

Mod en agil kultur

plan-drevet eller agil?



**Dette whitepaper udspringer af dialoger med deltagere på et agile forum etableret af Peak.
Forfatterne ønsker derfor at takke:**

Charlotte Gall, Digitaliseringsstyrelsen
Dan Sten Rexen, Arbejdsmarkedsstyrelsen
Jesper Rindshøj, Nordea Liv & Pension
Kim Holm, Forsvarets Koncern IT
Peter Lind Nielsen, Bird & Bird
Søren Stener, Erhvervsstyrelsen

For yderligere inspiration

- M. Sahota, An Agile Adoption and Transformation Survival Guide: Working with Organizational Culture. InfoQ, 2012.
- Economic Intelligence Unit, "Organisational Agility," Mar. 2009.
- D. Reinertsen and S. Thomke, "Six Myths of Product Development," Harvard Business Review, May-2012.
- B. Boehm, "Get ready for agile methods, with care," Computer, vol. 35, no. 1, pp. 64–69, 2002.
- Sutherland, J. (2010). Scrum Handbook. The Scrum Training Institute.

Indholdsfortegnelse

1.0	INDLEDNING	4
2.0	PLAN-DREVET ELLER AGIL ?	6
2.1	BARRIERER VED IMPLEMENTERING AF AGILITET	10
3.0	TO CASES FRA VIRKELIGHEDEN	13
3.1	SKAT IT	13
3.2	SCHULTZ	15
4.0	KONKLUSION	18
5.0	OM FORFATTERNE	19

1.0 Indledning

Udviklingsmetoder til komplekse opgaver må vælges med udgangspunkt i den enkelte organisations aktuelle projektmodenhed, kultur, styringsmekanismer samt strategi. Det er én af de centrale pointer i dette whitepaper, som også beskriver hvilke konkrete overvejelser enhver organisation bør tage i betragtning, i forbindelse med valget af udviklingsmetode, til en given opgave.

Valget mellem plandreven- eller agiludviklingsmetode er afgørende for projekternes succes og for hvornår gevinstrealiseringen kan indfries.

Det uendelige krav om en højere organisatorisk værdiskabelse leveret i et højere tempo ved brug af færre ressourcer udfordrer en lang række organisationers nuværende måde at arbejde i projekter på – Samtidig vinder den agile metode større indpas i ledelserne som et svar på at skabe værdi i højere tempo samtidig med at kunne håndtere forandringer og ændringer til prioriteter.

En undersøgelse fra Standish Group, Boston¹ der omfatter mere end 100.000 projekter, viser, at udviklingsprojekter, hvor den agile metode blev anvendt, var 3 gange mere succesfulde ift. aftalt tid og omkostninger, sammenlignet med projekter, der har benyttet sig af traditionelle metoder.

I dette whitepaper vil vi komme tættere på den agile metode, og se på hvornår og hvordan du som erhvervsleder kan skabe fordel af den agile måde at arbejde på.

Erfaringen viser, at udviklingsmetoder til komplekse opgaver må vælges med udgangspunkt i den enkelte organisations aktuelle projektmodenhed, kultur, styringsmekanismer samt strategi. Det er én af de centrale pointer i dette whitepaper, som også beskriver hvilke konkrete overvejelser enhver organisation bør tage i betragtning, i forbindelse med valget af udviklingsmetode til en given opgave.

Større udviklingsopgaver initieres ofte på baggrund af en positiv business case og en 'return of investment kalkyle'. Gevinsterne betyder mere og mere, uanset der er tale om en offentlig eller privat organisation.

I mange organisationer er valget af udviklingsmetode imidlertid styret af den enkelte projektleders interesser eller ønsker om at 'nørde' med metodikker, der er oppe i tiden, eller som den enkelte føler sig tryk ved, frem for at bygge på en af organisationen fastsat strategi for hvordan udviklingsopgaver skal håndteres.

Kravet til hvormed hastigheden en udviklingsopgave skal gennemføres med, og den kompleksitet den gennemføres med er steget i takt med teknologiens udvikling igennem de sidste årtier, hvilket også har haft den konsekvens, at der er mange 'tunge' viden-medarbejdere fra forskellige fagområder knyttet til opgaveløsningen. Samtidig er kravet om effektivisering og omkostningsminimering taget til i samfundet generelt.

¹ The Standish Group, "CHAOS Manifesto 2011,"

Valget af udviklingsmetodik har derfor en afgørende betydning for beregningen af udviklingsomkostninger, og skal derfor indgå i business case og return of investment kalkulationen.

De forskellige udviklingsmetoder rummer meget forskellige operationelle og strategiske muligheder for organisationens opgaveløsningsdesign, og stiller på samme tid forskellige krav til organisationens styring, modenhed, faglighed, kultur, omverden og ikke mindst opgaven og dens karakteristika.

Den plandrevne udviklingsmetode er i dag velkendt og vel-beskrevet, hvorimod den agile udviklingsmetode stadig synes at udgøre en størrelse mange taler om, men få er enige om, på trods af, at mange benytter sig af den.

Overordnet kan den agile udviklingsmetode beskrives som en meget kraftfuld tankegang. Den er et effektivt værktøj til at reagere på en situation, hvor organisationen er forpligtet til at levere værdi i et hurtigt tempo, og samtidig beskæftige sig med et højt niveau af forandringer og usikkerhed.

