



ØKONOMISTYRELSEN

Vejledning om anvendelse af agile udviklingsmetoder

Maj 2022

Indhold

| | |
|---|-----------|
| 1. Indledning | 3 |
| 1.1 Hvorfor benytte agile udviklingsmetoder | 3 |
| 2. Brug af agile udviklingsmetoder i statens it-projektmodel | 4 |
| 2.1 Arbejde agilt i fasemodellen | 4 |
| 2.2 Det bedste fra begge verdener | 4 |
| 2.3 Fuldskala agile metoder | 5 |
| 3. Krav til organisationen samt typer af projekter og it-systemer | 7 |
| 3.1 Hvad kræver det af organisationen at udvikle agilt | 7 |
| 3.2 Projektets karakter | 8 |
| 4. Minimumskrav for projekter, der ønsker at benytte agile udviklingsmetoder | 10 |
| 4.1 Prioritering | 10 |
| 4.2 Backlog | 11 |
| 4.3 Sprints | 12 |
| 4.4 Produktreview | 12 |
| 4.5 Procesreview | 13 |
| 4.6 Releaseplan | 13 |
| 4.7 Rollerne Product Owner og Scrum master | 14 |

1. Indledning

Formålet med denne vejledning er at sætte rammer for, hvordan agile it-systemudviklingsmetoder kan benyttes i statens it-projektmodel.

Vejledningen har ikke ambition om at lære myndigheder at arbejde agilt, men den giver forslag til, hvordan myndigheder, der ønsker at bruge agile udviklingsmetoder med de fordele, der kan ligge heri, kan komme i gang.

Målgruppen for vejledningen er ledelser i myndigheder, der enten allerede benytter agile udviklingsmetoder i it-systemudvikling eller som overvejer at gøre det.

Vejledningen retter sig desuden til de projektfolk, der skal bidrage til og/eller lede kompetenceudviklingen hos egen organisation, således at organisationen bliver i stand til at udvikle it-systemer efter agile metoder.

1.1 Hvorfor benytte agile udviklingsmetoder

Stadig flere myndigheder og it-leverandører benytter agile udviklingsmetoder i udvikling af it-systemer, og det er der gode grunde til. Under de rette betingelser og med optimalt brug, kan agile it-udviklingsmetoder lede til:

- *Større gevinster*, fordi de dele af it-systemet, der giver de største gevinster udvikles først, og at det er muligt at høste gevinster tidligere.
- *Bedre it-systemer*, fordi metoderne i langt højere grad involverer brugere.
- *Billigere it-systemer*, fordi kapaciteten bliver udnyttet bedre, så der udvikles mere pr. udviklingskrone. Derudover mindskes omkostninger ved, at kun det nødvendige udvikles (Minimum Viable Product).
- *Højere grad af agilitet*, fordi der hurtigere er kørende løsninger, og fordi der undervejs i udviklingsforløbet kan justeres, således at projektet hele tiden matcher de aktuelle forretningsbehov.
- *Mindre risici*, fordi agile udviklingsmetoder i sig selv er risikominimerende, og fordi at nogle af de risici, som knytter sig til it-systemet, kan mitigeres løbende på grund af justeringsmuligheder.
- *Bedre alignment mellem forretning og projekt*, fordi agil udvikling kræver tæt involvering af forretningen i hele processen. Agile udviklingsprincipper er enkle, intuitive og lette at forstå for både it-udviklere og medarbejdere fra forretningen.
- *Hurtigere afdækning af fejl*, fordi designfejl, kodefejl o.l. viser sig tidligere i agile it-produktudviklingsforløb. Det gør det muligt at nedbringe følgevirkninger af fejl og gør det lettere og billigere at rette fejlene.

I afsnit 2 herunder gives en beskrivelse af, hvordan agile udviklingsmetoder i udvikling af it-løsninger rummes i statens it-projektmodel.

I afsnit 3 drøftes, hvilke kompetencer organisationen bør opbygge for at få mest værdi af at bruge agile udviklingsmetoder.

I afsnit 4 gives nogle retningslinjer, der er obligatoriske for projekter over 15 mio. kr., der vil udvikle efter agile metoder.

2. Brug af agile udviklingsmetoder i statens it-projektmodel

De regler og retningslinjer, der gælder for it-projekter i staten, gælder uanset om der benyttes agile metoder i projektet eller ej. Det betyder bl.a., at projektet skal:

- Følge reglerne i budgetvejledningen (krav om at følge statens it-projektmodel, hvilke udgifter der kan afholdes hvornår, retningslinjerne for forelæggelse for Statens It-råd, og Folketingets Finansudvalg) samt ØAV (retningslinjer for kontering, hvad der kan aktiveres mv.).
- Risikovurderes i Statens It-råd, hvis de samlede udgifter udgør 15 mio. kr. eller mere.
- Følge fasemodellen i statens it-projektmodel.
- Have godkendt sit samlede styringsgrundlag i analysefasen inden gennemførelsesfasen kan påbegyndes.
- For it-projekter med samlede udgifter under 15 mio. kr. er det ikke et krav at anvende statens it-projektmodel. For projekter over 1 mio. kr. skal der dog som minimum etableres et styringsgrundlag tilsvarende til projektgrundlaget i statens it-projektmodel, men forenklet og i omfang afstemt med projektets størrelse.

