

Modernisering van uitzendorganisatie

Introductie Agile-werkwijze bij Randstad

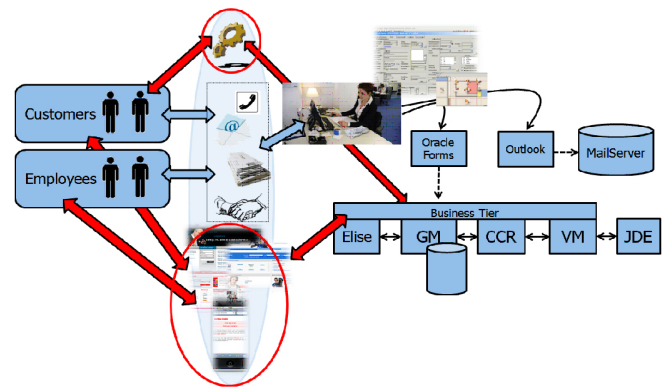
Het verhaal van de modernisering van Randstad is een verhaal met vele facetten. En een verhaal dat een soort van begin kent maar zeker geen einde. Modernisering is een voortdurende evolutie, met fasen en mijlpalen wellicht, maar niet met een concreet eindpunt. En modernisering heeft betrekking op tools en technologie en open industrie-standaarden, maar minstens zoveel op mensen, processen, communicatie en samenwerking. Tijdens OBUG Benelux Connect 2011 presenteerden Patrick Stevens en Lucas Jellema het verhaal van de invoering van een op services gebaseerde architectuur, de introductie van een agile werkwijze en de toepassing van Oracle Fusion Middleware om de business strategie realiseerbaar te maken. Dit artikel bevat een weergave van die presentatie.

De Randstad Groep Nederland is de grootste speler op de nationale uitzendmarkt, met ondermeer de labels Yacht, TempoTeam en Randstad. Met meer dan 750 vestigingen, vele duizenden intercedenten en zo'n 100.000 uitzendkrachten (employees in Randstad terminologie) ingezet bij duizenden klantorganisaties, zijn de uitdagingen rondom contacten, processen en informatie-uitwisseling interessant te noemen. Dat geldt al helemaal voor de wekelijkse piek op maandagochtend als het merendeel van de employees het urenbriefje van de afgelopen week invult en daarmee de facturatie en verloning in gang worden gezet.

De bedrijfsstrategie kent veel componenten die direct met IT samenhangen en daar eisen aan stellen - voor de handliggend gelet op de cruciale rol van geautomatiseerde processen en informatieuitwisseling. Sleutelonderdeel van de strategie is 'het vermogen om snel op veranderingen in te spelen'. Dat is in het Engels vertaald: agility. Met betrekking tot IT wordt daarop aansluitend gesteld: verhoog de betrokkenheid van de business bij IT. Andere elementen uit de strategie met directe consequenties op IT-gebied zijn ondermeer de gewenste verhoging van customer intimacy en de klanttevredenheid, meer grip op en sneller inzicht in het verloop van bedrijfsprocessen,

verhogen van efficiency door ondermeer self service voorzieningen aan te bieden aan employees en klanten.

Een belangrijk omslag die volgt uit deze strategie is geschetst in figuur 1.



Figuur 1.

De interactie tussen zowel klanten als employees met Randstad, Yacht en TempoTeam verliep voor 2009 voor het overgrote deel via de klassieke kanalen – de blauwe pijlen in deze figuur – telefoon, post, email en vooral mondelinge interactie in de vestigingen of bij de klant. Voor vrijwel iedere interactie was menselijke betrokkenheid van de kant van de Randstad onderdelen noodzakelijk – in de meeste gevallen via de Mondriaan Oracle Forms applicatie die alle vestigingsmedewerkers toegang geeft tot de enterprise database. Kantooruren bepaalden ook in veel gevallen de mogelijkheden voor klant of employee om contact te hebben.

