

Halverwege de jaren negentig kwam Microsoft tot de conclusie dat er aanpassingen gewenst waren op de wijze waarop intern de software-ontwikkeling werd aangepakt. Er werd besloten om projectleiders en ontwikkelaars uit verschillende teams ervaringen met elkaar te laten uitwisselen, om te bepalen welke werkmethoden wel of niet succesvol waren geweest. Het uiteindelijke resultaat van deze afstemmingen is uitgedomd in Microsoft Solutions Framework for Application Development (MSF AD). Het raamwerk wordt binnen het bedrijf inmiddels door alle ontwikkelteams van producten toegepast.

achtergrond

Microsoft Solutions Framework

Principes voor succesvolle software ontwikkeling

Men kwam tot dit inzicht nadat Office 95 Suite was uitgebracht. Het was de eerste maal dat producten uit verschillende productgroepen zoals Word en Excel als een geïntegreerde verzameling van samenwerkende producten op de markt werd gebracht. Omdat elke productgroep een eigen aanpak van software-ontwikkeling had, was niet het zo eenvoudig om de ontwikkelteams van de productgroepen te laten samenwerken. Na uitlevering van de suite bleek dat de klanten niet altijd even enthousiast waren over de stabiliteit van de uitgeleverde producten. Tevens bleken de producten veel features te bevatten, waar niemand echt behoefte aan had. Er werd besloten om een grondige evaluatie uit te voeren. Ook andere afdelingen binnen Microsoft, die met de producten van software-ontwikkeling te maken kregen, werden in de evaluaties meegenomen. Het MSF AD raamwerk dat hieruit voortvloeide, vindt inmiddels breed toepassing. Zo maakt Microsoft Consulting Services bij projecten gebruik van MSF AD. Het ambitieuze UK Government Gateway project is een aansprekend voorbeeld waarbij MSF AD succesvol buiten de muren van Microsoft is toegepast. Hierbij moest een team van zeventig personen, afkomstig uit verschillende organisaties, een aantal automatiseringssystemen van drie overheidsinstellingen aan elkaar koppelen. Binnen 75 dagen moest een werkende implementatie worden opgeleverd. Eerdere pogingen om dit voor elkaar te krijgen bleken steeds op een mislukking uit te draaien. Mede door de inzet van MSF AD bleek het toch moge-

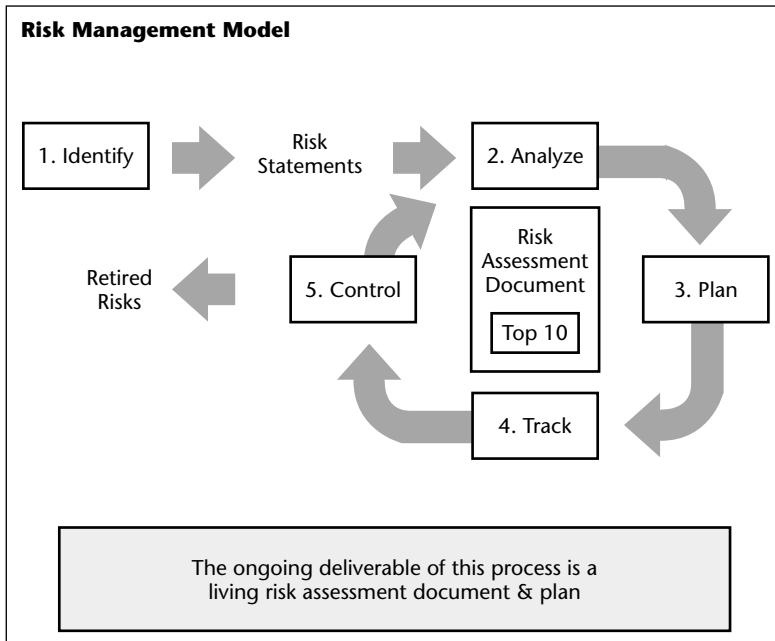
lijk, om in een complexe omgeving in korte tijd bruikbare oplossingen op te leveren.

