



Rational Unified Process: aandachtspunten voor de auditor

Systeemontwikkelingsorganisaties introduceren met regelmaat nieuwe of andere softwareontwikkelmethoden. Vaak ligt hier een ontevredenheid over de resultaten van de tot dat moment gebruikte methode aan ten grondslag, maar ook de invoering van nieuwe software of systemen kan een introductie bewerkstelligen. De Rijksoverheid gebruikt sinds enige tijd ook *Rational Unified Process* (RUP) als methodiek bij de ontwikkeling van software.

GERARD VAN DEN BERG

In het voorjaar van 2009 heeft een aantal auditors samen met RUP-coaches in een drietal bijeenkomsten kennis genomen van de essentie van RUP. Op basis daarvan zijn de verschillen tussen RUP en (andere) gebruikte softwareontwikkelmethodieken geanalyseerd en is vastgesteld welke gevolgen dit heeft voor het uitvoeren van audits en welke aandachtspunten er zijn voor de auditor. Dit artikel doet verslag van de onderkende gevolgen van de toepassing van RUP voor de werkzaamheden van de auditor binnen systeemontwikkelingsorganisaties. Volledigheid is niet het doel, de keuzen zijn gebaseerd op kennis en ervaring van de deelnemers van de bijeenkomsten.

Hierna wordt eerst ingegaan op de vraag hoe RUP als onderwerp in verschillende type audits kan voorkomen. Dan volgt een toelichting op de theorie van RUP en de condities waaronder de methodiek toegepast kan worden binnen een project. Daarna komen de onderwerpen *requirements*, *planning*, *testen*, *supplementary specifications* en de betekenis van RUP voor organisatieverandering aan de orde. Als afsluiting is een waarschuwing voor de auditor opgenomen.

TYPE AUDIT

Voor de uitgangspunten en werkzaamheden van de auditor is het van

belang vast te stellen wat het doel van de audit is. Bij een projectaudit¹ en/of een productaudit is RUP een van de auditaspecten waar je aandacht aan moet geven. Dit artikel is bedoeld om een deel van die specifieke aandacht duidelijk te maken. Ook kan een audit bedoeld zijn om inzicht te geven in het functioneren van RUP en de bijdrage die het levert aan de kwaliteit van het eindproduct. In dat geval gaat het om de toepassing van RUP zelf. Daar wordt in dit artikel niet op in gegaan.

THEORIE RUP

RUP is een softwareontwikkelmethode die in 1998 is ontstaan uit een verzameling *best practices* van geslaagde projecten. Het doel is om incrementeel² en iteratief³ software te ontwikkelen. De achterliggende gedachte is dat softwaresystemen zo groot en complex geworden zijn, dat deze niet in een keer kunnen worden ontwikkeld. Gedurende het traject kunnen producten veranderen door gewijzigde wensen, eisen, technische ontwikkelingen en/of beperkingen. RUP knipt het uiteindelijk te ontwikkelen softwareproduct daarom op in verschillende kleine, maar werkende deelproducten en deze worden afzonderlijk van elkaar ontwikkeld en opgeleverd in iteraties. Een iteratie is een afgebakende, korte periode (meestal 4 tot 6 weken) waarin een team een samenhangende hoeveelheid ■



functionaliteit ontwerpt, realiseert en accepteert. Iedere iteratie wordt afgesloten met onder andere een acceptatietest door de gebruikersorganisatie. Door de klant nauw te betrekken bij de ontwikkeling en acceptatie van de deelproducten voorkom je dat het product niet overeenkomt met de wensen van de klant.

RUP bestaat uit een aantal fasen die weergegeven worden in figuur 1. In de *inception*-fase wordt het doel van het project geformuleerd en vastgelegd in het *Vision document*.⁴ De activiteiten in deze fase bestaan uit het beschrijven van de scope, wensen en eisen, het in kaart brengen van de belangrijkste risico's en maatregelen en het maken van een realistische kosteninschatting en een globale planning. Vastlegging hiervan vindt plaats in diverse werkdocumenten. In de *elaboration*-fase vinden de eerste iteraties plaats, gericht op een stabiele en werkende softwarearchitectuur. Ook zijn hier de meest kritische *requirements* gedetailleerd uitgewerkt in zogenaamde *Use*

Cases,⁵ waarover later meer. In deze fase moet ook duidelijk zijn geworden dat de ontwikkelomgeving adequaat is. Realisatie van de overige functionaliteiten en het samenstellen van handleidingen en trainingsmateriaal vinden iteratief plaats in de *construction*-fase. Het toetsen van het volledige softwareproduct aan de acceptatiecriteria, het trainen van gebruikers en beheerders en evalueren gebeurt in de *transition*-fase.

