



Projecten evalueren

Wat we kunnen leren van ‘agile’ projecten

Projecten evalueren: het is lastig – en dus het komt er vaak niet van. In ons vorige artikel (Sigma 4, 2014), onderzochten wij hoe dat komt. In dit artikel zoomen we in op projecten waar het wél lukt om evalueren vast onderdeel te laten uitmaken van de routine: we noemen ze ‘agile’ projecten. Dit type projecten vinden we vooral in de software-ontwikkeling. Waarom lukt het daar wél stevast te evalueren?

In ons vorige artikel stelden we voor projectevaluatie als volgt te definiëren: 'Het tijdens het project met regelmaat met alle stakeholders vaststellen van de stand van zaken van het project waarbij twee vragen centraal staan: 1. Wat hebben we gedaan en gelaten zodat we op dit moment dit (tussen)resultaat hebben? En: 2. Welke acties zijn nodig om in het vervolg van het project en in volgende projecten, zeker zo effectief te werk te blijven gaan?' (Jacobs & Reichling, 2014).

Met deze definitie geven we het belang aan van product- én procesgericht evalueren, we trekken het perspectief van evalueren breder dan alleen het voorliggende project: door ons leren kunnen ook volgende projecten van ons evalueren profiteren.

Hiermee introduceren we het begrip 'leren'. En dat leren gaat om leren op twee niveaus en door verschillende actoren. Bij *first loop learning* gaat het erom er achter te komen wat de beste werkwijze is, gegeven de situatie. Bij *double loop learning* laat je ook de situatie meewegen in je analyse. Hoe komt het dat de situatie is wat hij is? Welke patronen herkennen we? Hoe voorkomen we dergelijke fouten in de toekomst? En: hoe zorgen we dat *good practices* in de toekomst behouden blijven? (Argyris, 1977).

Zoals gezegd, dit leren kan (moet!) gebeuren door alle actoren in en om projecten: individuen, het team, de betrokken organisatie(s) en soms ook ketenpartners. En dat is niet makkelijk, want zoals De Winter (2010) zegt: 'Mensen houden niet van fouten. Niet van hun eigen fouten, maar ook niet van die van een ander. Fouten maken staat per definitie voor incompetentie in een omgeving die verwacht dat mensen foutloos werken. Aangezien niemand zichzelf als incompetent wenst te zien, ontstaat cognitieve dissonantie wat al snel resulteert in het beschuldigen van anderen, het wijzen naar de omstandigheden of het zichzelf aanrekenen.'

'Agile' en evalueren

Softwareontwikkelingsprojecten kenmerken zich door een ingewikkelde relatie met de opdrachtgever en eindgebruiker. De mogelijkheden van software lijken eindeloos, maar zijn slecht voor te stellen als je het niet voor je ziet. Dat maakt

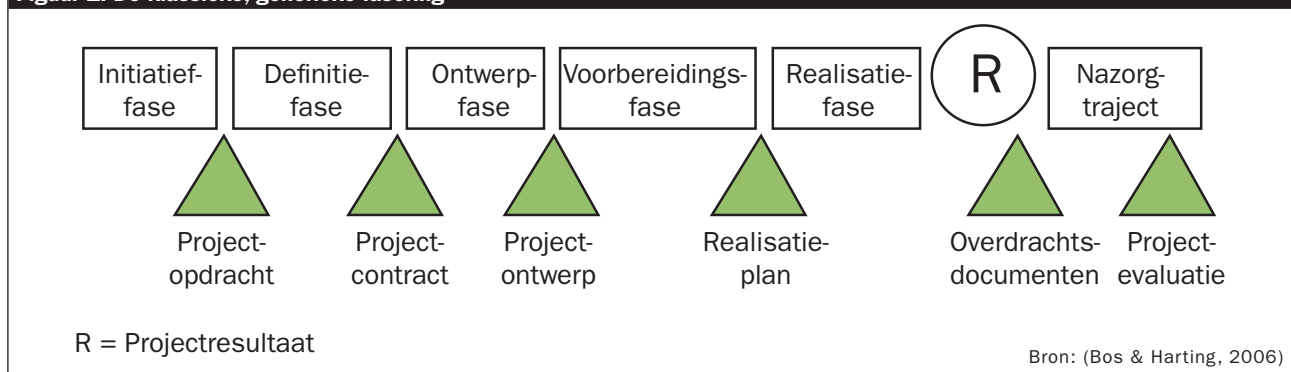
In minder dan 50 woorden

- Het evalueren van projecten wordt vaak overgeslagen, behalve in softwareontwikkelingsprojecten, waar het vaak wél lukt om projecten te evalueren.
- In deze projecten wordt een flexibele – 'agile' – projectstructuur gehanteerd, bijvoorbeeld Scrum.
- Deze methodiek is ook te gebruiken in andere dan IT-projecten.

