

Microsoft Solutions Framework versie 4

HET SOFTWAREONTWIKKELPROCES VOLGENS MICROSOFT

Het Microsoft Solutions Framework beschrijft Microsoft's filosofie over het softwareontwikkelproces. MSF is inmiddels 11 jaar oud. Het werd in 1994 geïntroduceerd als een verzameling best practices gebaseerd op het interne Microsoft-ontwikkelproces en de ervaringen van Microsoft Consulting Services. Het is door de jaren heen geëvolueerd op basis van opgedane inzichten en trends in de softwareontwikkeling en staat momenteel op versie 3.

MSF is relatief onbekend gebleven bij softwareontwikkelaars, hoewel het onderdeel is van het Microsoft training- en certificatieprogramma. Dit gaat allemaal veranderen met de introductie van versie 4. MSF zal dan brede publiciteit krijgen. Het wordt onderdeel van Visual Studio Team System, komt prominent op MSDN staan en zal het onderwerp worden van boeken. Dit artikel biedt een eerste blik op de belangrijkste aspecten van MSF versie 4. Kennis van versie 3 wordt niet verondersteld, maar de verschillen met deze versie worden wel aangegeven voor de lezers die daarmee bekend zijn.

Agile & CMMI

MSF 4 komt in twee smaken, te weten: Agile en CMMI. Organisaties hebben een uiteenlopende dynamiek en tegenovergestelde prioriteiten. In sommige is snel een product kunnen opleveren en het aanpassen het belangrijkste. In andere organisaties heeft de standaardisatie en de herhaalbaarheid van projectuitvoering hogere prioriteit. Vandaar dat MSF 4 in twee smaken komt. Het 'Microsoft Solutions Framework for Agile Software Development' is gericht op teams die streven naar snelle softwareontwikkeling die zich vlug moet kunnen aanpassen aan veranderingen in de omgeving. Het 'Microsoft Solutions Framework for CMMI Process Improvement' is bedoeld voor organisaties die niveau 3 willen bereiken, zoals is gedefinieerd door het Capability Maturity Model van het Software Engineering Institute. (Meer informatie over CMMI is te vinden op <http://www.sei.cmu.edu/cmmi/>.) Dit artikel behandelt beide vari-

anten en geeft de verschillen aan. Laten we beginnen met wat MSF over de mensen in het softwareteam te zeggen heeft.

Het teammodel

Het teammodel beschrijft de verdeling van de verantwoordelijkheden in het project. Microsoft gelooft dat het belangrijkste voor een succesvol software ontwikkelproject niet het gereedschap, de technologie of de documentatie is, maar de mensen en hun samenwerking. Vandaar dat binnen MSF het teammodel centraal staat en de op één na belangrijkste best practice is. (De belangrijkste is de daily build.) Het model bestaat uit groepen (advocacy groups) en rollen zoals is weergegeven in afbeelding 1.

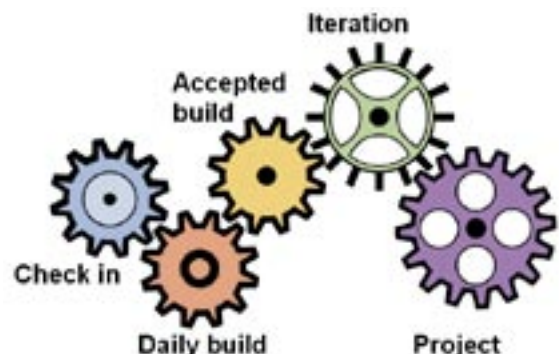
MSF Agile

MSF Agile bevat zeven groepen en zeven rollen – één per groep. MSF CMMI bevat dezelfde zeven groepen, maar heeft meer rollen per groep. De groepen komen overeen met de disciplines van MSF versie 3. De verantwoordelijkheden van iedere groep zijn:

