

Bij projecten waarbij met een iteratieve aanpak software wordt ontwikkeld wordt soms het Unified Process gehanteerd als methode en steeds vaker ook Scrum. Hoewel deze methoden niet nieuw zijn, hebben projectmanagers vaak nog bezwaren tegen een iteratieve aanpak. Hoog tijd dus om alle argumenten van projectmanagers tegen een iteratieve aanpak op een rij te zetten samen met de argumenten voor een iteratieve aanpak.

De iteratieve aanpak in projecten loont

Projectmanagers moeten worden overtuigd

Argumenten tegen zijn er veel als een nieuwe aanpak wordt geïntroduceerd voor het ontwikkelen van software. Dat is ook het geval bij de iteratieve aanpak. Soms zijn de bedenkingen van de projectmanager terecht, vooral als de organisatie niet zorgt voor een goede voorbereiding. Bijvoorbeeld in het geval van regressietesten, als de organisatie vooraf in de voorbereiding niet investeert in automatisch testen kan dat een groot probleem worden. Maar de baten op het gebied van kwaliteit en de betrouwbaarheid van leveren zijn dermate groot, dat de iteratieve aanpak loont. Tijd dus om de misvattingen weg te nemen.

Angst voor het nieuwe

Voor sommige projectmanagers zijn complexiteit en risico's van een project reden om aan te geven dat de software niet in iteraties moet worden ontwikkeld. De angst is dat een nieuwe aanpak nieuwe risico's met zich meebrengt, namelijk de onbekendheid met de nieuwe werkwijze, waardoor mensen niet meer weten wat ze moeten doen.

Projectmanagers hebben hier gelijk in: een nieuwe werkwijze brengt risico's met zich mee. Het team moet opgeleid worden en moet de nieuwe werkwijze leren hanteren. Toch is het de moeite waard om te veranderen als het project net begonnen is of nog moet beginnen.

Risico's bij watervalmethode

Ervaring leert ons, en daar is ook al veelvuldig over geschreven, dat de watervalmethode grote risico's met zich meebrengt. Risico's die meestal tot uiting komen in de laatste fasen van het project,

vaak tijdens het testen van de software. De problemen ontstaan vlak voor de oplevering, net als iedereen denkt klaar te zijn. De software werkt niet, of de performance is niet acceptabel. Op dat moment is helaas vaak onbekend of de problemen kunnen worden opgelost en zo ja, wanneer. Het project loopt vertraging op, zoals laatst bij een project waarbij vijf miljoen was geïnvesteerd en het nog onbekend is wanneer het systeem in productie kan vanwege een op dat moment onoplosbaar performanceprobleem.

Vooraf bij complexe en bedrijfskritische software is het van belang dat de risico's vroegtijdig worden weggenomen. Een aanpak die risicogedreven is, die als doel heeft om risico's vroegtijdig weg te nemen, is daarom bij uitstek geschikt.

Een risicogedreven aanpak

De iteratieve aanpak zoals deze binnen het Unified Process is opgenomen is risicogedreven. Op basis van de top tien aan projectrisico's worden de doelen van de iteratie bepaald en daarmee de activiteiten. De risico's bepalen dus welke activiteiten worden uitgevoerd in een iteratie.

Risico's kunnen verschillend van aard zijn. Ze kunnen bijvoorbeeld bedrijfsmatig van aard zijn of technisch. Neem als voorbeeld de performance-eis. De eis levert door de complexiteit van de software een technisch risico op. In een iteratie moet het team dit risico wegnemen, oftewel ze moeten voldoende vertrouwen hebben dat het geen risico meer is. Lukt dat niet, dan zou het kunnen betekenen dat het project moet worden stopgezet. Het wegnemen van het risico gebeurt niet op papier, maar



René Hietkamp

is Senior Consultant bij Info Support en werkt als trainer en mentor aan trajecten voor de invoering van het Unified Process.

door werkende software te ontwikkelen die waarde heeft voor de klant en bewijs vormt dat het risico is weggenomen.

Baat bij beheersen risico's

Projecten met veel risico's zijn dus zeer gebaat bij een methode die actief risico's beheerst. Daarom zijn bedrijfskritische projecten zeer gebaat bij een iteratieve aanpak. Zij zullen de meeste waarde verkrijgen bij het toepassen van de nieuwe werkwijze. Het risico van de nieuwe werkwijze zelf kan worden weggenomen door deskundige begeleiding.