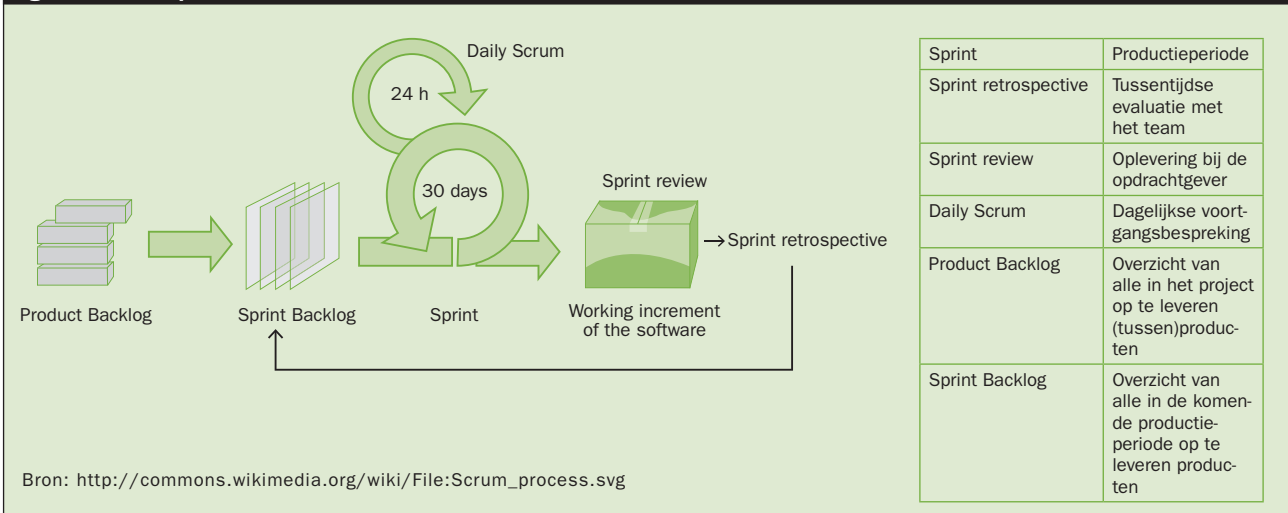
'klassiek' faseren (zie figuur 1), waarin het eindproduct in de vroege stadia van het project volledig gedefinieerd wordt, lastig. Dit vraagt immers een uitgebreide definitie- en ontwerpfase, waarna – na akkoord – de maanden durende voorbereiding- en bouwfase vrijwel buiten het zicht van de opdrachtgever plaatsvinden. Op het moment dat de resultaten bereikt zijn en de gebruiker voor het eerst het resultaat ziet, blijkt vaak dat die zich er toch iets anders van had voorgesteld. Een ander veel voorkomend 'euvel' is dat de te automatiseren processen inmiddels nét even wat anders zijn ingericht, waardoor het resultaat – ook al voldoet het exact aan de specificaties – niet meer direct toepasbaar is.

Dit gegeven heeft ertoe geleid dat een aantal *thoughtleaders* in softwareontwikkeling een *Agile Manifest* met bijbehorende *agile* principes heeft opgesteld (Agile Manifesto, 2001). De globale gedachtegang was: uit ervaring weten we dat producteisen gedurende het project zullen veranderen. Dat vraagt om een flexibele (agile) projectstructuur met bijbehorende flexibele attitude van alle deelnemers. Er mogen geen concessies gedaan worden aan de productgerichtheid. Hierop voortbordurend is een aantal methodieken ontwikkeld. Hiervan is *Scrum* inmiddels het meest bekend en wordt ook buiten de softwaresector het meest toegepast. *Scrum* ontleent zijn naam aan de dagelijkse teambijeenkomst van 10 minuten. Hierin staan ieders resultaten van de voorgaande dag op de agenda, net als de te verwachten resultaten voor vandaag en de directe bedreigingen voor het niet realiseren van de *sprint*doelen (Scrumguides.org). We nemen *Scrum* nu als kapstok, ons realiserend dat veel van wat we in *Scrum* tegenkomen ook in andere agile methoden wordt toegepast.

