

Agile: realiseren dat de wereld verandert

Arjen Uittenbogaard (DNV):

Agile werken gaat over continu leren. En de principes van cyclisch werken geven je de mogelijkheid om te bedenken wat je wilt gaan doen, te meten of je het gehaald hebt en om je plannen bij te stellen. Niet pas aan het einde van je project, maar voortdurend, vindt Arjen Uittenbogaard, senior trainer en consultant bij DNV Cibit. Hij gaf tijdens een seminar in Bilthoven een presentatie over agile die zo boeiend was dat we u deze niet willen onthouden.

“Een projectleider zei ooit: “De klant had last van voortschrijdend inzicht”. Dat klopt niet. Voortschrijdend inzicht gebeurt. Als je iets doet, dan leer je. Bewust of onbewust. De klant die denkt misschien aan het begin van het project dat hij weet wat hij wil. Jij denkt misschien aan het begin van het project dat je snapt wat de klant denkt dat hij wil. Je kunt het misschien nog wel goed uitleggen aan de ontwikkelaars. En zelfs als ze het snappen en maken wat de klant denkt dat hij wil, dan nog moet je hopen dat er in de tussentijd niets veranderd. In de markt, in de organisatie, zodat datgene wat je oplevert alsnog niet meer is waar de klant het meeste aan heeft. Dingen veranderen altijd. En daar kun je maar beter gebruik van maken”, stelt Uittenbogaard .

Slechts in zeer uitzonderlijke gevallen zal een organisatie weinig verandingsgevoelig zijn. Als een project moet

worden gerealiseerd met een team dat al jarenlang samenwerkt; en op een systeem dat het team van binnen en van buiten kent; én met technieken, tools en technologie die ze al jaren gebruiken; én die ze nog jaren zullen gebruiken; én voor een klant die ze goed kennen; én waar ze goed mee samenwerken; én in een omgeving, waar voorlopig niets gaat veranderen,

Procesbeschrijvingen

In deze uitzonderlijke situatie moet je vooral op dezelfde manier blijven werken. Vooropgesteld dat het tot dusver succesvol was. Uittenbogaard: “Dan kun je misschien zelfs procesbeschrijvingen maken die ook nog kloppen. Wat heel vaak gebeurt is dat dingen anders gaan en we hebben dan de neiging om de procesbeschrijvingen uitgebreider te maken en ook de alternatieven erbij te zetten en ook voor de alternatieven op de alternatieven en de uitzonderingen op de uitzonderingen speciale templates te gaan maken. Procesbeschrijvingen, die alsmat aangroeien en in de praktijk gaat het toch weer anders”.

