

# Visual Studio Team System en RUP

## EEN GOEDE COGNAC OF TOCH ALLEEN MAAR TWEE WIJNEN?

Microsoft begeeft zich met Visual Studio Team System (VSTS) op een voor haar geheel nieuwe markt, namelijk die van Application Lifecycle Management (ALM). Bij ALM wordt een sterke integratie tussen de verschillende disciplines van systeemontwikkeling nagestreefd. Hierbij is de ondersteuning door tools een belangrijk aspect. De eerste ervaringen wijzen uit dat Team System hierin een goede start heeft gemaakt. In dit artikel wordt uiteengezet of VSTS en het populaire Rational Unified Process (RUP) met elkaar kunnen samenwerken. Is het een ideale combinatie zoals bij een goede cognac of moeten we ze gewoon als aparte wijnen zien die voor verschillende situaties geschikt zijn?

Als je het ontwikkelen van software uit de 'hack & pray'-sfeer wilt halen, dan is het toepassen van Application Lifecycle Management een goede keuze. Het werken met een integraal ontwikkelproces is een belangrijke succesfactor voor het realiseren van haalbare en gewenste systeemoplossingen. Een goed samenhangende ondersteuning van methoden, technieken, tools en templates is daarbij een must. Visual Studio Team System is een ontwikkelomgeving die je in staat stelt om te gaan met de huidige dynamiek en complexiteit van ontwikkeltrajecten.

Tien jaar geleden zagen de 'Three Amigo's', Ivar Jacobson, Jim Rumbaugh en Grady Booch al dat moderne softwareontwikkeling een dergelijke ondersteuning nodig heeft. Deze visie is goed terug te zien in RUP. Tooling speelt een prominente rol in RUP, maar omdat destijds geen geïntegreerde omgeving beschikbaar was, werd RUP oorspronkelijk ondersteund door een bonte verzameling van tools. Iedere tool dekte een specifiek deel van het proces af waarbij een zekere integratie aanwezig was om de gewenste dynamiek bij softwareontwikkeltrajecten voldoende te ondersteunen. De integratie was hierbij echter niet altijd stabiel en de samenhang niet altijd logisch. Inmiddels zijn de stabiliteit en samenhang sterker geworden. Ook zijn er meer leveranciers in de markt verschenen. Naast IBM Rational hebben bijvoorbeeld Telelogic, Borland en nu dus ook Microsoft een geïntegreerde toolomgeving beschikbaar.

Met de komst van Visual Studio Team System is Microsoft ook in de markt van de procesondersteuning gedoken en doet daarmee een poging de verschillende disciplines van een systeemontwikkeltraject bij elkaar te brengen. Team System gooit daarbij hoge ogen met het integraal aanbieden van de mogelijkheid om ieder willekeurig proces te ondersteunen door middel van de 'process template'. Deze template is een XML-bestand waarin je allerlei policies, flows (proces- en workflows), document-templates en standaarden kunt vastleggen die samen het proces vormgeven. Team System wil de mogelijkheid bieden om moderne dynamische softwareontwikkeling te ondersteunen. De hamvraag is of Team System – de procestemplate – een proces als RUP kan ondersteunen?

RUP wordt bij Atos Origin sinds eind jaren negentig intensief gebruikt bij het ontwikkelen van moderne software. Door de jaren heen is er een variant van RUP ontstaan die ook elementen van Prince 2 kent, met inmiddels een stevige installed base. Tijdens het intensieve Readiness Programma, dat het afgelopen jaar is uitgevoerd voor Team System, is volop aandacht besteed aan het samenbrengen van de tool en het proces. Hierbij is ook de nodige aandacht geschonken aan de combinatie Team System en RUP.

RUP is een ontwikkelproces dat is gericht op productontwikkeling van maatwerksoftware, met veel aandacht voor de interactie tussen stakeholders zoals gebruikers, netwerkbeheerders en businesspartijen. RUP kent vier fasen met vast omschreven doelstellingen:

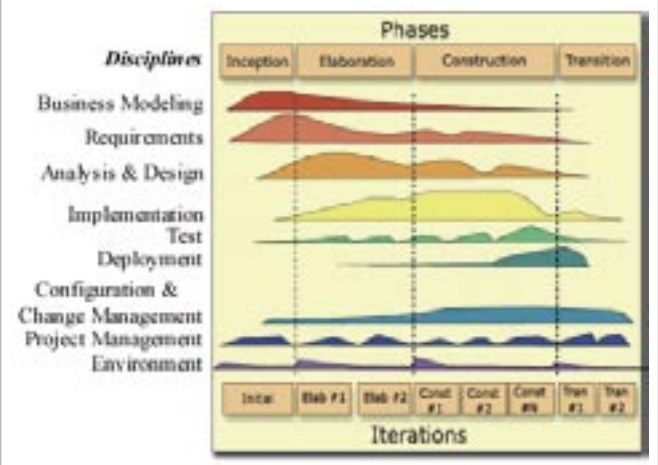
**Inception-fase:** Het doel van deze fase is de doelstellingen van het project helder te hebben, zodat een gefundeerde go/no-go-beslissing kan worden genomen. Proof of Concepts zijn een middel dat veel wordt toegepast om architectuurkeuzes te onderbouwen.

**Elaboration-fase:** Bij deze fase gaat het erom de doelstellingen en omvang van het systeem, de gekozen architectuur en de belangrijkste risico's gefundeerd te kunnen beoordelen. Kenmerkend hierbij is dat een prototype wordt gerealiseerd waarin de architectuur en de belangrijkste functionaliteit zijn geïmplementeerd.

**Construction-fase:** De construction-fase levert een product op dat gereed is om over te dragen aan de beheer- en gebruikersorganisatie. De grootste risico's zijn al afgedekt in de vorige fase, dus hier kan vooral op efficiëntie en productiviteit worden gestuurd.

**Transition-fase:** Het doel van deze fase is de software te accepteren, inclusief de materialen die gewenst zijn voor implementatie. Denk hierbij aan manuals en trainingen.

Afbeelding 1. RUP



Duidelijk is dat er tijd nodig is om aan elkaar te wennen: net als bij cognac is er een periode van rijping nodig. Toch kunnen we, ten tijde van de launch van beta-3, al enige uitspraken doen over de kwaliteit van deze 'cognac'.

RUP is te uitgebreid voor een platte XML-codering van de proce-  
stemplate. Daarom is gezocht naar ondersteunende tools voor het  
configureren van de proce-  
stemplate. Het Canadese bedrijf Osellus  
biedt deze ondersteuning in de vorm van een tool om processen  
te modelleren; zie afbeelding 2. Een aardige bijkomstigheid is dat  
RUP en Osellus dezelfde modelleertaal (SPeM) hanteren. Osel-  
lus was als partner van Microsoft al bezig met een integratie naar  
Team System van hun tool 'Iris'. De resultaten hiervan waren voor  
de RUP-modellering goed bruikbaar. Inmiddels is het 'eigen' RUP-  
proces met de Prince 2- elementen in een proce-  
stemplate vastge-  
legd. Deze wordt nu 'gerijpt' met een aantal pilot-projecten en is  
overtuigend ook al operationeel gebruikt.

Hoe goed gaan RUP en Team System samen? Er zijn uiteenlopende  
ervaringen uit de projecten naar boven gekomen. Laten we begin-  
nen met Team System dat de doelinfrastructuur als uitgangspunt  
heeft voor het ontwikkelproces. RUP echter laat zich leiden door  
de risico's en hanteert de stakeholders (gebruikers, belanghebben-  
den) als uitgangspunt. Voor RUP leiden de risico's en requirements  
vanuit alle stakeholders tot een stabiele architectuur. De doelinfra-  
structuur maakt hier onderdeel van uit en is niet het doel op zich;  
een wezenlijk verschil. Dit leverde een spanningsveld op tussen de  
implementatie van RUP in Team System en de RUP-denkwijzen  
van de projectleden. Een andere belangrijke constatering is dat  
Team System geen fasering kent zoals RUP. De vier fasen van RUP  
hebben vaste doelstellingen. Hierbij wordt met op maat samenge-  
stelde teams gewerkt. In Team System kan dit maar marginaal ver-  
taald worden, en dit leidt tot extra administratief werk. Dit zien we  
graag aangevuld worden in een volgende versie van Team System.  
Een opvallende eigenschap van RUP is het integraal toepassen van  
al haar disciplines, tegelijkertijd en met een zeer hoge interactiviteit.  
Nu kan tot op heden geen enkele geïntegreerde toolomgeving en  
geen enkel projectteam daar 100% invulling aan geven. Belangrijk  
voor het beantwoorden van de vraag is om te zien in hoeverre Team  
System hier beperkingen heeft. Per RUP-discipline worden huidige  
ervaringen in de relatie tussen RUP en Team System toegelicht:

## Businessmodellering

Businessmodellering is in RUP vaak een 'vreemde eend in de bijt'.  
RUP wordt meestal toegepast op applicatieniveau en dat vertaalt  
zich bijna altijd naar een enkel element in een businessmodel.  
Team System biedt het Software Factory-concept dat via de DSL's  
codegeneratie mogelijk moet maken. Businessmodellering kan  
hiervoor mogelijkheden bieden.