- **Product Management.** Deze groep vertegenwoordigt de klant. De groep moet de business van de klant begrijpen en uitleggen aan de rest van het team. Program Management moet ook zeker stellen dat de klant het project aan het eind succesvol vindt.
- **Program Management.** Deze groep is verantwoordelijk voor de oplevering van de oplossing binnen de vastgestelde projectkaders. Program Management moet ervoor zorgen dat de juiste oplossing wordt opgeleverd op het afgesproken moment en dat aan de verwachtingen van alle belanghebbenden is beantwoord.
- **Architecture.** Deze groep is verantwoordelijk voor het systeem in zijn totaliteit. Totaliteit betekent hier de technische over-



Afbeelding 1. MSF Agile teammodel



Afbeelding 2. MSF 4 Projectcycli

Cyclus	Werkstroom	Rol
Project	Leg projectivisie vast	Business Analyst
	Rol product uit	Release Manager
Iteratie	Beschrijf scenario	Business Analyst
	Beschrijf niet-functionele eis	Business Analyst
	Creëer architectuur	Architect
	Plan iteratie in	Project Manager
	Begeleid project	Project Manager
Geaccepteerde Build	Sluit bevinding	Tester
	Test niet-functionele eis	Tester
	Test scenario	Tester
Daily Build	Bouw product	Developer
	Begeleid iteratie	Project Manager
Check In	Implementeer ontwikkeltaak	Developer
	Los bevinding op	Developer

Tabel 1. Cycli en werkstromen

wegingen en de standaarden waar de oplossing rekening mee moet houden; de infrastructuur waarop zij gaat draaien; haar plek in de bedrijfsportfolio en een roadmap voor toekomstige versies. De architectuurgroep moet er ook voor zorgen dat het systeem aan alle niet-functionele eisen gaat voldoen en dat zij toekomstvast is.

- **Development.** Deze groep is verantwoordelijk voor de technische oplossing. Development is de groep die de oplossing primair bouwt. Hieronder vallen doordachte technische keuzes, strakke ontwerpen, unit tests, onderhoudbare broncode van hoge kwaliteit en nauwkeurige bottom-up urenschattingen. (Development is verantwoordelijk voor de urenschattingen en Program Management is verantwoordelijk voor de overall planning op basis van deze schattingen.)
- **Test.** Deze groep is verantwoordelijk voor de kwaliteit van de oplossing vanuit het klantperspectief. De testgroep anticipeert op, signaleert en rapporteert alle kwesties die de kwaliteit van de oplossing omlaag brengen in de ogen van de eindgebruikers of klanten.
- **User Experience.** Deze groep is verantwoordelijk voor de effectiefste oplossing vanuit het perspectief van de eindgebruikers. De groep moet verstand hebben van de belangen en prioriteiten van de eindgebruikers, bewust zijn van de diverse behoeftes en ervoor zorgen dat de rest van het team rekening houdt met de bruikbaarheid van de oplossing.
- **Release Operations.** Deze groep is verantwoordelijk voor de soepele uitrol van de oplossing op de infrastructuur. De groep moet er ook voor zorgen dat de infrastructuur op tijd klaar is en dat deze compatibel is met de oplossing.

De meeste rollen in de groepen zijn vanzelfsprekend en worden daarom hier verder niet toegelicht. Alleen de volgende twee vereisen verdere uitleg:

- **Release Manager.** Het doel van deze rol is de succesvolle uitrol van het systeem. De Release Manager is verantwoordelijk voor het uitrolplan en geeft goedkeuring aan het systeem dat uiteindelijk in productie wordt genomen.
- **Business Analyst.** Deze rol is verantwoordelijk voor het beschrijven van de benodigde functionaliteit van het product door scenario's te schrijven en personas te definiëren. (Beide worden toegelicht verderop in het artikel.)