Trods ovenstående regler og retningslinjer, er der fortsat vide rammer for at arbejde agilt i statens it-projektmodel.

2.1 Arbejde agilt i fasemodellen

I rent vandfaldsbaserede projektmodeller ses ofte en 'anskaffelses'- eller 'specifikationsfase' foran 'gennemførelsesfasen', der er tiltænkt kravspecifikation og udbud. Der er ikke en sådan fase i statens it-projektmodel, netop for at give de statslige it-projekter større frihed til at designe sit udviklingsforløb.

Faserne i Statens it-projektmodel fremgår af Figur 2.1.

Figur 2.1: Projektmodellens faser



Der kan arbejdes agilt i både idé-, analyse- og gennemførelsesfasen.

I *idé- og analysefasen* kan der bruges værktøjer, som også bruges i de agile metoder til at tilvejebringe de obligatoriske beslutnings- og styringsdokumenter, der skal være på plads inden projekter overgår fra analysefase til gennemførelsesfase. Værktøjer og metoder som proof of concept, prototyping, feasibility studies o.l. kan fint bruges til at scope det it-system, der bedst understøtter ønskede mål og gevinster.

I *gennemførelsesfasen* kan agile udviklingsmetoder bruges fuldt ud. Jo flere elementer der bruges fra den agile metode organisationen har valgt, jo bedre. For projekter over 15 mio. kr. er der nogle minimumskrav, der skal følges, hvis en myndighed ønsker at benytte agile metoder til udvikling af hele- eller dele af it-løsningen. Se uddybning i afsnit 4.

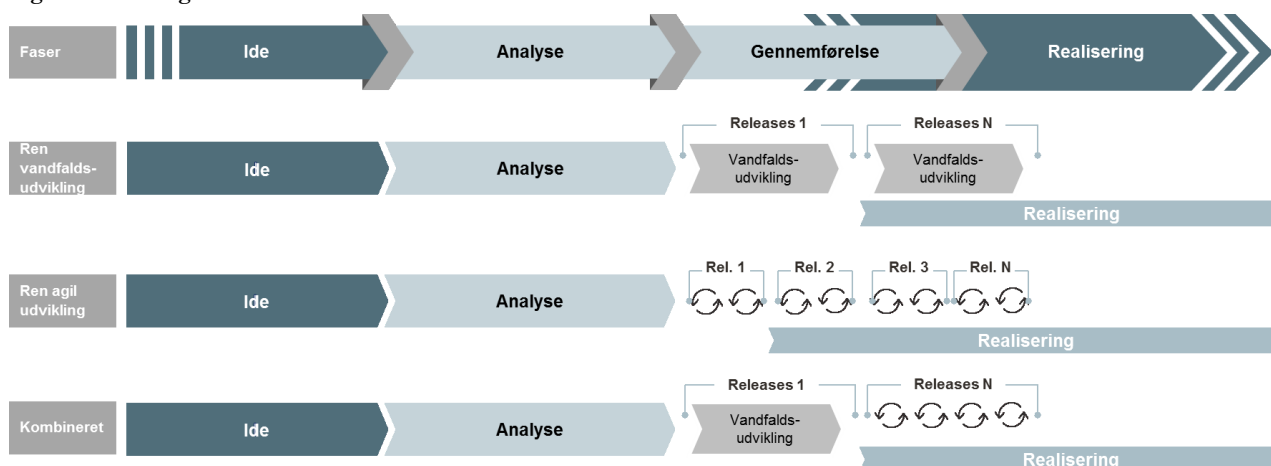
For projekter *under* 15 mio. kan alle retningslinjer opfattes som anbefalede. Mindre projekter kan være velegnede til at afprøve agile udviklingsmetoder af på. Dermed kan gennemførelse af mindre projekter give organisationen den erfaring, der er en forudsætning for senere også at kunne benytte agile udviklingsmetoder på større projekter over 15 mio. kr.

Som i vandfaldsbaserede metoder, stiller brug af agile udviklingsmetoder også krav til planlægning, styrende principper, værktøjer og governance ift. supplerende roller, styregruppens rolle samt ansvar og beslutningsstrukturer, om end det på nogle områder er nogle andre krav.

2.2 Det bedste fra begge verdener

Der kan tilrettelægges med traditionelle udviklingsmetoder, rent agile metoder, eller kombinationer heraf, se Figur 2.2.

Figur 2.2: Oversigt over metoder



Inden for de udbudsretlige rammer kan det også vælges at lave et funktionsudbud, hvor en ekstern leverandør tager ”ansvaret” for løsningen eller man udvikler selv, evt. suppleret med eksterne udviklere, hvor myndigheden selv er ansvarlig for løsningen.

I statens it-projektmodel er det ikke enten eller; ikke kun agilt eller kun vandfald, idet der søges mod at forene det bedste fra begge verdener.

