

Indledning

Af Poul Staal Vinje
Certified Scrum Master
Certified Scrum Practitioner

poul.vinje@gmail.com

www.agile-metoder.dk

Indhold

Indledning	1
1. Hvad er Agile	2
2. Hvad betyder det at arbejde agilt	3
3. Effekten	3
4. Små og store organisationer	5
5. Agile i praksis	5
6. Risikostyring	6
7. Hurtig levering	6
8. Teamet	7
9. Kunders og brugeres roller	7
10. Kontrol og synlighed	8
11. Agile processer	8
12. Værdier	9
13. Alt er anderledes	9
14. Ekstremt	9
15. Kom godt i gang	10
16. Tro på teamet	10

1. Hvad er Agile

Det simpleste svar er, at tage udgangspunkt i det agile manifest. I Wasatch i 2001 blev 17 personers tanker og ideer bragt sammen. Manifestet blev det fysiske resultat.

Det er værd at læse. Og især den ledsagende tekst til de centrale linier. Det ligger på: www.agilemanifesto.org.

Den centrale tekst i 'Manifesto for Agile Software Development' lyder således:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Den første line lægger vægt på at det er mennesker der arbejder sammen, mere end procesbeskrivelser der følges.

Den anden linie lægger vægt på at fremstille software der kan vises og prøves, mere end beskrivelser af det.

Den tredje linie lægger vægt på at samarbejde med kunder, mere end kontraktforhandlinger.

Den fjerde linie er den mest kontroversielle, og den uden sammenligning mest diskuterede. Den lægger vægt på at reagere på ændringer, mere end at holde sig til en plan. Ændringer diskuteres åbent, og deres konsekvenser lige så åbent. Men indledningsvis er der en parathed til at diskutere ethvert ændringsønske.

I hver linie ligger der muligheder for at yde et godt stykke arbejde. Muligheder for at skabe et godt produkt, der giver glade kunder og brugere.

I hver linie ligger der også muligheder for at opfatte det suboptimeret. Kun se linien snævert. I den tredje linie kan man for eksempel vælge at lade kernen i den være at nu skal vi holde op med at dokumentere. Hvad der er det rene vrøvl. Det læses kun ind af mennesker der i forvejen prøver at undgå at dokumentere. Realiteten er at dokumentationen blot skal aftales som ethvert andet produkt. Den opstår ikke af sig selv. Men hele dokumentationsspørgsmålet har i øvrigt intet med linien at gøre. Den har at gøre med at en del af systemet kan vises frem, snarere en at hele systemet er dokumenteret i UML. At der fore-

trækkes en iterativ konkretisering, snarere end en fasevis behandling af det samlede produkt.

De 17 bag manifestet havde haft tanker og ideer hver for sig. Tanker der gik i mange forskellige retninger. Kent Beck havde arbejdet med Extreme Programming. Martin Fowler havde tidligere kritiseret de alt for komplicerede processer der teoretisk blev anset for de bedste til design. Dave Thomas havde haft nogle ideer om at vedligeholdelse og videreudvikling kunne gøres bedre, med den samme investering i tid og penge.

De 17 fandt dog et fælles udgangspunkt for deres diskussioner. Nemlig deres succesoplevelser. Sammen prøvede de at beskrive de forhold der havde eksisteret i projekter der var gået godt. Deraf kom manifestet.

2. Hvad betyder det at arbejde agilt

De 17 var enige om at der var rigeligt med metoder og teknikker. I hvert fald på et konceptuelt plan. Det handlede mere om at anvende dem simpelt og ligefremt.

Fra XP havde de eksemplet med kodereviews. Det er en god ide at mere end én udvikler har set koden igennem. Detaljeret vel at mærke. Så langt så godt. Men det er ufremkommeligt at lade nogen skrive koden, og derefter bede nogen om at læse den igennem for at checke at den er rigtig. Kodereview er fint, men det skal gøres agilt, som PairProgramming. Reviewet kommer ved at have skiftende par ind over koden.

Et andet eksempel er test. Alle kan være enige om at det er en god ide at kode svarer til forventningerne. Men det er ufremkommeligt at bede nogen om at teste om et færdigt system svarer til forventninger. Hellere gøre det agilt. Ved at detaljere sine forventninger, før koden skrives, og kun for en lille del af gangen. Det er lettere at skrive kode der svarer til testen, end omvendt. Altså: Testdrevet udvikling.

