Scrum.

Scrum er en agil udviklingsmetode, der dog alt afhængig af hvilken kontekst den nævnes i, betegnes som enten et framework til agil udvikling, eller en agil udviklingsmetode. Det der kendetegner Scrum er, at den som metode ikke beskæftiger sig de praktiske aspekter af softwareudvikling. Scrum fokuserer derimod mere på managementmæssige og projektledelsesmæssige aspekter af softwareudvikling. Dette gør, at Scrum i praksis kommer til, at fungere som et projektstyringsredskab som selve softwareudviklingsprocessen styres med. Da Scrum ikke foreskriver noget omkring den praktiske side af softwareudvikling, ser man ofte i praksis, at Scrum anvendes i kombination med elementer fra en eller flere mere praksisorienterede udviklingsmetoder. Eksempelvis kan nævnes extremeprogramming og unified process, der begge trods deres forskelle, begge beskæftiger sig med den praktiske side af softwareudvikling. På denne måde kan Scrum i kombination med andre metoder danne en optimal praksis for udvikling, hvor der er fokus på både styring af projektet, samt taget højde for hvordan man vil håndtere de mere praksisnære dele af udviklingen.

Scrum rummer tre artefakter som er kerneelementer i metoden. Der opereres med en product backlog som er den vigtigste af artefakt. Product backlog er en prioriteret liste af krav til det system der skal udvikles. Elementerne i product backlog kaldes items, og de er i product backloggen prioriteret efter deres forretningsværdi også kaldes business value. Rent praktisk sker det, at efterhånden som der bliver fjernet items fra backloggen i takt med, at items bliver omsat til kode står man efter denne ændring med en ny version af product backloggen. På denne måde vil man komme til, at stå med en ny version af product backloggen hver gang der er blevet færdigudviklet nogle items. I kontrast til at der typisk fjernes items fra backloggen efterhånden som der udvikles mere og mere på systemet, kan det også ske at der føjes nye items til backloggen. Der kan være flere årsager til dette. Det kan være at kunden får nye forretningsbehov, der kan ske lovændringer der gør, at der skal implementeres yderligere funktionalitet på et område eller der kan for eksempelvis opstå nye konkurrencemarkeder som kunden vil satse på. Ofte er det i praksis sådan, at behovet for et stykke softwares kunnen, ændrer sig i løbet af, at det er i gang med at blive udviklet. Dette sker på baggrund af, at det scenarie som softwaren skal fungere i, ændrer sig med korte mellemrum. Netop derfor er en af hovedtankerne bag Scrum, at man havde behov for en udviklingsmetode, der kunne håndtere ændringer i krav til produktet undervejs i udviklingsprocessen.

Den anden artifakt i Scrum er sprint backlog. Inden man går i gang med en udviklingsfase flytter man items fra product backloggen over i sprint backloggen. Dette foregår sammen med kunden eller en repræsentant for denne. Alt efter hvad prioritet de forskellige items i product backloggen har, flyttes de mest relevante af dem over i sprint backloggen. Det der praktisk sker når items føres fra product backloggen til sprint backloggen er, at de omsættes til user stories. Sprint backloggens formål er, at være en liste af user stories hvor hver story er nedbrudt i en række konkrete udviklingsopgaver kaldes tasks.

Den tredje artifakt er burn down chart. burn down chart er en graf der indikerer hvor godt det lykkedes udviklingsteamet at løse de opgaver de har vurderet de kunne, indenfor den givne ramme af tid. På burn down chartet anføres der først en teoretisk linje der indikerer hvad man teoretisk set som team bør kunne brænde hver dag. Derudover påfører teamet hvad dag hvor meget arbejde de har nået at lave, altså hvor meget de reelt kan brænde. Burn down chart er en vigtig artifakt da den kan bruges til, at fortælle eller afsløre om teamet generelt ikke kan levere det estimerede arbejde, eller den kan lede opmærksomheden hen på om det eksempelvis er en særlig type funktionalitet, der ofte er svær at implementere til tiden.