De agile metoder er oprindeligt udsprunget af et ønske om at styrke *udviklingsprocesser* og har ikke haft særskilt fokus på *projektskyring* eller *projektledeelse*. Der er de senere år udviklet videre på nogle af de agile metoder, så de adresserer dele af projektledelsesdisciplinerne, men de agile metoder giver ikke svaret på alle behov i forhold til at beslutte, lede og styre statslige it-projekter. Fx håndterer de agile metoder ikke systematisk opstilling af mål, forretningsmæssige gevinster, interessanthåndtering, risikostyring, økonomihåndtering, organisatorisk implementering eller planlægning af gevinstrealisering.

De agile udviklingsmetoder gør op med den opfattelse, at man på forhånd kender alle krav til det endelige it-produkt. De agile udviklingsmetoder kredser om den fleksibilitet, der opnås ved, at krav kan ændre sig undervejs, og at der styres efter opstillede forretningsmæssige gevinster, rammer og mål, snarere end efter detaljerede krav.

Selv når man alene ser på it-systemudviklingen, er der ikke nødvendigvis tale om alt eller intet: En myndighed kan vælge at udvikle dele af en it-løsning efter traditionelle metoder og andre efter agile metoder.

2.3 Fuldskala agile metoder

Der er også varianter og videreudvikling af agile metoder, der har sat sig for at kunne opskalere fra et enkelt eller nogle få udviklingsteams, der favner et enkelt it-system (fx Scrum) til mange samtidige udviklingsteams, der udvikler store it-systemkomplekser (fx Scrum of Scrums, eller SAFe). De har alle agile fællestræk, men varierer også i scope og tilgang.

Lidt forenklet kan det beskrives som, at jo mere en metode skalerer op til håndtering af store systemkomplekser eller hele porteføljer af systemer, jo mere koordination og ledelse er der indbygget i dem. Dermed overlapper – og konkurrerer - de mere med de traditionelle projektstyringsdiscipliner, som også statens it-projektmodel står for.

Nogle fuldskala agile metoder som fx SAFe (i fuldt udfoldet konfiguration) er i sin grundudformning ikke umiddelbart forenelig med statens it-projektmodel. Der opereres ikke med projektbegrebet, der er ikke traditionelle faser og faseovergange, og der er ikke som i statens it-projektmodel, et naturligt sted, hvor et samlet projekt- og business case grundlag besluttet som rammer for projektet.

Disse fuldskalametoder retter sig mod udvikling af store, sammenhængende systemkomplekser, hvor en særlig organisatorisk linjeenhed opererer med værdistrømme (value streams) og løsningstoge (solution trains).

Hvor basale agile it-udviklingsmetoder og brug af agile værktøjer fint lader sig integrere med statens it-projektmodel, så skal de agile fuldskalametoder tilpasses på en række områder, før de er kompatible med rammerne for statslige it-projekter, herunder statens it-projektmodel.

Hvis I som myndighed overvejer at gøre brug af fuldskala agile metoder på it-udviklingsopgaver over 15 mio. kr., skal I derfor kontakte Kontor for it-styring, så relevante tilpasninger kan drøftes.

3. Krav til organisationen samt typer af projekter og it-systemer

Der er en række forhold, som en myndighed, der overvejer at udvikle hele eller dele af en kommende it-løsning med agile metoder, bør bringe ind i sine overvejelser. Nogle forhold knytter sig til organisationens erfaring og modenhed i brug af agile metoder, andre til karakteren af det projekt, der skal gennemføres og endelig til karakteren af den/de it-løsninger, der skal udvikles.

3.1 Hvad kræver det af organisationen at udvikle agilt

Det er i sig selv et betydeligt forandringsprojekt for en organisation at blive i stand til at arbejde efter agile udviklingsmetoder, og det bør derfor gribes an som sådan. Denne forandring kræver ledelsesbeslutning, planlægning, uddannelse, afklaringsaktiviteter og betydelige investeringer på lige fod med andre større forandringsprojekter. At udvikle it-systemer efter agile metoder er ikke kun en beslutning, det er også en kompetence, som organisationen og dens samarbejdspartner skal erhverve sig.

At arbejde efter agile udviklingsmetoder kræver blandt andet, at:

- ledelsen, og særligt styregruppeformanden og gevinstejeren skal have en høj grad af afklaringsparathed og være indstillet på at deltage i flere, men kortere beslutningskrævende fora og dermed træffe flere løbende beslutninger hurtigt samt er villig til at uddelegere nogle typer af beslutninger, fx et produktansvar til en Product Owner.
- den/de dele af forretningen eller driftsorganisationen, der bliver berørt af det it-projekt, der skal gennemføres, er meget aktiv i hele processen og er hele tiden klar til at prioritere forretningsbehov og godkende udviklede dele af it-løsningen. Forretningen skal stille med en eller flere medarbejdere, der skal varetage dialogen mellem forretningen og it-udviklingsteamet (en Product Owner). Hvis projektet gennemføres i et samarbejde mellem flere organisatoriske enheder i myndigheden, eller mellem flere myndigheder/samarbejdspartner, skal alle parter være indforståede hermed.
- en evt. intern it-organisation oplæres og geares til agil it-systemudvikling. Hvis al systemudvikling varetages af eksterne leverandører skal der aftales en meget tæt koordination med leverandørens udviklingsteam. Her skal også centrale roller besættes, bl.a. en Scrum Master.
- de teams, der arbejder med teknisk og organisatorisk implementering af løsningen skal magte og aktivt udnytte, at løsningen ideelt set leveres i løbende releases, som skal sættes i produktion. Dermed kan høst af gevinster påbegyndes allerede fra første release.
- organisationens økonomifunktion og juridiske enhed skal ikke bare forstå økonomistrømme og juridiske aspekter af agile metoder, men også være gearret til at spille ind og støtte projekterne der, hvor de agile tilgange kan blive udfordret af retningslinjer for statslig økonomistyring, regler om udbudspligt osv.
- brugere af it-løsningen skal forstå, hvorfor den samlede løsning kommer i etaper og at arbejdsgange tilsvarende ændres og tilpasses i etaper.