Kodestandarder er et tredje eksempel. Kode skal være let at læse for alle. Men det er vanskeligt at få udviklere til at følge en standard. Gør det agilt. Standarden følges når mennesker skal læse og anvende fælles kode. Udviklerne vil enes om en standard, fordi det er nyttigt.

Kodereview, test og standarder er nyttige. Man skal bare finde den rigtige måde at implementere dem på. Agilt betyder at gøre det rigtige på en simpel måde. Der altid har at gøre med de agile værdier: Kommunikation, Feedback, Enkelhed, Mod.

3. Effekten

Dette afsnit beskriver de effekter der kan opnås ved at bruge agile metoder.

Time-to-Market

Reduceres ved hjælp af iterationer der kan releases enkeltvis eller samlet. Det er kunden der bestemmer release hyppighed og omfang. Men den tidsmæssige afstand fra krav til kode reduceres uanset om iterationen releases med det samme.

Den almindeligste iterationslængde er 1 måned. Med et team der sammensættes til iterationen. Den almindeligste releasehyppighed er 3-6 måneder. Kunden kan beslutte release af hver iteration, og kan variere det efter behov.

Produktiviteten

Et benchmarking program skal vise en forøgelse på minimum 100 %.

De væsentligste kilder til øget produktivitet er:

- produktejerens rolle og effekt
- teamet, der er selvstyrende
- ScrumMaster, der fjerner forhindringer

Kvaliteten

Den tekniske kvalitet øges. Ved at antallet af fejl der skal findes og fjernes, reduceres.

Kvaliteten i form af velegnethed til sin brug, øges. Denne form for kvalitet er delvist objektiv, fordi noget kan måles. Noget er subjektivt fordi det opfattes forskelligt

Kvaliteten, i form af interessenternes rene subjektive vurdering, øges.

De væsentligste kilder til øget kvalitet er:

- On-site Customer
- Userstories
- Testdrevet udvikling
- PairProgramming
- Refactoring
- Automatiseret Test

Kilderne til øget kvalitet erstatter det meste af den afsluttende test. Der bruges "Pay as you go" som bærende princip. Det vil sige at testen udføres under vejs.

Arbejdsglæde

Den går i vejret. Blandt dem der producerer systemet. Blandt alle der deltager.

Ændringer

Det er muligt at tage alle de ændringsønsker ind i forløbet under vejs der ønskes. Teamet vurderer og kommunikerer konsekvenserne, men det er produktejerens afgørelse.

Vidensdeling

Nyansatte oplever at blive sat ind i domænet, og blive en del af teamet, på rekordtid.

Tab af viden ved opsigelser kan imødegås.

Kollegacoaching kan bringes op på det niveau der er brug for.

Det er en kombination af team og Pairing der giver den daglige vidensdeling på kryds og tværs. I takt med nye Sprints og teams, bliver det vidensdeling i hele organisationen.

Estimere

Teamet lærer at estimere hvor meget de kan nå. Det foregår dels på SprintPlanning-Workshop, og dels i det daglige.

Konkrete effekter

Disse effekter er konkrete. Det er dem vi går efter og dem vi opnår. Det er stort set effekter der har været målet med enhver ny metode i tidens løb. Forskellen her, i agile metoder, er at det er sund fornuft, Best Practice, og vidensdeling i praksis, der bruges til at skaffe sig dem.

4. Små og store organisationer

Agile metoder er ikke specielt knyttet til små eller store organisationer. De kan bruges til at organisere teams og iterationer uanset størrelsen af den organisation de bruges i.

Agile metoder er ikke specielt knyttet til små eller store opgaver. De indeholder teknikker til at skære opgaven til, så store opgaver skæres i skiver. Og omvendt, at mange små opgaver kan samles.

Agile metoder opstod i store organisationer på store opgaver. Jeff Sutherland arbejdede på Patientkeeper da han definerede Scrum. Og XP opstod på C3 (Chrysler). Senere var det mindre organisationer der fortsatte med agile metoder. Nu er det både små og store der er aktive, herunder Google, Microsoft, Yahoo, Amazon, SAP og Oracle.

5. Agile i praksis

Agile kan beskrives filosofisk, som værende adræt og behændigt. Det kan desuden beskrives konceptuelt som værende drevet af forretningsværdi. Begge synsvinkler er rigtige. Men Agile kan også beskrives i praksis. Som bestemte metoder og teknikker. Dem er der mange af i den agile værktøjskasse, men disse er de vigtigste:

Scrum, til styring og opfølgning
Userstories og Storypoints til planlægning og estimering
Testdrevet udvikling
Extreme Programming

Når disse fire områder udføres i et agilt forløb, er det ved hjælp af processer der er små i omfang, enkle i indhold, og lette at forstå og bruge.