GEEN METHODE Als aanvulling op Microsoft Solutions Framework for Application Development, heeft Microsoft nog een aantal andere raamwerken ontwikkeld. Voorbeelden hiervan zijn: MSF for Infrastructure Deployment en Microsoft Operations Framework (MOF). Bij ontwikkeling van deze raamwerken hebben de basisprincipes uit MSF AD als uitgangspunt gediend.

MSF AD wordt doelbewust een raamwerk genoemd en geen methode voor software ontwikkeling. Binnen het raamwerk worden een aantal op elkaar afgestemde modellen aangeboden, die in uiteenlopende situaties,

MSF geeft richtlijnen over het combineren van meerdere rollen door een persoon

op een aangepaste wijze kunnen worden ingezet. De modellen hebben als doel om een aantal basisproblemen, die nagenoeg binnen elk software ontwikkelproject te onderkennen zijn, het hoofd te bieden. Voor elk onderkend basisprobleem worden oplossingsprincipes voorgesteld, die soms op hoofdlijnen worden uitgewerkt en in een aantal gevallen wordt een meer gedetailleerde uitwerking gegeven. Er is nadruk gelegd op



FIGUUR 1: Het MSF Risk management model

problemen binnen een project die in de basis zijn terug te voeren tot een psychologische en/of sociale oorsprong. In mindere mate wordt aandacht geschonken aan “harde” software engineering aspecten. Om die reden is er veel vrijheid om te bepalen op welke wijze analyse en modellering wordt ingevuld. Dit betekent dat de mogelijkheid aanwezig is om voor analyse en ontwerp gebruik te maken van methoden en technieken, die bijvoorbeeld door (Rational) Unified Process of Select Perspective worden aangedragen. Belangrijk hierbij is dat de oplossingsprincipes, die binnen MSF gedefinieerd zijn, worden gerespecteerd.

De belangrijkste modellen die worden aangeboden zijn:

- Risico management model
- Team model
- Proces model

Geen van de modellen zijn echt uniek te noemen, echter wel de wijze waarop ze zijn gecombineerd. De centrale rol die risico management binnen het raamwerk inneemt is wel uniek.

RISICO MANAGEMENT MODEL Een van de hoofdoorzaken van projectmislukkingen is, doordat er niet goed met risico's wordt omgegaan. Het kan zijn dat risico's niet van tevoren of te laat onderkend zijn. Het kan ook zijn dat de kans op optreden en de eventuele gevolgen tijdens het optreden verkeerd zijn ingeschat. Uitgangspunt is dat elk project met veel een zeer uiteenlopende verzameling risico's te maken zal krijgen, die de voortgang van het project of de kwaliteit van het eindresultaat in gevaar kunnen brengen. Wanneer men

echter geen risico's durft te nemen zal er echter nooit een resultaat worden opgeleverd en kan men ook niet meer spreken over een project. MSF gaat er vanuit dat risico's onlosmakelijk aan een project verbonden zijn en dat deze op een pro-actieve wijze en via een gestructureerde methode benaderd moeten worden. MSF risico management model is een voorbeeld van een dergelijke methode.

Risico management is een gedeelde verantwoordelijkheid van het gehele ontwikkelteam. Iedereen zal tijdens de opstart van het project risico's moeten identificeren en inbrengen. Na het identificeren zal gezamenlijk worden ingeschat wat de kans is dat een risico optreedt en wat de verwachte gevolgen bij een eventueel optreden zullen zijn. Op basis van die analyse worden de risico's op een lijst gerangschikt. Per risico zal gezocht worden naar mogelijkheden om het risico te voorkomen of maatregelen te nemen om eventuele schade te beperken. Gedurende het gehele project zal iedereen actief moeten monitoren of bepaalde risico's daadwerkelijk gaan optreden. Dit monitoren is nog wat meer geformaliseerd, doordat een ieder, voor een aantal risico's verantwoordelijk kan worden gesteld.