Waar de veel gebruikte watervalmethode achtereenvolgens de fasen ontwerp, realisatie, test en acceptatie eenmalig gedurende het hele project doorloopt, is dit bij RUP onderdeel van iedere iteratie. Binnen het project zullen er over het algemeen meerdere iteraties tegelijkertijd plaatsvinden.

Voor de auditor is het van belang om vast te stellen dat er binnen het project een proces aanwezig is dat de iteraties managet, rekening houdend met de doelstelling van het project.

DE KEUZE VOOR RUP

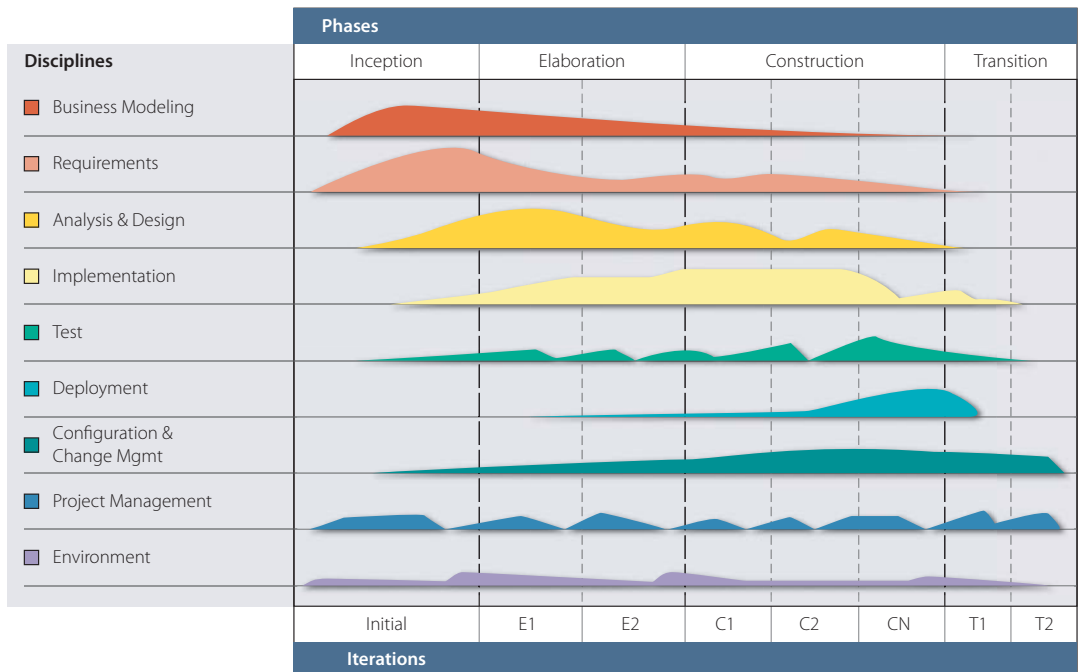
De eerste vraag die je als auditor moet stellen, is of de keuze van het project voor RUP als ontwikkelmethodiek gerechtvaardigd is.

RUP is een softwareontwikkelmethodiek en moet ook als zodanig worden ingezet. Dat betekent dat RUP niet geschikt is voor bijvoorbeeld pakketselectie of infrastructuurele projecten.

Binnen de projectaudit moet de auditor dan ook vaststellen in welke mate het project randvoorwaarden zoals basisvoorzieningen, infrastructuur et cetera meeneemt.

Daarnaast moet de auditor beoordelen of het project aansluit op de strategische doelstellingen van de organisatie als geheel.

Deze laatste punten gelden overigens voor alle projectaudits, maar voor een project met RUP is het belangrijk te beseffen dat deze onderwerpen niet tot de scope van RUP behoren.



