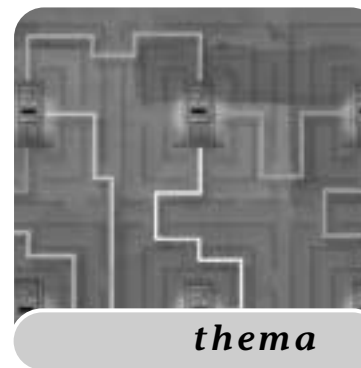


In twee opeenvolgende artikelen beschrijven Tom Jacobs en Alexander Hermanns ervaringen bij het invoeren van RUP (Rational Unified Process) als ontwikkelmethodiek bij LASER, een agentschap van het Ministerie van Landbouw, Natuurbeheer en Visserij. In dit eerste deel leggen de auteurs de nadruk op “het invoerproces”, in het afsluitende artikel zullen zij ingaan op de “lessons learned”.



Invoering van RUP bij agentschap van landbouwministerie (1)

Generiek systeem transformeerde organisatie LASER

LASER is belast met het uitvoeren van (financiële) regelingen voor de overheid. Denk daarbij aan uitbetalingen bij wateroverlast, subsidies op de Europese boterberg en zelfs aan de vuurwerkramp in Enschede. In totaal gaat het om een 150-tal verschillende regelingen, vanuit de historie veelal ondersteund met hun eigen, specifieke applicatie. Tijdens een architectuurtraject (medio 2000) is vastgesteld dat het achterliggende proces grotendeels gelijk is: aanvragen, beoordelen, berekenen, uitbetalen.

Reden om ook bij de ondersteuning door geautomatiseerde systemen te opteren voor een eenduidige aanpak. Via een generiek systeem zouden nieuwe regelingen binnen twee weken ingevoerd moeten kunnen worden. Bovendien zou door zo'n generieke applicatie de uitwisselbaarheid en inleertijd van de medewerkers verbeterd worden. Kortom: een generiek systeem past uitstekend in de transformatie naar de organisatie die voorzien is.

PROJECTSTART Begin 2002 startten de kwartiermakers met het ICT-project. LASER had zich voor systeemontwikkeling tot dan toe beperkt tot “Functionele Specificaties”, de bouw was altijd uitbesteed aan een centrale afdeling binnen het Ministerie of aan externen. Daarbij gebruikte men een lineaire methode: LIA, de Landbouw Informatica Aanpak.

In het transformatieplan van LASER is voorzien dat er een eigen systeemontwikkelafdeling zal ontstaan. Vooruitlopend daarop is door het project gezocht naar

een ontwikkelaanpak die goed past bij zelf ontwikkelen. In een korte selectie, met onder andere LIA en RUP is RUP verkozen.

De belangrijkste argumenten zijn:

- Ondersteunt een iteratieve aanpak;
- Ondersteunt systeemontwikkeling *in de volle breedte*: van Requirements Management tot operationaliseren (Deployment);
- Ondersteunt het ontwikkelen van componenten;
- Bekend in de markt (van belang bij het werken met externen).
- Het te bouwen systeem zal een web-applicatie worden; de programmeertaal is Java.

Na de keuze voor RUP is door Cap Gemini Ernst & Young en Rational Software een gezamenlijke offerte opgesteld, waarbij Cap de ondersteuning van het implementatieproces voor haar rekening neemt en Rational de toolspecifieke ondersteuning.

DE EERSTE ITERATIE Het invoeringsplan omvatte de volgende stappen:

- Opstellen “Development Case”;
- Omvormen van projectplan naar “Software Development Plan” conform RUP;
- Inrichten tools;
- Opleiding;
- Evaluatie RUP en toolgebruik.

Een essentieel product in het vormgeven van een RUP-project is de Development Case. Daarin wordt bepaald welke RUP-producten (*artifacts*) vervaardigd zullen worden. Dit is altijd een selectie uit het totale RUP-assortiment. Het Software Development Plan koppelt vervolgens resources en (iteratief) tijdsplan aan deze artifacts.

Een bijzonderheid bij LASER was dat het projectteam al samengesteld was vóór de keuze om RUP te gebruiken. Zwaartepunt van de expertise binnen dit team lag op "Software Development". Als we daarbij optellen de beschikbaarheid van een uitgebreide architectuurdefinitie en het ontbreken van echte gebruikers als gespreks-

dichtgetimmerd, waar de werkzaamheden overgedragen moeten worden van de ene persoon naar de andere. RUP kent immers vele "rollen" en het is niet op voorhand duidelijk wat de meest gewenste toedeling van rollen aan personen is. In het volgende artikel, dat zal handelen over "lessons learned", beschrijven we hoe in de loop van de tijd het proces is aangescherpt en verantwoordelijkheden duidelijker zijn toegewezen: in plaats van een teamverantwoordelijkheid voor producten werden eerstverantwoordelijke personen aangewezen.