Figuur 1. De klassieke, generieke fasering



Figuur 2. Scrum-proces



Scrum

Figuur 2 geeft de structuur van een scrum-project weer. Daarin valt op dat het project is opgedeeld in *sprints*. Dit zijn iteraties van 4-6 weken waarin steeds een '*working increment*' wordt opgeleverd. Dit is een werkend, en dus door de opdrachtgever beoordeelbaar (deel)product. Het contact met de opdrachtgever is een inhoudelijke bespreking, de *sprint-review*, waarin de opdrachtgever het product toetst aan de overeengekomen opdracht voor die sprint. Het contact kan zich niet beperken tot (schriftelijke) voortgangsrapportages. Na afloop van iedere sprintreview houdt het team, zonder de opdrachtgever, een zogenaamde *sprint retrospective*. Hierin staat de vraag centraal: 'gezien de reactie van de opdrachtgever op de (kwaliteit) van het product: welk gedrag moeten we veranderen, wat moeten we vasthouden en waar moeten we mee stoppen?' Hierbij wordt nadrukkelijk aandacht besteed aan werkvormen die leiden tot leren (Derby & Larsen, 2006), want: '*retrospectives that don't result in change are sterile and frustrating*' (Schwaber, 2003).

Hoe verhoudt 'agile' zich tot evalueren?

Volgens onze definitie moet evalueren tussentijds, productgericht, gericht op double loop learning en met en voor alle stakeholders, inclusief de eindgebruiker. In hoeverre voldoet het evalueren uit scrum aan deze eisen?

Tussentijds

Ja dat doet het. Door de korte cycli waarin evaluatie gekoppeld is aan oplevering, is evalueren een vast onderdeel van de routine. Bij 'klassieke' projecten zijn de geijkte evaluatiemomenten meestal fase-overgangen, die veel minder veelvuldig voorkomen, waardoor inslijten in de routine veel lastiger is. Het effect: evaluaties worden uitgesteld en op zijn best alleen nog aan het einde van het project gehouden. En daarmee zijn we terug bij ons vorige artikel.

Productgericht

Jazeker. Evaluatie van het proces gebeurt pas als een (of meerdere) concrete (tussen)producten zijn opgeleverd in de sprintreview met de opdrachtgever. Daarin wordt vastgesteld in hoeverre het product af is en hoe goed het is. Alle feiten over Tijd, Geld, Kwaliteit, Informatie, Organisatie (TGKIO) en de businesscase komen hier op tafel. Bij de klassieke projectfasering ligt de focus in het contact met de opdrachtgever vaker bij managementproducten zoals voortgangs- en *exception*-rapportages, ontwerpdocumenten, realisatieplannen. De (tussen)producten, waar het eigenlijk om gaat, zijn dan vaak al tijdens de fase gereed gekomen, zonder dat daar een evaluatiemoment aan was verbonden. Voortgangsrapportages waarin wordt vermeld dat 'iets' voor 60% gereed is, leiden ertoe dat conclusies alleen nog maar over het proces kunnen gaan. Ze 'zingen los' van dat waar het om gaat: een resultaat neerzetten en een doel behalen.

Double loop learning

Ja. De retrospectives zijn bij uitstek het moment waarop het team zoekt naar onderliggende patronen in de samenwerking. Het zijn procesgerichte bijeenkomsten die verder bouwen op de resultaten van de review. Doordat alle (TGKIO-)feiten daar op tafel lagen, kun je in de retrospective focussen op de analyse van hoe het zo gekomen is (en wat daarvan te leren valt).

Double loop learning van individueel tot en met ketenniveau.

Nee. Het leren binnen Scrum is heel nadrukkelijk gericht op projectniveau. Primaire oorzaak is het feit dat het team zonder 'pottenkijkers van buiten' de retrospective houdt. Natuurlijk zullen al doende het individuele en het organisatieniveau aan bod komen, maar daar ligt niet de focus.

