

In het eerste deel van dit artikel (Software Release Magazine, nr. 4, juni 2003) zijn de karakteristieken van RUP, het Rational Unified Process, beschreven. Daarbij is het proces rond invoering van RUP toegelicht aan de hand van een praktijkervaring. In dit artikel gaan we dieper in op de verschillende aandachtsgebieden binnen een RUP-systeemontwikkel-project en beschrijven we de lessen die we geleerd hebben tijdens enkele achtereenvolgende iteraties.

## Achtergrond

# Invoering van RUP - Lessons learned (2)

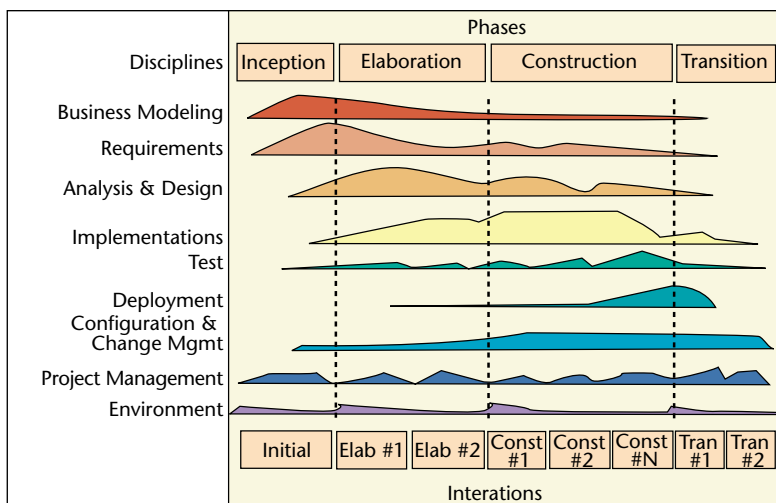
## Uitdagingen en kritieke succesfactoren

Als indeling volgen we de disciplines van RUP. Het hier beschreven project heeft de volgende karakteristieken:

- Systeemontwikkeling wil men in-house gaan uitvoeren. Dat was tot dusver uitbesteed.
- RUP is nieuw voor de projectmedewerkers. Daarom zullen we ook de ontwikkelmethode stapsgewijs (en 'just-in-time') gaan invoeren.
- De modelleerdisciplines maken gebruik van de tools van IBM-Rational. Bouwen gebeurt met tools van Oracle.
- De invoering van RUP vindt bottom-up plaats: het is de keuze van het software ontwikkelproject en RUP zal pas bij gebleken geschiktheid tot keuze op organisatieniveau uitgroeien.

**BUSINESS MODELING** Als je Business Modeling oppakt moet je het aan het begin van je project doen. Je geeft de medewerkers daarmee een eenduidig referentiekader. In ons geval is het bewust achterwege gelaten. Er lag al een uitgebreide architectuurstudie, inclusief procesbeschrijvingen. We zagen er onvoldoende toegevoegde waarde in om deze processen nu op andere wijze (met UML) in een ander tool (Rational Rose) te beschrijven.

**REQUIREMENTS MANAGEMENT** In de allereerste iteratie waren de contacten en procedures naar de gebruikersvertegenwoordigers nog niet op orde. Met de opdrachtgever is toen afgesproken dat het projectteam zich zou richten op een prototype waarin de mogelijkheid van een "generieke applicatie" werd bewezen. Inzet van RequisitePro als hulpmiddel is bewust niet geïntroduceerd in deze iteratie, die toch al bol stond van de nieuwe zaken. In de tweede iteratie is Requirements Management serieus gestart (met RequisitePro en bijbehorende training). Toch zagen we al gauw dat de eerste insteek "te licht" was en niet de gewenste traceability zou opleveren. Traceability staat hierbij voor de mogelijkheid om het spoor te volgen van Business Needs via Product Features naar onderdelen van het Use Case Model en het Design Model en uiteindelijk naar softwarecomponenten. Halverwege de tweede iteratie is via een aanscherping van het Requirements Management Plan een degelijker werkwijze doorgevoerd. Een wijzigingsprocedure kan overigens in dit stadium, waarin er nog geen operationele software is, heel licht gehouden worden. Het is ook van belang om terughoudend te zijn met het aantal



FIGUUR 1. RUP kent een aantal disciplines die elk een eigen workflow hebben.

attributen dat per requirement wordt vastgelegd, alles moet immers onderhouden worden. Een statusattribuut kan zinvol zijn, maar tracht daarmee niet aan te geven of het requirement 'in ontwikkeling', 'in test' of 'in productie' is. Geef aan in welke build/release een requirement wordt geïmplementeerd en registreer de status van zo'n build elders.

