

Groep: BC\_06

**project management in systeembeheerprojecten**

Cédric Denoo

Cedric Detemmerman

Levi Goessens

Robin Peerlinck

**TABLE OF CONTENTS**

**Voorwoord**

Project management methodologieën worden bij heel wat projecten gebruikt om een zekere structuur in de aanpak ervan te verkrijgen. Zo wordt het project in goede banen geleid en weet elke partij wat er van hem verwacht wordt. Binnen deze casus wordt onderzoek gedaan naar de verschillen in aanpak tussen systeembeheerprojecten en projecten die draaien rond softwareontwikkeling. Er wordt onderzocht of de methodologieën gebruikt bij softwareontwikkeling ook toepasbaar zijn op systeembeheerprojecten, alsook wat de verschillen zijn tussen de twee.

**De verschillende fasen van project management**

Project management kan worden opgedeeld in vijf fasen, de fasen geven een blik op het verloop van het project. Ze worden op de volgende manier opgedeeld: de initiatie, de planning, de uitvoering, de prestaties van het project alsook het monitoren van het project en dan de afronding van het project. Na het doorlopen van deze stappen wordt een afgerond en geslaagd project opgeleverd. Hieronder worden de fasen uitgebreid besproken:

De openingsfase, ook wel de Initiatiefase genoemd, is waar elk project mee start. Deze fase is meteen van groot belang, want er wordt meteen onderzocht of het project slaagkans heeft en het wel ondernomen kan worden. Op die manier wordt voorkomen dat een project al faalt nog voor het effectief begonnen is. Wanneer het project groen licht krijgt kan het officiële contract worden opgemaakt, met daarin onder andere de nodige hulpmiddelen, mensen die aan het project zullen werken, alsook het uiteindelijke doel van het project.

De tweede fase is de planningsfase, in deze fase worden de uiteindelijke doelen besproken. Voor deze fase kan gebruik gemaakt worden van een techniek die tijdens de lessen analyse reeds aan bod kwam, namelijk de S.M.A.R.T. methode. Tijdens deze fase is het ook belangrijk de balans te vinden van de ‘Project Management Triangle’: het budget, de beschikbare resources en personeel, de tijd en natuurlijk de kwaliteit van het project. Deze fase is pas afgesloten als iedereen weet wat er van hem of haar verwacht wordt en als enkele belangrijke documenten opgesteld zijn, zoals een Gantt Chart en een Risk Management Plan.

Bij de derde fase wordt het project effectief uitgewerkt, het is de uitvoeringsfase. Iedereen gaat aan het werk om de doelen van het project te behalen. Natuurlijk moeten de eerder besproken methoden en documenten in het achterhoofd gehouden worden om het project zo vlot mogelijk te laten verlopen.

De vierde fase heeft als doel de voortgang van het project te verzekeren. Hiervoor wordt gebruik gemaakt van Key Performance Indicators. Hiermee kan gecontroleerd worden of het project wel binnen de tijd en binnen het budget blijft dat vooraf bepaald werd, of de kwaliteit overeenstemt met de eisen, enz.

De vijfde en laatste fase is het afsluiten van het project. Het project is nu volledig afgerond en er kan nagegaan worden of het resultaat voldoet aan de vooropgestelde eisen. Indien het project management doorheen de voorgaande fasen gerespecteerd werd zouden hier geen onaangename verassingen mogen opduiken. Na het afsluiten van het project is ook de nazorg van het project van groot belang. Eventueel kunnen handleidingen worden gemaakt of trainingen worden voorzien en er het resultaat moet natuurlijk ook onderhouden worden, zelfs na de afsluiting van het project.

**De meest gebruikte Methodologieën bij project management**

# Agile

Agile betekent letterlijk vertaald behendig of lenig. In IT staat Agile voor het ontwikkelen van projecten in kort perioden die hooguit een maand duren. Deze korte perioden heten ‘iteraties’ en kunnen gezien worden als kleine projecten.

Agile is zelf geen methode voor project management, maar kan beter bekeken worden als een gedachte over hoe projecten kunnen worden aangepakt. Bekende methoden die gebruikmaken van de Agile manier van denken zijn Scrum en Kanban. Deze zijn niet hetzelfde als Agile, maar passen het op hun eigen manier toe. Elk framework heeft zijn eigen regels en richtlijnen om met een project om te gaan.

