

Scrum: where Business drives IT

De simpelste oplossingen zijn meestal de beste

Nu op veel plekken de Oracle Developer en Designer ontwikkelstraat aangevuld wordt met, of vervangen wordt door JDeveloper, komt de vraag naar boven welke projectmanagementaanpak daar het best bij past. Veel bedrijven die traditioneel met Oracle Developer en Designer ontwikkelen, maken gebruik van CDM (Oracle's Customer Development Method) en dan vooral CDM Classic. Naast de Classic approach (waterval methode) is er een Fast Track approach als een iteratieve methode, gebaseerd op DSDM. Is er een reden om voor een andere project aanpak te kiezen als u intensief met Java of SOA aan de slag gaat?

Switchen van Oracle Client Server naar Webbased of Service Oriented betekent niet alleen objectgeoriënteerd werken, maar in veel gevallen vooral ook het realiseren van extern beschik-

bare applicaties. Klanten en leveranciers werken direct met úw software. Hierdoor wordt snel reageren op marktveranderingen en concurrentie veel belangrijker. Dit betekent dat een ontwikkelmethodiek nodig is waarmee snel ingespeeld kan worden op veranderende wensen van de markt en van gebruikers.

De complexiteit van nieuwe applicaties is groter dan van traditionele client server applicaties. Java applicaties bestaan uit meer lagen, meer servers, meer frameworks, meer interfaces en meer programmeertalen zoals XML, HTML en JavaScript. De volgende stap, applicatie-integratie, binnen uw bedrijf maar ook met externe partijen, op basis van open standaarden zoals webservices, SOA en BPEL, brengt ook een hoop extra complexiteit met zich mee.

Door onervarenheid van de business met de nieuwe mogelijkheden zal het moeilijker zijn precies te specificeren wat



In de projectkamer van het Scrum-team wordt voor iedereen duidelijk zichtbaar op een taskboard de status van alle taken getoond.



Tijdens de sprint houdt het team iedere dag een Daily Scrum Meeting.

gewenst is. Het gebruik van nieuwe technologie is een leerproces, zowel voor de business als voor de IT-afdeling. De beste manier om te leren is proberen, met Proof-of-Concept trajecten, prototypes maar vooral ook met life applicaties waarbij geleerd kan worden van de ervaringen van gebruikers.

Product development of product manufacturing?

Een veelgehoorde opmerking over de IT is dat het een onvolwassen industrie is omdat het projecten niet, zoals in andere industrieën, goed voorspelbaar kan afleveren in vergelijking tot het bouwen van bruggen, auto's of vliegtuigen.

De realisatie van producten bestaat uit twee fasen, product development en product construction, ontwerp en bouw dus. In de IT wordt deze tweedeling ook gemaakt, waarbij Requirements Analyse, Functioneel Ontwerp en Technisch Ontwerp tot product development behoren en Realisatie en Implementatie tot product construction. Het coderen van een applicatie is dan dus vergelijkbaar met de constructie van een brug of het in elkaar zetten van een auto op de lopende band, waarvoor een gedetailleerde planning gemaakt moet worden. Maar is dit eigenlijk wel een goede vergelijking? Tijdens de ontwerpfasen van een auto of een vliegtuig wordt niet alleen een functioneel en technisch ontwerp gemaakt. Er worden ook meerdere prototypes gemaakt, zowel virtueel, in CAD-programma's, als reëel, echt testbare prototypes. Het ontwerp wordt zo meerdere malen getest. Dit leidt tot aanpassingen in het ontwerp en tot verbeterde prototypes.

Dit is een groot verschil met de IT. Hier worden FO en TO meestal niet getest. Een groep van senior engineers en architecten beoordeelt het ontwerp, hun mening en ervaring bepaalt

of aan de constructie fase begonnen kan worden. Echt getest is het ontwerp niet. Dit betekent dat het opleveren van een detail planning voor de constructie ook niet mogelijk is.