Den agile udviklingsmetode forsøger at tage hånd om den plandrevne metodes akilleshæl, hvor det ikke er muligt at vide alt fra opgaveinitieringstidspunktet. Der er derfor brug for en udviklingsstrategi, der understøtter muligheden for, at opgaveløsningsdesignet løbende kan tilpasse sig forandringer, og kunden undervejs skal kunne indfri dele af realiseringen og ikke først i slutningen af den samlede opgaveløsning.

Omdrejningspunktet i den agile tankegang er således den tidlige indfrielse af realiseringsgevinsten, baseret på en løbende leverancestrøm til kunden, og det med udgangspunkt i kundens aktuelle behov på rette tid og sted. Dette fordrer i langt større grad, end i den plandrevne model, et tæt tillidsbaseret samarbejde mellem kunden og leverandøren.

Vi vil i det efterfølgende beskrive, hvordan og hvornår organisationen kan drage fordel af den agile måde at arbejde på. Herunder hvilke overvejelser enhver direktion eller opgave- hhv. gevinstejer med fordel bør gøre sig i forbindelse med initiering af en udviklingsopgave, organisationen påtænker at gennemføre efter de agile principper. Og i samme regi beskrive hvilke krav det stiller til centrale organisatoriske faktorer.

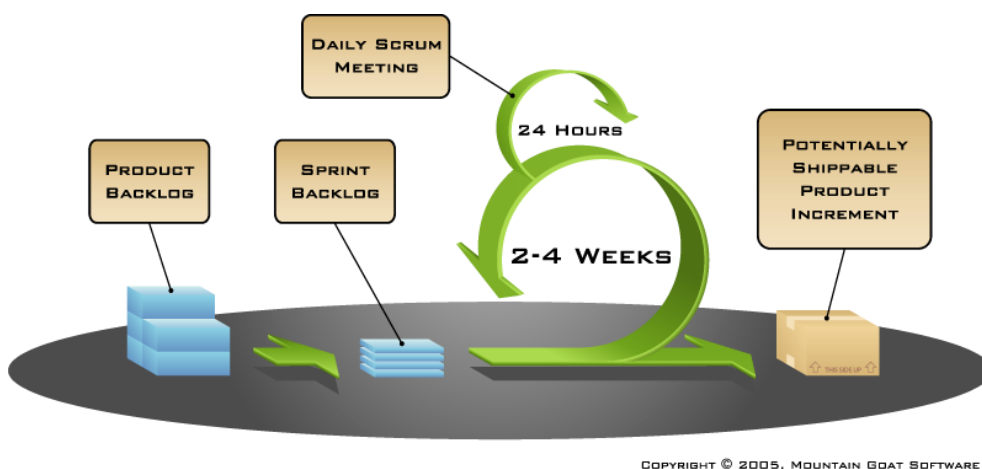
2.0 Plan-drevet eller Agil ?

I 2001 mødtes 17 repræsentanter for en række forskellige softwareudviklingsmetoder i Snowbird Ski Resort, Utah, USA. Gruppen som kaldte sig 'The Agile Alliance', bestod bl.a. af Alistair Cockburn, Jim Highsmith, Ken Schwaber, Kent Beck og Jeff Sutherland, der hver især var blandt ophavsmændene til Agile metoder, heriblandt SCRUM, XP og Crystal. Resultatet af gruppens anstrengelser var 'Det Agile Manifesto'², der lige siden har præget landskabet indenfor softwareudvikling i IT branchen enormt.

Blandt de væsentligste faktorer bag manifestet var ønsket om et større fokus på, at softwareudviklingsarbejdet i højere grad blev samordnet og strømlinet med forretningen, samt behovet for et skifte i de sociale strukturer i, og omkring, det enkelte softwareudviklingsteam. Manifestet havde 4 hovedpunkter, hvor man så 'værdi i punkterne til højre, men vi værdsætter punkterne til venstre højere'. Hovedpunkterne var:

- Individier og samarbejde frem for processer og værktøjer
- Velfungerende software frem for omfattende dokumentation
- Samarbejde med kunden frem for kontraktforhandling
- Håndtering af forandringer frem for fastholdelse af en plan

For mange er SCRUM den metode der først tænkes på når ordet 'Agile' bliver nævnt, og SCRUM er i dag repræsenteret i op mod 74% af organisationers metodevalg³ når der udvikles software agilt. Kendetegnende for SCRUM og andre agile metoder (så som DSDM, XP og Crystal) er en iterativ tilgang, hvor man leverer produktet i mindre bidder, fremfor at vente til alle features er færdigudviklede. Dette opnås ved at aftale såkaldte 'iterationer', ofte med en længde fra 14 dage til 4 uger, hvor indholdet er udsat for en skarp prioritering, der er aftalt mellem kunden og teamet.



Figur 1:
SCRUM forløb

² Kent Beck et al, "Agile Manifesto," *Agile Manifesto*, 2001

³ VersionOne, "7th-Annual-State-of-Agile-Development-Survey.pdf," 2013.

Fordelene ved dette er bl.a., at kunden hurtigere kan vurdere de leverede features og hvis ønsket foreslå ændringer, der gør produktet, eller den givne feature, endnu bedre. Effekten af denne tilgang er større fleksibilitet og omstillingsparathed, da man kontinuerligt specificerer, estimerer, udvikler og tester mindre bidder af funktionalitet. Derefter vurderes de færdige features sammen med kunden, for at verificere om leverancen stemmer overens med forventningerne. Hvis ikke, beskrives de nødvendige ændringer således de kan prioriteres med i en kommende iteration.