Figuur 1: De vier hoofdfasen en negen disciplines van RUP

RUP is een *process framework*. Dit framework verlangt dat het project analyseert welke rollen, taken en werkproducten van RUP van toepassing zijn binnen het project en welke niet. Het achterwege blijven van deze analyse, en daarmee het volledig overnemen van het framework, leidt waarschijnlijk tot het mislukken van het toepassen van RUP, omdat er dan eenvoudigweg te veel onnodig werk wordt gedaan. Het succesvol afronden van het project loopt daarmee gevaar.

De auditor dient de opzet van het project op basis van het RUP framework te beoordelen.

RUP is gefundeerd op nauwe samenwerking binnen de hele systeemontwikkelingsorganisatie met een hoge participatie van gebruikers en *stakeholders* (belanghebbenden) binnen iedere iteratie. Onderlinge communicatie is daarbij essentieel, iets waar een gemeenschappelijk discussiedomein en bijeenkomsten aan bij kunnen dragen. Stakeholdermanagement is een manier om stakeholders, hun belangen, betrokkenheid en mandaat vast te leggen en te onderhouden. De auditdienst zou hier ook als stakeholder kunnen deelnemen om het project te wijzen op relevante wet- en regelgeving voor bijvoorbeeld informatiebeveiliging, procesbeheersing en bestuurlijke informatievoorziening. Indien de voorschriften voor projecten op orde zijn, is deze deelname eventueel niet nodig.

De auditor moet vaststellen dat alle belanghebbenden op de juiste wijze betrokken zijn bij het project.

De cultuur binnen de systeemontwikkelingsorganisatie is erg belangrijk om RUP succesvol toe te kunnen passen: geen wantrouwen, maar gedurende het gehele project nauw willen samenwerken. Ook indien er binnen de organisatie al projecten met RUP geslaagd zijn, is dit geen garantie voor succes. Een nieuw project betekent namelijk ook vaak nieuwe medewerkers die mogelijk niet op de hoogte zijn van specifieke eisen die

RUP stelt. Het inzetten van een RUP-coach en de eerder genoemde communicatiemiddelen zijn mogelijkheden om de risico's op dit vlak sterk te verminderen.

Stel als auditor vast dat het project structureel aandacht geeft aan communicatie over RUP, bij voorkeur door iemand die voldoende kennis en ervaring heeft met RUP (bijvoorbeeld een RUP-coach).

Inherent aan RUP is dat voortschrijdend inzicht kan leiden tot koerswijzigingen binnen het project. Dit vereist een flexibele en proactieve instelling van de hele systeemontwikkelingsorganisatie en de gebruikersorganisatie. Daarnaast vereist RUP een ervaren RUP-projectleider die volledig *commitment* van het management heeft om vast te kunnen houden aan de principes van RUP.

De auditor moet een goede inschatting kunnen maken of de projectleider en de cultuur van de organisatie past bij het principe van RUP.

REQUIREMENTS

De visie van het project, het *business object-model*⁶, de op te leveren functionaliteiten en niet-functionaliteiten⁷ worden requirements genoemd en zijn bedoeld om de gezamenlijke visie en uitgangspunten helder te krijgen. Voor de auditor is dit een belangrijke discipline: de requirements dienen volledig te zijn maar niet te gedetailleerd. De verdere detaillering vindt pas plaats in de iteraties. Van belang is vast te stellen of de juiste medewerkers betrokken zijn bij het beslissingstraject, denk hierbij aan de projecteigenaar, domeindeskundigen et cetera. De functionaliteiten liggen vast in Use Cases, de niet-functionaliteiten in de Supplementary Specifications, zie verderop in dit artikel. Het Use Case-model bevat een overzicht van alle Use Cases, hun samenhang, gewicht, classificatie en een beknopte beschrijving. Het Use Case-model kan door de auditor mede gebruikt worden om de juistheid en volledigheid van de planning te beoordelen.

Bij de requirements is het voor de auditor belangrijk om vast te stellen dat het juiste proces is doorlopen om de functionaliteiten te bepalen en welke prioriteiten er zijn gesteld.