Periodiek zal de risicolijst geëvalueerd worden en eventueel worden aangepast, op basis van aangepaste inzichten of bepaalde bevindingen. Risico management is daarom een aspect waarbij het gedurende het gehele project en bij het gehele team onder de aandacht blijft. Op die wijze wordt de kans op schadelijke effecten zo minimaal mogelijk gehouden. In figuur 1 is het risico management schematisch weergegeven:

Ook ten aanzien van de tijdsplanningen speelt het risico aspect een belangrijke rol. Zo wordt geadviseerd om risicovolle onderdelen zo vroeg mogelijk in de planning op te nemen. Op die wijze wordt het snelste inzicht gekregen in de daadwerkelijke gevolgen van een risico.

HET TEAMMODEL Een tweede model dat binnen MSF beschreven wordt is het teammodel. Er wordt uitgegaan van relatief kleine teams met gelijkwaardige leden, waarbij maximaal gebruik kan worden gemaakt van alle kennis en ervaring die binnen het team aanwezig is. In dit model worden ten eerste een aantal algemene principes en doelstellingen beschreven, die door elk teamlid nageleefd moeten worden. Voorbeelden van die principes zijn :

- Problemen en de oplossingen moeten vanuit het perspectief van de klant worden benaderd
- Iedereen moet gericht zijn op de realisatie van zichtbare eindproducten

Rol	Verantwoordelijk voor	Toelichting
Product management	Tevreden klanten	De product management rol zorgt ervoor dat product wordt uitgeleverd dat voldoet aan de verwachtingen van de opdrachtgevers/klanten
Process management	Oplevering volgens afspraak	De proces management rol zorgt ervoor dat een product volgens de afgesproken specificaties en binnen de afgesproken periode wordt opgeleverd
Development	Ontwikkelen volgens specificaties	De ontwikkelaars rol is ervoor verantwoordelijk dat het product volgens de afgesproken specificaties gerealiseerd wordt
User education	Bruikbaarheid voor gebruikers	De gebruikers educatie rol moet ervoor zorgdragen dat de productiviteit van de eindgebruikers door het product verbeterd wordt
Logistics management	Probleemloze uitrol	De logistiek management rol is verantwoordelijk voor een probleemloze uitrol van het product in de uiteindelijke productie-omgeving
Testing	Uitrol nadat alle problemen geadresseerd zijn	De test rol moet ervoor zorgdragen dat na in productie name van het product geen onverwachte zaken meer optreden

TABEL 1

- Elk lid moet er naar streven dat de vastgestelde opleverdatum wordt behaald
- Het leveren van kwaliteit moet als vanzelfsprekend worden ervaren
- Iedereen moet bereid zijn om voortdurend van elkaar en de opgetreden situaties te leren

Naast deze algemene principes zijn een aantal principes gedefinieerd, die direct gerelateerd zijn aan een zestal gespecialiseerde rollen. In elk ontwikkelteam zullen deze rollen vertegenwoordigd moeten zijn. Elke rol wordt als even belangrijk ervaren en heeft een even zware mate van inbreng. Elke rol heeft duidelijk omschreven verantwoordelijkheden, waarvoor hij verantwoordelijk kan worden gehouden. Daarnaast wordt het gehele team verantwoordelijk gehouden voor het uiteindelijke eindresultaat. Belangrijke beslissingen moeten daarom op basis van consensus genomen worden.

Tabel 1 geeft aan welke rollen binnen het teammodel worden onderkend. De namen die voor de rollen bedacht waren geven soms niet een direct beeld waarvoor een rol daadwerkelijk verantwoordelijk is. Zo is de user educatie rol niet alleen verantwoordelijk voor de opleidingen maar ook voor de handleidingen. En ook de gebruiksvriendelijkheid van de software valt onder de verantwoordelijkheid van deze rol.

Er is geen eenduidige leidende rol vastgelegd. Het is afhankelijk van de fase waarin het project zich verkeerd, welke rol op dat moment een leidende positie heeft. Alleen wanneer er intern onoverbrugbare verschillen optreden, is de procesmanager verantwoordelijk om tot

een oplossing te komen. Binnen het teammodel zijn tevens de communicatielijnen met de buitenwereld vastgelegd.