Alleen objectgeïntendeerd

Grote bedrijven hebben vaak te maken met oude systemen. Deze oude systemen zijn meestal niet geschreven in een objectgeoriënteerde taal, waardoor ze veel afhankelijkheden in zich hebben, zoals bijvoorbeeld afhankelijkheden met het datamodel. De afhankelijkheden maken het lastig om functionaliteit incrementeel toe te voegen. Het toevoegen van een veld in de database kan bijvoorbeeld gevolgen hebben voor de gehele software en niet alleen voor dat deel dat gewijzigd moet worden. Het vraagt dus om een goede strategie van het team om deze afhankelijkheden te managen.

Projectaanpak is niet gebonden aan programmeertaal.

De iteratieve aanpak, met iteraties als een vaste tijdsperiode waarin activiteiten worden gerealiseerd, is niet gebonden aan een programmeertaal. Het is een risicogestuurde projectaanpak en heeft dus niks met objectoriëntatie te maken. Met behulp van objectoriëntatie is het eenvoudiger functionaliteit incrementeel toe te voegen, maar dat maakt niet dat het voor oudere talen onmogelijk is.

Momenteel zijn we bijvoorbeeld iteraties aan het introduceren voor een project in een Tandem mainframe-omgeving. De risico's zijn anders, en de lengte van de iteratie is wat langer om alle afhankelijkheden te kunnen managen, maar het principe van een risicogedreven aanpak is hetzelfde.

Al succes met watervalmethode

Projectmanagers geven soms aan dat hun projecten al zeer succesvol zijn. Ze leveren hun projecten vrijwel altijd binnen de afgesproken tijd en geld op en hanteren daarbij de watervalmethode. Een aantal vervolgesprekken met onder andere het team levert helaas een ander beeld op. Het is de definitie van succes die hierbij komt kijken. Wanneer is het project succesvol? De projectmanager hanteert hiervoor een andere definitie dan de klant.

Projectsucces moet gelijk zijn aan klantsucces

Het hele project was voor de projectmanager succesvol als het budget voor een projectcode niet overschreden werd en de software op het afgesproken

Wat is iteratief software ontwikkelen?

Wat wordt eigenlijk bedoeld met iteratief ontwikkelen? Een iteratie is een vaste tijdsperiode waarin activiteiten voor software ontwikkeling worden uitgevoerd. Iteratief staat echter niet los van incrementeel. Incrementeel betekent dat de software groeit, in iedere iteratie wordt er een stukje software bijgebouwd. Met behulp van iteraties wordt incrementeel gebouwd aan een software release. Iedere iteratie levert een werkend deel op en met alle iteraties samen is de gewenste oplossing voor de klant gereed.

Het totale project wordt opgedeeld in kleinere eenheden (een iteratie) en in vaste tijdsblokken gerealiseerd. Door in iedere iteratie de belangrijkste risico's te adresseren en weg te nemen, is het project eerder onder controle, en meer voorspelbaar te maken.

Een belangrijke gedachte bij de iteratieve aanpak is het Pareto Principe waarbij de gedachte is dat slechts 20% van de gewenste functionaliteit al 80% van de waarde voor de organisatie vertegenwoordigt. Door de prioriteiten goed te stellen hoeft daarom niet alle gewenste functionaliteit gerealiseerd te worden.

Een populaire iteratieve aanpak van dit moment is Scrum. Maar ook binnen het Unified Process, zoals bijvoorbeeld het Essential Unified Process van Ivar Jacobson, is een iteratieve aanpak gemeengoed.

tijdstip werd opgeleverd. Met aanvullende wijzigingsverzoeken kon het budget worden verhoogd en bleef hij dus binnen budget. Door de extra functionaliteit die gerealiseerd moest worden kon ook de oplevertermijn worden uitgesteld. Knap gemanaged kun je stellen.

De klant vond het project minder succesvol. Hij werd met onverwachte extra uitgaven geconfronteerd en met uitstel van de oplevering, omdat de software toch niet bleek te doen wat hij ervan had verwacht. Voor hem was het project dus niet binnen tijd en geld gebleven.

Als het project voor de klant niet succesvol is, mogen wij het ook niet succesvol noemen. Het project moet waarde leveren voor de klant en voorspelbaar zijn. De iteratieve aanpak doet dat.



Een standup meeting om de voortgang en het werk van de dag te bespreken.

**Ook als niet
alles kan
worden
geleverd
heeft de
software
waarde voor
de klant.**

Iteraties leveren werkende software

Iedere iteratie behoort werkende software op te leveren, wat betekent dat de software na de vaste tijdsperiode van een iteratie volledig getest is. Door de software te demonstreren aan de klant kan deze direct beoordelen of de software voldoet aan de verwachtingen en kan hij direct zien dat de software waarde oplevert.