Hvis organisationen, eller dele af den gerne vil påbegynde en modningsproces frem mod at kunne udvikle it-systemer agilt, anbefales det at:

- tage kontakt til en sammenlignelig ”søsterorganisation” som allerede er i gang med agil it-udvikling. Hør om fif, faldgruber, do's and don'ts og få inspiration til implementeringsforløbet.
- se nærmere på de agile frameworks, der er relevant for it-systemudvikling (Scrum, Agile PM, PRINCE 2 agile, DSDM...). Vælg det, der synes at matche bedst og vurder, hvilke af elementerne i frameworket, der kunne startes ud med, og hvilke, der kan tilføjes hen ad vejen.
- uddanne relevante medarbejdere til de nøgleroller, der skal varetages. Uddan gerne interne agile coaches, der kan støtte alle deltagere i de agiles forløb.
- starte med projekter, hvor de løbende resultater er meget tydelige, fx systemer med meget brugerrettet dialog. Det giver produktreviews, hvor drøftelser om sprintets resultater bliver meget håndgribelige og giver et godt visuelt billede af om det udviklede understøtter de ønskede user stories.
- arbejde agilt, prøv tingene af i mindre pilotprojekter, juster ind løbende, osv.

- Hvis der skal inddrages en ekstern leverandør er der udviklet en særlig standardkontrakt, K03, til regulering af samarbejdet mellem myndighed og en ekstern leverandør i projekter med agil udvikling. Der er et omfattende vejledningsmateriale til kontrakten, bl.a. om hvordan ”kravspecifikationer” kan se ud i agile projekter. [Klik her](#) for mere info.

Statens og Kommunernes Indkøb, SKI, har udviklet en aftale, ”02.18 It-løsninger og –projekter”, med skabeloner og leveranceaftaler, som specifikt understøtter agile udviklingsparadigmer. Se nærmere herom på ski.dk.

På de første agile projekter kan det være relevant, at en såkaldt time & material aftale med leverandøren, dvs. hvor der afregnes faktisk forbrug, frem for en fastprisaftale. Det kræver en vis modenhed i agile processer at samarbejde med en ekstern leverandør på en fastpriskontrakt. Her har leverandøren typisk ansvaret for løsningen og kan og skal (med rette) kræve, at myndigheden spiller effektivt ind i de agile processer.

Som altid, når der introduceres nye metoder, kan der opstå nye risici. Vær derfor meget opmærksomme på eventuelle faresignaler. Der kan ikke opstilles en udtømmende liste over eventuelle faresignaler, men fra nogle af de myndigheder, der allerede er påbegyndt en agil rejse, har vi noteret følgende:

- Da fokus af gode grunde er på it-systemudviklingen og de nære processer rundt om den, kan fokus tilsvarende blive trukket væk fra nogle af de andre vigtige projektdiscipliner, fx implementerings- og gevinstrealiseringsaktiviteter.
- Nogle agile teams kan have en tendens til at lukke sig så meget inde om sig selv, at den nødvendige interaktion med resten af organisationen kan blive udfordret.
- Projektet og det agile team skal arbejde i en organisation, hvor andre dele af organisationen, stabsfunktioner, ledelser, it-driftsorganisationen osv. ikke nødvendigvis arbejder agilt. Det kan give nogle gnidninger, der skal håndteres i gensidig forståelse.
- Product Owner’ens rolle er et meget vigtigt, men også sårbart led i kæden. Hvis Product Owner’en ikke magter opgaven, ikke har tilstrækkelige ressourcer, ikke kan have den fornødne adgang til de rette personer i forretningen på det rette tider, ikke er dubleret ved sygdom o.l., kan hele fødekæden til de(t) agile team bryde sammen.
- Det kan selvfølgelig give problemer, hvis Scrum Master’en og det agile team ikke leverer som forventet.

3.2 Projektets karakter

Når organisationen eller den del, der er kandidat til at gennemføre projekter med agil it-udvikling, har udviklet en passende modenhed (eller har lagt en passende plan for at nå derhen) er der nogle projektmæssige overvejelser, der bringes i spil.