MSF geeft tevens richtlijnen over de wijze waarop meerdere rollen door een persoon gecombineerd zou kunnen worden of een opsplitsing van een rol over meerdere personen. Er worden tevens adviezen gegeven over de eindproducten, die een bepaalde rol bij een mijlpaal van een project zou moeten opleveren. Deze mijlpalen en de op te leveren producten, staan in het procesmodel beschreven.

HET PROCESMODEL MSF is te plaatsen onder de familie van iteratieve systeem ontwikkelingsmethoden, die begin jaren negentig als Rapid Application Development (RAD) werden aangeduid. Het gaat uit van korte overzichtelijke iteraties met een beperkte of gelimiteerde doorlooptijd. Hierbij moeten eindproducten van een iteratie gebruikt kunnen worden als basis voor een vervolgitatie. Wanneer de gewenste functionaliteit niet in een iteratie gerealiseerd kan worden, kunnen meerdere iteraties parallel en/of achtereenvolgens worden uitgevoerd. Het resultaat van een iteratie zal echter wel een afgeronde verzameling van producten moeten zijn die, technisch gezien, in productie genomen zou kunnen worden. Om er voor te zorgen dat de iteraties functioneel en technisch op elkaar blijven aansluiten wordt geadviseerd om vooraf een informatie architectuur en een technische architectuur te bepalen.

In tegenstelling tot een aantal evolutionaire RAD-methodes, zijn binnen een MSF AD iteratie een viertal

Adv.
Twice IT Training

Fase	Zichtbare mijlpaal	Toelichting
Envisioning	Vision approved	De visie over wat er gebouwd moet gaan worden en op welke wijze dat gaat gebeuren is door alle betrokkenen onderschreven
Planning	Project plan approved	Alle ontwerpen en de plannings zijn goedgekeurd
Development	Scope complete	De functionaliteit die uitgeleverd moet gaan worden is gebouwd en binnen het ontwikkelteam getest
Stabilizing	Release	Er is een versie van het product beschikbaar dat naar de klanten kan worden uitgeleverd

TABEL 2

hoofdmijlpalen gedefinieerd waarbij zichtbare tussenproducten opgeleverd moeten worden. Deze mijlpalen vormen tevens een zichtbaar bewijs naar de buitenwereld dat een mijlpaal gehaald is. Deze zichtbaarheid wordt van belang geacht om het vertrouwen binnen het team en van de opdrachtgevers te bevorderen. Zichtbare resultaten spreken mensen namelijk meer aan dan beloftes.

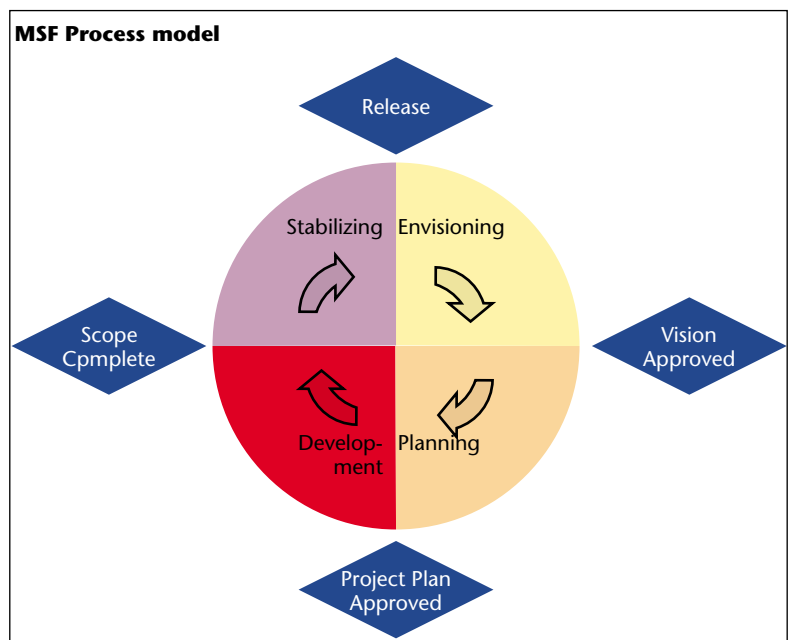
Naast deze extern gerichte mijlpalen worden intern gerichte mijlpalen gedefinieerd, die als synchronisatiepunt fungeren voor de interne teamrollen. Zo zal een functioneel ontwerp op een bepaald moment voldoende uitgekristalliseerd moeten zijn, voordat er wordt gestart met het technisch ontwerp. Het daadwerkelijk bevroren van documenten wordt zo lang mogelijk uitgesteld om te voorkomen dat resultaten van de activiteiten zonder voldoende afstemmingsmogelijkheden over de muur worden gegooid. Dit houdt in dat er ook binnenin een fase geen waterval aanpak gewenst is, en dat ook hier iteratieslagen worden aanbevolen. In de tabel 2 zijn de hoofdmijlpalen en hun bijhorende fase weergegeven; figuur 2 geeft het procesmodel nog eens grafisch weer. In de volgende vier paragrafen zal kort op de hoofdfasering van een software ontwikkel cyclus worden ingegaan.