Klant bepaalt de prioriteit

Het team krijgt bij een iteratieve aanpak de mogelijkheid te leren, bijvoorbeeld hoe snel ze zijn met het ontwikkelen van software. Het team verkrijgt daarmee de mogelijkheid voorspelbaar te zijn en kan dus beter communiceren over wat wel en wat niet mogelijk is.

Ook bij een iteratieve aanpak gebeurt het wel eens dat niet alle software binnen de afgesproken termijn kan worden geleverd en dat de kosten om alles te realiseren daarom hoger zijn. Dit betekent bij een iteratieve aanpak dat de klant niet alle functionaliteit krijgt. De klant bepaalt de volgorde van software-ontwikkeling, zodat de functionaliteit met de meeste waarde vooraan staat. Hierdoor heeft de software, ook als niet alles kan worden opgeleverd, veel waarde voor de klant.

Kostenschatting

In een traditionele watervalmethode is er eerst een definitiestudie, dan een basisontwerp en vervolgens een detailontwerp. Aan het ontwerp wordt steeds meer detail toegevoegd, wat resulteert in een steeds nauwkeuriger kostenschatting. Op basis van een detailontwerp zal vrijwel iedere softwareleverancier een vaste prijs durven af te spreken voor de realisatie.

Een iteratieve aanpak heeft een andere manier om tot een goede kostenschatting te komen.

Een iteratieve aanpak, zoals in het Unified Process,

heeft een andere benadering om tot een nauwkeurige kostenschatting te komen. De benadering is gebaseerd op de veronderstelling dat zolang er geen software gerealiseerd is, er ook geen nauwkeurige schatting kan worden gemaakt. Een papieren ontwerp geeft dus per definitie een onnauwkeurige schatting. Een iteratieve aanpak vereist dus een andere denkwijze van projectmanagers. Dit zorgt voor de misvatting.

Het Unified Process kent vier fasen, namelijk: inceptie, elaboratie, constructie en transitie. De inceptiefase heeft, net als de definitiestudie, tot doel de haalbaarheid van het project te beoordelen en dan vooral de economische haalbaarheid. Na de inceptiefase kan een globale kostenschatting worden gegeven. Om meer nauwkeurigheid te verkrijgen moet de elaboratiefase worden uitgevoerd. Deze fase beoordeelt de grootste technische risico's op haalbaarheid. Het bewijs van de haalbaarheid is werkende software. De elaboratiefase heeft als resultaat: alle grote technische risico's zijn weggenomen en het belangrijkste deel van de software-eisen is beschreven. Op dat moment is een nauwkeurige kostenschatting mogelijk. De realisatie vindt vervolgens plaats in de constructiefase en de invoering in de transitiefase.

Papierwerk en micromanagement

Ook met het iteratief ontwikkelen van software blijft het nodig plannen te maken, met onder andere de definities van de mijlpalen, de releases en de iteraties. Vervolgens is er per iteratie een iteratieplan en een evaluatie. Het iteratieplan bevat de doelstellingen, de evaluatie beoordeelt welke doelen gehaald zijn en wat het team heeft geleerd van de iteratie. Zowel het iteratieplan als de evaluatie wordt op papier vastgelegd. Voor projectmanagers lijkt het veel papierwerk en met name de doelen per iteratie lijken op micromanagement. Wat is het nut?

Effectieve besturing

Projectplannen, iteratieplannen en evaluaties lijken veel papierwerk. Projectplannen zijn echter niet nieuw en zijn vaak voor de projectmanager nodig als overeenkomst met de klant. Door de iteratieplannen heeft het team meer focus en kan het team de voortgang effectiever bespreken en werkelijk inzicht geven. De evaluatie is voor het team een manier om inzicht te hebben in de status van het project en te kunnen leren. Het maken van de plannen levert met andere woorden veel op en is daarom zeer nuttig. Ik zal hieronder het nut van het opstellen van een projectplan en het overeenkomen en evalueren van een iteratie verder toelichten.

Opstellen projectplan

Een project moet gedefinieerd worden. Dat blijft ook bij een iteratieve aanpak het geval. Het moet bekend zijn welk doel het project heeft, welke



(c) 2009 Visual Management Blog

Een scrumboard, waarop het werk aan het project wordt bijgehouden.

afhankelijkheden er zijn en hoe de projectstructuur is. Standaardzaken voor een projectmanager. De iteratieve aanpak verandert daar niets aan. Het projectplan wordt aangepast door de iteraties te definiëren. Tijdens de eerste fasen van het project (de fasen inceptie en elaboratie uit het Unified Process) worden de tijdlijnen bepaald. Binnen deze fasen verandert het projectplan dus nog. Daarna zal het project de iteraties gaan uitvoeren volgens plan.

De iteratie met de klant overeenkomen.