Af samme årsag, og dette er med til at betegne agile metoder som 'lette', venter man med at producere den nødvendige dokumentation til sent i forløbet, for ikke at spille tid og ressourcer på dokumentation, der skal ændres flere gange. Det er denne fleksibilitet, og mulighed for tidligt feedback, der er med til at gøre forretningsområder med stor dynamik, kompleksitet og behov for innovation, til agile metoders hjemmebane.

Det agile manifest har dog ikke kun bragt ændringer i måden udviklingsarbejdet nedbrydes og eksekveres på, men også i strukturerne omkring softwareudviklingsteams. Det enkelte team har i agile metoder en meget selvstændig og selvstyrende rolle og SCRUM Master rollen er først og fremmest en facilitator for teamet, mere end en reel projektleder, der dikterer gennemførelsen af aktiviteter.

Denne arbejdsform stiller til gengæld også større krav til kompetencerne i teamet hvis man ønsker at arbejde agilt, da man beror sig på en høj grad af implicit viden, forankret i de enkelte medlemmer, frem for eksplicit viden, i form af dokumentation og beskrevne processer. Samtidig forsøger man, i agile metoder, at tilbyde generelle retningslinjer, i stedet for specifikke processer, hvilket også er afspejlet i manifestets relativt løse formuleringer.

Idet indholdet af de enkelte iterationer aftales mellem kunden og teamet, ud fra kundens prioritering, stiller anvendelse af agile metoder også større krav til kundens evne til at tage ansvar og prioritere. Her ligger netop nøglen til agile metoders fleksibilitet; I princippet revurderes prioriteringen af det arbejde der skal udføres, for hver iteration. Man fastlåser altså ikke rækkefølgen af funktionalitet eller features langt frem i tid, som der ellers er tradition for i en plan-drevet tilgang.

Den klassiske plan-drevne (vandfaldstankegang) for udvikling af software, fungerer ved at afvikle sekventielle faser med hver deres individuelle fokus på kravspecifikation, analyse, design, udvikling, test og overdragelse til drift. I sin mest rigide form tilgår plan-drevet udvikling opgaven ud fra en forventning om, at det er muligt at specificere brugerens krav før næste fase indledes og man designer løsningen ud fra kravene. Herefter udvikler man den designede løsning, efterfulgt af test og fejlretning, inden man til slut overleverer til drift. Den plan-drevne tilgang har sit udgangspunkt i ingeniørfeltet og hårdere systemisk tankegang, og målet er at kunne eksekvere forudsigeligt ud fra optimerede processer og arbejdsgange.

Kort sagt kræver denne tilgang anderledes forudsætninger end den agile: Der skal være stabilitet omkring kravene, da omkostningerne ofte er høje grundet den sekventielle tilgang. Den plan-drevne tilgang kaldes også for 'procesdrevet' og 'tung', da man her håndterer viden meget eksplicit i kraft af dokumentation som forretningsgange, Use Cases og UML diagrammer.

Dokumentationen er et bærende element i at kunne levere produktet forudsigeligt. Dette bidrager også til at skabe større omkostninger i forbindelse med ændringer, som skal afspejles i dokumentationen. Typisk er den plandrevne tilgang stærkt knyttet til en hierarkisk opdelt organisation, hvor det er projektlederen som uddelegerer, kontrollerer og følger op på fremdriften i forhold til projektets plan. Planen, og dens data, bliver således hurtigt omdrejningspunktet for projektet.

Nedenstående tabel beskriver i opsummeret form en række af de forskelligheder, der er mellem de plan-drevne og agile tilgange. Bemærk, at det er yderpunkter, som beskrives, og at det i mange organisationer er lykkedes at finde gode kompromisser mellem de to tilgange⁴:

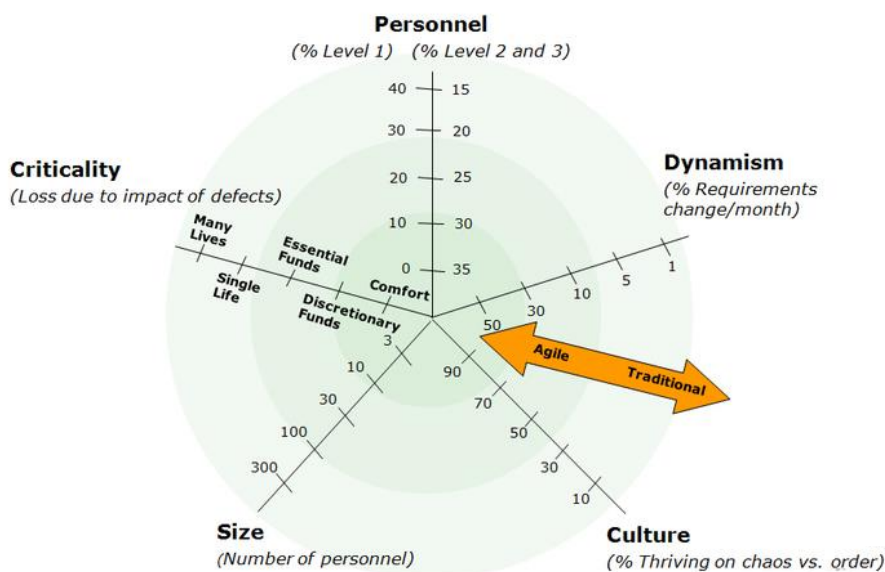
	Agile	Plan-drevet
Fundamentelle formodninger	Adaptiv software af høj kvalitet kan udvikles af små teams ved anvendelse af principperne om kontinuerlig designforbedring og test baseret på hurtig feedback og ændringer	Systemer er fuldt specificerbare, forudsigelige og kan leveres ved detaljeret og omfangsrig planlægning
Kontrol	Personbaseret	Procesbaseret
Ledelsesstil	Ledelse og samarbejde	Kommando-og-kontrol
Videnstyring	Implicit	EksPLICIT
Rolletildeling	Selv-organiserende teams, opfordrer til vekslen i roller	Individuel, ser positivt på specialisering
Kommunikation	Uformel	Formel
Kundens rolle	Kritisk	Vigtig
Projekt cyklus	Styret af produktfeatures	Styret af opgaver og aktiviteter
Ønsket organisationsstruktur	Organisk (Fleksibel og deltagende, opfordrer til samarbejde om sociale aktiviteter)	Mekanistisk (Bureaukratisk med høj grad af formalisering)