ENVISIONING PHASE De leiding in deze fase ligt bij de product management rol. Er wordt bepaald welke doelstellingen voor de komende release worden gesteld en er wordt op globaal niveau bepaald op welke wijze de doelstellingen behaald kunnen worden. Er zal tevens een duidelijke afbakening worden afgesproken over wat niet en wat wel in de release moet worden opgenomen. In deze fase wordt het kernteam samengesteld en er wordt gezamenlijk met de teamleden en de belanghebbende een visie bepaald over wat een nieuwe release aan verbeteringen en uitbreidingen moet gaan opleveren. Hierbij komt ook de globale technische oplossingsrichting aan de orde. Dit is bijvoorbeeld van belang voor de uiteindelijke samenstelling van het team dat het project zal gaan uitvoeren. Het kernteam zal van het begin tot het einde bij een iteratie betrokken blijven.

PLANNING PHASE Op de mijlpaal van deze fase zullen twee onderdelen in detail duidelijk moeten zijn. Het ontwerp van de toepassing die gebouwd gaat worden en het projectplan waarmee dit ontwerp gerealiseerd zal worden. Voordat deze onderdelen kunnen worden ingevuld zal eerst een requirements analyse gedaan moeten worden.

Elk teamlid zal zelf de plannings voor de vervolgfases moeten maken, die door de proces manager tot een totaalplan wordt omgevormd. De hoofdverantwoordelijkheid in deze fase ligt bij de proces management rol. Hij is namelijk er eindverantwoordelijk dat er eindproduct volgens requirements wordt opgeleverd, binnen de afgesproken tijdslijn.

DEVELOPMENT PHASE Deze fase staat nagenoeg volledig in het teken van het ontwikkelen en testen van de toepassing. Elke ontwikkelaar is verantwoordelijk voor de eerste testen van zijn eigen resultaten. Wanneer de bouw van zijn functioneel deel is afgerond, wordt dit deel aan de test role overgedragen. Aan het einde van de dag zullen die ervoor zorgen dat alle opgeleverde deel-



FIGUUR 2: Het MSF proces model

producten geassembleerd worden, waarna er 's nachts geautomatiseerde tests op uitgevoerd kunnen worden. Het iedere nacht geautomatiseerd testen van de (dagelijkse) tussenresultaten beschouwt Microsoft als een van belangrijke eigenschap van MSF AD, om aan het einde van de rit stabiele software te kunnen opleveren. De fouten die de volgende dag gerapporteerd worden, moeten zo snel mogelijk door de ontwikkelaar van het component worden opgelost zodat andere ontwikkelaars er zo min mogelijk last van zullen ondervinden. Deze aanpak vereist dat ontwikkelaars hun eigen werkzaamheden ook in een aantal kleine iteraties hebben ingedeeld en op elkaar hebben afgestemd..