Het grote verschil tussen een ontwerpfasen en een constructiefase is dat de ontwerpfasen een leerproces is, terwijl de constructiefase het zo nauwkeurig mogelijk repliceren van een eerder bedacht ontwerp is. Tijdens een ontwerpproces moet kennis gegenereerd worden over hoe het product er precies uit moet zien om te voldoen aan de vraag. Dit is een leerproces en laat zich dus niet zo gemakkelijk plannen. Product development moet dus op een andere manier gepland worden dan product construction.

Een project aanpak kiezen

"De methodologie keuze is niet het probleem. Het probleem bij veel projecten is dat de keuze niet correct of volledig gehanteerd wordt." Dit is een veelgehoorde opmerking als je over een nieuwe methode begint. Dit is gedeeltelijk waar, veel projecten lopen inderdaad mis omdat een methode niet goed gebruikt wordt. Maar ook omdat het project en de toegepaste methode niet dezelfde doelen hebben. De doelen van bekende methoden variëren sterk: RUP (risico's elimineren), Waterval (requirements volledig kennen), Prince2 (focus op inzicht voortgang), CMMi (hoewel geen methode an sich: herhaalbaarheid en verbetering). Het klopt dat een projectaanpak correct gebruikt moet worden om het beoogde effect te bereiken.

The New New Development Game

In de jaren tachtig is er een onderzoek gedaan naar bedrijven die succesvol producten ontwikkelden en effectief inspeelden op snel veranderende (concurrentie) omstandigheden. De resu-

Itaten van dit onderzoek zijn beschreven in “The New New Development Game”. Volgens de schrijvers maken succesvolle bedrijven onder andere gebruik van self organizing teams, die in korte cycli producten opleverden.

Deze manier van werken werd vergeleken met de Scrum uit de rugby wereld. Hier werkt een team samen om in korte perioden steeds een aantal stappen vooruit te komen. Dit contrasteert sterk met de traditionele manier van werken van veel andere bedrijven, waar gewerkt werd alsof het een estafette race was. Iedereen deed zijn ding en gaf het dan door aan de volgende. Deze manier van werken heeft een negatief effect op de time-to-market, en maakt het dus moeilijk om snel concurrerende producten te leveren.

Scrum

Scrum is een Agile software development proces dat een antwoord probeert te geven op bovenstaande uitdagingen: hoe kun je product development zo goed mogelijk plannen; hoe zorg je voor een maximaal leereffect; hoe zorg je ervoor dat je zo snel mogelijk kunt reageren op continu veranderende business doelen? Scrum is eenvoudig van opzet, in tegenstelling tot veel andere projectmanagement methodieken kent het slechts een klein aantal rollen, meetings en documenten (zie kader).

Scrum is midden jaren 90 ontwikkeld door Ken Schwaber en Jeff Sutherland. Scrum wordt door duizenden bedrijven voor een groot aantal projecten en producten toegepast. Bij Scrum draait alles om korte iteraties, meestal van een maand. Het is de bedoeling dat iedere iteratie (Scrum: Sprint) resulteert in werkende software die ook écht af is: niet alleen gecodeerd, maar ook getest en gedocumenteerd zodat het feitelijk direct in productie genomen kan worden.

Wat tijdens een Sprint wordt gerealiseerd, wordt bepaald door de Product Owner. Deze persoon zorgt ervoor dat er altijd een op prioriteit gesorteerde lijst met requirements is. Requirements worden in Scrum User Stories genoemd en de prioriteiten lijst de Product Backlog. De product backlog bepaalt wat er tijdens een sprint gerealiseerd wordt. Zo staat de product owner dus echt aan het stuur van een project. Bij iedere sprint kan de product owner opnieuw bepalen waar de prioriteiten liggen, en dus reageren op veranderingen in de markt. Aan het begin van een sprint bepaalt het ontwikkelteam welke user stories ze kunnen realiseren. Het hele team is verantwoordelijk voor het schatten van de activiteiten. Dit schatten wordt dus gedaan door de mensen die de activiteiten ook gaan uitvoeren.