Een belangrijke wens van de business was de ontsluiting van aanvullende kanalen: webapplicaties en webservices. De rode pijlen geven aan hoe via die nieuwe kanalen de klanten en employees zonder tussenkomst van intercedenten eigen gegevens kunnen inzien en onderhouden, zowel handmatig via de selfservice webapplicatie als programmatisch via de B2B webservice interface (de laatste alleen voor klanten). Uit de

afbeelding blijkt ook dat de realisatie van deze wensen grote gevolgen heeft – zowel voor de technische architectuur van het IT landschap als ook voor de processen in de organisatie: data die voorheen alleen via het filter van de intercedent bij klanten en employees belandde kan nu direct worden ingezien. Dat betekent dat intercedenten te maken krijgen met beter geïnformeerde en meer "empowerde" contacten en bovendien dat hogere eisen worden gesteld aan de correctheid en volledigheid van de vastgelegde gegevens.

Naast de strategische business wensen is er binnen de IT-teams behoefte aan grotere invloed op het plannen en uitvoeren van het (team)werk, een soepele(re), meer evolutionaire introductie van nieuwe technologieën en het benutten van de bestaande mix van (Fusion) Middleware en klassieke Oracle Forms en PL/SQL kennis en vaardigheden.

Proces

De modernisering van Randstad's IT en de realisatie van de genoemde strategische wensen rusten op twee pijlers: proces en architectuur. De introductie van een agile totstandkomingsproces op het fundament van een op ont koppeling gerichte service georiënteerde architectuur zijn de twee kern-elementen in wat zo langzamerhand een succesverhaal mag worden genoemd.

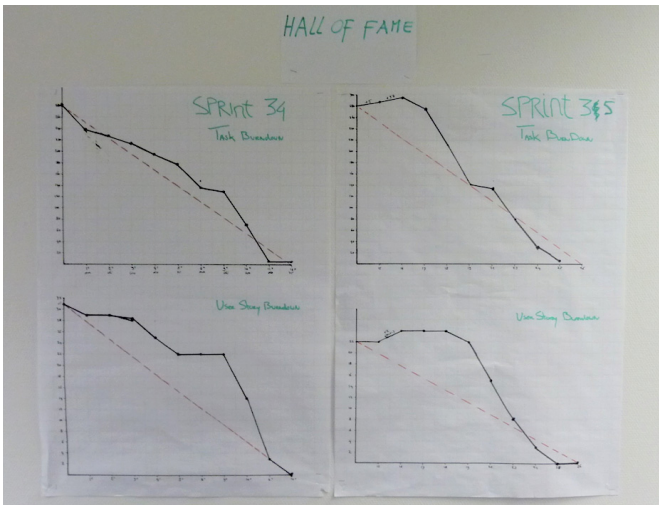
Software wordt binnen Randstad ontwikkeld in een agile proces volgens de Scrum-methode – uitgevoerd binnen meerdere, parallel opererende multi-disciplinaire teams. Teams hebben een omvang tussen de 6-10 personen. Alle disciplines

– architectuur, functioneel ontwerp, testen, web design, programmeren – zijn in het team vertegenwoordigd. Daarnaast staat het team in nauw overleg met de infrastructuur teams die de acceptatie- en productieomgevingen beheren en verantwoordelijk zijn voor uitrol. Er is geen sprake van projectleiders – de teams zijn zelfsturend. Scrum-masters faciliteren het proces – op heel praktische wijze voor wat betreft werkplekken en hulpmiddelen maar ook door het organiseren van planning-, refinement- en retrospective-bijeenkomsten, het bijhouden van de burndown-charts (zie afbeelding) en het met de product owner onderhouden van de product backlog.

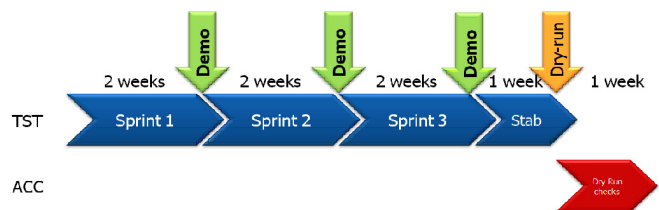
De sprints zijn kort: slechts twee weken hebben de teams om de meest belangrijke user stories – zoals gedefinieerd door de product owner, de vertegenwoordiger van de business, te implementeren tot een werkende demonstratie van software die in principe productierijp is. Dit vereist een intensieve samenwerking tussen de teams en de business, een flinke betrokkenheid van het team bij de business en de concrete waarde van de user stories en – misschien nog wel het meest vernieuwend – een voortdurende betrokkenheid van de business bij wat de teams doen en een continue beschikbaarheid voor toelichting, bijsturing en beoordeling. De wederzijdse betrokkenheid van business en IT en het daaruit voortvloeiende focus op business value binnen de teams is een belangrijke meerwaarde van het agile proces.