Het teammodel is verder gebaseerd op het zogenoemde 'team of peers'- (gelijken) principe. Alle teamleden delen de verantwoordelijkheid voor het projectresultaat. Iedere rol neemt een specifiek aspect van de totale kwaliteit van de oplossing voor zijn rekening. Er is geen hiërarchie - geen rol is belangrijker dan een andere rol. Het team gebruikt een besluitvormingsproces waarin iedere groep even goed vertegenwoordigd is - het

consensusmodel. Iedere vertegenwoordiger moet de belangen van zijn groep verdedigen. Het principe van een team van gelijken leidt tot een gezonde balans tijdens de projectuitvoering, omdat alle kwaliteitsaspecten continu aandacht krijgen. Op deze manier kan de projectmanager geen activiteit overslaan die development bijvoorbeeld essentieel vindt voor de kwaliteit van de broncode (bijvoorbeeld code reviews of refactoring). Noch kan architectuur een webgebaseerde applicatie opleggen als user experience van mening is dat deze niet productief zal zijn voor de eindgebruikers. In kleinere teams is het noodzakelijk dat één persoon meer rollen aanneemt. Daarbij is het belangrijk dat groepen met tegenstrijdige belangen niet samengevoegd worden. Stel dat development en user experience worden samengevoegd, dan gaan de ontwikkelaars bepalen hoe de gebruikersinterface eruit ziet. Dit leidt meestal tot de eenvoudigste te implementeren, maar niet de gebruiksvriendelijkste interface.

De cycli, werkstromen en activiteiten

Nu we weten wat de verantwoordelijkheden van de teamleden in het project zijn, wordt het tijd te kijken wat iedereen moet doen. De cycli, werkstromen en activiteiten beschrijven wie, wanneer, wat moet doen en hoe. Tijdens het verloop van een project worden diverse activiteiten met verschillende frequentie uitgevoerd. Zo wordt het document dat de projectvisie beschrijft éénmalig gecreëerd aan het begin van het project. Het projectplan wordt vaker bijgewerkt en de broncode nog vaker. De frequentie waarmee de diverse activiteiten in MSF 4 moet worden uitgevoerd, is beschreven aan de hand van de zogenoemde projectcycli. De vijf cycli zijn weergegeven in afbeelding 2.

Iedere cyclus beschrijft de werkstromen (works streams) die een rol moeten uitvoeren gedurende het project. MSF Agile bevat 14 werkstromen die als volgt verdeeld zijn over de cycli en de rollen¹; zie tabel 1.

De werkstromen op hun beurt bestaan uit activiteiten die gedetailleerde instructies bieden voor de uitvoering van de werkstroom. De laatste werkstroom in tabel 1, 'Los bevinding op', bijvoorbeeld bestaat uit negen activiteiten zoals is weergegeven in afbeelding 3. Iedere activiteit bevat stapsgewijze instructies om het doel te bereiken.



Afbeelding 3. Stapsgewijze instructies

Sommige van deze concepten zoals de daily build-cyclus en iteraties bestonden in MSF versie 3, maar de meeste zijn helemaal nieuw. Kenners van MSF 3 vragen zich misschien af wat er gebeurd is met het procesmodel en zijn vijf fases, namelijk envisioning, planning, build, stabilize, deploy. Deze bestaan nog steeds, alleen heten ze nu 'tracks' en worden ze vooral gebruikt om 'go/no-go' momenten in het project in te bouwen. Bijvoorbeeld, de 'go/no-go' aan het eind van de envisioning-track heeft te maken met de beslissing of het juiste product gebouwd gaat worden. De 'go/no-go' aan het eind van de stabilisatietrack moet binnen de werkstroom beslissen of het project klaar is voor uitrol, enzovoort.

De combinatie van cycli, werkstromen en activiteiten zorgt ervoor dat iedere rol precies weet wat hij moet doen en hoe hij het moet doen op ieder moment tijdens het project. Het enige dat hij hoeft te doen is zijn activiteiten in de juiste volgorde uit te voeren en de verwachte resultaten te produceren. Deze resultaten zijn de werkproducten.

De werkproducten zijn de resultaten van het project

Het belangrijkste resultaat van het project is uiteraard de software zelf, maar om zover te komen zijn er eerst andere zaken als projectplan, prototype, architectuurdiagrammen, tests, releaseplan en dergelijke nodig. Deze worden gezamenlijk werkproducten genoemd. MSF 4 Agile beschrijft in totaal 24 werkproducten. Ze zijn weergegeven in tabel 2.