Tijdens een sprint staan alle taken duidelijk zichtbaar op een Taskboard, en bepalen de teamleden onderling zelf wie welke

taken op een bepaald moment uitvoert. De product owner mag tijdens een sprint geen prioriteiten wijzigen of requirements toevoegen. Wijzigingen moeten wachten op de volgende Sprint Planning. Dit garandeert de stabiliteit die nodig is om te zien of schattingen ook kloppen en waargemaakt worden. Hierdoor is het mogelijk om de productiviteit van een team te bepalen tijdens een sprint en dit te gebruiken als leereffect voor de volgende sprint planning.

Dat een sprint productie-klare software oplevert, betekent niet dat er na iedere sprint een release is. De product owner bepaalt wanneer er gereleased wordt. Dit kan na iedere sprint zijn, maar ook na een aantal sprints omdat dan pas een complete set van functionaliteit af is. Toch is het belangrijk om ervoor te zorgen dat iedere sprint werkende software oplevert. Problemen kunnen zo niet naar achter geschoven worden, en de productiviteit die gemeten wordt per sprint is ook realistisch.

Rollen, Meetings, Documenten

Scrum kent 3 rollen:

- Product Owner
- Team lid
- Scrum Master

Scrum kent 4 bijeenkomsten:

- Sprint Planning
- Sprint Review
- Sprint Retrospective
- Daily Scrum Meeting

Scrum kent 4 documenten:

- Product Backlog
- Sprint Backlog
- Taskboard
- Burndown Chart.

Rollen

Zoals al eerder aangegeven stuurt de product owner feitelijk het project. Hij vertaalt business doelen naar user stories, en zorgt dat er altijd een up-to-date product backlog is. Het Team is een cross-functional ontwikkelteam dat zorg draagt voor planning, ontwerp, implementatie, testen en documentatie. Een Scrum-team bestaat bij voorkeur uit 5 tot 9 mensen, die gezamenlijk alle taken kunnen uitvoeren die nodig zijn om functionaliteit op te leveren die ook echt af is. Voor de efficiëntie van het project is het zinvol indien teamleden meerdere rollen kunnen vervullen, dus zowel architect als program-

Scrum in de praktijk

Als projectleider ben ik binnen IT-eye verantwoordelijk voor de realisatie van Service Oriented Architecture projecten. Eind 2006 waren er binnen IT-eye interessante discussies over ontwikkelmethodieken. Scrum was één van de methodes die ter sprake kwamen.

De week voor de training werkte ik met mijn toenmalige projectteam hard om een belangrijke deadline te halen. Op zondagavond om 23:30 (na een dag overwerk) nam ik het programma van de Scrum-training door, die maandag van start ging. Mijn interesse werd nog groter: overwerken bestaat namelijk niet in een Scrum-project.

De training heeft mij enorm geïnspireerd. Zelden heb ik twee dagen zo productief samengewerkt met medecursisten. Na een thema-avond bij IT-eye over Scrum ben ik elementen gaan toepassen in een lopend project. Samen met het projectteam heb ik alle nog uit te voeren taken geïnventariseerd, geschat en in een sprint backlog opgenomen. Dagelijks bespreken we de voortgang in een daily scrum en werken we de sprint backlog bij. Ook medewerkers van de opdrachtgever worden actief bij de daily scrum betrokken.

In april zijn we met een nieuwe projectfase van start gegaan. Het doel was om in één sprint een proces van de klant met BPEL, ESB en SOA te realiseren. Hierin hebben we zoveel

mogelijk elementen van Scrum gebruikt. Tijdens de Sprint Planning hebben we op basis van user stories de gewenste functionaliteit doorgenomen en geschat door middel van planning poker. Vervolgens zijn de user stories in taken van maximaal 16 uur opgedeeld en is de sprint gestart. De eerste paar dagen verliepen moeizaam. Steeds meer taken kwamen boven tafel. De burndown chart ging omhoog in plaats van omlaag. Maar in de loop van de sprint ging het steeds beter. Er ontstond een ritme in de groep en al snel wist iedereen wat er van hem of haar werd verwacht.