Scrum rummer også tre roller der er centrale i metoden. Som repræsentant for kundens behov har man en product owner. Dette kan være en person kunden selv har stillet til rådighed, eller det kan være en person internt i udviklingsvirksomheden. Product owner er den vigtigste af de roller der er i Scrum, da det er dennes opgave, at sørge for, at det der bliver udviklet faktisk er det kunden har behov for, også hvis der skulle opstå ændringer i kundens behov. Product owner er den person der kender produktet der skal udvikles og, de behov kunden har, og gerne forretningen som produktet skal varetage. Ifølge Jeff Sutherland, grundlæggeren af Scrum, har product owner nogle direkte ansvar som er følgende :

* Han har som ene person ansvar for product backloggen.
* Items i product backloggen skal være tydeligt og præcist formuleret.
* At hele tiden, og løbende omprioritere items i product backloggen, så deres hierarki korresponderer med et virkelighedens snapshot af kundens behov.
* Hvis der er interessanter der kunne ønske ændringer i product backloggen, skal de gå til product owner, da han er den eneste person, der har ret til at ændre på prioriteringen af items.
* Han er en nøgleperson der er ansvarlig for, at definere krav og features til systemet.
* Han har til opgave, at prioritere features og dermed de items der findes i product backloggen, så de er reflekterer hvad der har mest business value/forretningsværdi.
* Når nogle features er udviklet, er det hans opgave, at acceptere eller afvise de features der er blevet udviklet af udviklingsteamet. Dette sker i forbindelse med en særlig Scrum aktivitet der betegnes sprint review.

Product owner er på mange måder central, og man kan sige, at hvis ikke man har en product owner der har et solidt kendskab til forretningsbehovene, som det færdige produkt skal understøtte, kan det være svært for udviklingsteamet, at ramme rigtigt med de features de udvikler. Product owner har som udgangspunkt, som den eneste person ret til at varetage ovenstående, men må gerne uddelegere

opgaverne til udviklingsteamet. Dog er han stadig den endeligt ansvarlige for. at opgaver udføres tilfredsstillende.

Den anden centrale rolle i Scrum er scrum master. Scrum master har overordnet set en servicerende rolle overfor selve udviklingsteamet, mere end der er tale om en dikterende rolle. De primære ansvar i rollen består i, at hjælpe teamet fri af blokeringer, der hindrer dem i at arbejde effektivt, og levere de estimerede features indenfor den afsatte tidsramme. I Scrum opererer man med begrebet 'impediments', som er et udtryk der dækker over alt tænkeligt, der hindrer en udvikler i, at få leveret et stykke arbejde indenfor estimeret tidsramme. Det er ligger scrum masters rolle, at tage sig af impediments og få skabt bedre betingelser for udviklingspraksis hvis der er behov for det. Han fungerer på denne måde som en slags firewall der beskytter teamet.

Det er scrum masters ansvar, at facilitere scrum processen, via en slags projektlederrolle. Dette betyder, at han skal sørge for at scrum processen bliver overholdt. Dette indebærer, at de forskellige scrum events bliver afholdt på korrekt vis, at scrums indbyggede regler, værdier og praktikker overholdes. Scrum master kan både være en del af udviklingsteamet, men han kan i praksis også være en dedikeret person, der som scrum master kun beskæftiger sig med sikring af scrum processen. Scrum master leder det daglige Scrum møde der kaldes daily scrum, hvor alle i teamet bliver opdateret på hvordan det går andre i teamet. Generelt fungerer scrum master som et bindeled mellem teamet og product owner.

Den sidste af de tre rollet er selve udviklingsteamet hvor alle udviklere i teamet under et betegnes som udviklingsteamet. Det er udviklingsteamets ansvar under hele processen at blive mere erfaren i hvordan de kan arbejde med Scrum, ligesom det er helt central at de får mere erfaring i at estimere tasks og vide med sikkerhed hvad de kan udvikle indenfor en given tidsramme.

Scrum består udover roller og artifakter af en række foruddefinerede events, som hver især har sit formål og sin speficikke plads i scrum processen. Scrum anvender i forbindelse med events et princip om 'timeboxing', hvilket betyder at der er afsat et maksimalt tidsrum som en given event må tage. Hver Scrum Event har sit eget navn. Meningen med at have disse events er, at minimere behovet for ad-hoc møder, der antageligt vil have en forstyrrende effekt på udviklingsprocessen, samt sikre kontinuitet og fremdrift i arbejdet. En Event i Scrum kan godt slutte før dens 'timebox' er brugt, hvis dens formål er opfyldt.

Den helt centrale event i Scrum er Sprint. Sprint er en fastsat tidsramme på typisk 1-4 uger, som er der hvor selve softwareudviklingen foregår. Indenfor sprintet har teamet comitted sig til, at udvikle en bestemt mængde features, som de gerne skulle kunne nå indenfor denne periode. Sprint består af en række underaktiviteter der betegnes som, sprint planning, daily scrums, sprint review og sprint retrospective.