Alleen een paar van de werkproducten zijn terug te vinden in MSF 3, de meeste zijn nieuw. Een volledige beschrijving van alle werkproducten valt buiten het kader van dit artikel. Wat wel opvalt, is het gebrek aan UML-diagrammen. UML wordt noch in MSF Agile, noch in MSF CMMI gebruikt. Het alternatief dat MSF gebruikt, heet Domain Specific Languages en wordt toegelicht in het artikel over Software Factories en DSLs in dit nummer van .NET Magazine. Iets anders dat zeker genoemd moet worden, is de analogie van use cases die in veel andere ontwikkelprocessen worden gebruikt. In MSF 4 dienen de scenario's hetzelfde doel. Een scenario is een tekstuele beschrijving van één pad die één eindgebruiker volgt door het systeem heen om een bepaald doel te bereiken. Hieronder wordt een voorbeeldscenario vanuit de MSF-documentatie gegeven:

Arnold heeft gehoord over het nieuwe lichtgewicht frame voor zijn mountainbike. Hij gaat naar de website van Adventure Works en gebruikt de zoekfunctionaliteit om zijn product te vinden. Hij typt 'mountainbike frames' in het zoekvenster en krijgt een lijst van alle producten die aan zijn zoekcriteria voldoen. Arnold kiest de 'Excalibur XJ11'. Hij ziet dan een nieuwe pagina met het plaatje van de Excalibur XJ11, een nieuw product. Hij ziet in de specificaties ook dat het nieuwe frame gemaakt wordt van een lichtgewicht metaalmengsel. De prijs van 677 euro wordt ook getoond. Arnold voegt het product aan zijn winkelmandje toe en gaat naar de 'kassa'.

Het scenario is geschreven in de taal van de eindgebruiker en wordt expres kort gehouden. Het biedt voldoende informatie, zodat een ontwikkelaar de gewenste functionaliteit kan implementeren en een tester deze kan testen zonder daarbij in details te

• Application Diagram	• Persona	• System Diagram
• Changeset	• Project Checklist	• Team Build
• Class Diagram	• Prototype	• Test Approach
• Code	• Quality of Service	• Test Result
• Iteration Plan	• Requirement List	• Threat Model
• Load Test	• Release Plan	• Unit Test
• Logical Datacenter Diagram	• Scenario Description	• Vision Statement
• Manual Test	• Scenario List	• Web Test
	• Storyboard	

Tabel 2. MSF 4 Agile werkproducten

vervallen. Dit is de Agile-filosofie. Wat eruitspringt in het scenario is het gebruik van een naam – Arnold. Arnold is geen echte eindgebruiker maar een persona. Personas zijn fictieve gebruikers die de typische skills, behoeftes, werkpatronen en achtergronden van een specifieke groep eindgebruikers samenvatten. Personas worden samengesteld op basis van onderzoek en interviews met echte gebruikers en ze maken het voor het projectteam eenvoudiger om de eindgebruikers te begrijpen en aan hen te relateren. De Arnold van het bovenste voorbeeld zou de volgende persona kunnen zijn:

Arnold leest tijdschriften over mountainbikes en is altijd op de hoogte van de nieuwe ontwikkelingen in dit veld. Hoewel mountainbikes alleen zijn weekendpassie zijn, koopt Arnold zijn uitrusting het liefst bij gespecialiseerde winkels. Hij koopt vaak nieuwe fietsen en heeft momenteel er drie – een mountainbike, een stadsfiets en een racefiets. Hij shopt graag op de Adventure Works-website vanwege zijn assortiment en de recensies, maar is vaak gefrustreerd door de beschikbaarheid van artikelen. Arnold is single en 25.