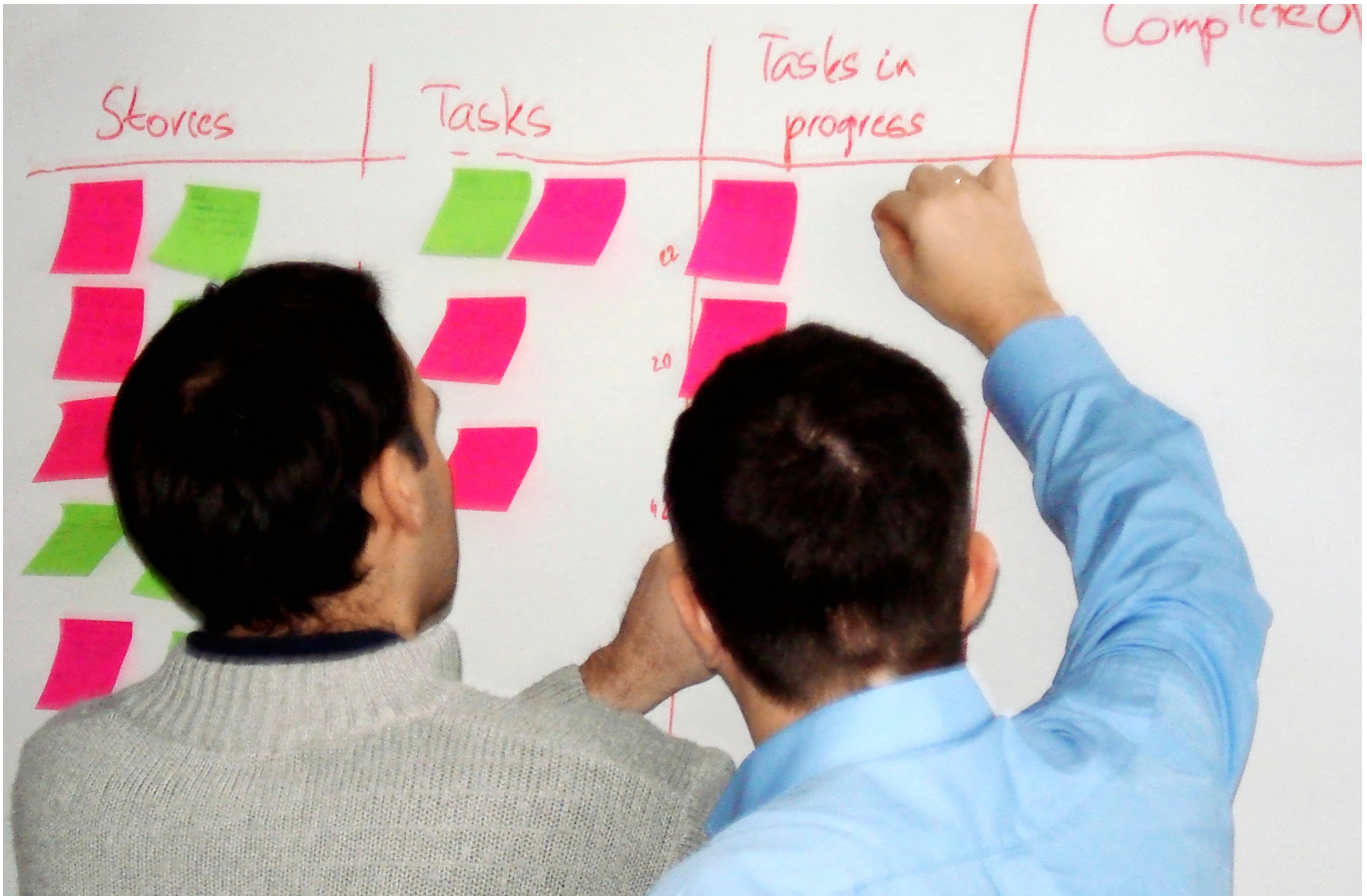
Er zijn verschillende manieren om daarmee om te gaan. Je kunt proberen de werkelijkheid in beton te gieten. Proberen te zorgen dat die veranderingen niet optreden. Doen alsof er geen veranderingen zijn. Je kunt contracten sluiten, met bloed ondertekenen, zodat je iets hebt om mee te slaan als er dingen anders gaan dan je had verwacht.

Arjen haalt een voorbeeld aan van een klus bij een klant, waar lineair werd gewerkt.

“Als we begonnen met programmeren gingen we dingen interpreteren die we in het design hadden opgeschreven. Op het moment dat de klant zag hoe we het interpreterden en het was niet goed, dan zei hij dat we hadden opgeschreven dat het zó moest worden gemaakt. Wij zeiden dan: wij hebben het opgeschreven en we bedoelden het zus. Ja, zei de klant, maar het staat er zo en zo willen we het hebben. Daar kun je dus heel lang over discussiëren, maar uiteindelijk gaat de klant op zijn strepen staan, verwijst naar het contract en dreigt niet langer met ons in zee te willen gaan. Je had ook veel eerder kunnen zeggen: we passen het gewoon aan. Maar dat kon in dit specifieke geval niet volgens het contract”. Uittenbogaard wil hier vooral mee aangeven dat het niet verstandig is om te veel zaken in een contract vast te leggen. “Daarmee kun je jezelf in de voet schieten”.

Veranderingen

“Agile werken wil zeggen dat je mee gaat met de veranderingen. Dat je weet dat veranderingen komen en dat je daarop anticipeert. Het wil niet zeggen dat je een windvaan bent, die alle kanten op wappert. Je hebt een doel en je checkt elke keer of je nog op weg bent naar dat doel en of dat doel nog steeds het goede doel is”.