Figur 2:
Sammenligning af agile med plan-drevet tilgang

Udover at have forskellige karakteristika, har de to tilgange også forskellige hjemmehavere. De fungerer bedst i forskellige miljøer, der ofte har forskellige forudsætninger. Boehm og Turner har defineret en række faktorer, der kan anvendes til at hjælpe med at vurdere, hvilken tilgang, eller blanding, der egner sig bedst til opgaven⁵. Jo tættere man kommer ind mod midten i figuren, desto bedre vil en agil tilgang trives.

⁴ S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of migrating to agile methodologies," *Communications of the ACM*, vol. 48, no. 5, pp. 72–78, 2005.

⁵ B. W. Boehm and R. Turner, *Balancing agility and discipline: a guide for the perplexed*. Boston: Addison-Wesley, 2004.



Figur 3:
"Balancing Agility with Discipline". Barry Boehm, Richard Turner.

Faktor	Agile	Plan-drevet
Størrelse	Et godt match til mindre produkter og teams. Kravet til implicit viden begrænser evnen til at skalere.	Metodikker der er udviklet over tid til at håndtere store produkter og projekter. Svært at skalere ned til mindre produkter eller projekter.
Kritikalitet	U-testet på sikkerhedskritiske produkter. Potentielle problemer ved simpelt design og manglende dokumentation.	Metodikker der er udviklet over tid til at håndtere stærkt kritiske produkter. Svært at skalere ned til knapt så kritiske produkter.
Dynamik	Simpelt design og kontinuerlig re-faktorering excellerer i stærkt dynamiske miljøer, men er en kilde til potentielt dyrt omarbejde i meget stabile miljøer.	Detailede planer og 'Meget Design Før Vi Begynder' excellerer i stabile miljøer, men er en kilde til meget dyrt omarbejde i dynamiske miljøer.
Personale	Kræver kontinuerlig tilstedeværelse af en kritisk masse af Cockburn Level 2 eller 3 eksperter. Risikabelt at anvende ikke-agile Level 1B personale.	Kræver en lille kritisk masse af Cockburn Level 2 og 3 eksperter under definitionen af projektet, men kan arbejde med færre senere i projektet. Medmindre konteksten er meget dynamisk. Kan som regel håndtere noget Level 1B personale.
Kultur	Trives i en kultur hvor personer er komfortable og motiverede af en stor grad af frihed.	Trives i en kultur hvor personer føler sig komfortable og motiverede ved at deres roller er defineret af klare politikker og procedurer.

Figur 4:
Perspektivering på agile og plan-drevet tilgang

Den agile tilgangs hjemmebane findes, når antallet af ressourcer og interessenter er begrænsede, og når konsekvensen af fejl i systemet ikke har for voldsomme konsekvenser. Derudover håndterer den agile tilgang bedre dynamik i form af ændringer, i en kultur som er præget af turbulens, mens kravet om kompetente ressourcer stadig er et must, grundet den lette grad af dokumentation.

Den plan-drevne tilgangs hjemmebane findes, når kulturen trives bedst med orden, og når der er få ændringer til det planlagte. Derudover håndterer den plan-drevne tilgang bedre større projekter med sin hierarkiske struktur, og pga. dokumentationsniveauet kan man bedre håndtere ressourcer med en mindre implicit viden. Systemer, hvor konsekvensen af fejl er høj, er også en hjemmebane for den plan-drevne tilgang i det både proces og dokumentation er beskrevet eksplicit.

Ovenstående tabel og diagram demonstrerer også, at et projekts anvendelse af den agile eller plan-drevne tilgang bør vurderes og tilpasses. Det er ikke nødvendigvis 'one size fits all', hvor man kun kan vælge den ene eller den anden tilgang. Men for at man kan anvende den rigtige blanding, er man nødt til at kende og have erfaring med begge tilgange. Mange organisationer kender i dag til den plan-drevne tilgang, men har lidt, eller slet ingen erfaring med den agile tilgang. Vi har derfor valgt at fokusere på de barrierer der kan besværliggøre implementeringen af en agil tankegang.

2.1 Barrierer ved implementering af agilitet

Ser man for en stund bort fra konkrete metodikker, værktøjer og organisatoriske faktorer, hvad betyder det så at have en 'agil tankegang'? Og hvorfor er der et stigende behov, blandt organisationer, for at ændre tilgang?

Alistair Cockburn⁶, en af Det Agile Manifests forfattere, har forsøgt at sammenfatte ånden bag agil metodik som: "*A system of methods designed to allow the development team to match and track the business needs, especially in a context where business needs change frequently, important facts change, or where we are obliged to adapt to important uncontrolled factors.*"

Blandt årsagerne til denne tankegang ligger den globale udvikling gennem de senere år. Udviklingen har betydet, at mange markeder fungerer ved højere og højere tempo, ofte drevet af en aggressiv teknologisk udvikling. Disse forhold skaber turbulens, og derfor er det for mange organisationer svært at skabe succes med en plan-drevet tilgang, idet man har behov for at kunne foretage ændringer uden for store omkostninger for at kunne følge med markedet.