Overigens: hoewel er na iedere sprint release-bare software wordt opgeleverd door de teams, wordt er in de praktijk met grotere tussenpozen daadwerkelijk uitgerold naar productie. Iedere sprint wordt afgesloten met een demonstratie aan de product owner en andere betrokkenen en als enkele sprints zijn afgerond en een samenhangende set user stories is afgerond worden in een stabilisatiesprint de producten afgerond waarna een Dry-Run deployment en daaropvolgende acceptatietesten de daadwerkelijke productierijpheid moeten vaststellen.

Het succes van deze agile aanpak binnen i-bridge is mede te danken aan het grote vertrouwen dat er inmiddels wederzijds tussen IT en business bestaat. Dit zorgt er onder andere voor dat ook technische user stories – stories die geen directe business



Figuur 2: enkele burndown charts die laten zien hoe gedurende de sprints - twee weken, afgezet langs de horizontale as - user stories - uitgedrukt in story-points langs de verticale as - werden opgeleverd. In alle vier gevallen was na twee weken het volledige pakket storypoints waaraan het team zich vooraf had gecommiteerd ook daadwerkelijk als geteste, gedemonstreerde software opgeleverd.



Figuur 3: het release-proces- waarin de producten van meerdere sprints worden gecombineerd tot een productie-uitrol.

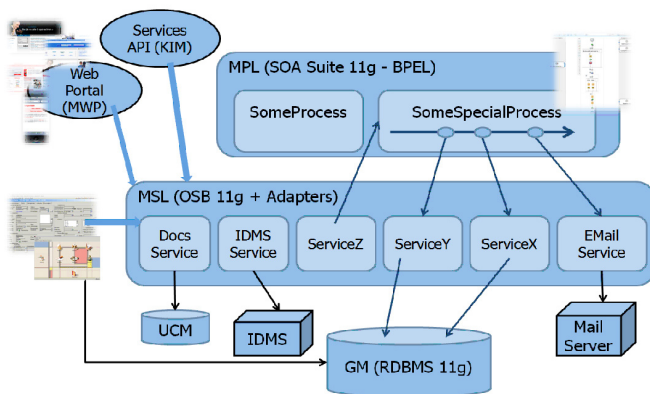
value opleveren, zoals refactoring, software upgrades en ontwikkelen van unit tests – niet alleen op de product backlog komen maar ook daadwerkelijk in sprints worden opgepakt. Dit vertrouwen schept ook de ruimte om naast de functionele user stories te werken aan over-all architectuur en non-functionele aspecten van het landschap. De aanvankelijke angst dat een agile aanpak en een consistente architectuur moeilijk te verenigen zijn blijkt bij i-bridge in de praktijk ongegrond.

Architectuur

Al in een vroeg stadium van het moderniseringstraject werd het belang van een solide architectuur onderkend. In lijn met industrie-trends en de voorziene functionele trajecten werd door de architecten van i-bridge een service architectuur ontworpen. Uitgangspunt in deze architectuur is dat alle kernsystemen – databases, document systemen, mail server en overige enterprise resources – uitsluitend via een service laag benaderbaar zijn voor andere systemen. Door deze ont koppeling via deze Mondriaan Service Layer (MSL) kunnen de interne systemen ondermeer geoptimaliseerd en heringericht worden zonder dat de afnemende systemen daardoor geraakt worden. Een voorbeeld daarvan is de verplaatsing van documenten uit de BLOB kolommen in de enterprise database naar UCM – de Oracle Universal Content Manager. De services worden in eerste instantie (her)gebruikt door zowel de WebServices API (KIM) als de WebApplicaties (MWP). Meer hergebruik ligt in het verschiep gezien de verdere toekomstplannen.

De MSL wordt geïmplementeerd door middel van de Oracle Service Bus 11g – recent geïntroduceerd als de opvolger van de OESB uit SOA Suite 10g. Van OSB 11g wordt met name betere schaalbaarheid en performance verwacht en ook betere administratie-faciliteiten en meer functionaliteit dan OESB biedt.