De daily stand-up meeting in een Scrum werksituatie.

“Je maakt afspraken over hoe je daar naartoe gaat werken en wat je gaat doen als er dingen veranderen. Dan is het handig als je cyclisch of iteratief werkt. Als je iedere maand of iedere twee weken kunt kijken of wat je gemaakt hebt past in het plaatje van de klant. Dit wil niet zeggen dat je helemaal geen processen hebt. Agile werken is misschien wel de meest gedisciplineerde manier van werken die er bestaat. Dat er dingen veranderen is alleen maar mooi. Het betekent dat de klant uiteindelijk precies krijgt wat hij nodig heeft”.

Methodes

Scrum, Atern (voorheen DSDM) en XP zijn methodes waarmee tegenwoordig vaak wordt gewerkt door agile-aanhangers. RUP is veel uitgebreider. Alles is in RUP gedefinieerd; de kunst is om datgene wat je niet kunt gebruiken weglaat.

Een ander punt, waarop RUP zich onderscheid van de andere methodes is dat de bedenkers er in 2001 niet bij waren toen de topmannen van de methodes bij elkaar kwamen. Ze concludeerden dat ze het met elkaar eens waren, ook al werden andere namen aan hun manier van werken gegeven: iteratief, incrementeel, interactief. Bij DSDM ligt de nadruk sterk op business value. Bij XP heel sterk op goed programmeren. Maar in essentie waren de opstellers van het Agile Manifesto het eens over de volgende punten:

- Mensen die samenwerken zijn belangrijker dan processen en tools.
- Er moeten wel processen en tools zijn, maar als we moeten kiezen dan kiezen we voor goede mensen die goed samenwerken.
- We willen software die het doet, liever dan een pak papier en geen werkende software. Papier is misschien

nodig voor documentatie, accounting en dergelijke. Kies echter eerst voor goede software.

- Speel liever in op verandering dan dat je een vastgesteld plan volgt.

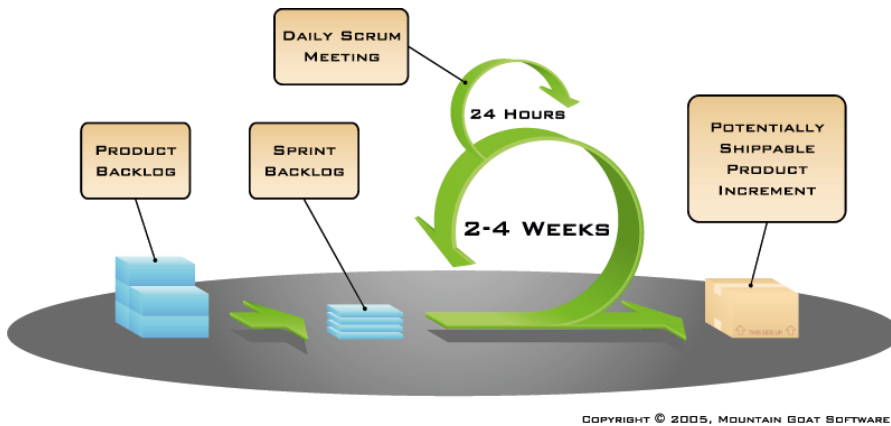
Nog een mooie in het Manifesto is het begin: ‘We are uncovering better ways of developing software’.

Met andere woorden: We zijn aan het ontdekken hoe we het beter kunnen doen.

“Het is een soort nederige houding. Het werkt goed voor ons, deze waarden vinden we belangrijk, maar elke ervaring is weer anders. Laten we kijken wat hier werkt. Laten we kijken hoe goed onze schattingen hier zijn. Laten we kijken en leren”, zo interpreteert Arjen Uittenboogaard deze regel.

Principes

Als je op internet zoekt naar het manifesto (www.agilemanifesto.org) vind



je een knop met 12 principes van agile software. Een voorbeeld: onze hoogste prioriteit is de klant tevreden te stellen door vroeg en continu waardevolle software opleveren. En: werkende software is de beste maatstaf voor de voortgang. Ook heel belangrijk: de kunst om de hoeveelheid werk die je niet doet te maximaliseren. Dat geldt ook voor de zelforganiserende teams in tegenstelling tot command-and-control manier van werken, die heel onnatuurlijk aanvoelt als je agile ingesteld bent.