Sprint planning er den første aktivitet der er i et sprint. Alle i teamet deltager i denne aktivitet. Formålet med denne aktivitet er, at planlægge alt det arbejde der skal foregå i det kommende Sprint. Tidsrammen for sprint planning er maksimalt 8 timer for sprint på 4 uger, men for kortere sprint er tidsrammen kortere. Formålet med aktiviteten er af finde ud ad :

* Hvilke features kan vi nå at lave i løbet af dette sprint.
* Hvad for noget arbejde skal der til, for at disse features er udviklet

På dette møde tager man udgangspunkt i product backloggen. Det første der sker på sprint planning meeting er, at teamet sammen med product owner gennemgår product backloggen og drøfter de items der har størst prioritet. Med udgangspunkt i, at udviklingsteamet som skal udvikle, er dem der bedst ved hvad de kan nå indenfor en given tidsramme, er det dem der i samarbejde med product owner tager items fra den prioriterede product backlog, som de comitter sig til, at kunne færdigudvikle indenfor sprintets tidsramme. En af nøgleprincipperne i Scrum er, at teamet selv vurderer hvad de kan nå, frem for at få påtvunget arbejdsmængden udefra.

Det næste der sker er at det gennemgåes hvor meget tid hver udvikler har til det konkrete udviklingsarbejde, fraregnet eksterne møder, pauser og andet 'fravær'. På denne måde udregnes den samlede timekapacitet for udviklerholdet der er til rådighed til selve softwareudvikling. Når denne er specificeret begynder de, at nedbryde hvert item i product backloggen til tasks, som er konkrete udviklingsopgaver. Disse tasks påføres et estimeret timetal, som man skønner det vil tage, at udvikle den features den pågældende tasks gælder for. Udviklerholdet bliver ved med, at omsætte items fra product backloggen til tasks indtil de ikke har flere timer til rådighed.

De tasks som udviklerholdet nu står med, danner tilsammen det der kaldes sprint backlog – en samlet liste over features, der skal udvikles indenfor Sprintet. Når alle valgte items er omsat til tasks i sprint backlog, kan selve udviklingen begynde. Udviklere kan herefter frivilligt påtage sig tasks, og begynde selve softwareudviklingen. Udviklerne bør dog være opmærksomme på indbyrdes afhængigheder mellem tasks, og eventuelle hensigtsmæssigt rækkefølge at implementere dem i. Når først sprintet er startet, kan product owner ikke presse nye tasks ind i sprintet. Dermed kan eventuelle nye behov, først komme med i næste sprint, ved at de blevet indført og prioriteret i product backloggen. Dog kan product owner her i ekstreme tilfælde aflyse et sprint. Alternativt kan man sige, at product owner kun er et sprints varighed eller mindre, fra at kunne kræve ændringer implementeret. For at holde styr på product backlog, sprint backlog og arbejdets fremdrift, anvendes der ofte visuelle hjælperedskaber, der gør det muligt for alle i teamet at få overblik over hver enkelte tasks status, om den er påbegyndt, i gang, afsluttet, eller skal godkendes. Til dette formål bruges der oftest et scrum board, eller task board.

I løbet af et sprint afholdes der hver morgen et møde styret af scrum master. Mødet betegnes både som daily scrum eller daily stand up meeting. Scrum master sørger for, at det kun er teamet der deltager i daily scrum. Mødet afholdes samme tid hver dag, og har samme varighed hver gang.

Hensigten med mødet er ikke diskussion, men derimod afrapportering af status for arbejdets fremgang. På mødet er der tre obligatoriske punkter alle teammedlemmer svarer på :

* Hvad lavede jeg i går, som bragte teamet tættere på vores mål for Sprintet?
* Hvad er det min intention at lave i dag, for at bringe teamet tættere på at nå vores mål?
* Er der nogle impediments jeg synes, at kunne se, enten for mig selv eller teamet?

Mødet har til formål, at kaste lys over, om teamet arbejder hen imod at nå den arbejdsmængde der ligger i sprint backloggen. Daily scrum er også underlagt timebox- princippet og må tage maksimalt 15 minutter. Mødet skal også fungere, som en event der sikrer følgende :

* Forstærke kommunikation i teamet.
* Identificerer impediments så de kan fjernes.
* Mindske behov for andre møder.
* Styrke teamets viden og fremskynde dets evner til beslutningstagen.

Om mødet beskrives der også i Scrum-litteraturen, blandt i The Scrum Papers af Jeff Sutherland, at det er et 'inspicér og tilpas-møde'. Hvis der er behov for diskussion, finder det sted efter mødet.