Uiteindelijk heeft het team meer functionaliteit gerealiseerd dan ik in vorige SOA-projecten in een vergelijkbare doorlooptijd heb kunnen realiseren. In de sprint review werd waardevolle feedback gegeven en deze nieuwe manier van werken heeft ook voor veel plezier gezorgd. Bijkomend effect van Scrum is dat de onderlinge samenwerking én communicatie aantoonbaar is verbeterd. Uit de sprint review bleek daarnaast dat een aantal Scrum-elementen nog niet helemaal uit de verf zijn gekomen. De groep heeft deze zelf als verbeterpunten voor de volgende sprint benoemd.

Bij de demo bleek dat we volledig hadden voldaan aan de verwachtingen van de klant. Het project is inmiddels begonnen met de zevende sprint.

Ronald Doelen, Certified Scrum Master

meur, als tester. Natuurlijk is het zo dat iemand bijvoorbeeld meer architect dan tester zal zijn, maar het is goed als iedereen kan bijspringen als ergens extra mensen nodig zijn.

Meetings

Iedere Sprint begint met een Sprint Planning sessie. Deze bestaat uit twee delen, een backlog selectie en de workload planning. Tijdens deel 1 licht de product owner de belangrijkste user stories toe. Vervolgens bepaalt het team de complexiteit van deze user stories. De complexiteit wordt uitgedrukt in Story Points. Op basis van deze story points en de Velocity van de vorige sprints bepaalt het team welke user stories er tijdens de sprint gerealiseerd kunnen worden. Het ontwikkelteam bepaalt de story points met behulp van Planning Poker. Tijdens planning poker heeft ieder teamlid een aantal kaarten met mogelijke story points. Om te beginnen kiest het team een user story waarvan iedereen het eens is dat deze eenvoudig te realiseren is. Deze user story krijgt een waarde van 1 story point. Vervolgens wordt de complexiteit van de overige user stories bepaald, in verhouding tot de eerste user story. Het resultaat van planning

poker is dat in korte tijd duidelijk wordt wat er allemaal bij komt kijken om een user story te implementeren, en dat het hele team ook achter de schatting staat.

Tijdens het tweede deel van de sprint planning vertaalt het team de user stories in taken. Per taak wordt ingeschat hoeveel uur hiervoor nodig is. Alle taken komen vervolgens op de Sprint Backlog.

Tijdens de sprint houdt het team iedere dag een Daily Scrum Meeting. Tijdens deze meeting van 15 minuten brengen teamleden elkaar op de hoogte van voortgang en eventuele problemen. Door dit iedere dag te doen, zorg je ervoor dat iedereen gefocussed is op het behalen van de doelen van de sprint, en dat problemen snel gesignaleerd worden, en dus aangepakt kunnen worden.

Een sprint wordt afgesloten met een Sprint Review en een Sprint Retrospective. Tijdens de sprint review geeft het team een demo van de opgeleverde software. Hierbij mag het team alleen de software tonen indien het ook echt af is. Ook wordt de velocity van het team bepaald. Dit bepaalt de productiviteit van het team. Vervolgens wordt de Sprint

tijdens de sprint retrospective geëvalueerd, en wordt afgesproken welke punten tijdens de volgende sprint verbeterd worden.

Documenten

Alle Scrum artefacts hebben tot doel dat voor iedereen duidelijk is wat er gedaan moet worden. Als een team zichzelf wil managen, kan dat alleen als voor iedereen duidelijk is wat er moet gebeuren en wat de status is. Dit wordt bereikt met de Product backlog, de Sprint backlog, het Taskboard en de Burndown Chart.

De product backlog is een op business value gesorteerde lijst van alle product wensen. Dit heeft wat weg van de MoSCow categorisering, waarbij het een probleem is dat alles een Must have wordt. Doordat Scrum van de product owner verlangt dat hij de exacte volgorde van stories in de backlog aangeeft, is er meer duidelijkheid.

Alle taken van een sprint worden door het team zelf in de sprint backlog bijgehouden. Deze bevat per taak voor iedere dag een schatting hoeveel uur er nog nodig is om de taak te voltooien. Indien een taak lastiger is dan vooraf werd ingeschat, kan het zijn dat het aantal uren oploopt. Ook komt het voor dat er nieuwe taken ontdekt worden tijdens een sprint, deze worden dan aan de sprint backlog toegevoegd.