Jim Highsmith, een van de ondertekenaars van het manifesto, zegt dat de essentie van agile werken een cyclus is van 'speculate, collaborate, learn' ofwel schatten, samenwerken en leren. Dat schatten klinkt eng, maar misschien is het wel goed om te gokken wat de komende maanden het belangrijkste is. Misschien is het de eerste keer dat je in deze combinatie van programmeurs, testers en anderen aan de gang gaat. Realiseer je dat een schatting geen commitment is. Een schatting is een poging om zicht te krijgen op wat je gaat doen. Als je een schatting moet maken kun je soms een requirements-traject in gaan. Snellere feedback kan ook leiden tot een betere volgende schatting. Samenwerken is heel belangrijk, zowel onderling – in XP werken twee programmeurs altijd samen – als met de klant. En als je dan hebt geschat en samengewerkt kun je gaan leren. Dat leren is in veel methodes heel expliciet

gemaakt. In Scrum hebben ze aan het eind van een sprint een sprintreview. Daar kom je met zijn allen bij elkaar, kijk je hoe goed het product is dat werd opgeleverd en hoe werd samengewerkt.

Neigingen

Uittenbogaard: "We hebben de neiging om bij veranderingen terug te grijpen op processen. We hebben de neiging om te doen alsof veranderingen niet plaatsvinden. We hebben de neiging om dingen in de klauwen te willen houden. Als je een veranderende omgeving hebt en je moet beter en sneller software maken, dan moet je proberen die neigingen te onderdrukken of af te leren".

"We hebben het te vaak zien gebeuren dat in een organisatie RUP werd toegepast. Als het niet werkte met RUP en dan werd niet alleen RUP in de ban gedaan, maar ook het agile werken. De vraag is of ze wel echt agile werkten, of alleen maar het RUP aan de slag waren gegaan. Maar het gaat over werkende software boven documentatie".

Scrum

Scrum is de meest minimalistische methode om een project te doen. Het is meer een projectmanagement filosofie dan dat het een software development methode is.

Dat is het grote verschil met RUP. RUP gaat over software development. Het gaat ook over requirements engineering en analyse en design, over

programmeren en testen. Scrum is alleen het raamwerk, waarbinnen je dingen gaat doen. Scrum kent drie rollen: de product owner, die uiteindelijk de keuzes voor de prioriteiten maakt. De scrummaster, ofwel het agile geweten, en het team member. Die werkt daar in het team waar dat nodig is. Wanneer je als tester even niets te doen hebt, kun je bijvoorbeeld kijken of je de analist of een programmeur ergens mee kunt helpen.

Scrum is ook minimalistisch omdat je maar drie soorten meetings hebt: de dagelijkse stand-up meeting, de meeting voor de planning aan het begin en een review meeting aan het einde. Op een burndown chart staat aangegeven hoe het project verloopt. Als je die dagelijks bijhoudt kun je zien of het project op schema ligt.

Combineren met XP

Scrum is volgens Uittenbogaard goed te combineren met Extreme Programming, wat veel handvatten geeft om goede code te ontwikkelen, of met RUP. Je moet je bij RUP wel goed realiseren dat dit heel uitgebreid is. RUP kent een hele lijst met rollen en verantwoordelijkheden. De waarde ervan is dat alle rollen zijn gedefinieerd en dat per rol en per fase allerlei activiteiten zijn beschreven per rol en per fase. De waarde is dat er allerlei templates en beschrijvingen zijn hoe je ermee kunt werken. Maar het is wel veel. Het geeft je de handvatten om goede requirements op te stellen en niet alleen use cases.

"Agile werken gaat over het realiseren dat dingen veranderen. In- of externe prioriteiten, organisatie, marktomgeving. Dan kun je ook maar beter rekening houden met verandering. Als je in die omgeving sneller en goedkoper producten wilt maken dan is agile werken de geëigende aanpak", zo besluit hij.

Robert de Ruiter