Microsoft zelf gebruikt vaak scenario's en personas tijdens de ontwikkeling van zijn eigen producten. De personas voor bijvoorbeeld de Visual Basic-ontwikkelaar beschrijven een mens die graag code ontwikkelt in veel kleine 'Edit-Compile-Debug'-iteraties. Vandaar dat Visual Basic als eerste de 'Edit and Continue'-feature aanbod. De persona voor de C#-ontwikkelaar beschrijft iemand die veel aandacht schenkt aan de structuur van zijn broncode – vandaar de nieuwe refactoringfunctionaliteit in Visual Studio 2005. Net als de persona en het scenario zijn de rest van de MSF-werkproducten praktisch en minimalistisch gehouden. Dit is in lijn met de Agile-filosofie waar de documentatie een communicatiemiddel is en geen controlemiddel.

MSF CMMI

MSF CMMI is een uitbreiding bovenop MSF Agile. De tweede smaak van MSF 4 heeft de codenaam 'MSF Formal' gehad, wat misschien de perceptie geeft dat MSF CMMI een zwaar en formeel proces is. Dit is zeker niet het geval. De oude codenaam komt niet meer overeen met de inhoud van het product dat verre van zwaar en formeel is. MSF CMMI is een uitbreiding bovenop MSF Agile en is niet een volledig apart verhaal. Met andere woorden, MSF CMMI blijft een Agile-methodologie met net genoeg extra's om aan de eisen van CMM niveau 3 te kunnen voldoen. Bijvoorbeeld, MSF CMMI heeft 50 werkproducten in vergelijking met de 24 van Agile. Dit is weliswaar tweemaal zo veel, maar blijft piepklein in vergelijking met de rond de 400 producten van een typische CMMI-implementatie.

Het teammodel van MSF CMMI is gebaseerd op dezelfde zeven groepen als MSF Agile. De rollen zijn echter gegroeid zoals is aangegeven in afbeelding 4.



Afbeelding 4. MSF CMMI Teammodel

- Analyse
- Inrichten omgevingen
- Beschrijf producteisen
- Manage change requests
- Issue management
- Risicomanagement
- Baseline configuratiemanagement
- Verifieer functionele eis
- Verifieer operationele eis
- Schrijf documentatie

Tabel 3. MSF 4 CMMI extra werkstromen

De extra rollen zijn:

- **Integrated Project Management Officer.** Deze rol is verantwoordelijk voor het management en de coördinatie van meer projecten in een organisatie en neemt de overkoepelende planning en resourcetoekenning voor zijn rekening.
- **Development Manager.** Deze rol is de lijnmanager van de ontwikkelafdeling. De developers en lead developers rapporteren aan haar/hem. De development manager is verantwoordelijk voor de capaciteit van het team en voor de constante verbetering van zijn skills. Deze rol heeft geen directe verantwoordelijkheid voor de voortgang van het project, maar wel voor de resources die de projectmanager aanstuurt.
- **Build Engineer.** Deze rol is verantwoordelijk voor integratie van de broncode en het succesvol uitvoeren van de dagelijkse builds van het systeem. Door deze specifieke skills in een aparte rol te beleggen, kunnen de andere ontwikkelaars zich richten op het realiseren van functionaliteit.
- **User Experience Architect.** Deze rol is verantwoordelijk voor de bruikbaarheid en de esthetica van de oplossing via prototypes, storyboarding, usability testing, enzovoort. GUI-ontwerpers en grafische ontwerpers vallen typisch onder deze rol.
- **User Education Specialist.** Deze rol is verantwoordelijk voor de opleiding van de eindgebruikers, zodat zij het systeem efficiënt kunnen gebruiken. De User Education Specialist ontwikkelt producthandleidingen, on-line help, beheerhandleidingen, opleidingsplannen, cursussen, enzovoort.
- **Sponsor.** Deze rol is verantwoordelijk voor de business case en de financiering van het project. Een project moet niet beginnen als er geen sponsor is.
- **Auditor.** De auditor is extern voor het project en is verantwoordelijk voor een objectieve beoordeling van de kwaliteit van de producten en processen van het project ten opzichte van de afgesproken standaarden.
- **Subject Matter Expert.** Deze rol heeft domeinspecifieke kennis en helpt de rest van het team de functionaliteit van het systeem op de juiste manier te realiseren.
- **Product Manager.** Deze rol bewaakt dat het product op één lijn zit met de strategische doelen van de organisatie en dat deze de oorspronkelijke visie realiseert.