STABILIZING PHASE Tijdens de stabilisatie fase is alles ien iedereen er op gericht om tot een release kandidaat van het op te leveren product te komen. In deze fase worden testen door de afnemers en gebruikers uitgevoerd en probeert men zoveel de bevindingen van de

Microsoft meent dat XP binnen MSF AD in de development phase kan worden toegepast

testers in een volgende bètarelease opgelost te krijgen. Wanneer de voorraad aan op te lossen bevindingen een neergaande lijn gaat vertonen, zal men een definitieve releasedatum gaan vaststellen. Naast de release van een definitief product vormen de eindevaluaties van het product en het gevolgde proces een belangrijk eindproduct van de fase. Met behulp van deze evaluaties kunnen verbeteringen worden ingebracht voor de vervolgitaties.

VERGELIJKINGEN MET ANDERE METHODES In Nederland zijn er weinig organisaties te vinden waar MSF AD wordt toegepast. De afgelopen tijd weet de leverancier Rational met de methode Rational Unified Processing (RUP) de nodige aandacht te trekken. De laatste methode is een zeer uitgebreide verzameling aan richtlijnen, hulpmiddelen en beschrijvingen om een software ontwikkelproject in te richten waarbij uitgebreid gebruik wordt gemaakt van allerlei UML diagramtechnieken. RUP komt sterk uit de software engineering hoek en gaat nauwelijks in op de wat "zachtere" projectmatige aspecten. Het is voor veel organisaties erg moeilijk om te bepalen welke onderdelen ervan voor welke type project relevant zijn.

MSF AD is beter vergelijkbaar met Dynamic Systems Development Method (DSDM), een leveranciers onaf-

hankelijke aanpak die ook behoorlijk in de belangstelling staat. Beide aanpakken leveren een raamwerk waarbij de "zachtere" projectmatige aspecten meer worden benadrukt. In beide raamwerken worden nogal wat keuzes opengelaten, waardoor ze gemakkelijk aan verschillende situaties en omstandigheden zijn aan te passen. Wanneer de teamleden de principes snappen en daarna handelen, kan men relatief snel met de geboden handreikingen aan de slag. Over de wijze waarop systemen gemodelleerd en gedocumenteerd moeten worden, doen beide raamwerken weinig uitspraken. Binnen MSF AD vindt men modellering en documentatie wel van belang.

De essentie van eXtreme Programming (XP) is, dat direct gebruikers requirements zo snel mogelijk in werkende software wordt omgezet met een minimum aan modellering en documentatie. Minimalisering van overhead is een van de leidende principes van deze methode. Binnen MSF AD wordt overhead als minder negatief ervaren; als het maar volledig in dienst staat van alle andere principes die men heeft gedefinieerd.

Microsoft geeft zelf aan dat XP binnen MSF AD in de development phase kan worden toegepast, wanneer in de planning phase geconstateerd wordt dat het te moeilijk is om concrete requirements boven water te halen en daarbij de benodigde ontwerpen te maken.

CONCLUSIE MSF AD is een aanpak die gebaseerd is op "best practices" en die ook nog daadwerkelijk door organisaties succesvol wordt toegepast. De keuze om MSF AD te gebruiken hoeft geen enkel verband te hebben met de keuze om wel of geen Microsoft technologie te gaan toepassen. Het kan als nadeel worden ervaren dat niet alle aspecten van een software ontwikkelproces tot in detail zijn uitgewerkt. Het biedt echter de mogelijkheid om die details, die voor een omgeving daadwerkelijk van belang zijn, aan de meer algemeen geldende modellen binnen MSF AD toe te voegen. Hierbij kan gebruik worden gemaakt van principes, methodes die door andere software ontwikkelmethodieken en raamwerken worden aangedragen.

Men kan zich natuurlijk afvragen waarom men niet beter kan kiezen voor een volledig in detail uitgewerkte methode. Dergelijke methodes houden er meestal geen of weinig rekening mee dat software ontwikkeld wordt door een groep van mensen en voor een groep van mensen. Door MSF DA als basis te gebruiken krijgen deze aspecten de aandacht die nodig is, om software ontwikkelprojecten tot een succes te maken.

Richard Claassens

Ing. R. Claassens is werkzaam als ICT Architect en MSF Certified Trainer bij Info Support B.V.