Beskrivelsen af hvordan man udfører testdrevet udvikling kan forstås på fem minutter. Og læres ved at gøre det sammen med andre i teamet på en halv time.

Få enkle procesbeskrivelser er grundlaget. Teamet gør det derefter bedst muligt. Lad os udtrykke det positivt: Teamet får mulighed for at gøre deres ypperste. Når showet er forbi skal teamet sidde med en klar og oplevet fornemmelse af at have gjort det bedste de kunne. Som det givne team de var, og under de omstændigheder de havde at arbejde under.

Det er ikke noget under at agile forløb leverer en høj produktivitet og en høj kvalitet.

Agile metoder leverer paradoksalt nok produkter der i høj grad bygger på normer og standarder. Med kode der er let at læse. Med automatiseret test der kan bruges til regressions-

test. Fordi mange personer har deltaget i at udvikle den fælles kode. Og mange personer har brug for at videreudvikle den.

6. Risikostyring

Kent Becks tidlige arbejde omkring Extreme Programming gik i høj grad ud på at reducere risici. For på den måde at gøre uforudsigelige forløb mere produktive.

Kundens krav skal 'tvinges' frem. De skal stå synligt, detaljerede og enkle for udviklerne. Fysisk. Det er nok at processen er uforudsigelig. Der skal oven i det ikke være tvivl om hvad opgaven går ud på.

Kundens forventninger skal erkendes af kunden selv. Gerne sammen med teamet.

Udviklingstiden skal ned. Der skal leveres håndfaste resultater, konstant, løbende. Kort tid mellem krav og kode er afgørende for at holde risici nede.

Forhindringer skal fjernes systematisk. De er risici for ikke at nå deadline. Risici fjernes derfor håndfast, ikke bare konceptuelt. Risici checkes hver dag, det er ikke noget vil skal komme i tanke om.

Spikes som det kendes fra XP er at slå søm i. Dagligdagens "Proof of Concept". Hvis der er usikkerhed om en løsning, eller om en foreslået løsning holder, så undersøg det. Slå det fast, eller drop det.

Fejl på grund af teamets uvidenhed skal fjernes. Ingen arbejder alene. Ikke nok med det, alle skal lære af hinanden, dagligt.

Suppleret med systematisk afdækning og imødegåelse af:

- risici for kunden ved at bruge systemet
- risici for udviklingsforløbet
- risici for at systemet ikke er velegnet til sin brug
- risici for ikke at levere det vigtigste først

Håndfast risikostyring gøres dagligt i agile forløb. Helt konkret, dagligt, med hele teamet som deltagere og tilhørere.

7. Hurtig levering

Kunde og team ønsker begge en høj produktivitet. Et projekt efter traditionelle metoder på 18 måneders varighed, skal kunne afsluttes på ni måneder. Med et noget større forbrug end 50 % af timerne. Men med et rigeligt overskud.

Der giver kunden mulighed for meget tidligt at afgøre om produktet der udvikles er det rette. Både for at checke at teamet leverer det ønskede. Og at det ønskede er det rette.

Det næste skridt er at levere det vigtigste efter 1 måned. Ved hjælp af iterativ udvikling. Efter ni iterationer kan der releases, eller kunden kan have valgt at release dele under vejs.

Kundens prioritering er en forudsætning for at løse op for den fasevise udvikling. Prioriteringen er i virkeligheden også en forudsætning for at have den rigtige retning.

Teamets produktivitet er dernæst en forudsætning for at have den rigtige marchhastighed. Den skaffer teamet sig på tre måder:

- produktivitetsforøgelse i det daglige arbejde
- beslutningshastighed i det daglige arbejde
- ingen fejl at rette

8. Teamet

Betydningen af det agile team står centralt. Teamet kunne nævnes som en af kilderne til øget produktivitet og/eller øget kvalitet. Men det tåler også at blive fremhævet selvstændigt.

Teamet er på 7-9 personer. Der kan sættes så mange parallelle og overlappende teams i gang, som opgaven og deadlines kræver.

Det agile team er selvstyrende. Det påtager sig den daglige ledelse i form af aktivitetsstyring og opfølgning.