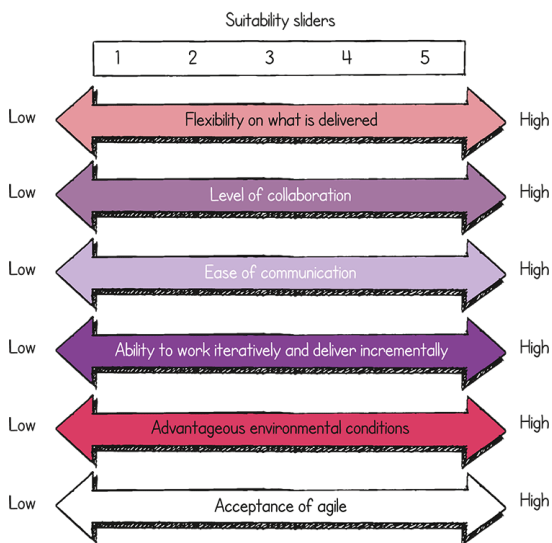
Bemærk, at de karaktertræk, der nævnes herunder ikke kan stilles op som ja/nej, eller ”alt eller intet”, ligesom man ikke kan sige, at hvis der fx sættes kryds ud for to forhold, så skal eller bør der automatisk arbejdes med agil udvikling. De forskellige forhold kan tale i en bestemt retning, men til syvende og sidst, er det en samlet vurdering (projekt-)organisationen skal tage.

- Jo mindre man ved om, hvordan det færdige system skal se ud, jo mere oplagt er det at benytte agil udvikling. Man bør have en god idé om, hvad der forretningsmæssigt skal it-understøttes, men ikke nødvendigvis *hvordan, det skal understøttes*. De agile udviklingsmetoder er effektive til at prøve nogle ideer af og justere ind om nødvendigt.
- Heraf følger også, at jo mere komplekst et it-system, der skal udvikles, jo mere oplagt at udvikle agilt. Det kræver dog, at koordinationen mellem forskellige udviklingsaktiviteter skal skaleres i relevant omfang afhængig af størrelse og kompleksitet af det system, der skal udvikles. Der findes agile metoder, der netop adresserer denne koordination, men det vil kræve tilsvarende mere samordning med de styringsdiscipliner, der i øvrigt bruges i projektet.
- Jo mere foranderlige projektets omgivelser (og deres krav) er, jo mere oplagt er det at udvikle agilt, netop fordi der hele tiden justeres ind efter de forretningsmæssige mål og gevinster, der giver bedst mening i forhold til forretningen løbende.

- Jo mere presset projektet er på tid i gennemførelsesfasen, jo mere oplagt at udvikle agilt. Med agil udvikling, hvor udvikler projektet i hvert sprint det mest værdiskabende. Dermed har projektet større chance for at nå i mål med et system, der rammer højt på forretningsunderstøttelse.
- Jo mere man kan samle de centrale roller og udviklingsteamet på fælles lokationer, også når der er en ekstern leverandør, jo mere oplagt er det at udvikle agilt.

I PRINCE2 Agile har man ligefrem udviklet et Agilometer¹, hvor man på seks parametre kan score organisationen og en kommende opgaves velegnethed til agil udvikling. De parametre PRINCE2 Agile her benytter kan også meget vel bringes i overvejelse, når organisationen (eller en del af den) og dens rammer skal vurderes for ”agil-parathed”:

Figur 3.1 PRINCE2 Agile Agilometer



¹ Se fx http://prince2agile.wiki/The_Agilometer

4. Minimumskrav for projekter, der ønsker at benytte agile udviklingsmetoder

For projekter over 15 mio. kr. er der nogle minimumskrav, der skal følges, hvis en myndighed ønsker at benytte agile metoder til udvikling af hele- eller dele af it-løsningen.

For projekter under 15 mio. kr. kan disse retningslinjer opfattes som anbefalede. Mindre projekter kan være velegnede til at afprøve agile udviklingsmetoder på. Dermed kan mindre projekter give organisationen den erfaring, der er en forudsætning for senere også at benytte agile udviklingsmetoder på større projekter over 15 mio. kr.

Minimumskrav for projekter over 15 mio. kr. kan sammenfattes i følgende retningslinjer:

Krav, behov og ønsker til it-løsninger **prioriteres** ud fra, hvor meget forretningsmæssig værdi (gevinster), de hver især bidrager til. De prioriterede krav opføres sammen med nødvendige tekniske krav i en **backlog**, som giver den rækkefølge it-udviklingen følger.

Der udvikles i antal **sprints**, i princippet indtil der er udviklet tilstrækkeligt, eller tid og/eller ressourcer/økonomi slipper op. Projekter med agil udvikling har i udgangspunktet låst de to dimensioner fra projekttrekanten; tid og økonomi.

Hvert sprint afsluttes med et **produktreview**, hvor den funktionalitet, der udvikles i sprintet fremvises og godkendes. Der gennemføres et efterfølgende **procesreview**, hvor teamets egne interne processer evalueres, og hvor det vurderes, hvordan teamet kan arbejde endnu bedre fremadrettet.

Projektet udarbejder en **releaseplan**, der med en relevant kombination af agile og ikke-agile udviklingsmetoder viser hvilken hovedfunktionalitet, der forventes frigivet hvornår.