Er zijn 4 grote kenmerken aan methoden die op een Agile-manier werken.

De eerste is dat projecten worden opgesplitst in iteraties. Elke iteratie kan op zichzelf bekeken worden als een project dat zijn eigen analyse, planning en testing heeft. Door met iteraties te werken wordt de kans op problemen kleiner. Het tweede kenmerk is dat bij Agile de nadruk ligt op directe communicatie. Bij voorkeur wordt er altijd persoonlijk gecommuniceerd en wordt er zo weinig mogelijk geschreven.

Het derde kenmerk is dat het project steeds wordt verbeterd. Na iedere iteratie wordt er gekeken naar wat er beter zou kunnen gaan in volgende iteraties. Er wordt steeds een beoordeling gegeven over de iteratie en over het gehele project tot op dat punt. Zo leert men bij over hoe men het best omgaat met het project. Het laatste kenmerk is dat men steeds de nadruk legt op werkende software. Na iedere iteratie wordt er een werkend product afgeleverd. Je kan een iteratie hierbij vergelijken met een feature die wordt toegevoegd aan een app. De app werkt zowel voor als na de toevoeging van deze feature en zal altijd blijven werken. Wat er ook wordt toegevoegd in volgende iteraties.

# Waterval

Deze methode is afgeleid van de manier waarop men werkt in de constructie. Het project wordt opgedeeld in meerdere fases. Wanneer men een fout ontdekt in een vorige fase, keert men terug naar deze fase om de fout op te lossen en voert men vanaf die fase opnieuw alle opeenvolgende fases uit.

Projecten worden meestal opgesplitst in 5 fasen:

* Analyse
* Ontwerp
* Implementatie
* Testen
* Integratie

Het grootste voordeel van het gebruiken van deze methode is dat alles zeer goed wordt gedocumenteerd. Indien er mensen bij het project bijkomen, zullen deze kunnen bekijken wat er allemaal al gebeurd is en zullen ze direct kunnen verder werken.

Het grootste nadeel van deze methode is dat er niet correct kan ingeschat worden hoelang een project gaat duren. Indien er op het einde van het project, bij het testen of integreren van het product, zich fouten voordoen, moet men terug naar vorige fasen gaan kijken waar het precies is misgelopen. Indien de fout zich al in de analyse-fase voordeed, zullen ze dus opnieuw beginnen met de analyse-fase en daarna alle daaropvolgende fasen opnieuw uitvoeren. De tijdsduur van het project zal dus sterk afhangen van de fouten die ontdekt worden en in welke fase deze ontdekt worden.

**Hypothese omtrent project management in systeembeheerprojecten**

Na gericht onderzoek te hebben gedaan naar project management specifiek voor systeembeheerprojecten kan het volgende besloten worden: er is theoretisch weinig te vinden over dit onderwerp en de verschillen tussen projecten voor softwareontwikkeling en systeembeheerprojecten worden nooit echt aangekaart. Er kan dus vanuit worden gegaan dat het bij software-ontwikkelingsprojecten gebruikte project management ook van toepassing is op dat van systeembeheerprojecten. Ook is duidelijk geworden dat ook voor systeembeheer een planmatige aanpak van groot belang is. Het merendeel van de projecten heeft nood aan een gestructureerde opvolging en wordt voorafgegaan door een risicoanalyse om zoveel mogelijk addertjes onder het gras tegen te gaan.

Aangezien de theorie rond dit onderwerp eerder algemeen is, zal onderzocht moet worden hoe hiermee in de praktijk wordt omgegaan. Hiervoor zullen twee interviews gedaan worden met personen die hier dagelijks mee in contact komen, namelijk Yoshi De Boeck van Capgemini en Peter Keymeulen die werkt bij C.M.C. Door deze interviews kan achterhaald worden of er in de praktijk al dan niet gebruik wordt gemaakt van de verschillende methodologieën die in de theorie beschreven staan en of effectieve verschillen zijn tussen de aanpak van software-ontwikkelingsprojecten en systeembeheerprojecten.