Derudover er det, når der er stor omskiftelighed eller usikkerhed i parametrene for succes, svært at forudsige, hvordan projektets produkt bedst leverer værdi, og derved drage fordel af langsigtet planlægning. I sådan et tilfælde fungerer en tilgang med hyppig 'trial and error' bedre.

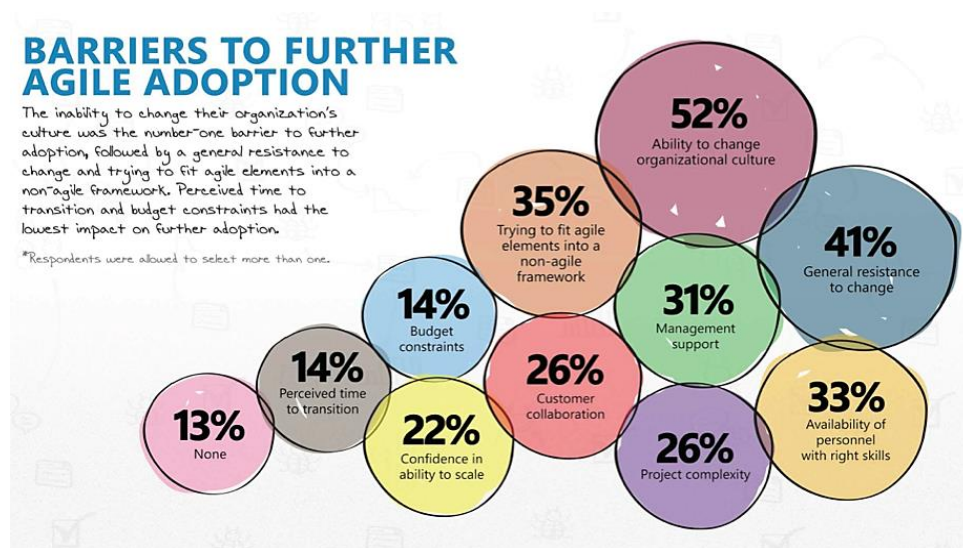
Usikkerhed og dynamik er altså i mange forretningsområder normen, og det fremtvinger en anderledes tilgang når IT udviklingsprojekter skal eksekveres. Men det er ikke blot projektet, men også den organisation og kultur det eksisterer i, der skal evne at håndtere den øgede grad af dynamik og usikkerhed.

Netop tankegangen er derfor vigtig for den forståelse, der skal være ledestjerne for at skabe en kultur, der evner at være agil. Ikke blot i det enkelte udviklingsteam, men også i projektets styregruppe og interessentkreds.

⁶ Alistair Cockburn, "Notes on the writing of the Agile Manifesto," Alistair Cockburn, 2009.

Dette giver derfor anledning til at definere en agil tankegang i et bredere perspektiv: *Et forretningsorienteret fokus på at skabe produkter der leverer den ønskede værdi, i kraft af en transparent og videns baseret samarbejdsstruktur, der effektivt kan skabe, og reagere på, forandring i en turbulent og dynamisk verden præget af omskiftelige faktorer.*

For de fleste organisationer kræver overgangen til en sådan tankegang en decideret strategisk prioriteret kulturændring, hvis der skal opnås succes med at implementere agile metodikker. Og det kan være svært, som understreget i en undersøgelse foretaget af Version One⁷ i 2013. Her nævnte de adspurgte organisationer således 'Evnene til at ændre organisationens kultur' som værende den største forhindring for yderligere forankring af agilitet, ligesom 'Generel modstand over for forandring' lå højt placeret.



Figur 5:
Barrierer imod anvendelsen af agile principper

En undersøgelse udført af Peak Consulting Group, blandt deltagere på et 'Agile Forum', fandt følgende vigtige erfaringer fra deltagernes organisationer, i relation til anvendelse af en agil tilgang:

- Organisationens kultur skal evne at tænke agilt
- Det kræver en moden organisation at slippe 'vandfaldsillusionen' (om at kunne forudse alt fra dag 1)
- Tættere og mere integreret samarbejde i projektet mellem kunde og leverandør
- Organisationen skal turde fejle; der skal være en margin for fejltagelser
- Tillid til projektorganisationen og ligeledes mellem kunde og leverandør
- Uddelegering af ansvar og beslutningskompetence giver råderum
- Fokus på løbende prioritering / omprioritering inden for det etablerede råderum
- Opgaven, og ikke politik, i centrum

⁷ VersionOne, "7th-Annual-State-of-Agile-Development-Survey.pdf," 2013.

Samtidig viser Boehm og Turners arbejde, at der ikke er tale om at vælge sort-hvidt mellem enten en plan-drevet eller en agil tilgang.

Men skal en overgang til at anvende agile metoder i organisationens tilgang til softwareudvikling, have succes, kræver det langt mere end procesbeskrivelser og værktøjer. Agile skal altså ikke ses som et værktøj, et produkt eller en proces i sig selv. Det er en tankegang, der kræver en specifik kultur for at skabe værdi.

3.0 To cases fra virkeligheden

I dette afsnit beskriver vi to cases fra virksomheder, der er to forskellige steder i implementeringen af en agil tankegang og tilgang.