## Requirements

De discipline Requirements is een van de belangrijkste en tegelijk  
een van de moeilijkst inpasbare disciplines. Enerzijds is ze belang-  
rijk omdat de requirements de link vormen naar de opdrachtge-  
vers, business of andere stakeholders. Anderzijds is ze moeilijk  
inpasbaar omdat ze de basis vormt voor het bouwen en testen van  
het systeem. In RUP wordt hieraan veel aandacht besteed. In Team  
System nauwelijks! Met de nodige inleving en configuratie is het  
mogelijk om een work item een directe relatie te laten hebben met  
een requirement. Ook kan via de optie voor het toevoegen van  
documenten een work item aan een use case-beschrijving worden  
gekoppeld. Toch is dit vooral een work-around en is een zoektocht  
naar alternatieven een betere optie. Een van de alternatieven die  
zich aandient, is de aangekondigde en al getoonde integratie van  
Borlands CaliberRM. Zeker iets om nauwlettend te volgen.

## Analysis & Design

Zoals al eerder is aangegeven houdt Team System bij het ontwerpen  
rekening met deployment. De DSL's die nu beschikbaar komen zijn  
alle daarop gericht. Maar Team System biedt geen directe link tus-



Afbeelding 2. Osellus

sen bijvoorbeeld use case-beschrijvingen en statische en dynamische  
modellen zoals we die in UML kennen. Deze hebben in RUP een  
prominente plaats. Een gemiste kans? Nee, de ervaringen laten juist  
zien dat er een goede scheiding is tussen concept en implementatie.  
De conceptuele analyse, de designdiagrammen en beschrijvingen en  
de applicatiearchitectuur kunnen door deze scheiding goed apart  
worden gehouden van de class-diagrammen in Team System. De  
class-diagrammen hebben een 1-1-relatie met de code. Vooral in  
situaties waarbij specificeren en implementeren organisatorisch ver  
van elkaar staan is dit een prettige scheiding. In andere situaties zou  
ook hiervoor naar andere tools gekeken kunnen worden.

## Implementation

De discipline Implementation is de kern voor Visual Studio Team  
System. Na de vorige versie (2003) wordt een groot aantal extra  
features aan de programmeur geboden met het nieuwe .NET 2.0  
Framework. De ervaring leert dat vanuit de rol van een program-  
meur de procesondersteuning redelijk goed werkt. Redelijk, omdat  
toch wel veel muisklikken nodig zijn om de gewenste ondersteu-  
ning te krijgen. Kijken we naar ideeën voor de toekomst, dan gaat  
dit zeker verbeteren in een volgende versie van Team System. RUP  
besteedt relatief weinig aandacht aan deze discipline. Een belangrijke  
reden daarvoor is dat RUP niet gericht is op een specifieke technolo-  
gie. Team System gaat bijvoorbeeld heel ver met het integreren van  
de tool FxCop. FxCop biedt de mogelijkheid standaarden en richtlij-  
nen voor het bouwen te verplichten, zodat alle code bij het opleve-  
ren (inchecken) pas wordt geaccepteerd als de validatie correct is.

## Test

De mogelijkheden om te testen in Team System zijn uitgebreid.  
Zeker als je kijkt naar de mogelijkheid tot het afdwingen van  
unittests en de code coverage-feature. Code coverage maakt het  
mogelijk een policy in te stellen die builds niet toestaat als er in  
onvoldoende testcases in de code zijn voorzien. Kijkend vanuit het  
RUP-proces is testen in Team System lastig. RUP hanteert een ster-  
ke koppeling tussen testcases en requirements dat in Team System  
niet wordt geboden. Het aanpassen van de proce-  
stemplate op dit punt is in ontwikkeling. Op dit moment worden de testtools van  
Team System wel ingezet voor white box testing. Voor black box  
testing wordt (nog) uitgeweken naar andere tools.

## Deployment

In RUP levert iedere iteratie een werkende applicatie op. Deze  
oplevering vindt plaats binnen de discipline Deployment. In Team  
System wordt deployment goed ondersteund met uitgebreide  
mogelijkheden voor policies en het automatiseren van bijvoorbeeld  
bijbehorende documentatie en scripts voor verschillende platfor-  
men. De dunne basis die Team System voor iteratief ontwikkelen  
biedt, is in deze discipline wel goed merkbaar, en daar zou Micro-  
soft zeker iets aan moeten doen.