De Database en AQ Adapter worden veelvuldig toegepast om PL/SQL packages in de database te ontsluiten. Deze packages ontvangen en retourneren XMLType parameters. Deze opzet



Figuur 4: De service architectuur van i-bridge

houdt ondermeer in dat de service bus nooit rechtstreeks tegen tabellen of views praat en ook geen SQL bevat.

Naast services die met de database communiceren zijn er services die andere resources ontsluiten, zoals de Mail Server, UCM en het Identity Management System.

De Klant Integratie Module (KIM) is geïmplementeerd met SOA Suite 11g – Mediator en BPEL. De SOA composite applicaties in de KIM maken gebruik van de services in de MSL. Voor meer proces-georiënteerde composite applicaties en echte composite services wordt in de architectuur ook de MPL (Mondriaan Process Layer) onderkend. Deze is geïmplementeerd met SOA Suite 11g – ook met name BPEL – en maakt ook gebruik van de MSL Services om enterprise resources te benaderen. Op dit moment wordt er geen gebruik gemaakt van BPM – maar een verdere oriëntatie op business processen is wel aanstaande.

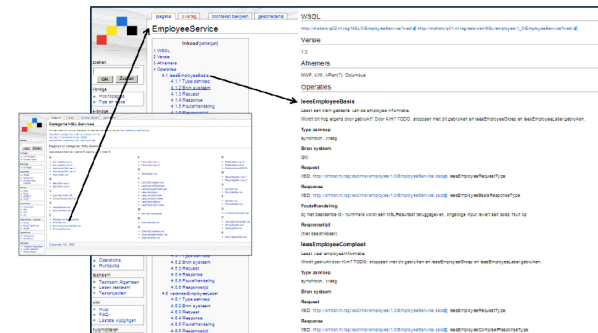
Het MWP (Mondriaan Web Portal) is gebouwd met ADF 11g en WebCenter. De JSF componenten zijn geassocieerd met managed beans die op hun beurt via een tussenlaag gekoppeld zijn aan web service proxies gegenereerd op basis van de MSL services. Data in een web pagina wordt dus via de proxies uit de MSL verkregen – en indirect uit de database maar dat is voor de ADF web applicatie volledig afgeschermd.

Op het eerste gezicht – zie de figuur hieronder – is er sprake van drie verschillende web applicaties, één voor elk van de drie marketinglabels Randstad, TempoTeam en Yacht.



In werkelijkheid is er sprake van slechts één applicatie, die door middel van dynamische geselecteerde skins, stylesheets en customizations het bij de gebruikte url passende uiterlijk toont. Met voor ieder label nog een aanzienlijke vrijheid in de definitie van functionaliteit en uiterlijk levert deze werkwijze een enorme efficiency-winst op, vergeleken met het bouwen en onderhouden van drie volledig gescheiden applicaties. De mobiele applicatie maakt gebruik van dezelfde web service proxy laag en uiteraard daaronder dus ook de MSL. De Forms applicatie werkt grotendeels direct op de database, maar is recent uitgebreid met een web service gebaseerde koppeling met UCM voor opslag en download van documenten die tot voor kort in BLOB kolommen in de database stonden. De recente migratie van de MSL van SOA Suite 10g naar OSB 11g is aangegrepen voor een inventarisatie van de service catalogus – alle beschikbare MSL services – en een verbeterde

uitwerking van het canoniek model – zeg maar het bedrijfsbrede data model van alle XML berichten. Van veel services is een opvolger gedefinieerd – een verbeterde variant die we in de namespace aanduiding positioneren als nieuwe versie van een bestaande service. In de praktijk is het echter een compleet nieuwe service die alleen in naamgeving en meta-documentatie een associatie heeft met de oorspronkelijke service. In dit traject hebben we ook de service administratie opgezet. Deze bestaat uit een verzameling Wiki-pagina's waarop per service en per operatie een aantal kern-gegevens vermeld staat -inclusief referenties naar de WSDL en XSD documenten. Om de vindbaarheid van services te vergroten en hergebruik te bevorderen wordt in de eenvoudig toegankelijke service administratie beschreven wat iedere operatie aan functionaliteit biedt, wat invoer en response-bericht structuren zijn (en welke faults kunnen worden geretourneerd), wat de gemiddelde responsetijd is, een indicatie van testresultaten, en eventueel andere SLA kenmerken zijn – zoals beschikbaarheid, autorisatie-eisen en headers met meta-data. Ook wordt de levenscyclus van de service beschreven: is de service de opvolger van een andere service, welke plannen zijn bekend rond de service en wat is de huidige status en tenslotte kent de service zelf een opvolger.