På den ene side er SKAT en virksomhed som er meget tidlig i forløbet i implementering af den agile fremgangsmåde. Desuden er SKAT et eksempel på, hvordan den agile tilgang ses fra en kundes side.

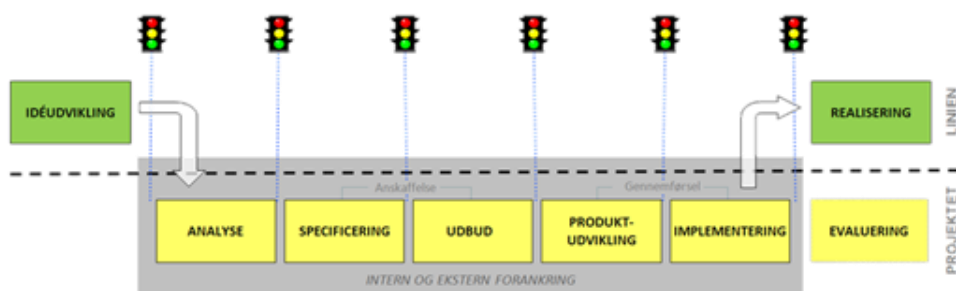
I den anden case har beslutningen for Schultz om at gå den agile vej været en del af en strategi som IT leverandør til det offentlige marked. Denne case er et eksempel set fra en leverandørs side og efter at den agile tilgang er taget i brug igennem et stykke tid. Casen for Schultz giver også eksempler på de effekter, man har opnået.

3.1 SKAT IT

Kontaktperson i SKAT, Niels Zachariassen, kontorchef for Metodekontoret, SKAT IT.

Baggrund for agil i SKAT

SKAT IT driver nyudvikling på en større portefølje af IT systemer. Dette gennemføres som en række projekter, typisk ud fra en plandreven metode, som er inspireret af Statens IT projektmodel bla. baseret på metoden PRINCE2. Fokus på det agile støtter op omkring SKATs overordnede resultatmål: 'Indsatsområder for at effektivisere SKAT' og 'Sikre, at SKAT forbliver en moderne og effektiv (*projekt*)organisation'.



Figur 6:
Projektmodel hos SKAT IT

Derfor har der i et stykke tid været forskellige projekter, der har arbejdet individuelt med en agil tilgang, og i lyset heraf, samt en større efterspørgsel for AGIL fra leverandører, igangsættes et fælles initiativ til at definere Agil i SKAT IT version 1.0.

Initiativet går i første omgang ud på at definere og implementere et metodeapparat til at supportere agile udviklingsprojekter, så SKAT IT er i stand til at samarbejde med leverandører, der arbejder agilt.

Forventede effekter ved agil i SKAT

Formålet med at arbejde og udvikle agilt i SKAT er at levere et mere rigtigt produkt hurtigere, ved hyppigere og mindre trinvis leveringer, med større og tættere interaktion med brugerne/kunden. Der forventes færre misforståelser og en minimering af risici samt maksimering af værdi og produktivitet gennem en iterativ proces samt oftere feedback på leveringer.

SKATs særlige kontekst for et agil setup

Det agile setup i SKAT tager udgangspunkt i de kendte agile værdier og principper, men er designet til de særlige rammer, som projekter i SKAT er underlagt. Således skal det agile setup ses i lyset af flere kontekstmæssige forpligtelser, herunder regler om involvering af Statens IT råd for større projekter, samt kontraktlige og udbudsmæssige afhængigheder. Desuden skal det agile setup ko-eksistere med SKATs plandrevne projektmodel og allerede foruddefinerede projektroller.

Det agile setup i SKAT

Scopet for det agile setup version 1.0 i SKAT består af definering og implementering af metoder, værktøjer og vejledninger til et agilt setup til støtte for SKATs projekter, når man samarbejder med leverandører, der er agile, dvs. primært i fasen Produktudvikling, men også i de tilstødende faser "Udbud" og "Implementering".

Det betyder, at et projekt er reguleret af en kontrakt tilpasset agil og opsplittet i en række del-leverancer. Hver del-leverance har et fastsat scope inden for hvilket, der kan arbejdes agilt ift. prioritering og gennemførelse af en product backlog.

Det agile setup er inspireret af SCRUM og Metodekontoret, som er ejer af projektmetoder i SKAT. Det agile setup kan rådgive og støtte projekterne med viden, processer, roller og skabeloner.

SKAT tager product owner rollen. Product owner arbejder tæt sammen med projektlederen på den ene side og det agile udviklingsteam (Scrum-teamet), som er leverandørens team. Undervejs i gennemførelsen af del-leverancerne forsøger SKAT i høj grad at sidde tæt på leverandøren.

Det agile setup er blevet integreret som en ekstra byggeklods til SKATs eksisterende plandrevne projektmodel, så man fortsat understøtter de kontekstmæssige forpligtelser herfor, og samtidig arbejder agilt.

På dette punkt af den agile rejse, som SKAT er i, er der flere ting, som bryder med den agile tanke og tankegang, herunder fx:

- Faste krav i kravspecifikationen, defineret detaljeret på et tidligt tidspunkt
- Underlagt faste rammer i kontrakten
- Produkter i kontraktens delleverancer kan ikke flyttes fra fx delleverance 1 til delleverance 2
- SKAT har ikke agile teams

For SKAT er nuværende agile setup et skridt på vej mod næste punkt. Dette vil gå ud på at undersøge følgende områder:

- Udbudsdelen
- Kravspecifikation kontra behovsopgørelse
- Use cases kontra user stories
- Arkitekturdelen/dokumentering
- Tættere leverandørsamarbejde
- Modenhedsniveau

3.2 Schultz

Schultz leverer IT-systemer, informationsløsninger og kundetilpassede IT-projekter. Schultz løsninger omfatter fagsystemer til sagsbehandling, lovportaler, selvbetjeningsløsninger, afklaringsværktøjer til job- og uddannelsesvalg samt en lang række offentlige vidensportaler.