De auditor moet ook beoordelen of het project niet te ambitieus is. Het is niet de eerste keer dat een project ten onder gaat aan zijn ambities. Daarnaast moet de auditor vaststellen in welke mate in de Use Cases ook de kwaliteitsaspecten beschikbaarheid, integriteit en vertrouwelijkheid zijn verwerkt.

PLANNING

Binnen RUP bestaan twee soorten planningen. De eerste (globale) planning vindt plaats in de inception-fase. Dit *Software Development Plan* bevat een beschrijving en globale planning van de fasen en mijlpalen voor de gehele duur van het totale project. Hierin zijn de doelen, kosten, *resourceing*, aantal iteraties, doorlooptijden en procedures aangegeven. Hier ligt ook de relatie met het eerdergenoemde Use Case-model, dat een overzicht bevat van alle Use Cases, hun samenhang, gewicht, classificatie en een beknopte beschrijving. De prioritering van de Use Cases vindt plaats via het MoSCoW-principe⁸, waarbij het project zich committeert aan (een deel van) de eerste drie prioriteiten, de *Must have*, *Should have* en *Could have*. De auditor dient er voor te waken dat het project dit principe gebruikt om bij tijdgebrek te snijden in de functionaliteiten, bijvoorbeeld door een flink aantal *Could-* en *Should have's* te laten vervallen.

Was de auditor bij de watervalmethode gewend aan een volledig dichtgetimmerde planning van kop tot staart, hier moet hij (als RUP goed wordt toegepast) het doen met een globaal inzicht. De totstandkoming en het voortdurend onderhouden van de planning en de betrokkenheid van de business bij de prioritering zijn belangrijke aandachtspunten voor de auditor.

De tweede planning is de planning behorend bij een iteratie, het *Iteration Plan*. Voorafgaand aan de te starten iteratie worden eerst de werkproducten, evaluatiecriteria, mijlpalen en de aan medewerkers gekoppelde ■



taken vastgesteld en gepland. Per iteratie wordt vastgesteld welke op te leveren functionaliteit essentieel en welke optioneel is. Dit is van belang om keuzen te kunnen maken, bijvoorbeeld als er sprake is van vertraging. Binnen de iteratie zijn de processen management, ontwerp, realisatie, acceptatie en kwaliteit aanwezig om de doelstellingen van de iteratie te halen. Deze processen worden parallel uitgevoerd (zie figuur 2). Na afloop van de iteratie vindt er een evaluatie plaats en een detailplanning voor de volgende iteratie.

Is er bij veel ontwikkelmethoden vaak sprake van één zeer gedetailleerde planning, bij RUP kom je deze dus niet tegen.

Het planproces is onderdeel van de continue processen en, zoals al eerder aangegeven, verdeeld in twee soorten plannings: globaal op projectniveau als een Software Development Plan en in detail per iteratie via het Iteration Plan. Voor de auditor is het van belang beide plannings onder de loep te nemen. Via het Software Development Plan krijgt de auditor snel inzicht en overzicht van de functionaliteiten die het project gaat opleveren evenals het belang er van. Dit plan mag geen speeltje zijn van een

planner of het projectmanagement, maar is een product van de prioriteiten die gemaakt zijn door het project én de business.

Met betrekking tot het Iteration Plan is het voor de auditor van belang om het kortcyclisch karakter ervan te beoordelen. Wordt de planning van de iteratie gehaald of worden functionaliteiten (regelmatig) doorgeschoven naar volgende iteraties. Het resultaat van iedere iteratie moet geprojecteerd worden op het Software Development Plan en gevolgen moeten inzichtelijk gemaakt worden.