**ANALYSE & DESIGN** Analyse en Design zijn vanaf het begin ondersteund met Rational Rose. In de eerste iteratie werd toepassing van Rose ook als grensafbakening gebruikt tussen medewerkers: degenen die de use case diagrammen vervaardigden, maakten ook de sequence diagrammen en de "view of participating classes". De bouwers gingen op basis hiervan aan de slag met de Oracle bouw-tools. Later is vastgesteld dat Design en Bouw beter in een team verenigd kon worden dan Requirementsanalyse en Design. Kortom, het overdrachtpunt tussen personen verschoof naar het punt tussen use case (= requirement) en sequence diagram (= design).

**IMPLEMENTATION (BOUW)** De programmerings-/bouwactiviteiten vonden plaats in een Oracle-omgeving. Database-ontwerp en -generatie verliepen via Oracle Designer; terwijl Oracle JDeveloper gebruikt is voor coderen, debuggen en build. In JDeveloper is het maken van UML-diagrammen beperkt mogelijk (activity en class diagram). Wij hebben echter besloten om alle gewenste diagrammen in Rose vast te leggen, zodat daar maximale integratie (en traceability!) ontstaat. Tussen de modellen in Rose en de uitwerking in Oracle bestaat geen harde koppeling. Onderzoek naar zo'n koppeling via forward of reverse engineering is op de activiteitenlijst gezet, maar met een lage prioriteit. De koppeling verliep in ons geval dus via de "human interface": de designer en de implementer waren gewoonlijk één en dezelfde persoon. Dat brengt natuurlijk risico's voor een toekomstige onderhoudsfase met zich mee. Derhalve is geïnvesteerd in de "Design Guide", het RUP-artifact dat Designers en Implementers richtlijnen geeft. De beschreven stappen bij de overgang van Design naar Implementation zijn eenduidig en straightforward, zodat ze begrijpelijk zijn voor een toekomstige onderhoudsploeg. Daarbij nam het gebruik van wizards in JDeveloper om classes te genereren veel werk uit handen.

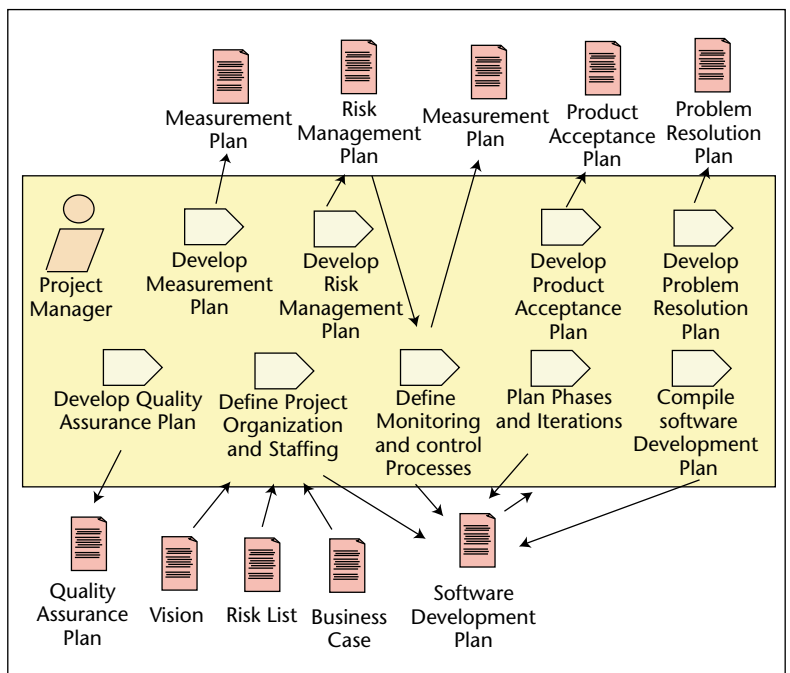
**TEST** Test is in de eerste iteratie bewust low profile gehouden: er is nog niet getest aan de hand van RUP-testartifacts. In de tweede iteratie is er wat organisatorische onrust geweest: moet het testen wel belegd worden binnen het projectteam, of moet het juist onafhankelijk zijn? Dat kunnen we zien als een uiting van de spanning tussen "agile" (lichtvoetig en wendbaar) werken zoals hoort bij een iteratieve aanpak waarin je kort-

cyclisch wilt werken en "formeel", zoals past bij een organisatie die gewend was de bouw van systemen uit te besteden maar daarbij het testen in eigen hand te houden.