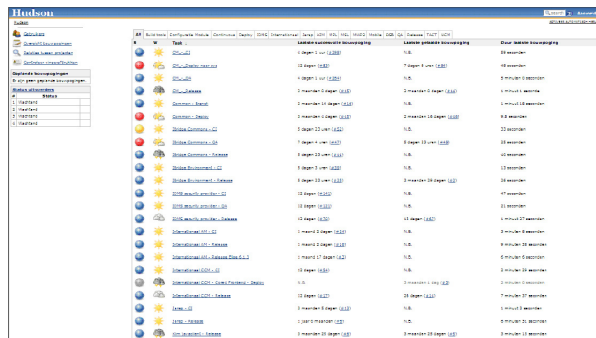


Figuur 6.

Software Engineering

Een multi-team, multi-technologie-project dat op agile wijze aan software ontwikkeling doet kan alleen succesvol zijn met gestroomlijnde software engineering. Alleen door gestructureerd en geautomatiseerd de omgevingen en ontwikkelde producten te beheren en manipuleren kan een zo complexe constellatie tot in productie worden gebracht. Het centraal beheer van source code bijvoorbeeld- over vele versies en releases heen, met ondersteuning van gelijktijdige, van elkaar geïsoleerde branches. Het i-bridge team maakt hiervoor gebruik van Subversion en de Tortoise client. Een ander belangrijk element wordt gevormd door automatische build-voorzieningen die van een samenhangende set artefacten op ieder gewenst moment een uitrolbaar eindproduct kunnen samenstellen – met ondermeer compilatie, automatische

test, omgevingspecifieke transformaties en samenstelling van deploybare JAR, WAR en EAR files. Het gebruikte build-tool is Maven en de Continuous Integration server is Hudson. Via de web interface van Hudson kunnen ontwikkelaars maar ook testers en beheerders builds laten uitvoeren en eventueel laten uitrollen naar een specifieke omgeving. Resultaten van een (mislukte) build kunnen geïnspecteerd worden.



Figuur 7.

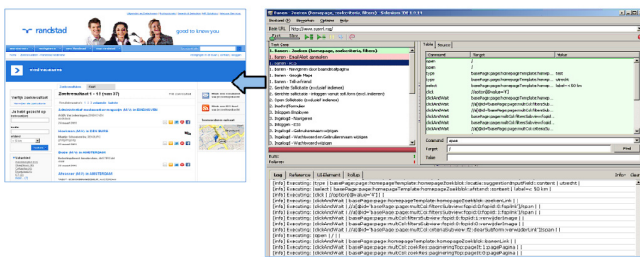
Het inrichten en beheren van een serie omgevingen met elk een eigen specifieke functie is noodzakelijk en tegelijk potentieel tijdrovend. Om tegelijk ontwikkeling, integratie, systeemtesten (binnen de teams) en acceptatie testen (door de business) te kunnen uitvoeren zijn voor die activiteiten afzonderlijke omgevingen opgezet. Met in elke gedeelde omgeving ondermeer een database, OSB Ilg instance, mail-server, Identity Management systeem, SOA Suite Ilg server en WebLogic Server domain.

De afbeelding toont de omgevingen-straat bij i-bridge, met uiterst links de lokale, individuele ontwikkelomgeving waar ontwikkelaars producten compileerbaar en executeerbaar maken in een snelle, lichtgewicht, uitprobeeromgeving waar ondermeer OSB services en Web Applicatie onderdelen worden ontwikkeld. De DEV, SYS en TST omgeving worden beheerd door het ontwikkelteam zelf. De TST omgeving wordt gebruikt door de eigen testers om in de loop van de sprint de productierijpheid van de team-deliverables in samenhang met deliverables van andere teams te verifiëren. In SYS brengen de teams de ingecheckte producten samen en verifiëren de ontwikkelaars en geautomatiseerde tests de correctheid van de programmatuur. Op DEV kunnen nog niet ingecheckte artefacten worden uitgetoond, bijvoorbeeld als een lokale omgeving niet beschikbaar is voor een bepaalde technologie. Deployment op DEV, SYS en TST gebeurt vooral vanuit Hudson. Het beheer van en deployment naar ACC en PRD ligt in handen van de operationele beheerteams.