ZELFSTANDIG PROJECT Voor het invoeren van een nieuwe werkwijze wordt vaak een zelfstandig project ingericht. Dat zou voor LASER te zwaar zijn: er was nog geen ontwikkelorganisatie, maar slechts een eerste systeemontwikkelpject, LARIS geheeten. De organisatie wacht op resultaten van LARIS en liever niet op al te veel initiërende schijnbewegingen: er moeten al werkplekken ingericht worden, ontwikkelsoftware geselecteerd en geïnstalleerd worden et cetera.

Kortom, de invoering van RUP vindt in eerste instantie plaats binnen het LARIS-project. Een plan om RUP organisatiebreed in te voeren kan op zijn vroegst gemaakt worden na de evaluatie van de eerste ervaringen met RUP. Die evaluatie is gepland aan het eind van de eerste iteratie, na drie maanden. Toch hoeft het invoeren van RUP vanuit een softwareontwikkelpject geen bezwaar te zijn. Het omgekeerde is wel ongewenst: je moet geen methode invoeren wanneer daarbij geen software-ontwikkelpject als pathfinder of pilot beschikbaar is. Dan verwordt het gauw tot het uitvaardigen van richtlijnen vanuit een ivoren toren; zonder binding met de praktijk zullen die moeilijk geaccepteerd worden. De standaardaanpak van de adviesgroep IT Process Consulting van Cap Gemini Ernst & Young bij invoering van een nieuwe methode zien we in afbeelding 1.

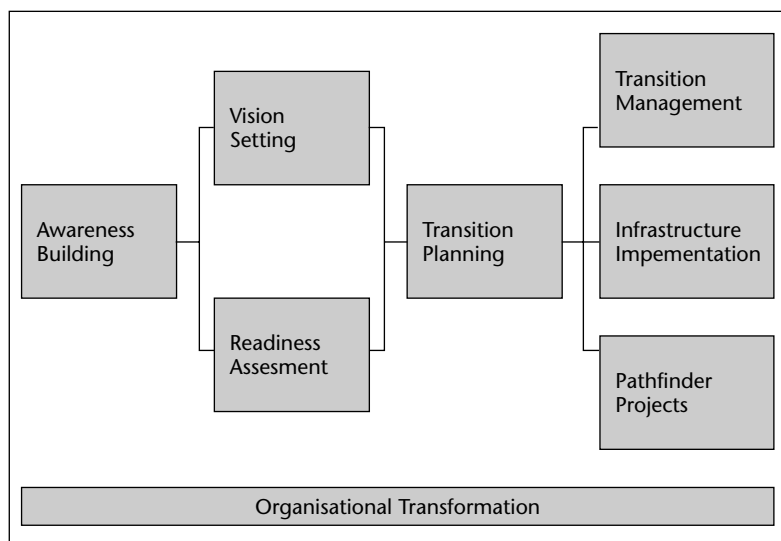
Als we de aanpak binnen LASER hierop projecteren zien we dat met name de eerste stappen (Awareness building, visie en ambitie neerzetten) nu onderbelicht blijven of althans: beperkt blijven tot de scope van het eerste project. Ook de transformatie van de organisatie valt buiten de scope, pas in een later stadium zijn ook (awareness-)presentaties en trainingen voor de organisatie opgestart.

Een methode moet je stapje voor stapje invoeren. Alles tegelijk zou een te grote belasting voor de eerste iteratie van LARIS worden en bovendien niet passen in de timebox van drie maanden. In de eerste iteratie hebben we onze aandacht gericht op het "RUPpificeren" van de primaire disciplines Requirements, Analysis & Design en Implementation (= Bouw) en de ondersteu-

Je moet geen methode invoeren wanneer daarbij geen software-ontwikkelpject als pathfinder of pilot beschikbaar is

partners om "generieke functionaliteit" mee te bediscussieren, dan ligt de pragmatische keuze voor de hand om de Inception fase over te slaan en direct met een iteratie in de Elaboration fase te starten. Wel is gesteld dat enkele essentiële artifacts uit de Inception fase alsnog vervaardigd moesten worden. Denk daarbij aan de Business Case, het Vision document (met requirements en gewenste "product features") en de Glossary (begrip-lijst).