RUP kent "continuously verify quality" als een van de onderliggende best practices en dus hoort er in ieder geval vanuit het project het nodige aan testen te worden gedaan. Zo levert vroegtijdig testen bijvoorbeeld een bijdrage aan een goede (dat wil zeggen: een eenduidige en toetsbare) formulering van de requirements. Omdat het testen zo moeizaam van de grond kwam en ook omdat geschikte huisvesting ontbrak is op zeker moment besloten om het testen uit te besteden aan het Advanced Test Center van Cap Gemini Ernst & Young. Hoewel aan de late kant bood dat wel een vliegende start: er kon snel een professionele en ingewerkte club aan de slag. Nadelig was de afstand; die vereiste wat formelere afspraken over tijdstip en wijze van terugkoppeling.

Een van de lessons learned is dat het voor gebruikers die voorheen gewend waren om te testen op nagenoeg afgeronde software een cultuuromslag betekent om tussenoplevingen te bekijken. Zonder juiste voorbereiding en instelling overheerst dan de teleurstelling om wat nog niet gerealiseerd is. Kortom, verwachtingsmanagement, goede voorlichting over het iteratieve proces en duidelijkheid over wat wel en wat niet opgeleverd wordt, zijn essentieel voor een positieve insteek.

**DEPLOYMENT** Min of meer gelijktijdig met de start van het softwareontwikkelpject is een team in het leven geroepen dat zich op het voorbereiden van de gebruikersomgeving en het operationaliseren ging



FIGUUR 2. RUP voor het ontwikkelen van een Software Development Plan

richten. Op zich is zo'n splitsing best verdedigbaar. Maar de samenwerking en afstemming met dit invoeringsteam is bemoeilijkt omdat RUP aanvankelijk alleen binnen het softwareontwikkelpject is ingevoerd, terwijl het invoeringsteam een eigen terminologie en proces hanteerde. Later, bij een reorganisatie op overkoepelend niveau is dit overigens op een lijn getrokken: RUP.

**PROJECT MANAGEMENT** Zoals we in figuur 2 kunnen zien schenkt RUP voldoende aandacht aan de diverse aandachtsgebieden die als deelplan van zo'n SDP ontwikkeld moeten worden. In de eerste iteratie is vanuit de RUP-invoering wat te weinig aandacht besteed aan het ondersteunen van het projectmanagement. Er is vertrouwd op ervaring rond plannen en voortgangsbewaking. In de beginperiode is de aandacht gericht op het RUPpificeren van de disciplines Requirements, Analysis & Design, Implementation en de ondersteunende discipline Environment. Maar ook voor de projectmanagers lagen er grote uitdagingen: nieuwe methode, iteratief werken, nieuwe deliverables (RUP-artifacts), gebaseerd op nieuwe technieken (UML), te realiseren met nieuwe tools (Rational en Oracle).

Wenselijk in zo'n periode is de beschikbaarheid van praktijkervaring, maar de combinatie van methode en ontwikkeltools was niet alleen voor de klantorganisatie nieuw, ook de ingeschakelde consultants van Cap Gemini Ernst & Young, Rational en Oracle moesten hier hun weg nog vinden: we waren trendsetters, pioniers. Dan moet je in elk geval tijdens de elaboratiefase zorgen voor een klein team met ervaren medewerkers. Bij zo'n opstartfase horen wellicht de "local heroes" die we kennen uit het eerste niveau van CMM (Capability Maturity Model). Maar we hadden vanuit de kennis van CMM moeten beseffen dat projectmanagement en voortgangsbewaking (Tracking & Oversight) al bij de eerstvolgende stap op de CMM-ladder aan bod komen en wellicht nog belangrijker zijn als randvoorwaarden voor succes dan het inzoomen op de Software Engineering disciplines.