Kontaktpersoner i Schultz,
Martin Sønderkær Larsen, IT-Udviklingsdirektør

Baggrund for agil i Schultz

Schultz oplever, at agile udviklingsmetoder bliver mere og mere efterspurgt af markedet, og Schultz har allerede benyttet agile udviklingsmetoder på adskillige projekter med stor succes. Schultz benytter også denne udviklingsmetode til udviklingen af Schultz' egne produkter. Agil udvikling i Schultz er baseret på det agile manifest, hvor der er fokus på forandringsvillighed, samarbejde og transparens i udviklingsprocessen.

Schultz kombinerer agil metode med mere klassiske principper fra PRINCE2. Det sker i kraft af, at der indgås aftaler, som har faste krav, tidsplaner og økonomi. Derfor vil det iterative forløb kontinuerligt blive evalueret mod en masterplan – således, at man for hver iteration laver en ny vurdering af, hvorledes det samlede projekt kan gennemføres, inden for projektets resterende tid og scope.

Agil udvikling som et strategisk og bærende valg i Schultz

Schultz leverer bedst på komplicerede forretningskritiske løsninger. Processen frem mod forankring agilitet har medført at Schultz skulle genopfinde sig selv i en ny model for derigennem at stå som et fornyende element i markedet.

Et kardinalpunkt på rejsen har været et stærkt ønske om at kunne agere på kundernes input, og en stærk kundeinvolvering med det formål at ramme mere rigtigt. Det har været helt afgørende at komme kontrolleret og sikkert i drift, så synlighed i fremdrift og kvalitet i implementeringen har været en væsentligt integreret del af det agile projekt.

Det agile setup i Schultz

Schultz anvender Scrum som den agile udviklingsmetode i de fleste af de udviklingsprojekter Schultz gennemfører.

Schultz anvender Scrum på en klassisk måde med de alment udbredte processer, roller og artefakter, herunder fx Product Backlog Grooming, Sprint Planning, Daily Scrum, Retrospective, Product og Sprint backlogs, Burn down charts, og de almindelige roller som Scrum Master og Product Owner.

At implementere Scrum er et organisatorisk forandringsprojekt, og skal styres som et sådan. Det betyder, at:

- Der skal være én, der bærer det igennem, og virkelig brænder for det
- Der skal folk om bord, som virkelig kan og tror fuldt og fast på metoden
- Man skal forstå metoden. Forstå styrkerne og svaghederne – i relation til opgaverne
- Man skal tale samme sprog: Få styr på roller/ansvar og på forskellene fra tidligere
- Man skal få styr på styringsmekanismerne, på nøgletallene og på afrapporteringen
- Der skal være fokus på uddannelse, certificering, netværk, udbredelse, læring

Hvad har været svært

Skal Schultz pege på svære områder, nævnes følgende:

- Projektledelsesrollen – at slippe kontrollen ... (det ligger ligesom ikke i rollen)
- Scrum og 'fastprisleverancer'
- At finde niveauet for designdokumentation og for beskrivelse af backlog items
- Øgede krav til udviklerne – Du skal ikke kun kunne tage imod udviklingsopgaver og færdiggøre dem. Som udvikler i et agilt team skal man være parat til at deltage aktivt i en større kontekst og med mere selvbestemmelse og – initiativ.

Erfaringer med Scrum i Schultz

Igennem arbejdet med Scrum på flere projekter har Schultz gjort sig følgende generelle erfaringer:

- Sprints bør have en varighed på 2-3 uger, da det giver koncentrerede sprints med stor fokus på at nå sprintmålet, og hvor de faste mødeaktiviteter er overskuelige.
- User stories estimeres i story points, som baseres på en række af de første tal i Fibonacci talrækken (1, 2, 3, 5, 8, 13, 20). Der benyttes oftest planning poker som estimeringsmetode.

- Taskboard og burn down chart bør fremstå tydeligt og fysisk i projektlokalet, da det giver en transparens i sprintets fremdrift. De kan også "digitaliseres" alt efter projektets kontekst.
- Burn down charts kan med fordel afspejle antallet af tasks, som resterer i sprintet. Det er Schultz erfaring, at det ikke giver mening at restestimere i timer undervejs i sprintet - Omkostningen ved at restestimere opgaverne løbende står ikke mål med den værdi, dette giver for styringen af projektet.
- Tasks kan med fordel estimeres i små, mellem og store (t-shirts størrelser) – Og det anbefales at holde task størrelser i små niveauer for at sikre synlighed omkring fremdrift.

Realiserede effekter af agil i Schultz

Nogle år efter starten på implementeringen af agilitet, vurderer Schultz, at man har nået målet med det agile initiativ. Der gives et klart og rungende JA, til at agil metode fungerer godt for Schultz. Schultz opererer i dag i en rendyrket agil udviklingsmodel, og agerer desuden agilt i vedligeholds- og support-funktionerne

Ledergruppen kører en agil styring af fælles tiltag og fokusområder, og strategiprocesser styres agilt med korte forløb, præsentation af resultat, evaluering og planlægning af næste sprint.