Met en voor de eindgebruiker

Een beetje. In tegenstelling tot in veel andere projecten, wordt

De tips in de praktijk

	Bouwproject	Reorganisatie	Certificeringsproject
Gebruik de Product Breakdown Structure als primaire structuur voor de planning van je project-overleg	Bijvoorbeeld: <ul style="list-style-type: none"> – Fundering gereed – Kozijnen geplaatst – Dak gedekt 	Bijvoorbeeld: <ul style="list-style-type: none"> – plaatsingsplan gereed – OR-zitting voorbereid – informatiebijeenkomst personeel voorbereid – financiële reservering voor outplacement e.d. gemaakt 	Bijvoorbeeld: <ul style="list-style-type: none"> – oplevering procesbeschrijvingen – (concept) kwaliteitsbeleid opgesteld
Betrek écht iedereen	Van aannemer en onderaannemer de uitvoerder én een bouwvakker die betrokken was; de koper van het huis en/of zijn adviseur, de projectontwikkelaar, architect, (constructeur)	Directie, personeelszaken, OR, projectmanagement, (externe adviseurs/ondersteuning)	Directie, kwaliteitsmedewerkers, vertegenwoordiger van de afdeling(en) waarover de procesbeschrijvingen gaan, eventuele externe ondersteuning
Geef voldoende inhoud aan je opleveringen	Aannemer toont: <ul style="list-style-type: none"> – kwaliteit van de opgeleverde producten met behulp van het bestek (K); – gemaakte kosten (G); – evaluatie van geschatte en gerealiseerde tijd (T) – het plan voor de komende oplevering (wat opleveren en wanneer) (T) – met wie er gewerkt is en met ze de volgende periode ingaan (O) 	Projectleider presenteert het product en geeft alternatieven en consequenties daarvan. Hij gaat expliciet in op de minimaal vereiste kwaliteit (juridisch voldoende 'zeker' en voldoende draagvlak voor verandering bij relevante stakeholders). (K) Hij geeft tevens aan: <ul style="list-style-type: none"> – gemaakte kosten (G) – evaluatie van geschatte en gerealiseerde tijd (T) – het plan voor de komende oplevering is (wat opleveren en wanneer) (T) – met wie er gewerkt is en met ze de volgende periode ingaan (O) 	Afdelingsvertegenwoordigers presenteren de procesbeschrijvingen en geven het verschil met de eerdere situatie aan waarbij zowel positieve als negatieve punten aan de orde komen (K). Projectleider geeft aan: <ul style="list-style-type: none"> – evaluatie van geschatte en gerealiseerde tijd (T) – het plan voor de komende oplevering (wat opleveren en wanneer) (T) – met wie er gewerkt is en met wie ze de volgende periode ingaan (O)
Koppel evaluaties aan opleveringen, maar blijf weg bij afrekenen	Dit is niet eenvoudig, want de bouwsector is hier niet aan gewend. Bewoners komen veelal indirect – via projectontwikkelaars – aan het woord. Toch is hier winst te behalen, maar dat vraagt vertrouwen over en weer. Doelen van bewoners en bouwers liggen uiteindelijk dicht genoeg bij elkaar (mooie goede huizen bouwen voor een redelijke prijs voor alle partijen) om dit te kunnen laten slagen.	In een reorganisatie liggen doelen van individuele 'eindgebruikers' vaak te ver af van het doel van het project om constructief met elkaar te kunnen evalueren. Daarom is ook niet iedereen betrokken bij de evaluatie (zie bij Betrek echt iedereen). De OR als vertegenwoordiger van het personeel is het hoogst haalbare, maar zij vertegenwoordigen geen individuele medewerkers.	In dit type project is dat relatief makkelijk: het zijn veelal interne projecten waarin iedereen uiteindelijk hetzelfde belang heeft: de certificering voor elkaar krijgen.

de opdrachtgever binnen agile zeer nadrukkelijk aangesproken als vertegenwoordiger van de eindgebruiker. Zijn rol heet niet voor niets 'productowner'. Er wordt op dit punt ook echt wat van hem verwacht. Toch blijft het lastig, want de productowner is lang niet altijd degene die in de praktijk 'achter de knoppen zit'. En dus komen, alle goede bedoelingen ten spijt, de eindgebruikers er ook in scrum vaak wat bekaaid van af.

Tips voor evalueren

Het voorgaande, gecombineerd met de in ons vorige artikel (Jacobs & Reichling, 2014) beschreven oorzaken voor 'niet evalueren', leiden tot de volgende praktische tips. In het kader 'De tips in de praktijk' (vorige pagina) geven we een indruk van de toepassing op andersoortige dan software-ontwikkelingsprojecten: een (huizen)bouwproject, een reorganisatie en een certificeringsproject.

Gebruik de Product Break Down Structure als primaire structuur voor de planning van je projectoverleg

We hebben geconstateerd dat evaluaties zinvoller worden als ze een productoplevering als startpunt hebben. Dat is in ieder project te organiseren wanneer je de *Product Break-down Structure* als uitgangspunt neemt. Waar in de klassieke projectfasering het voortgangsoverleg tussen opdrachtgever en projectleider met vaste tussenpozen (wekelijks, maandelijks) plaatsvindt, wordt dit nu primair gepland tijdens oplevermomenten. Daarbij tekenen we aan dat oplevermomenten natuurlijk niet de enige contactmomenten zijn als de opleveringen in de tijd erg ver uit elkaar liggen.