**Analyse van het interview**

De methodologieën die we reeds eerder besproken hebben bleken ook in de praktijk het vaakst terug te komen. Zo achterhaalden we dat er ook binnen systeembeheer vaak aan de hand van agile gewerkt wordt. Deze methode wordt vooral gebruikt door de jongere generatie, terwijl de oudere generatie iets terughoudender is en teruggrijpt naar het oude bekende, namelijk de waterval methode. Soms wordt er ook een soort hybrid gebruikt tussen de verschillende methodologieën om zo een ‘best of both words’ te krijgen. Ook scrum werd bij de projecten toegepast in de vorm van dagelijkste stand-up meetings en het gebruik van een burndown chart.

Welke methodologie gebruik wordt bij een project bleek af te hangen van het budget en de scalability in het ene bedrijf, terwijl bij het andere bedrijf de methodologie van bovenaf opgelegd wordt. Hetzelfde fenomeen trad op bij de vraag of er sprake was van een ‘method prison’. Zo werd er weldegelijk agile ingevoerd in oudere projecten, maar werd er toch vastgehouden aan sommige procedures. Er werd gewerkt met sprints van telkens 3 weken, gevolgd door een planning, een review, een eventuele retrospective en tenslotte sprint refinement. In het andere bedrijf daarentegen waren er wel enkele standaard afspraken, maar kon men individueel bepalen welke werkmethode hun het best ligt en deze waarmee je je het comfortabelst voelt. De vrijheid van keuze voor een bepaalde aanpak hangt af van bedrijf tot bedrijf.

Op de vraag of er gebruik gemaakt word van custom management tools kregen we uit beide kampen volmondig het antwoord dat men gebruik maakt van off-the-shelf pakketten. Weliswaar met consultants met bedrijf-specifieke aanpassingen. Er zijn voldoende tools beschikbaar die voor elk project opnieuw gebruikt kunnen worden. Als ze al bestaan moet je ze niet zelf ontwikkelen, je moet ook niet het warm water opnieuw uitvinden.

Bij projecten waarbij het moeilijk is om met iteraties te werken probeert men toch om vast te houden aan het agile principe, al zijn sprint van 3 à 4 weken in die gevallen te lang. Toch blijft men te allen tijde demo driven werken. Er werd ook aangegeven dat indien je agile werkt aanpassingen maken aan het project eens je begonnen bent wordt vergemakkelijkt, maar dat dat niet wegneemt dat indien er steeds opnieuw veranderingen gemaakt worden dit wel als irritant wordt ervaren.

Bij de bedenking of er projecten zijn waarbij management tools volledig overbodig was kregen we als antwoord dat dat slecht in enkele gevallen voorkomt. Een voorbeeld hiervan zijn dan hulpsoftware of kleine hulptools. Voorts zullen altijd tools gebruikt worden om het project vlotter te laten verlopen.

Volgens de geïnterviewden zijn de verschillen tussen project managment bij software-ontwikkelingsprojecten en systeembeheer projecten eerder klein. Ze volgen beide gelijkaardige principes, maar het verschil zit hem in hoe snel ze afwijken van bepaalde methodologieën. In vele gevallen zullen ze echter op precies dezelfde manier van start gaan.

Wat betreft de opdelingen binnen het bedrijf inzake IT management kregen we weer uiteenlopende antwoorden door de grootte van de bedrijven. Enerzijds was er een standaard IT afdeling, de opsplitsing tussen front-end, back-end, hardware en software en de wisselwerking tussen back-end en front-end. Anderzijds, in een kleiner bedrijf, zijn de teams te klein om zich degelijk te specialiseren, elk lid van het team vervult dan verschillende rollen.

Ten slotte werd de vraag gesteld wat volgens hen zou mogen veranderen aan project management. Hierop werd geantwoord dat men van agile te vaak denkt dat men naar het einde van een project toe nog makkelijk wijzigingen kunnen doen omdat de theorie het zo beschrijft, maar dat in de praktijk te weinig rekening wordt gehouden met de analyses en de administratieve overhead dat zo’n veranderingen met zich meebrengen. Ook het verschil tussen de domeinen business en IT werd aangekaart, de werking tussen de twee domeinen verloopt minder nauw dan zou moeten doordat niet genoeg rekening wordt gehouden met de verschillen tussen de twee. Tenslotte werd ook nog aangehaald dat het niet altijd makkelijk is om aan alle informatie, zoals netwerk schema’s en documentatie te komen omdat ze niet op één centrale plaats beschikbaar zijn. Documentatie zou dus makkelijk beschikbaar moeten zijn om zo efficiënter te werken.