In de projectkamer van het Scrum-team wordt voor iedereen duidelijk zichtbaar op een taskboard de status van alle taken getoond. Per taak wordt met een geeltje aangegeven bij welke user story de taak hoort en wat de status van de taak is: staat gepland; wordt geïmplementeerd; wordt getest; is klaar. Op het geeltje staat een korte beschrijving van de taak en meestal ook het aantal geschatte uren dat nodig is voor de taak. De daily scrum wordt bij het taskboard gehouden, zodat snel zichtbaar is of iemand langer dan gepland bezig is met een bepaalde taak en of iedereen wel met geplande taken bezig is.

De burndown chart laat ten slotte zien of de geplande doelen gehaald gaan worden. Het is een grafiek die voor iedere dag van een sprint laat zien hoeveel uur werk er nog gepland is. Door de lijn te extrapoleren is vrij snel in te schatten of de planning gehaald gaat worden.

Iedereen die een Scrum-projectkamer binnen loopt is dus binnen één minuut op de hoogte van de status van een Scrum-project. Alle taken met status zijn zichtbaar op het taskboard en de burndown chart laat zien of alle taken van een sprint op tijd af zijn. Omdat de voortgang altijd bekend is, is een formele voortgangsrapportage niet nodig.

Voordelen

Een belangrijk voordeel van de Scrum-aanpak is dat plannen is verankerd als een continu terugkerende activiteit. Scrum probeert niet op een onrealistische manier aan het begin van het project de toekomst te voorspellen, maar continu op basis van ervaring de

schattingen te verbeteren. Iedere sprint wordt de productiviteit van het team bepaald en wordt deze kennis gebruikt om de volgende sprint nauwkeuriger te kunnen schatten.

Het schatten van taken wordt door het team zelf uitgevoerd, en iedereen binnen het team doet daaraan mee. Ook dit heeft twee voordelen. Ten eerste is de schatting gebaseerd op de kennis en visie van alle projectleden en daardoor realistischer. Ten tweede committeert iedereen zich aan de schatting, doordat iedereen moet instemmen met een schatting tijdens planning poker.

Een ander belangrijk voordeel van Scrum is dat het altijd in dienst staat van de business. Een Scrum-team is altijd bezig om de op dat moment belangrijkste business-doelen te realiseren, en de business kan (indien nodig) ook snel bijsturen. Dit maakt dat IT een belangrijk middel wordt om de concurrentiepositie van een bedrijf te verbeteren.

Veel ontwikkelteams die op Scrum overstappen, melden dat hun productiviteit fors omhoog gaat. Een van de redenen hiervoor is dat de teamleden gefocust zijn op het realiseren van business-doelen, in plaats van het uitvoeren van taken. Het is dus eenvoudiger om te bepalen of je iets doet omdat het voorgeschreven is, of omdat het waarde voor de klant genereert. Een Scrum-project biedt veel minder mogelijkheden voor verrassingen.

Fixed price en date projecten

Als nadeel van Scrum wordt genoemd dat het geen zekerheid biedt over tijd, kosten en functionaliteit. Deze zekerheid zou alleen mogelijk zijn als alles vooraf tot in detail wordt gepland. Maar in feite verschilt Scrum hierin niet veel van andere methodieken. Je kunt van tevoren alle user stories bepalen, en vervolgens de complexiteit van deze user stories. Wanneer je dat weet, kun je een uitspraak doen over tijd van oplevering, kosten en functionaliteit. Tot zover is er geen verschil met de meeste andere methodieken.

Scrum stelt echter dat het bouwen van software zo complex is, en dat business-eisen zo dynamisch zijn, dat je de toekomst niet tot in detail kunt voorspellen. Het is dus nodig om continu je plannen bij te sturen op basis van behaalde resultaten. Dit om te zorgen dat je jezelf niet voor de gek houdt. Scrum onderkent ook dat deadlines en financiën belangrijk zijn, en hanteert daarom een op prioriteit gesorteerde requirements lijst. Door flexibel te zijn in je requirements, en te weten welke het minst belangrijk zijn, kun je zekerheid bieden in tijd en kosten.

Cowboy toestanden?