Geef voldoende inhoud aan je opleveringen

Om een evaluatie met alle partijen voor iedereen zinvol te maken, zul je die goed moeten voorbereiden. Voor de productoplevering moeten de feiten over Tijd, Geld, Kwaliteit, Informatie, Organisatie én de businesscase op tafel liggen. Een product presenteren dat er 'goed' uitziet is onvoldoende, de kwaliteit moet worden aangetoond. Alle relevante partijen mogen hierop hun zienswijze geven, waarna de opdrachtgever een definitief waardeoordeel uitspreekt. Uiteraard is dat alleen mogelijk als vooraf duidelijk is gedefinieerd wat zou worden opgeleverd en met welke kwaliteit.

Koppel evaluaties aan opleveringen, maar blijf weg bij afrekenen

Wees consequent en laat iedere oplevering volgen door een evaluatie waarbij alle stakeholders aanwezig zijn. Zorg er voor dat de evaluatie het (lopende) proces en het verbeteren daarvan tot doel heeft. Zodra evaluaties samengaan met verantwoorden of afrekenen, zal het misgaan. In een juridische context zal het (h)erkennen van de mogelijkheid tot verbeteren te makkelijk als schuldbekenenis kunnen worden opgevat. Dat gezegd hebbende: Laat iedereen toelichten hoe hij denkt dat het product is geworden, hoe hij daarop invloed heeft uitgeoefend. Het perspectief is, in

tegenstelling tot bij de oplevering, procesmatig: samen op zoek gaan naar onderliggende oorzaken zoals kwaliteit van samenwerking en belangenbehartiging.

Betrek écht iedereen

Zoals gezegd, scrum doet een poging de eindgebruiker te betrekken, maar slaagt daar net niet in. De kunst is om ervoor te zorgen dat er werkelijk eindgebruikers aan tafel zitten als je gaat evalueren. Het maken van een product dat aan de specificaties voldoet is al geen sinecure, maar de meeste kosten worden meestal gemaakt in de vorm van 'beheerkosten'. Een product is geïmplementeerd, maar moet al doende steeds weer een beetje worden aangepast. Onze stelling is dat een deel van die aanpassingen te voorkomen is als de eindgebruiker nadrukkelijker wordt betrokken in het ontwikkelproces.

Maak goede afspraken

Ga op zoek naar ieders persoonlijke commitment. Het gaat daarbij om de vraag: 'Wat kan ieder als individu doen?'. Op die manier leg je binnen het acroniem SMART de nadruk op de A van acceptabel of geaccepteerd, in combinatie met de R van realisme. Specifiek, Meetbaar en Tijdgebonden formuleren geeft het nodige houvast en maakt het makkelijker om op de afspraken terug te komen. Houd er rekening mee dat met name professionals niet makkelijk leren (Argyris, 1977).

Literatuur

- Agile Manifesto. (2001). Verkregen op 08 mei 2014 van <http://www.agilemanifesto.org/iso/nl/>
- Argyris, C. (1977). Double Loop Learning in Organizations. *Harvard Business Review, Volume 55 Issue 5*, 115-125.
- Bos, J., & Harting, E. (2006). *Projectmatig Creëren 2.0*. Schiedam: Scriptum.
- Derby, E., & Larsen, D. (2006). *Agile retrospectives - making good teams great*. Dallas: The Pragmatic Bookshelf.
- Jacobs, T., & Reichling, A. (2014, augustus). Projecten evalueren - Zes redenen waarom het er te vaak niet van komt. *Sigma (4-2014)*.
- Schwaber, K. (2003). *Agile Projectmanagement with Scrum*. Redmond, Washington: Microsoft Press.
- Scrumguides.org. (sd). The official scrumguide. Verkregen op 30 december 2014, van [scrumguides.org: http://www.scrumguides.org/scrum-guide.html](http://www.scrumguides.org/scrum-guide.html)
- Winter de, P. (2010). Leren van fouten blijft vaak een holle frase. *Slow management-Stilstaan bij organiseren (Winter 2010)*, 52-58.

Auteurs

Tineke Jacobs is projectmanagementspecialist met een focus op ICT-projecten. Zij is trainer en projectcoach, en staat ook regelmatig zelf 'met de voeten in de klei' als projectmanager. www.projectpraktijk.nl

Antonie Reichling helpt met zijn expertise in programma- en projectmanagement organisaties hun verandercapaciteit optimaal te benutten en oplossingen van strategische waarde te creëren. Als Certified-beoordelaar is organisatie-leren voor hem speerpunt. www.reichling.nl

Antonie en Tineke ontwikkelden samen met collega Bart Terlingen De ProjectCheck, een evaluatie-instrument voor projectmanagement. www.deprojectcheck.nl