant Product Backlog Items at the top; he has also made sure that these are Estimated and Valued. The Team has beforehand assessed their capacity for the upcoming Sprint.
Sprint Planning #2		The second planning meeting at the beginning of a Sprint, where the Team analyzes and designs how to deliver the selected Product Backlog Items. This produces a breakdown of work, typically a series of Tasks. All this is placed on the Sprint Backlog which is the result of this meeting. At the end of this meeting, we have a commitment (forecast) from the Team and it is made public.
Sprint Retrospective		A meeting at the very end of the Sprint (after the Sprint Review), where the Team and Scrum Master analyze the completed Sprint and come up with ideas for improvement. They assess these and decide on which to do something about in the next Sprint. When this meeting is finished the Sprint is completed. This closes the PDSA cycle with respect to how work is performed.
Sprint Review		A meeting at the end of the Sprint, where the Team presents the results of the Sprint, the completed Product Backlog Items. The Product Owner and stakeholders (could be Manager, Customer, User and others) are present, the Product Owner approves (or disapproves) the results, new ideas or changes to the produced features are actively sought from Stakeholders. This closes the PDSA cycle with respect to how the Product should be like.
Story		A short form of User Story.
Story Point	SP	A unit-less measure of size of Product Backlog Items typically used in Scrum. It is used to estimate relative size of Product Backlog Items compared to each other. It is

	customary to use the Fibonacci numbers as a scale of Product Backlog Items called Stories (1, 2, 3, 5, 8 and 13) for larger ones (called Epics) it is customary to use 20, 40 and 100 as a scale.
Task	A very specific example of a Sprint Backlog Item.
Task-board	A typical implementation of the Sprint Backlog.
Team	A primary role in Scrum. The Team have assumed responsibility of transforming Product Backlog Items into finished pieces of final Product. The Team owns the Sprint Backlog, and keeps this updated at any time. The Team self-organizes during the Sprint and finds the right ways to implement the Product. The Team estimates Product Backlog Items, typically during an Backlog Refinement Meeting. The Team selects a collection of Product Backlog Items during Sprint Planning #1 together with the Product Owner and breaks these down into smaller pieces of work during Sprint Planning #2. After this a forecast and a commitment from the Team is presented. The Team is also called the Development Team.
Theme	A collection of Product Backlog Items that deal with the same general functional area of the Product Backlog.
Time boxing	The concept of allowing a specific time to an activity. When the time has elapsed, the activity is over, period.
Time-Line	The top-level view of time-based commitments in the whole Project. Normally the Time-Line consists of 1) "Milestones", where certain achievements must be reached, 2) "Deliveries In", where something is delivered to the Team and 3) "Deliveries Out", where the Team must deliver something to an outside party. Often Milestones take the form of a "Release Milestone", where the Product must exist in some well-defined form having reached a completeness making it useable to the User and Customer.
User	A secondary role in Scrum, who uses the Product, the result of the Project. User acceptance test are performed by Users or their proxies.
User Story	A way of describing what a Product Backlog Item is about, it often follows this template: "As such-and-such a user, I want to be able to do this-and-that in order to achieve jada-jada-jada." A User Story can also have one or more Acceptance Criteria connected to it, giving evidence as to when a User Story is Done.
Valuation	The process of assigning a numerical value indicating Business Value to a Product Backlog Item. The Product Owner is responsible for the Valuation, but may call on others to supply him with information (such as sales and marketing people or a customer forum).
Velocity	The Speed at which the Team delivers Product Backlog Items, typically measured in