## Configuration & Change management

Naast Requirements is CCM een pijler van het RUP-proces. In Team  
System is dit goed geregeld. Team System biedt een integraal proces,  
waarin wijzigingenbeheer is opgenomen. Daarbij heeft Microsoft de  
repository in SQL Server 2005 geplaatst waardoor veel mogelijkheden

---

worden geboden voor source-control. Toch zijn er nog dingen die beter kunnen. Zo is het inpassen van Word-documenten in de repository lastiger dan gewenst. Verder merken we dat het (nog) niet mogelijk is om een hiërarchie van work items te hanteren. Met een hiërarchie van work items kunnen meer mensen in verschillende rollen aan eenzelfde hoofd-work item werken. RUP is geënt op het gezamenlijk werken met verantwoordelijkheden. Binnen Team System kan dit in de praktijk een woud aan work items opleveren, waarbij het een hele uitdaging is om het overzicht te houden.

## Projectmanagement

Projectmanagement is bij RUP voornamelijk gericht op het ontwikkelteam en veel minder op projectmanagement van de staande organisatie. Vandaar ook dat RUP vaak in combinatie met een methodiek als Prince2 wordt toegepast. Team System biedt een goede integratie met Microsoft Project en Excel. Deze integratie is goed vanwege de mogelijkheid om het zelf uit te breiden en dat bleek ook nodig omdat de standaard integratie te dun was. Met extra macro's en aanpassingen in de procestemplate is de combinatie met Microsoft Project goed werkbaar. Prince2 en RUP zijn samen operationeel binnen de Team System-omgeving.

## Environment

Wat RUP Environment noemt, is afgezien van de technische onderdelen van de omgeving, vooral terug te brengen naar de procestemplate. De procestemplate biedt, als ze zover geconfigureerd en/of gemodelleerd is, een goed en dynamisch alternatief voor de development-case van RUP. Benodigde templates, documenten, flows en zelfs de mogelijkheid om bijvoorbeeld een referentie-architectuur en/of een framework in de omgeving te plaatsen, worden geboden. RUP gaat uit van iteratief ontwikkelen en biedt hiervoor veel ondersteuning. Maar ook in RUP moet het iteratief werken in de eerste plaats door het team zelf gepland en uitgevoerd worden. Team System biedt vanuit haar view op deployment, source-control en integraal testen goede handvatten voor het opleveren van de benodigde kwaliteit per iteratie. Alleen is het toewijzen en afhandelen van de work items in de praktijk nogal basic. Dit vergt meer discipline van het team dan je zou willen. Misschien dat in een toekomstige versie van Team System de mogelijkheden worden uitgebreid.

## Wijn moet rijpen

Geconcludeerd kan worden dat RUP en Team System twee goede wijnen zijn. Team System is nog wel jong en vraagt enige tijd voor verdere rijping. We hebben hier wel te maken met twee wijnen die uiteenlopende karakters hebben. De centrale vraag in dit artikel was of ze gezamenlijk een basis kunnen vormen voor een goede cognac. Of moeten we ze toch apart houden? Onze ervaringen en inzichten hebben geleerd dat er zeker potentie is tot het maken van een goede cognac. De tijd zal hierbij leren of het ook een gewilde cognac gaat worden.

**Peter Mazereeuw** is Rational Unified Process (RUP) trainer/consultant en product manager Visual Studio Team System bij Atos Origin. Peter werkt al meer dan vijf jaar als RUP-consultant. Sinds VSTS twee jaar geleden het daglicht zag, heeft Peter zich in VSTS verdiept en is hij de drijvende kracht achter de invoering van Visual Studio Team System bij Atos Origin. [peter.mazereeuw@atosorigin.com](mailto:peter.mazereeuw@atosorigin.com)

**Roy van Reijssen** is projectleider en RUP-trainer/consultant bij Atos Origin. Roy werkt al sinds 1999 aan projecten waarin RUP wordt ingezet. Hij is een van de grondleggers van een RUP Awareness-programma dat al bij veel organisaties in Nederland is uitgevoerd. Sinds de invoering van VSTS bij Atos Origin is Roy nauw betrokken bij de procesinrichting. [Roy.vanReijssen@atosorigin.com](mailto:Roy.vanReijssen@atosorigin.com)

### Nuttige internetadressen

<http://www.gotdotnet.com/team/fxcop/>

<http://www.osellus.com/>

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