MSF CMMI heeft dezelfde vijf cycli als Agile - Check In, Daily Build, Geaccepteerde Build, Iteratie en Project. Er zijn echter tien extra werkstromen bovenop de veertien werkstromen van Agile; zie tabel 3.

De meeste werkstromen hebben te maken met de focus op organisatorische aspecten van MSF CMMI. De grootste uitbreiding is op gebied van werkproducten. MSF 4 CMMI definieert vijftig werkproducten tegen de veertien van Agile. Ze zijn weergegeven in tabel 4.

De focus op standaardisatie en herhaalbaarheid van MSF CMMI is hier terug te zien in werkproducten voor risicomanagement, audits, peer reviews en diverse richtlijnen. Er is ook meer planning vooraf in de vorm van tien verschillende planningdocumenten. Verder zijn er documenten om diverse afspraken vast te leggen, zoals projectorganisatie, notulen, betrokkenheid van de belanghebbenden, enzovoort. Ten slotte, het is belangrijk te weten dat het gebruik van MSF CMMI niet automatisch tot CMMI-certificatie leidt. CMMI is een certificatie voor bedrijven en niet voor processen.

- Application
- Connection Diagram
- Architecture
- Alternative Solutions Proposal
- Asset List
- Audit Plan
- Audit Report
- Changeset
- Commitments
- Communication Plan
- Corrective Action Plan
- Delivery
- Documentation
- Deployment Plan
- Documentation Plan
- Domain Model
- Improvement
- Suggestion
- Iteration Effort and Cost Estimate
- Iteration Plan
- Knowledge and Skills Plan
- LAAAM Assessment Matrix
- Lessons Learned
- Lifestyle Snapshot
- Load Test
- Logical Data Center Diagram
- Manual Test
- Master Schedule
- Meeting Minutes
- Operations Review
- Presentation
- Organization Coding
- Guidelines
- Peer Review Checklist
- Peer Review Exception
- Review Minutes
- Persona
- Product
- Documentation
- Product Integration
- Plan
- Product Integration
- Sequence
- Project Coding
- Guidelines
- Project Organization
- Chart
- Project Plan
- Risk Categories and Sources
- Shared Vision
- Evangelization
- Presentation
- Stakeholder
- Involvement Plan
- Static Analysis
- Guidelines
- Static Analysis Rules
- Storyboard
- System Diagram
- Team Charter
- Document
- Test Approach
- Threat Model
- User Interface Flow Model
- Unit Test
- Vision Document
- Web Test

Tabel 4. MSF 4 CMMI werkproducten

Welke MSF-smaak past bij mij?

De grote gelijkenis tussen MSF Agile en MSF CMMI maakt het niet eenvoudiger om tussen de twee versies te kiezen. Het goede nieuws is dat het niet een of/of-beslissing is, omdat MSF CMMI een uitbreiding op Agile is. Met andere woorden, een organisatie kan altijd beginnen met Agile en op een gegeven moment overstappen naar CMMI. Vice versa, als CMMI te zwaar blijkt te zijn, kan de organisatie terugvallen op Agile. Het doel van CMMI-niveau 3 is gestandaardiseerde voorspelbare uitvoering van meer projecten in een organisatie. Dit betekent dat MSF CMMI een goede basis is voor algemene ontwikkelstraten die zijn bedoeld voor meer projecten. Als standaardisatie niet de belangrijkste prioriteit is, en de focus meer ligt op het snel en praktisch uitvoeren van één of meer losstaande projecten, dan is MSF Agile de betere kandidaat.