De top tien aan risico's vormt steeds de basis voor de doelstellingen van een iteratie. Het team spreekt deze doelen af met de klant. Daarna kunnen de werkproducten en taken afgeleid en verdeeld worden onder het team. Plannen en afspreken van de iteratie is iets wat op één dag plaatsvindt, de planningsdag. Na de planningsdag weet iedereen wat hem of haar te doen staat.

Gedurende de iteratie zorgt de projectmanager ervoor dat de focus van het team op de doelstellingen blijft, belemmeringen weggenomen worden, de voortgang wordt gemeten en er een gezonde verdeling van werklast bestaat tussen de teamleden. Het iteratieplan helpt hierbij, omdat doelstellingen en taken gedefinieerd zijn en daardoor meetbaar.

Evalueren van de resultaten.

Op de laatste dag van de iteratie gaat het team de iteratie samen met de klant evalueren. De projectmanager bepaalt aan de hand van het iteratieplan welke doelen gedefinieerd zijn. Het team moet vervolgens bewijzen dat een doel is gehaald en toont het bewijs aan de klant. Ook bespreekt het team de leermomenten en de tegenslagen, waardoor de klant al vroeg daarover wordt geïnformeerd.

De projectmanager legt de evaluatie vast en borgt hiermee de kennis die wordt opgedaan. De projectmanager kan hiermee de teamprestatie verder optimaliseren.

Duurder dan waterval

Met een iteratieve aanpak wijzigt het team iedere iteratie de software om bijvoorbeeld functionaliteit toe te voegen. Dit maakt regressietesten noodzakelijk en dat werk neemt toe naarmate er meer iteraties zijn geweest. Het is daarom duurder dan de watervalmethode als je geen maatregelen neemt. Naast het regressietesten wordt ook het gezamenlijk plannen en evalueren van de iteraties als overhead gezien en daarom als duur ervaren.

Een volledige regressietest handmatig uitvoeren is praktisch niet te doen, wat automatisch testen als maatregel dan ook noodzakelijk maakt. Een alternatief is om niet een volledige regressietest uit te voeren. In dat geval is een teststrategie nodig waardoor met een minimale set aan testgevallen het vertrouwen in de werking kan worden bevestigd. Het team zal dus moeten nadenken hoe zij wil omgaan

met regressietesten. Een goede oplossing is mogelijk zonder dat die heel duur is.

In de voorgaande paragrafen is al beschreven wat het nut is van een planningsdag en van een evaluatie. Dat het team dat gezamenlijk doet is nuttig, omdat iedereen dan op de hoogte is van de doelen en de behaalde resultaten. Het team verkrijgt daarmee direct focus en kan effectiever aan de slag, waardoor het tijd bespaart. En daarom is het niet duur.

De klant wil niet itereren

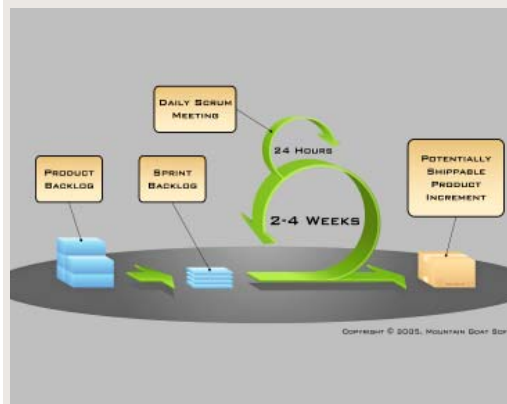
De klant wil het niet betekent vaak dat de klant geen extra risico's wil. Dit komt vaak voor als een iteratieve aanpak wordt geïntroduceerd. Een goede voorlichting aan de klant en hem laten meedenken in de beoogde wijze van samenwerking levert vaak positieve geluiden op.

De klant wil er natuurlijk graag zeker van zijn dat kwalitatief goede software op tijd en binnen budget wordt geleverd. Een iteratieve aanpak helpt daarbij en er zijn genoeg argumenten om de klant daarvan te overtuigen. Veel van die argumenten zijn in de voorgaande paragrafen al beschreven. Het begint echter bij de projectmanager die eerst zelf overtuigd moet zijn voordat hij zijn klant kan overtuigen. «

De klant laten meedenken en goed voorlichten helpt bij de introductie van de iteratieve aanpak.

Scrum: voorbeeld van een iteratieve projectaanpak

Met Scrum worden de plannen gemaakt vanuit een productbacklog. De productbacklog bevat de onderdelen die de klant in het product wil hebben opgenomen. De productbacklog wordt vertaald naar taken per iteratie, een sprint in Scrum termen. De evaluatie na een sprint wordt binnen Scrum de retrospectieve genoemd.



Iteratieve aanpak volgens Scrum.