Een nadeel dat aan Agile-methodieken wordt toegeschreven, is een gebrek aan discipline en respect voor standaarden. Zonder gedegen ontwerp vooraf, zouden ontwikkelaars maar wat in het wilde weg programmeren, en iedere keer voor de meest hippe tools en frameworks kiezen. Hierdoor zou de opgeleverde software niet voldoen aan bedrijfsstandaarden en kostbaar

in onderhoud worden. Het feit dat je voor de implementatie geen compleet ontwerp maakt, betekent echter niet dat je niet aan architectuur en framework standaarden hoeft te voldoen. Ieder Scrum-team moet op de hoogte zijn van standaarden. Bij grote projecten wordt aangeraden de eerste sprints te beginnen met één team, en nadat de eerste opzet van de applicatie is uitgewerkt, de teamleden van het eerste team te verdelen over extra teams. Deze mensen kunnen er dan voor zorgen dat de extra teams de standaarden en architectuur zoals gekozen in de eerste sprints goed toepassen.

Modellen als CMM en CMMi worden vaak gebruikt om de volwassenheid van een IT-organisatie te meten. Ken Schwaber heeft onderzocht in hoeverre Scrum past binnen CMM. Zijn conclusie was dat het goed toepassen van Scrum ervoor zorgt dat je voldoet aan de meeste eisen van CMM level 2 en 3. Andere praktijkresultaten tonen aan dat het bereiken van CMMi level 5 significant goedkoper en sneller gaat door gebruik te maken van Scrum.

Conclusie

De simpelste oplossingen zijn meestal de beste oplossingen: dat is één van de charmes van Scrum. Scrum is de meest uitgekilde project management methodiek die net voldoende biedt om te

kunnen werken: Een pragmatische aanpak voor requirements management, een pragmatische aanpak voor schatten en plannen, een realistische kijk op de toekomstvastheid van plannen, een realistische kijk op toekomstvastheid van business requirements, en de realisatie dat IT in dienst staat van bedrijfsdoelen. Scrum: where Business drives IT.

Meer info

Agile Software Development with SCRUM,

Ken Schwaber, Mike Beedle

Agile Project Management with Scrum, Ken Schwaber

Agile Estimating and Planning, Mike Cohn

User Stories Applied: For Agile Software Development,

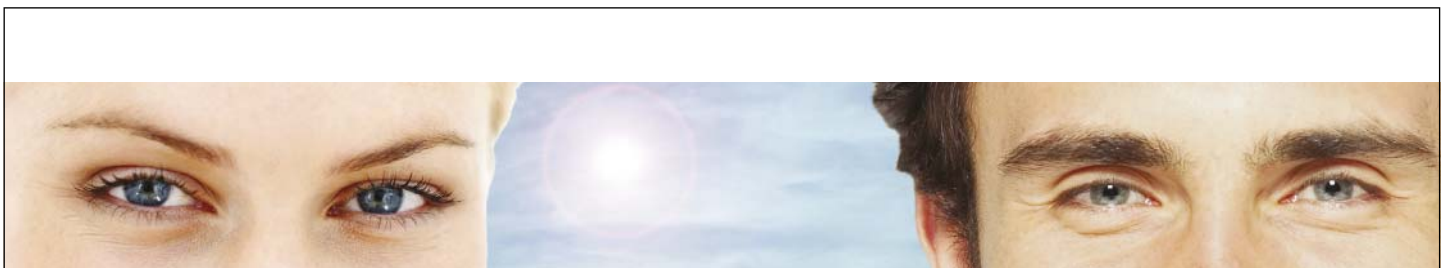
Mike Cohn

The New New Product Development Game,

Hiroataka Takeuchi, Ikujiro Nonaka

Andrej Koelewijn, Managing Consultant, IT-eye
(andrej.koelewijn@it-eye.nl)

Advertentie



Weet jij alles van Business Intelligence?

Doe mee met de grote BI-expert test.

10 vragen, 40 mogelijke antwoorden en een leuke beloning!

www.bi-expert.nl

**Hoe hoog
is jouw
'Business
Intelligence'?**

WIE DURFT...? KIJK OP WWW.BI-EXPERT.NL