TESTEN

De basis voor het testen, ligt beschreven in het *Test Plan*. Dit document beschrijft de testaanpak voor het hele project. Dit plan moet minimaal ingaan op testorganisatie, testscope, testhulpmiddelen, testomgeving en *testtooling*. Het daadwerkelijke testen vindt plaats in iedere iteratie. In de ontwerpworkflow (zie figuur 2) start de tester in overleg met de functioneel ontwerpers met het schrijven van het Testontwerp. In de realisatieworkflow worden de geautomatiseerde testgevallen gebouwd en uitgevoerd. In de acceptatieworkflow vinden de acceptatietest en Gebruikers Acceptatie Test (GAT) plaats. De acceptatietest controleert of de applicatie

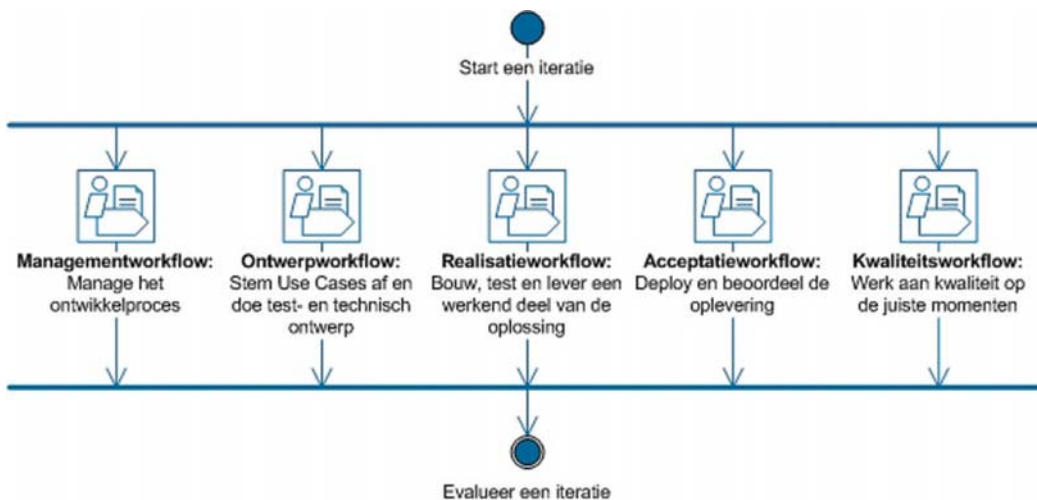
werkt conform de Use Case Specifications. De GAT controleert of de applicatie werkt zoals bedoeld door de business. De testmanager beoordeelt de bevindingen, die bij voorkeur binnen dezelfde iteratie opgepakt moeten worden. De activiteiten en de resultaten van de testen van iedere iteratie worden beschreven in het Testrapport.

De auditor moet alle werkzaamheden met betrekking tot het testen in kaart brengen en onderlinge verbanden vast stellen.

De (parallele) werkzaamheden en producten ten aanzien van testen zullen een weerslag moeten zijn van het Test Plan. Specifiek bij RUP is ook het anders omgaan met defects. Deze kunnen invloed hebben op volgende iteraties: een iteratie kan uitlopen of een onafgeronde functionaliteit schuift door naar een volgende iteratie.

De auditor moet inzicht krijgen in de mate waarin defects optreden, hoe er mee omgegaan wordt en wat de gevolgen er van zijn voor volgende iteraties.

Een ander punt is de regressietest. Het project moet ook controleren of nieuw opgeleverde functionaliteit een onbedoeld effect heeft gehad op de reeds eerder opgeleverde functionali-



Figuur 2: Iteratieworkflow

teiten. Omdat het gewenste eindproduct gedurende het project in omvang toeneemt, neemt het volume van de regressietest ook toe. De in de realisatieworkflow ontwikkelde geautomatiseerde testgevallen worden ook gebruikt bij de regressietest.

De auditor dient te verifiëren of het project een juiste invulling geeft aan de in volume toenemende regressietest.

Tot slot nog dit over testen: testen is vaak het kind van de rekening, dus als de planning in het gedrang komt, wordt er al snel op bezuinigd. RUP is hier geen oplossing voor en kan dit niet voorkomen. Dit is een cultuuraspect dat ligt binnen projectbesturing en -management.

SUPPLEMENTARY SPECIFICATIONS

Zoals al aangegeven, komen de gewenste functionaliteiten via het *Vision*-document terecht in de Use Cases. De niet-functionaliteiten komen in de Supplementary Specifications. Hierin zijn de onderwerpen beschreven zoals betrouwbaarheid, performance en eisen aan interfaces evenals de eisen die voortkomen uit wet- en regelgeving. De Supplementary Specifications kunnen voorkomen als een op zich zelfstaand document, maar ook gekoppeld zijn aan een specifieke Use Case indien het specificaties betreft die alleen daarop van toepassing zijn.