In de Development Case is voor de eerste iteratie het zwaartepunt gelegd bij de **producten** (artifacts) en is niet expliciet ingegaan op de activiteiten die daarvoor nodig waren. De standaardprocedure was: als je ondersteuning nodig hebt bij het maken van zo'n artifact gebruik je de toelichting in RUP. Ook is niet vooraf



AFBEELDING 1. Framework voor invoering van de methode

nende discipline Environment. Dat betekent niet dat er geen activiteiten werden uitgevoerd in disciplines als Test of Configuration Management, maar daar werd even vertrouwd op de vakbekwaamheid van de project-medewerkers.

RESULTATEN EVALUATIE De evaluatie na de eerste iteratie was met name gericht op de toegevoegde waarde van RUP. RUP zelf werd beoordeeld en gekeken moest worden naar de criteria waarvoor RUP oorspronkelijk was geselecteerd (compleetheid, ondersteuning CBD, ondersteuning iteratief werken). Conclusie van de evaluatie was dat we met RUP door moesten gaan. Vervolgens is ook na elke volgende iteratie een evaluatie gehouden zoals past in de activiteit "Assess Iteration" uit de RUP Project Management discipline.

In de tabel hiernaast tonen we de resultaten van het deel van de evaluatie dat speciaal op RUP gericht was. De resultaten van de twee opvolgende iteraties E1 en E2 (volgnummers binnen de Elaboration fase) staan naast elkaar, zodat er ook een eerste trend in de tijd onderkend kan worden. De respondenten uit beide enquêtes vallen niet samen: het projectteam heeft wisselingen gekend. Er is gevraagd naar een waardering tussen 1 (Onbelangrijk, Geheel mee oneens) en 5 (Uitstekend, Zeer belangrijk, Geheel mee eens), met 3 als neutraal oordeel. Let op: de vragen zijn zo gesteld dat een lage waardering soms positief, soms negatief voor RUP is.

TRENDS De uitgesproken hoge of lage waarderingen in de tabel zijn vet afgedrukt. Dat zijn er bij E2 minder dan bij E1, waarschijnlijk door de grotere groep respondenten. De verschillen tussen beide beoordelingen zijn niet groot. Opmerkelijke resultaten waren:

- De mogelijkheid in RUP om te "customizen" naar de projectsituatie wordt herkend.
- Men verwacht eerder een positieve invloed op de kwaliteit dan op de snelheid.
- Men ziet RUP niet als noodzakelijk vanwege de nieuwe technieken (OO, CBD, UML). Reden hiervoor is dat diverse projectleden al vertrouwd waren met die technieken voordat ze RUP leerden kennen.
- Aardig is ook de spreiding in de antwoorden (niet zichtbaar in de getoonde tabel). De stelling "het projectteam is belangrijker dan het proces" blijkt een controversiële vraag met uitgesproken voor- en tegenstanders. Het gemiddelde antwoord schommelt echter rond 3 (d.w.z. neutraal).
- "Ik heb inzicht in de RUP-artifacts die ik moet produceren" is in de tweede iteratie spectaculair hoger gewaardeerd dan in de eerste. Daarmee samenhangend is ook een toegenomen gebruik te zien van de RUP-website om informatie te halen.

Vragen	E1	E2
RUP biedt ondersteuning bij iteratief werken	3,54	3,50
Ik heb inzicht in de RUP-artifacts die ik moet produceren	3,08	4,07
Ik heb de RUP-site gebruikt om aanvullende informatie te halen.		
Over mijn rol binnen RUP	2,64	3,20
Over checkpoints /kwaliteitseisen aan mijn producten/artifacts	2,50	2,87
Over guidelines bij het werken	2,25	2,60
Over de activiteiten waarin de artifacts vervaardigd worden	2,80	3,13
Over andere zaken		2,86
RUP geeft aan hoe het aandachtspunt verschuift bij iteraties in verschillende fasen	4,00	3,36
RUP geeft duidelijk aan dat een project meer is dan software bouwen	4,23	4,00
RUP is een keurslijf en beperkt de creativiteit	2,00	2,21
RUP is goed toepasbaar bij wijzigende requirements	3,13	3,55
RUP onderkent veel te veel artifacts om snel en iteratief werken mogelijk te maken	2,46	2,80
RUP biedt mogelijkheden om proces en artifacts aan de projectsituatie aan te passen	4,00	3,86
Het projectteam is belangrijker dan het proces	2,92	3,13
Met RUP zal het eindresultaat uiteindelijk sneller worden bereikt	3,17	2,57
Met RUP zal het eindresultaat uiteindelijk een hogere kwaliteit hebben	3,92	3,57
RUP dekt het systeemontwikkelp proces in de volle breedte	3,75	3,73
RUP dekt ook het proces van herstructureren/aanpassen van de organisatie	2,33	2,46
RUP is soms overhead, maar wel nodig	3,38	3,40
Gezien de nieuwheid van de technieken (OO, CBD, UML) is ondersteuning met RUP nodig	2,77	2,40
RUP zorgt dat de producten van verschillende project-medewerkers op elkaar aansluiten	2,75	3,00
Ik verwacht dat ik de komende jaren RUP steeds vaker zal tegenkomen in de markt	4,15	4,00
<u>Conclusie</u>		
Per saldo denk ik dat RUP een goede keuze is voor LARIS	3,92	3,67