Det agile team er forsynet med alle nødvendige ressourcer til arbejdet i iterationen. Det er et såkaldt "Whole Team". Det kan udføre alt og beslutte alt.

Teamet sammensættes til hver ny iteration. Der er intet i vejen for at bruge samme team flere gange i træk, men overvejselsen er der hver gang.

Teams er et godt middel, både til den indledende nyudvikling, og til den efterfølgende videreudvikling og vedligeholdelse.

Teamet sammensættes af ScrumMaster og ProductOwner i fællesskab.

9. Kunders og brugeres roller

Der er brug for en ProductOwner og en On-site Customer.

Det agile forløb drives af forretningsværdi, og ProductOwner er derfor uundværlig til at prioritere indholdet af hver iteration.

Det agile forløb drives af beslutninger. Mod. On-site Customer er derfor uundværlig til at beslutte i det daglige arbejde. Teamet skal holde et højt tempo, og der er brug for beslutninger. Afstanden mellem erkendelse af et behov for afklaring og en beslutning, skal holdes på få minutter. Ikke uger.

De to, Produktejer og On-site Customer, håndterer alle konfliktende interesser. Herunder dem der vedrører retning, indhold og rækkefølge

On-site Customer kan være en rolle der udfyldes af en til formålet udpeget ressource. Det kan være en marketing- eller produktansvarlig. Eller simpelthen en person der får opgaven.

Når On-site Customer ved hjælp af de agile værdier, ikke mindst 'Mod', har truffet en beslutning, kan der være brug for at checke med baglandet. Det holder ikke teamet tilbage for at gå videre med den modige beslutning. Hvis baglandet bestemmer at det er en beslutning der skal gøres om, så må det være sådan. Vi satser benhårdt på at udbyttet af beslutningerne langt overstiger ulemperne ved omarbejde. Ved at øve sig kan man blive bedre. Alternativet er en projektorganisation med hørings- og koordineringsgrupper. Der erfaringsmæssigt er lang tid om at træffe beslutninger, der ikke af den grund er bedre.

Produktejeren er aktiv og besluttende i den workshop der ligger umiddelbart før sprinten. On-site Customer er aktiv og besluttende i sprinten.

On-site Customer behøver ikke være 100 % tidsmæssigt allokeret. Men skal være 100 % fysisk allokeret. Vedkommende kan passe sit normale arbejde fra teamet. Tidsforbruget kan være alt mellem nogle få og 100 %.

10. Kontrol og synlighed

Alt er synligt i et agilt forløb. Intet er skjult. Således sagt to gange. Gennemsigtighed er både ønsket og til stede.

Kontrol forudsætter synlighed. Kontrol er både ønsket og muligt. Kontrol via hyppige inspektioner. Endda konstante; løbende.

Inspektionerne er på tre niveauer:

- i par, når der bruges PairProgramming, partest og Testdrevet udvikling
- i teamet, især ved den indledende workshop, på det daglige møde og i Retrospective
- i organisationen når produktet vises frem dagligt, integreres dagligt, og ved iterationens afsluttende præsentation efter en måned. Alle er velkomne på det daglige statusmøde.

Inspektionerne er ikke kun en mulighed for kontrol. De bruges som praktiske muligheder for kommunikation og feedback. Også på de tre niveauer: Par, Team og organisation.

Produktet er synligt fra start til mål. Fra ProductBacklog til Working Product.

Styringen er synlig fra start til mål. Fra estimering, over daglig opfølgning til beregning af den opnåede Velocity. Der beregnes som antallet af Storypoints pr. Sprint.

11. Agile processer

De simple agile processer er en reaktion på de omfangsrige og detaljerede processer, der anses for god latin. Omfattende processer der kræver uddannelse og instruktion i at forstå.

Og som ikke lever op til det rationale der ligger bag. Der jo er at et forløb kan forudsiges ved hjælp af dem.

Hvis forløbet i et projekt bliver forudsigeligt, er rationalet på plads. Problemet med rationalet er, at det helst skal holde 100 %. Selv med mindre dele af processen der ikke dækkes på den rigtige måde, kan der spildes store mængder tid, skabes store mængder fejl og u hensigtsmæssigheder, og opstå behov for store mængder omarbejde.

Agile processer er enkle. De bygger på at det uforudsete håndteres af teamet.

12. Værdier

De agile værdier stammer oprindeligt fra Extreme Programming. Det er:

”Kommunikation, Feedback, Enkelhed, Mod”

De findes i varianter. Fred med det.