4.0 Konklusion

For mange organisationer er optimering af metode og eksekvering i højsædet, og i dette lys søger mange forbedringer ved anvendelse af agile metoder. Med dette whitepaper har vi søgt at belyse, hvori modsætningsforholdet til en plan-drevet tilgang til softwareudvikling består, da det er mange organisationers udgangspunkt. Den agile tilgang har konkrete fordele og en klart defineret hjemmebane. Ligeså den plan-drevne tilgang. Men der stilles så absolut krav til organisationen, der ønsker at implementere agilitet.

For som både emnets baggrundsmateriale og de fortalte cases beskriver, så skal implementering af agilitet i organisationen drives som et forandringsprojekt. Agile er ikke et produkt eller et værktøj, der kan rulles ud over en kort periode, ud fra en defineret plan. Det er en kultur der skal forankres, som en del af organisationens DNA, for at skabe værdi. Som tilgang stiller agile ligeledes meget konkrete krav til en række faktorer i organisationen, for at kunne lykkes, og dette er med til at belyse en meget vigtig pointe.

Agilitet er ikke nødvendigvis den rigtige tilgang for alle organisationer eller forretningsområder. Og som beslutningstager i en organisation, der overvejer at implementere agilitet, er det vigtigt at søge svar på en række specifikke spørgsmål:

Opererer vi i en kontekst og omverden, hvor en agil tilgang har hjemmebane og kan bidrage positivt?

- Hvis ja, hvordan skaber vi en kultur, der evner at arbejde agilt, med de kompetencer og den modenhed, organisationen besidder?
- Hvis nej, er der da elementer af en agil tankegang som kan anvendes i samspil med vores eksisterende processer?

Organisationer har ofte meget forskellige udgangspunkter, og derfor har vi fravalgt at anbefale en decideret vejledning i at implementere agile. Idét der er tale om en længerevarende kulturændring, og dermed et længerevarende forandringsprojekt, at implementere en agil tilgang i organisationen, vil det ganske enkelt være for stor en opgave. Men i Peak Consulting Groups arbejde med en række af vore kunder er følgende erfaringer trådt frem, når det kommer til at implementere en agil tankegang:

- Evaluer om organisationen har den nødvendige modenhed og de rette kompetencer til at arbejde agilt, eks. ved at gennemføre en modenhedsmåling.
- Sørg for at der i organisationen er en bemyndiget og kompetent person til at drive forandringen mod en agil kultur.
- Det er vigtigt at uddanne alle relevante personer i den metodik der vælges og at man løbende udforsker og supplerer med værktøjer.
- Evaluer hvilke typer af projekter, hvor de rette betingelser er til stede, for at kunne lykkes med agile udvikling (det er ikke nødvendigvis alle projekter der bør afvikles ud fra en agil tankegang).
- Find gerne et pilotprojekt af passende størrelse og vigtighed, hvor ideer og metoder kan prøves af, før den bredere implementering startes.

5.0 Om forfatterne

Nikolaj Christian Raahauge

Nikolaj er Management Konsulent i Peak Consulting Group, og har i flere år arbejdet med projekter inden for bl.a. Finans, Telekommunikation, Produktion og Service. Nikolaj leder program- og projektopgaver ud fra et holistisk perspektiv, og lader opgavens krav og karakter definere metodesammensætningen. Nikolaj motiveres af at skabe en projektorganisation, hvor individets kompetencer er i centrum. Drivkraften er det tværgående samarbejde mod opgavens mål. Nikolaj er PRINCE2 Lead Trainer for Peak Consulting Group, og er certificeret og har erfaring inden for Management of Risk, MSP, MoP, Kanban og kobling imellem projektledelse og agile leverancer.



Niels Zachariassen

Niels er kontorchef i SKAT med ansvar for at understøtte og kvalitetssikre organisationens samlede portefølje af udviklingsprojekter. Niels har gennem de sidste 20 år bl.a. specialiseret sig i, at implementere projektmodeller, metoder, styringsprincipper ind i organisationens samlede governancestruktur, så udviklings- og styringsprocessen hænger naturligt sammen med organisationens processer og strategi. Niels har omfattende erfaring med projektledelse og forandringsledelse fra både den offentlige- og private sektor, hvor han har arbejdet med at projektmodne organisationerne hen imod etableringen af et med organisationen integreret 'centre of project excellence'. Herudover har Niels gennem flere år fungeret som foredragsholder og underviser inden for projekt- og forandringsledelsesdisciplinen.



Mads Lomholt

Mads er Management Konsulent i Peak Consulting Group og har omfattende erfaring med ledelse af projekter, programmer og porteføljer, og med implementering af processer og metoder hertil. Ydermere har Mads rådgivet virksomheder i at forholde sig til hvornår et projekt, og forholdene omkring det, er egnet til gennemføre via agile leverancer. Som drivende kraft benytter Mads egne erfaringer med best practice metoder og erfaring med hvad der virker ud fra kultur og modenhed. Et vigtigt tema for Mads i sit arbejde med rådgivning og implementering af forandringer er det rette fokus på værdi og gevinster for de relevante interessenter. Mads har desuden mange års erfaring som underviser og PRINCE2 træner. Mads har evne til at motivere, og hans brug af humor kan gøre det sværeste forståeligt. Mads arbejder praktisk, er kvalitetsbevidst, struktureret og får tingene til at ske.





Peak Consulting Group

Sølvgade 10, 5
1307 København K

+45 3526 2880
info@peakconsulting.dk
www.peakconsulting.dk