**CHANGE & CONFIGURATION MANAGEMENT** Het werken in een team brengt met zich mee dat er een oplossing gevonden moet worden voor het gemeenschappelijk werken met dezelfde bestanden (artifacts). Soms biedt het tool die mogelijkheid (zoals RequisitePro met zijn onderliggende database), soms is het goed werkbaar om af te spreken dat er sequentieel aan een product gewerkt wordt, soms moet een bestand in kleinere stukken verdeeld worden (bijvoorbeeld bij de "repository" achter Rational Rose) om elkaar niet te hinderen. RUP vraagt hier om een "Configuration Management Plan", waarin niet alleen Configuratie management, maar ook Change Management aan bod komen. In onze omgeving konden we toe met een licht tool (Visual Source-



FIGUUR 3. De tree browser van het 'projectweb'

Safe), omdat we (nog) geen behoefte hadden aan parallelle ontwikkeling van programmasources. Bij de tools-evaluatie aan het eind van iteratie E1 (de eerste iteratie in de Elaboration-fase) is wel besloten om Rational's ClearCase te gaan inzetten, mede omdat er een goede integratie is vanuit JDeveloper. In de praktijk is de invoering echter enkele malen uitgesteld, omdat er geen urgentie gevoeld werd.

Ook procedures rond Change zijn zeer licht gehouden. Behoeft aan zwaardere processen en een invulling van een "Change Control Board" zal zeker komen als de software eenmaal een eerste operationele uitrol achter de rug heeft. Ook wensen rond een "one click install" procedure bij het promoveren van releases van Ontwikkelaar naar Test- naar Acceptatie- naar Productieomgevingen zijn naar de toekomst doorgeschoven.

**ENVIRONMENT** Environment bleek een zware discipline. Bij de start van het project was er nog geen proces, waren er nog geen (opgeleide) medewerkers, nog geen tools, nog geen ontwikkelinfrastructuur... In de praktijk is gebleken dat een "projectweb" een bindend

*Lees verder op pagina 49.*

element vormt in de communicatie tussen de medewerkers. RUP geeft hiervoor een template. In figuur 3 zien we via een afbeelding van de treebrowser van ons projectweb welke onderwerpen daar zoal aan bod kwamen.

Op zeker moment ontstond de behoefte aan een aanscherping van het proces. Via workshops zijn afspraken gemaakt, die uiteindelijk resulteerden in enkele proces-schema's op ons projectweb. Binnen RUP is de flow van activiteit naar activiteit vaak indirect (via tussenliggende artifacts) en opgedeeld over diverse schema's. Onze eigen compacte schema's boden een beter overzicht.

Binnen Environment valt ook het up-and-running houden van de ontwikkelomgeving. Daarvoor was weliswaar een system administrator uit de klantorganisatie toegewezen, maar bij diens afwezigheid vielen we terug op een helpdesk die ingericht was voor het ondersteunen van kantoorautomatiseringsapplicaties. Daar ging wel eens wat tijd verloren.

Een ander onderwerp binnen Environment zijn de "Guides". Wij hebben er uiteindelijk acht onderkend en vervaardigd: Requirements Management Guide; Use Case Modeling Guide; User Interface Guide; Design Guide; Programming Guide; Test Guide; Tools Guide; Configuration Management Guide.

Hierin zijn de projectspecifieke afspraken op de diverse gebieden vastgelegd. Voor enkele van deze Guides biedt RUP een goede eerste versie, enkele andere zijn van scratch opgebouwd. Belangrijk is wel om voor alle Guides een eerstverantwoordelijke voor het onderhoud aan te wijzen en de werkzaamheden in te plannen: de werkzaamheden aan zo'n guide moeten immers concurreren met de uitvoering van de primaire taak van deze trekkers.

**SAMENWERKING EN ONDERSTEUNING** Een project kan alleen succesvol zijn als er goede contacten met haar context zijn. Wij hadden te maken met de gebruikers, de interne kwaliteitsorganisatie (KO) en de (tool-) leveranciers.

Contacten met de eindgebruikers werden in eerste instantie bewust afgeschermd door de opdrachtgever, maar vanaf de tweede iteratie zijn deze intensief opgepakt.

De kwaliteitsorganisatie werd gevormd door een groepje medewerkers, dat weliswaar werd aangestuurd door dezelfde opdrachtgever, maar dat niet tot het softwareontwikkelproject behoorde. De taakafbakening tussen project en KO is aanvankelijk wat moeizaam: de KO heeft een kennisachterstand en er speelt een wij/zij gevoel. Later wordt een vorm van samenwerking gevonden, waarbij de KO al bij de kick off voor een nieuw artifact aanwezig is en daar op voorhand haar normen bekend maakt. Overigens is een werkwijze waarin alle

projectleden producten maken én betrokken zijn bij de review op de producten van teamgenoten eenvoudiger: de wij/zij-tegenstelling vervalt dan automatisch en een projectmanager heeft meer vrijheidsgraden wanneer medewerkers voor meer taken ingezet kunnen worden.