Agil it-udvikling kræver et meget tæt samarbejde mellem udviklingsteamet og den del af forretningen, der skal aftage it-løsningen. En række centrale opgaver i dette samarbejde er samlet i **rollen Product Owner**.

Minimumskravene er alle centrale retningslinjer for agile udviklingsmetoder.

Det er et krav, at myndigheder, der ønsker at udvikle it-systemer agilt i projekter til over 15 mio. kr. har en passende modenhed og erfaring i agil udvikling. Denne erfaring kan indhentes fra et antal mindre projekter, hvor agile retningslinjer, roller og evalueringer er prøvet af og løbende optimeret.

Minimumskravene er uddybet i de efterfølgende afsnit.

4.1 Prioritering

I en verden hvor projektet ikke har ubegrænset tid og ressourcer, er én af pointerne med agil it-udvikling at sikre, at projektet får udviklet det, der understøtter realiseringen af de største gevinster først og dermed har udviklet det mest værdifulde, når tiden og/eller ressourcerne slipper op.

De agile udviklingsmetoder anvender Skal/Bør/Kan-prioritering², hvor:

Skal

Skal-prioriteter hænger direkte sammen med de bærende gevinster, som projektet har prioriteret højest. Uden udvikling af Skal-prioriteterne vil projektet ikke kunne opnå de forventede gevinster og projektets berettigelse kan smuldre. Skal-prioriteter kan fx være opgaver, hvor:

² Inspireret af MoSCoW prioritering (Must, Should, Could, Would)

- Den pågældende del af systemet er grundforudsætning for resten af systemet.
- Det ikke giver mening at levere produktet uden den pågældende funktionalitet, da systemet i så fald ikke vil virke.
- Produktet ikke overholder gældende lovkrav uden den givne funktionalitet.
- Produktet er farligt eller uforvarligt at bruge uden den pågældende funktionalitet.

Det er afgørende, at udviklingen af Skal-kravene ikke forventes at bruge hele den tidsramme eller alle de ressourcer, som projektet råder over. Projektets risici stiger voldsomt, hvis hele udviklingsrammen er fyldt ud med Skal-funktionalitet, da projektet i så fald ikke har noget at tage bort, såfremt der sker noget uforudset.

Den samlede mængde af Skal-prioriterede krav er at betragte som projektets Minimum Viable Product (MVP) og består af det absolut minimale som projektet kan levere og samtidig være en succes.

Bør

Funktionalitet, som er Bør-prioriteret, er funktionalitet, som er vigtig, men som ikke er afgørende for at få en første version af systemet i produktion og som ikke bidrager til, at de højest prioriterede gevinster realiseres. Bør-prioriterede opgaver kan eksempelvis være opgaver, hvor:

- Funktionaliteten underbygger gevinster, som ikke er de bærende, højest prioriterede, men dog væsentlige for den samlede business case.
- Systemet kan lægges i produktion uden den pågældende funktionalitet, fordi der kan etableres en work-around på funktionaliteten.
- Støttefunktionalitet, som vil gøre systemet væsentligt mere anvendeligt, men hvor systemet kan anvendes i en kortere periode.

Kan

Den laveste prioritet, som den agile udviklingsmetodik arbejder med, er Kan-prioriteten³. Kan-prioriterede krav er krav på funktionalitet, som giver noget lækkert, men ikke bidrager til de prioriterede gevinster i nævneværdig grad.

Eksempler kunne være:

- At systemet automatisk formaterer tal på skærbilleder.
- At der er mouseover på enkelte felter eller funktionalitet, der automatisk retter stavfejl.
- At farver på et skærbillede ikke er helt ens ift. øvrige skærbilleder i systemet.

Fordeling af prioriterede opgaver

Projektet er pr. definition et højrisikoprojekt, hvis alle krav er Skal-krav, eller hvis tidsplanen er så stram, at projektet næsten med sikkerhed kun lige akkurat kan nå igennem Skal-kravene.

En tommelfingerregel er, at en god fordeling af Skal-, Bør- og Kan-funktionalitet er omkring 60 pct. - 20 pct. - 20 pct., hvilket både vil give en vis sikkerhed for, at alt Skal-funktionalitet bliver udviklet, og samtidigt at projektet når pænt ned i Kan-funktionaliteten før tiden eller ressourcerne slipper op.

4.2 Backlog

Alle krav i projektet – forretningsmæssige, tekniske og sikkerhedsmæssige - skal lægges i en backlog, som Product Owner'en med input fra forretningen prioriterer med Skal/Kan/Bør, så det til enhver tid er klart, hvad der er højest prioriteret fra forretningens side, og dermed det næste, der udvikles.

³ Would have (in another release), er reelt et fravalg, eller ”må ligge over til en anden gang”, er taget ud i statens it-projektmodel for at holde det simpelt.

En backlog er en komplet prioriteret liste over de produkter/user stories⁴, fejl, udeståender og hvad projektet ellers skal lave. Det er Product Owner, der ejer backloggen og har ansvaret for at sikre, at der altid er opgaver til teamet og at opgaverne er beskrevet tilfredsstillende.