- Opmerkelijk is de daling bij de vraag: "Met RUP zal het eindresultaat uiteindelijk sneller worden bereikt". Dit sluit overigens aan bij de breed gevoelde zorg ten aanzien van de voortgang die bij een ander onderdeel van de evaluatie naar boven kwam. Of de toerekening aan RUP van deze zorg terecht is, is discutabel, maar RUP staat voor het proces en wordt daarmee verantwoordelijk gehouden voor de resultaten.
- Tot slot blijft de conclusie van de respondenten dat RUP een goede keuze is voor LARIS, al is ook hier de uitspraak minder expliciet dan tijdens de eerste evaluatie.

HOGERE MATURITY Dat we met de invoering van RUP lerend op weg zijn gegaan, mag onder andere blijken uit de compleetheid van de evaluaties:

- in de eerste evaluatie werd uitsluitend gekeken naar

Wat is RUP?

RUP staat voor Rational Unified Process. Het is een methode voor systeem-ontwikkeling, gebaseerd op een zestal "best practices":

- Develop Iteratively
- Manage Requirements
- Use Components Architectures
- Model Visually (UML)
- Continuously Verify Quality
- Manage Change

In afbeelding 2 zien we een overzicht van RUP. RUP kent "disciplines" of aandachtsgebieden die elk een eigen workflow of proces hebben en eigen producten. Naast de zes primaire disciplines, die het gebied bestrijken van bedrijfsmodellering (Business Modeling) tot aan het operationaliseren van software (Deployment) zijn er drie ondersteunende disciplines: Configuratie- en Change Management, Project Management en Environment. Environment staat hierbij voor het inrichten van de ontwikkelomgeving en omvat het installeren/configureren van pakketten en ontwikkel-tools tot het schrijven van Standaards en Richtlijnen.

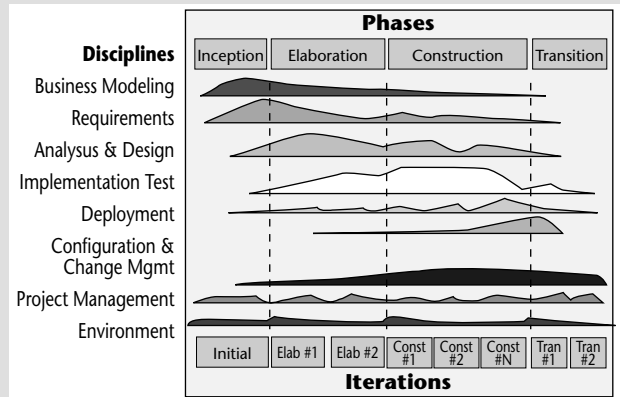
Wat is typerend voor RUP?

RUP wijkt af van andere methoden op de volgende punten:

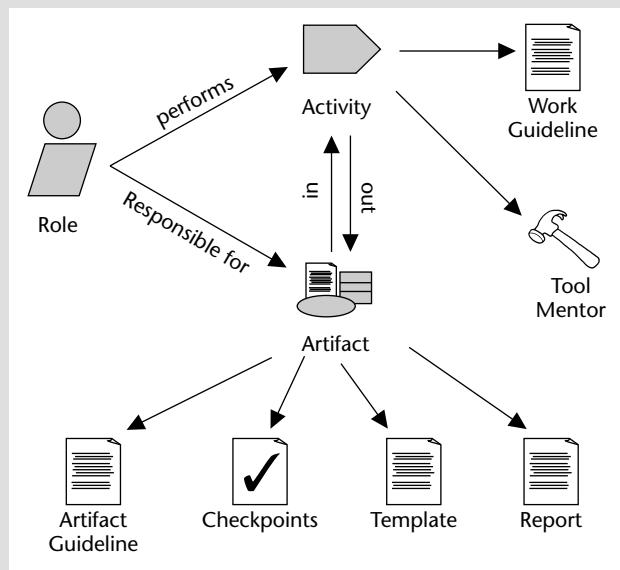
- Het kent fasen én iteraties. Hierdoor zien we de focus van iteratie tot iteratie verschuiven:
 - van: Bepalen scope en haalbaarheid (tijdens Inception),
 - via: Vaststellen Architectuur en Ontwikkelstraat (tijdens Elaboration),
 - naar: Produceren conform plan in een bekende omgeving (tijdens Construction),
 - eindigend met: Uitrollen in de gebruikersomgeving (Transition).
- Tijdens elke iteratie wordt werkende software gemaakt. Dus ook al vroeg in de Elaboration-fase. Daarmee wordt de mogelijkheid tot vroege feedback geboden aan alle stakeholders. De focus verschuift daarbij van "proof of concept" naar "puntjes op de i".
- De grootste risico's worden het eerst aangepakt. Onzekerheden rond architectuur, aansluitingen aan externe systemen etc. horen al tijdens de Elaboration-fase opgepakt te worden, uitmondend in een "proof of concept".
- RUP baseert zich in zijn richtlijnen op UML (Unified Modeling Language), een steeds breder geaccepteerde modelleringstaal.

RUP wordt ook wel getypeerd als het Unified Process met Rational tools, het wordt geleverd als een website en vormt met zijn 60 MB een zeer omvangrijke kennisbank.

De basisstructuur van RUP kent **Activiteiten** (als laagste niveau binnen de workflows), uitgevoerd door medewerkers met zekere **Rol** en leidend tot **Artifacts** (typische RUP-term voor Producten). Daarnaast biedt RUP praktische ondersteuning via Guidelines, Toolmentoren (voor de Rational Suite!, maar desgewenst uitbreidbaar), Checklists, Templates etc. (zie ook afbeelding 3).