De er ikke for sjov. De er ikke kun til festtaler. De bruges og bygges på. Værdierne giver alle mulighed for at gøre deres arbejde så godt som muligt.

På trods af at værdierne skal tages alvorligt, er det arbejdsglæde de sikrer. For alle involverede og berørte.

13. Alt er anderledes

Alt er enkelt, og det er jo glædeligt. Men alt er også anderledes. Det kan være en barriere. Man må aldrig undervurdere modstanden mod forandring. Den er dyb og stærk.

Agile metoder skal ikke håndteres pragmatisk. Stop dér, hvis udtrykket dukker op. Agilt betyder at gøre det ordentligt. Lad være med at tænke pragmatisk på den dårlige måde. Så man kun gør tingene halvt. ”Meget af det gør vi jo i forvejen, så det er kun nogle få ændringer”. Underforstået, de færrest mulige ændringer. Glem det.

Agilt betyder i de fleste tilfælde en radikal anderledes måde at arbejde på. Krav behandles og dokumenteres anderledes. Testen gennemføres anderledes. Konstruktionen ligner ikke den sædvanlige, ikke mindst på grund af den ændrede test. Alt gøres anderledes, og ser anderledes ud.

Stumper og dele er genkendelige. Fordi det er sund fornuft. Men vær åben for forandring. Lad være med at stoppe fordi det giver ændringer i måden der arbejdes på. Selvom dele kan genkendes, skal alt tænkes nyt.

14. Ekstremt

De agile metoder tager skridtet fuldt ud. Processerne er simple, men til gengæld anvendes de helt og fuldt. ”To the Extreme”. Der kan snige sig en oversættelsesfejl ind. Ordet ”Extreme” kan ikke oversættes til dansk ”Ekstremt”. Det skal oversættes til ”Helt og fuldt”.

Det er simplere, men skal gøres fuldt og helt. Det duer ikke at gøre det halvt. Halvhjertet. Det er simpelt, så gør det. Hvis det er værd at gøre, så er det værd at gøre helt og fuldt.

Komplicerede processer kan sjældent gøres helt og fuldt. Agile kan. De bygger på sund fornuft. Som ikke kun skal gælde for en del af et system. Eller kun skal gælde en del af tiden i et udviklingsforløb.

15. Kom godt i gang

Den nemmeste måde at komme i gang med Agile metoder er at hyre en coach til det. Det er ikke nødvendigt, men det kan være det nemmeste.

Agile metoder er nemme at lære. Der er desuden masser af hjælp at hente på nettet. Så det er bare at gå i gang. Hvis man alligevel gerne vil have noget hjælp til at komme godt i gang, er det et menneske der skal investeres i. Ikke procesbeskrivelser. Ikke værktøjer. Heller ikke kurser eller anden træning. Det kan komme i takt med at der er brug for det. Til et konkret team der er sat sammen til en konkret iteration.

En coach er det rigtige. De agile metoder handler om kommunikation, feedback, enkelhed og mod. Det kan en coach give.

Sammen med coachen kan de vigtigste opgaver til den første agile iteration fanges an. Det vil som oftest være en Scrum Sprint. Eventuelt en sprint 0 (nul) der skaber fundamentet for de efterfølgende sprints. Det er især en ProductBacklog der skal etableres, og de tre roller der skal afprøves: ProductOwner, ScrumMaster og Teamet.

Her er coachens aktiviteter opsummeret:

- etablere ProductOwners, og coache dem før, under og efter den første sprint
- sammensætte et godt team sammen med ProductOwner
- Coache teamet før, under og efter den første sprint
- Oprette en ProductBacklog sammen med ProductOwner
- Deltage i SprintPlanningMeeting
- Fungere som ScrumMaster i den første sprint
- Coache alle i forbindelse med Retrospective

Som det fremgår, er der en del udførende aktiviteter. Skal en coach have det? En coach er vel en der stiller spørgsmål, mere end giver svar! Sandt nok, men enten må man definere det som en udvidet coachrolle, eller også at det kun som coach, og at det egentlige udførende i hver pind, gøres af organisationens egne folk.

16. Tro på teamet

Teamet er den afgørende faktor. Håndtering af usikre forhold og imødegåelse af risici gøres bedst ved at lade mennesker arbejde sammen om det.

Giv teamet muligheder. Giv det ansvar og kompetence. Et team leverer fantastiske resultater. Når det får lov.

Overlad styringen af det til teamet selv. Der er ikke brug for detailstyring. Tro på teamet.