Ved ændringer i udviklingen eller udfordringer med at gennemføre Skal-prioriterede opgaver, vil det være Kan-prioriterede opgaver, som tages ud først og derefter Bør-opgaver, såfremt projektet ikke kan nøjes med at tage Kan-opgaverne ud.

4.3 Sprints

Sammen med Skal/Bør/Kan-prioriteringen gør sprints det muligt at sikre, at projektet løbende leverer værdi og hele tiden kan justeres ind efter forretningens behov. Ved at styre efter en kort tidshorizont og have prioriteret alt arbejde er det muligt at holde styr på fremdrift samtidig med, at de agile fordele med løbende tilpasning og prioritering er i fokus. Sprints anbefales at være på to til fire uger, maksimalt fire til seks uger. Længden af sprintet bestemmes både ud fra projektets beskaffenhed og længde; i korte projekter vil det som regel give mening at have korte sprint, for at få flest mulige gennemløb og mest mulig læring ud af tiden.

Under alle omstændigheder anbefales det, at kadencen fastlægges, således at det er muligt over tid at se, hvor produktivt teamet er inden for de enkelte sprints.

Det anbefales, at der afsættes tid til adskillige mindre møder internt i teamet gennem et givent sprint, hvor det udviklede produkt kan reviewes i teamet og tilpasses tilsvarende.

Ønskede produkter eller user stories brydes ned til et niveau, hvor det kan udvikles færdig i et enkelt sprint. På denne måde er det helt færdige produkter, som vil blive leveret i et enkelt givent sprint. Samtidigt vil nedbrydningen hjælpe projektet med at få gjort udviklingsopgaverne overskuelige og lettere at estimere.

Hvis det skal være muligt at lave helt færdige produkter i et givent sprint, så skal teamet være i stand til at udføre alle faser af udviklingen i selve teamet – analyse, design, udvikling, test og evt. idriftsættelse.

Et sprint er således et fast tidsrum, hvor outputtet er et eller flere helt færdige produkter eller delløsninger, som er testet, godkendt og klar til at blive lagt i produktion eller i et produktionslignende miljø. Hvis de udviklede produkter ikke er helt færdige i et givent sprint eller ikke bliver godkendt på produktreviewet, bliver de lagt tilbage i backloggen, så de kan blive prioriteret ind sammen med de øvrige opgaver efterfølgende.

4.4 Produktreview

Ved afslutningen af hvert sprint skal teamet lave et Produktreview af det udviklede produkt sammen med Product Owner og repræsentanter fra forretningen og eventuelle slutbrugere.

Formålet med et Produktreview er:

- at få godkendt de udviklede produkter af Product Owner og repræsentanter fra forretning og slutbrugere.
- at vise interessenterne, at teamet har konstant fremdrift gennem fremvisning af anvendelige produkter og funktionalitet.
- ikke kun beskrive, men faktisk vise det udviklede i en kørende løsning.
- at forventningsafstemme og få feedback på det udviklede fra en bredere interessentgruppe, inklusive forretningen, slutbrugerne og eventuelle øvrige inviterede interessenter. Derved identificeres misforståede krav og eventuelle ændringer noteres, så produkterne kan skabe mest mulig værdi.
- at få godkendt de udviklede produkter og få lukket sprintet officielt af Product Owner.

Det kan være en god idé at lave en demonstration internt i teamet før produktet vises frem for en større interessentskare, så eventuelle user stories eller produkter, som ikke fungerer, bliver identificeret og kan rettes op før afholdelsen af Produktreviewet.

⁴ User stories er en agil metodik til at bryde krav ned i en struktur, som sikrer sammenhæng mellem brugere, funktionalitet og forventede gevinster.

4.5 Procesreview

I forlængelse af projektreviewet skal der afholdes et procesreview⁵ i de enkelte udviklingsteams. I et procesreview arbejder teamet med deres interne processer og har mulighed for at afprøve forbedringer til processerne i de efterfølgende sprints, således at teamet konstant arbejder mod at blive bedre og mere effektive samt leverer det forventede.

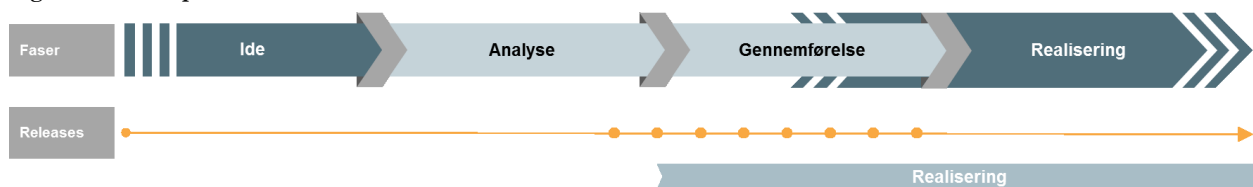
4.6 Releaseplan

Som en del af, eller et bilag til den overordnede projektplan, skal der udarbejdes en releaseplan⁶. Releaseplanen skal vise indholdet i de planlagte releases i projektet. Releaseplanen skal, ligesom resten af projektplanen, indeholde forudsætninger og overordnede estimater.