Ontwikkelstraat-in-a-box

Ongeacht hun formaliteit of vrijheid, blijven veel ontwikkelprocessen helaas een papieren exercitie en zijn ze niet terug te zien in de tools die de teamleden dagelijks gebruiken. MSF versie 3 heeft hetzelfde lot ondervonden, maar MSF 4 kan gelukkig rekenen op alle procesondersteuning die Visual Studio Team System te bieden heeft op basis van de zogenoemde processjablonen. Een procesjabloon beschrijft een ontwikkelproces op een manier die de tool begrijpt en vervolgens op diverse plekken kan faciliteren en afdwingen. Met behulp van VSTS kunnen de volgende MSF-aspecten tot leven komen:

- **Work items².** MSF Agile schrijft vijf work-items voor – bevin-ding, scenario, niet-functionele eis, taak en risico. MSF CMMI voegt nog change request, issue, review en eis toe. De gedefini-eerde work-items en de functionaliteit om hen te raadplegen en te bewerken komen in VSTS terug.
- **Metrieken².** MSF beschrijft de metrieken die gebruikt moeten worden om inzicht te verkrijgen in de voortgang en kwaliteit van het project. Voorbeelden zijn resterende werk, dekking-spercentage van de unit tests, enzovoort. VSTS biedt rapporta-ges voor deze metrieken aan op basis van de gegevens in zijn datawarehouse.
- **Werkproductsjablonen.** VSTS biedt sjablonen aan voor alle werkproducten die beide smaken van MSF voorschrijven.
- **Autorisaties.** Met behulp van de beveiligingsfunctionaliteit van VSTS kunnen de verantwoordelijkheden en rechten beschreven per MSF-rol ook daadwerkelijk afgedwongen worden. Bijvoor-

beeld dat iedereen een bevinding mag indienen, maar alleen een tester mag beslissen of hij is opgelost.

- **Handboek.** De beschrijving van MSF is beschikbaar in overzichtelijk referentieformaat als een website.

Ten slotte, misschien vanzelfsprekend, maar VSTS biedt ook de tools die de MSF-rollen nodig hebben om de van hen verwachte activiteiten te kunnen uitvoeren. Hierbij valt te denken aan de Team Build voor de Build Engineer, de Distributed System Designers voor de Architect, de Test-tools voor de tester, enzovoort. De processjablonen voor beide smaken van MSF 4 zijn te downloaden via de links aan het eind van dit artikel.

Best practices algemeen beschikbaar

Het is onmogelijk om alle aspecten van MSF 4 in één artikel te behandelen en daarom zijn zaken als mindsets, governance, qualities of service en how-to's in dit artikel buiten schot gebleven. Andere onderdelen als rapportages, work-items en queries zijn al behandeld in eerdere artikelen over VSTS. Het belangrijkste dat u moet onthouden is dat Microsoft het ontwikkelproces en de best practices aan iedereen beschikbaar stelt met Microsoft Solutions Framework versie 4. MSF komt voor in twee smaken - Agile en CMMI. Beide varianten zijn zeer praktisch ingestoken met hun wortels in de Agile-filosofie. Ten slotte biedt MSF samen met VSTS alles dat nodig is om als de basis voor een .NET-ontwikkelstraat te kunnen dienen.

Rossen Blagoev is werkzaam als Senior Developer Consultant bij Microsoft Services. Zijn e-mail adres is rossenb@microsoft.com.

Relevante links op het web:

<http://lab.msdn.microsoft.com/teamsystem/msf/>

<http://lab.msdn.microsoft.com/vs2005/teamsystem/>

<http://lab.msdn.microsoft.com/teamsystem/workshop/>

Noten

¹ Sommige werkstromen komen in meer cycli voor, maar dit is niet weergegeven in de tabel.

² Work-items en metrieken zijn uitgelegd in het artikel 'Visual Studio voor projectmanagers' in nummer 9 van het .NET Magazine.

(advertentie Microsoft Press)



Programming Microsoft® Windows® Forms

ISBN: 0-7356-2153-5

Auteru: Charles Petzold

Pagina: 400