På daily scrum mødet ajourføres teamets burn down chart gældende for sprintet. Det er vigtigt, at vide, at burn down chart ikke handler om, at vise hvor meget tid der er brugt, men derimod hvor meget arbejde der mangler, for at teamet har nået sit mål. Burn down chartet viser enten timer eller dage, eller story points hvis man kombinerer med extreme programming. Efter hvert daily scrum møde registreres der, hvilke tasks der er færdige, og det samlede antal timer for dem udregnes. Dette antal timer skal burn down chartet ajourføres med. Dette gøres ved, at tage værdien af resterende timer for den foregående dag, og føre grafen videre, til det nye antal resterende timer på den pågældende dag. Hvis den teoretiske linje ligger over den kurve man ajourfører, betyder det man arbejder hurtigere, end man have estimeret man ville. Hvis den teoretiske linje ligger under den der ajourføres, betyder det at man som team ikke formår, at følge med det arbejdstempo man har fastsat. Der er således 'gaps' på burn down chartet, og dem er det vigtigt at reagere på. Rent praktisk laver man ofte burn down chartet visuelt på et stykke papir eller karton på en væg, så alle altid har adgang til at se på det. Dog findes der elektroniske redskaber der kan generere burn down chart.

Der er to resterende event i Scrum, som hedder henholdsvis sprint review og sprint retrospective.

Sprint review, som er den første der afholdes af de to aktiviteter er timeboxed til 4 timer for et sprint på 4 uger. På sprint review demonstrerer teamet, hvad de har udviklet siden sidste sprint. Typiske vil der blive demonstreret nye features, ændringer i arkitektur eller andre signifikante ændringer i produktet. Sprint review bør ikke tage mere end 30 minutter at forberede, da det mere er en demonstration af funktionalitet end et foredrag. Deltagerne til sprint review vil typisk være interessanter, og andre mere perifere personer som kan inviteres af product owner som også deltager. Udviklerteamet svarer på eventuelle spørgsmål fra andre deltagere. Det er vigtigt, at udviklerne som skal demonstrere de nye features, har haft en generalprøve inden demonstrationen for at sikre at alt fungerer.

Sprint retrospective er en aktivitet, der foregår efter sprint review. Den er timeboxed til maksimalt tre timer. Det er scrum masterens ansvar, at aktiviteten finder sted. Sprint retrospective foregår internt i scrum teamet og formålet med sprint retrospective er selvinspektion, hvilket indebærer, at teamet reflekterer over det forgangne sprint og deres egen arbejdsprocess :

* Identificere hvad der hovedsageligt gik godt, som er vigtigt at holde fast ved.
* Finde ud ad hvad der gik mindre godt, som bør forbedres.
* Lave en plan for implementering af forbedringer, som der er nødvendige for teamets

evne til at arbejde effektivt.

Teamets medlemmer reflekterer således over det forgangne sprint, eventuelt med udgangspunkt i noter de har taget undervejs igennem sprintet. Denne event er det sted i scrum processen, hvor teamet kan drøfte, hvad der fungerer, og hvad der ikke fungerer, og om der eventuelt skal prøves noget nyt.

For at arbejde med de eventuelle ukendte områder der kan være af et udviklingsprojekt, opererer man i Scrum med et begreb der hedder spikes. En spike er en slags proof of concept der laves på noget funktionalitet man ikke er helt sikker på om man kan implementere. Hvis der på den måde er noget meget nyt i et projekt, bør man via en spike teste på om man reelt kan implementere det ønskede. Typisk vil en spike ligge i et sprint 0, hvor det er meningen at man afklarer om man kan implementere funktionaliteten. Dog kan man godt indarbejde spikes i løbet af sprints. Man kan også komme i situationer hvor man er nødt til det.

For at give et overblik over Scrum sammenfattes der afslutningsvis med et overblik på selve processen. Denne process er samtidig et billede på hvordan man skal gribe et scrum projekt an fra start til slut. Selveste scrum processen starter med, at en kunde har et it-mæssigt behov. En masse informationer fra alle der er involverede i produktet, samles et sted, nemlig hos productowner. Denne har ansvar for, at alle disse informationer føres i product backloggen som items. Når dette er gjort, eksisterer der den prioriterede liste af items som product backloggen er. Herefter kan scrum teamet begynde på sprint planning meeting, som indbefatter, at der bliver lavet en sprint backlog ud fra de eksisterende items i product backloggen. Derefter starter det pågældende sprint, og dets længde kan ikke ændres, efter det er startet. Under Sprintet er scrum master tovholder for selve scrum processen, og der afholdes daily stand up meetings dagligt. Når Sprintet er afsluttet, står teamet med et færdigt stykke arbejde. Efter Sprintet afholdes der sprint review, hvor de i sprintet producerede nye features fremvises ved en demonstration. Til sidst i sprintet, er der sprint retropective, hvor scrum teamet internt kan reflektere, og drøfte, hvad der gik godt, som skal holdes fast ved, og hvad der gik mindre godt eller problematiske forhold, som eventuelt skal forbedres.