Releaseplaner kan være i samme format som standard projektplaner, men viser indholdet i de releases, som projektet forventer at levere, og hvilke prioriteter, der er på de overordnede produkter/features/epics⁷ eller produkter og arbejdsplaner i planen.

Releaseplanen er et øjebliksbillede, som på et givent tidspunkt fortæller, hvad projektet forventer at nå i prioriterede blokke. I tråd med de agile udviklingsmetoder er der stor sandsynlighed for, at projektet og forretningen bliver klogere over tid og at projektets prioritering kan ændre sig.

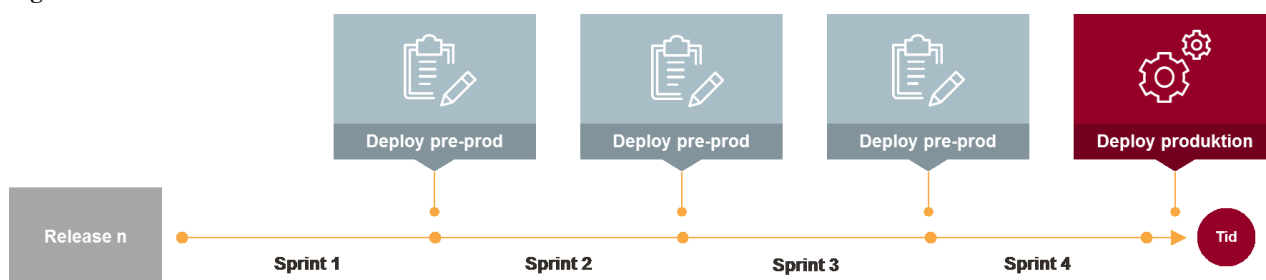
Figur 4.1 Releaseplan



Ovenstående illustration viser, hvordan de planlagte releases i eksemplet ovenfor findes inden i gennemførelsesfasen i statens it-projektmodel. Når der stilles skarpt på gennemførelsesfasen, så ses det, at hver release kan indeholde et antal sprints, således at en release består af et eller flere sprints, som skal være gennemført for, at produktet videreføres til produktion eller et produktionslignende miljø.

Antallet af releases samt sprints i releases er afhængigt af projektets beskaffenhed og organisations rytme⁸. Hvis organisationen laver fire årlige tværgående deployments til produktion og de valgte sprints er på tre uger, vil der altså typisk være fire sprints i en release:

Figur 4.2 Releases



Som det fremgår af illustrationen lægges der i det tænkte eksempel, i pre-prod eller lignende miljø efter de tre første releases, hvorefter der lægges samlet i produktion ved afslutningen af sprint 4, således at projektets releases rammer organisationens rytme.

⁵ Kaldes også for Retrospective

⁶ Nogen arbejder med begrebet roadmap

⁷ Features = en sammenhængende klump af user stories. Epics = overordnede produkter eller funktionalitet

⁸ Mange organisationer samler ny funktionalitet på tværs af projekter i fx fire årlige releases/deployments til produktion.

Releaseplanen skal dokumenteres i projektgrundlaget samt i den overordnede projektplan.

4.7 Rollerne Product Owner og Scrum master

Der er to nye roller i statens it-projektmodel, som understøtter det agile udviklingsarbejde, Product Owner og Scrum Master/teamleder. Begge roller er obligatoriske, men hvis der er en ekstern udviklingsleverandør på projektet, vil det som oftest være en af leverandørens medarbejdere der varetager Scrum Master-rolle.

Product Owneren har ansvaret for:

- at der er en aktiv forbindelse mellem udviklingsteamet og kravstillerne. Den ansvarlige Product Owner skal kunne fungere som gevinstejers forlængede arm og bærer gevinstejers mandat i det daglige arbejde og i forhold til at prioritere, hvad teamet skal arbejde med i de givne sprints.
- at indsamle og få prioriteret krav og behov og sikre, at disse bliver dokumenteret på en måde, der gør det muligt for udviklingsteamet at forstå opgaverne og stadig have plads til at udvikle den bedst mulige løsning.
- at fungere som kontaktpunkt i forhold til forretningsmæssige afklaringer og med hensyn til tilretning i den daglige udvikling. Som et effektivt bindeled mellem forretning og udvikling skal Product Owner have indgående forståelse og kendskab til domænet og gerne erfaring fra linjen i den kravstillende styrelse, således at krav og afklaringer bliver pragmatiske og anvendelige for slutbrugerne.

Hvis en ekstern leverandør udvikler agilt, og det aftales at samarbejde efter agile retningslinjer, anbefales det, at der udpeges en Product Owner i organisationen, som har den daglige kontakt med leverandøren og indgår i teamet på samme måde, som hvis der er tale om et internt agilt udviklingsprojekt.

I agile udviklingsorganisationer skal følgende rolle besættes, uanset om rollen udfyldes af en ansat i organisationen eller fra leverandøren:

Scrum Master er en rolle i selve udviklingsteamet, som faciliterer afholdelse af workshops og møder samt sikrer, at teamet får arbejdsro og de interne spilleregler overholdes.

Der findes en uddybende beskrivelse af disse to agile roller i temavejledningen *Projektorganisationens roller og ansvar*, der findes [her](#).