De auditor zal meer dan gemiddelde interesse moeten tonen in de Supplementary Specifications, omdat daarin de niet-functionaliteiten zoals betrouwbaarheidseisen zijn beschreven.

RUP EN ORGANISATIEVERANDERING

Zoals aangegeven is RUP een softwareontwikkelmethodiek. Er is aandacht voor training van eindgebruikers en overdracht van het product naar de beheeromgeving, maar altijd vanuit het oogpunt van de werkende software. Het managen van de orga-

nisatorische veranderingen valt buiten de scope van RUP en het project zal dit op een andere wijze moeten organiseren.

De auditor moet vaststellen dat het project verandermanagement heeft ingericht en dat belangrijke wijzigingen met gevolgen voor de softwareontwikkeling terugkomen bij de discipline *Business Modeling*.

SLOTWOORD

In dit artikel is een aantal aspecten van RUP beschreven en de gevolgen ervan voor de auditor. RUP is een incrementele en iteratieve wijze van softwareontwikkeling, die van de auditor vraagt daar rekening mee te houden. Het in een iteratie zoveel mogelijk parallel aan elkaar ontwerpen, realiseren, testen en accepteren, is een werkwijze waar de auditor mee moet leren omgaan. Als de auditor is opgegroeid met de watervalmethode zijn er veel en pijnlijke valkuilen die kunnen leiden tot *pain and suffering* [LARM01]. Niet begrijpen wat RUP inhoudt voor softwareontwikkeling en de organisatie eromheen kan leiden tot (in deze context) onjuiste of onbegrepen bevindingen, want een project met RUP is nu eenmaal geen vooraf tot in detail uitgewerkt en volledig gepland project. Om dit te begrijpen, maar ook om een beeld te krijgen van de ontwikkelorganisatie zou de auditor er hoe dan ook goed aan doen een (kritisch) oor te luisteren te leggen bij een RUP-coach.

Tot slot moet de auditor beseffen dat RUP alleen betrekking heeft op soft-

wareontwikkeling en het managen ervan, er is geen aandacht voor het managen van het project zelf.

Veel informatie over RUP is op het internet te vinden, onder andere op www.rupopmaat.nl, ook te verkrijgen in boekvorm. ■

Literatuurlijst

[LARM01] Craig Larman, Philippe Kruchten, Kurt Bittner, *How to fail with the Rational Unified Process: Seven steps to Pain and Suffering*, 2001.

Noten

- 1 Onder projectaudit wordt verstaan: het beoordelen of binnen een project voldoende waarborgen zijn om de doelstelling van het project te bereiken. Onder productaudit wordt verstaan: het beoordelen of de producten (gaan) voldoen aan de afgesproken kwaliteitskenmerken.
- 2 Met iedere iteratie wordt een volgend werkend deel van de oplossing gerealiseerd.
- 3 Aan het eind van iedere iteratie wordt de functionaliteit door stakeholders geaccepteerd, de feedback wordt verwerkt in volgende iteraties.
- 4 De Vision beschrijft het gezamenlijk perspectief van opdrachtgever en opdrachtnemer met betrekking tot het project. Dit document is dan ook normatief: de eisen aan en grenzen van het systeem, de context waarin de opdrachtgever het systeem zal gebruiken en de belanghebbenden worden vastgelegd.
- 5 Een Use Case beschrijft vanuit het gebruikersperspectief 'wie' met het betreffende systeem 'wat' kan doen.
- 6 Het business object-model legt de relatie vast tussen de meest relevante objecten en entiteiten die een rol spelen in de businessprocessen.
- 7 Van belang voor de auditor zijn met name de bedrijfsvoeringsfunctionaliteiten gericht op besturing, beheersing en verantwoording van de primaire processen, veelal op het gebied van bestuurlijke informatievoorziening, informatiebeveiliging en productiebesturing.
- 8 Must have, Should have, Could have, Want to have but won't have this time.



G. (Gerard) van den Berg RE is Senior Auditor bij de Rijksauditedienst en onder meer betrokken bij audits op (grote ICT-)projecten.