AFBEELDING 2

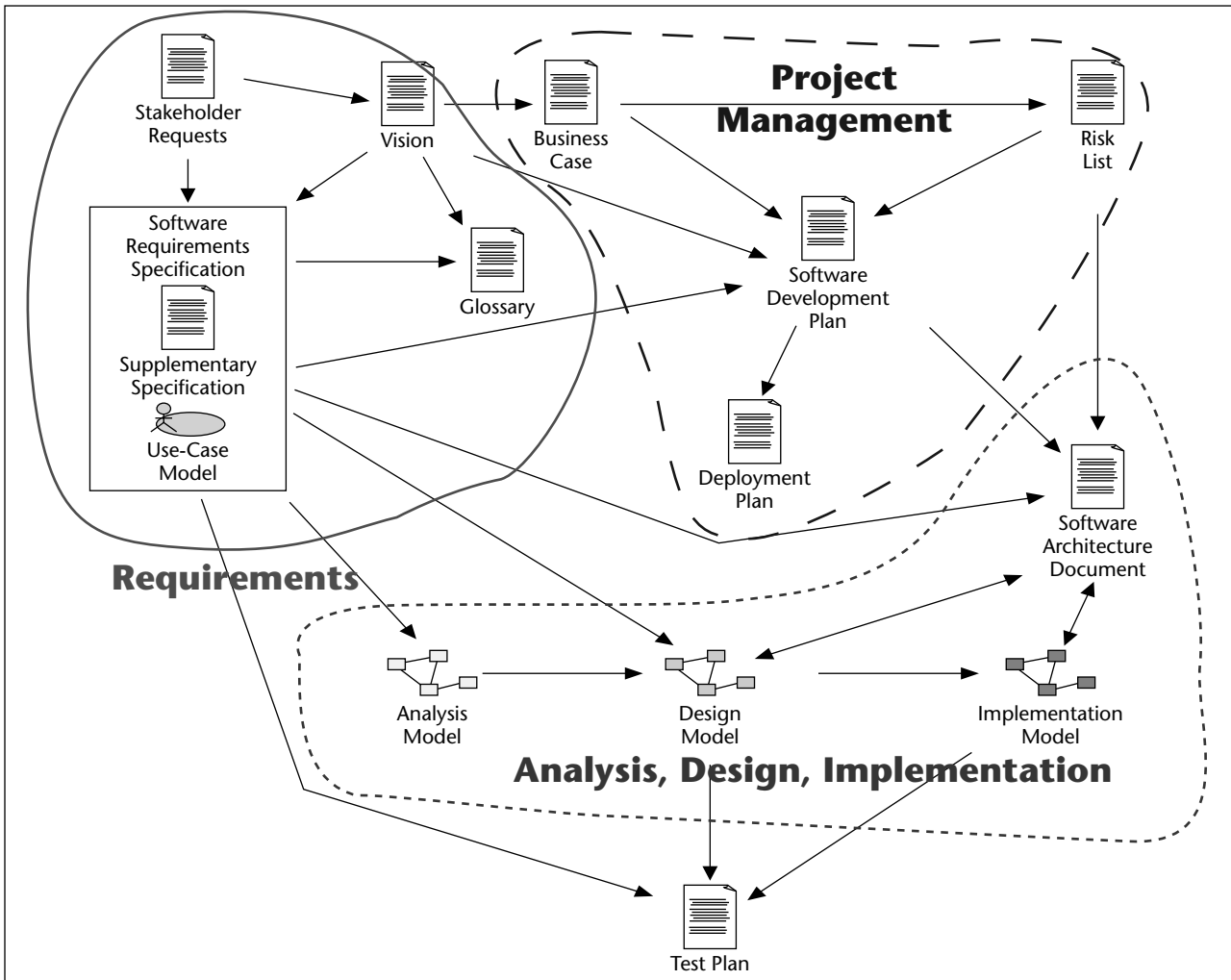


AFBEELDING 3

- de toereikendheid van RUP en de lessons learned;
- in de tweede iteratie werd dit uitgebreid met een "meetweek", waarin de kwaliteitsorganisatie ook een oordeel gaf over de bereikte resultaten via een quick scan op de artifacts;
- voor de volgende iteraties zou dit (conform de voorschriften in de RUP-activiteit Assess Iteration) worden uitgebreid met het vergelijken van budgets/plannen

versus realisatie op diverse andere aspecten zoals tijd, mankracht en geld.

Zo zie je dat we in rijpheid groeien van "hebben we het gevoel dat we goed op weg zijn" via "leveren we ook kwalitatief voldoende resultaten" naar "verloopt het bovendien op een economisch verantwoorde wijze".



AFBEELDING 4. De hoofdstructuur van artifacts in RUP

DE KERNSPRODUCTEN In afbeelding 4 zien we de hoofdstructuur van artifacts in RUP. Wel ontbreken hier diverse ondersteunende artifacts zoals de verschillende Guides en de Development Infrastructure (de ontwikkelstraat). Tijdens de invoering zijn die essentieel!

De getoonde producten zijn bij LARIS vervaardigd met de volgende kanttekeningen:

- Stakeholder Requests als *zelfstandig* artifact zijn aangehouden totdat Change Management goed ingevuld zou worden.
- De Business Case is in een conceptstatus blijven hangen. Intern voor het project is deze niet van belang, wel voor de economische motivatie naar partijen buiten het project.
- Het Analysis Model groeit door naar het Design Model en is dus niet een te onderhouden artifact.
- Het Implementation Model was geen zelfstandig *model*. De instructies in onze "Design Guide" gecombineerd met de vastlegging in het Software Architecture Document beschrijven de transformatie van Design Model tot software componenten. Een zelfstandig Implementation Model kan in complexe (gedistribueerde) omgevingen wel zinvol zijn.

CONCLUSIE Onze aanpak om RUP en een bijbehorende ontwikkelstraat vanuit een systeemontwikkelpject in te voeren is tot op heden succesvol. Wel is het noodzakelijk om daarbij realistische verwachtingen te scheppen: de invoering gaat geleidelijk en loopt mee met de iteraties; het project zal langere tijd met leer- en inrichtingsperikelen geconfronteerd worden en dus in een Elaboration fase verkeren. In ons volgend artikel zullen we de "lessons learned" per discipline toelichten.

Drs. T.M. (Tom) Jacobs is managing consultant bij de adviesgroep IT Process Consulting van Cap Gemini Ernst & Young. Hij ondersteunt organisaties bij Software Process Improvement en met name bij het invoeren of aanscherpen van methoden voor systeemontwikkeling. Bij LASER heeft hij de rol van Process Engineer vervuld.

A.K.E.O.M. (Alexander) Hermanns is gecertificeerd projectmanager bij LASER. Vanaf iteratie E2 is hij als projectmanager verantwoordelijk voor het LARIS-project.