Er zijn goede contacten met de leveranciers: een Oracle consultant maakt parttime deel uit van het projectteam, Rational consultants worden periodiek gevraagd om inzet en advies. Ook via het Rational Developer Network en de Rational Customer Support vinden we ondersteuning. Cap Gemini Ernst & Young levert een flink aandeel in de projectbemensing, maar ook is er toegang tot de kennis binnen hun Accelerated Delivery Center en Advanced Test Center.

**SPANNINGSVELD** Onderdeel van de iteratie-assessments was ook een onderzoek naar de ervaring op diverse aspecten, zoals weergegeven in figuur 4. Er is gevraagd naar waardering en belang van de genoemde onderwerpen. De waardering drukt daarbij uit of het onderwerp adequaat is aangepakt in de afgelopen iteratie; via het belang kon je aangeven of er in de komende periode meer of beter aan gewerkt moet worden. De resultaten zijn compact weergegeven: als waardering

#### Assessment Iteratie E2: Waardering versus belang per onderwerp

- ⊗ Requirements Management
- ⊗ Project Management  
Voortgangsbewaking / status assessment / metrieken  
Planning / Schatting  
Business Modeling  
Configuratie Management  
Change Management
- ⊗ Testen  
Deployment
- ☺ Architectuur  
Use case modeling  
Design  
Implementation (Bouw / Programming)
- ☺ Database design
- ☺ Guidelines (Opstellen richtlijnen)
- ☺ (Ontwikkel)tools  
Reviews  
Team overleg
- ☺ Opleidingen voor de projectmedewerkers
- ☺ Project Web
- ⊗ Externe communicatie  
Aansluiting /synchronisatie tussen ontwerp en bouw  
(tussen documentatie en software)

FIGUUR 4. Onderdeel van de iteratie-assessments was een onderzoek naar de ervaring op diverse aspecten.

sterk achterblijft bij belang leidt dat tot ☹, als waardering groter is dan belang is dat weergegeven als ☺.

De doelstelling van de Elaboration-fase in RUP is het neerzetten van een architectuur en ontwikkelstraat (proces en tools) die stabiel zijn, zodat er in de navolgende Construction-fase meters gemaakt kunnen worden. Binnen die primaire doelstelling mogen we de ☺ voor Architectuur, Guidelines, (Ontwikkel)tools en Opleidingen als compliment beschouwen: we hebben de focus bij de juiste onderwerpen gelegd.

## Vroegtijdig testen levert een bijdrage aan een eenduidige en toetsbare formulering van de requirements

Toch blijft er een spanningsveld tussen de interne doelstelling voor een fase en de verwachting bij de klantorganisatie: die wacht op een zichtbaar resultaat in de vorm van werkende, productierijpe software. Hier zie je het belang van een “cultuuromslag” in de gebruikersorganisatie: men moet vertrouwd raken met het iteratieve concept en de daaruit voortvloeiende tussentijdse opleveringen van incomplete software. De standaard-aanpak van Cap Gemini Ernst & Young uit ons eerste

artikel ondersteunt daarbij: Awareness Building, Vision Setting en Readiness Assessment zorgen voor een realistisch ambitieniveau. Kortom, verwachtingsmanagement is ook hier een belangrijk aspect. Daarmee voorkom je ook de druk tot een voortijdige opschaling van de resources: tijdens de Elaboration fase moet je met een klein en ervaren team werken.

### CONCLUSIE

Onze slotconclusie kan zijn dat RUP houvast biedt en past in een Java-ontwikkelomgeving, dat aandacht voor het menselijke aspect (communicatie, verwachtingsmanagement) en begeleiding door mensen met praktijkervaring kritieke succesfactoren zijn. Welke uitdagingen liggen er voor dit project nog in de toekomst? Allereerst moet het project en daarmee RUP ongeschonden een in gang gezet fusietraject doorstaan. Uiteindelijk zetten we een state-of-the-art ontwikkelomgeving neer en moet onze aanpak doorgroeien van projectmethode naar methode op organisatieniveau.

*Drs. T.M. (Tom) Jacobs is managing consultant bij het competence cluster Process Management van Cap Gemini Ernst & Young. Hij ondersteunt organisaties bij het (her)inrichten van processen en met name bij het invoeren of aanscherpen van methoden voor systeemontwikkeling. Bij het besproken project heeft hij de rol van Process Engineer vervuld.*

---

Advertentie