Figuur 8.

Het geautomatiseerd testen is een belangrijk element in de twee-wekelijkse opleveringscyclus. Met een zo snelle evolutie van applicatie-componenten en een zo groot aantal afhankelijkheden is het erg belangrijk om voortdurend vast te stellen dat vereiste functionaliteit nog steeds aanwezig is en werkt als verwacht en ook binnen de gestelde performance-grenzen. Regressie-testen van webapplicaties en web-services vormen een belangrijk onderdeel van het voortbrengingsproces. Hiervoor wordt ondermeer gebruik gemaakt van Selenium – een plugin voor de Firefox browser waarmee scripts kunnen worden vastgelegd en afgespeeld die een gebruiker van een webapplicatie en haar acties simuleren en de reacties van de applicatie vergelijken met de eerder als correct vastgelegde antwoorden.

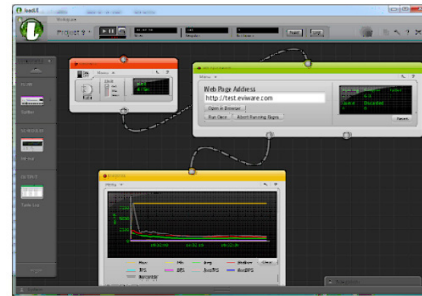


Figuur 9: een Selenium test-scenario voor de Randstad Nederland website.

Voor het verifiëren van de performance van de web applicatie wordt JMeter gebruikt – een tool waarmee een aantal 'robots' kan worden geactiveerd die gezamenlijk een load op het systeem zetten en de responsnelheden meten.

Nog verder geautomatiseerd is het test-proces voor de MSL web services. Met SoapUI zijn test-scripts gebouwd die voor iedere web service een happy flow en een bad flow testen. In de happy flow wordt iedere operatie aangeroepen met een 'normaal' verzoek en wordt de correctheid van het antwoord geverifieerd alsmede het verwachte 'side-effect' zoals een wijziging in de database of een verstuurde email. In de bad flow worden afwijkende scenario's met incomplete, incorrecte en onlogische aanroepen doorlopen en wordt gecontroleerd of de verwachte afhandeling optreedt – en in elk geval niet de ongewenste side-effecten optreden.

Een aan SoapUI complementair tool – LoadUI – wordt ingezet voor een load-test van de volledige MSL. Met LoadUI kunnen we gedurende een langere periode vanaf verschillende agents web service aanroepen uitvoeren via de SoapUI test cases in een belasting en onderlinge verhouding die min of meer representatief is voor de productie-situatie. De load of endurance test die we op die manier uitvoeren belast de MSL en de onderliggende systemen zoals database, document management systeem en mail server op een manier die overeenkomt met de werkelijkheid – en geeft ons een goed inzicht in de ontwikkeling van die load en de afhandeling daarvan tussen de verschillende releases.



Figuur 10: LoadUI.

Conclusies

Na de eerste -stevige- stappen in het evolutionaire moderniseringsproces kunnen diverse interessante conclusies worden getrokken. Als gevolg van de agile manier van werken zijn de business prioriteiten nu echt leidend geworden – en kunnen ze gaandeweg de ontwikkeltrajecten ook voortdurend worden bijgesteld op basis van nieuwe inzichten. De teams zijn beter gaan functioneren, meer als team en veel meer betrokken bij de business doelen van de applicaties. Het doen van software ontwikkeling op een agile manier blijkt prima mogelijk met Fusion Middleware. De service gebaseerde architectuur en de ontkoppeling op alle niveaus in de applicatie is overigens cruciaal, zowel om de teams naast elkaar te laten werken zonder grote wederzijdse impact als ook om binnen de teams ontwikkelaars van elkaar af te schermen.

Goed georganiseerde, sterk geautomatiseerde software engineering is essentieel om met korte sprints, meerdere parallel werkende teams en complex omgevingen met grote aantallen software artefacten op een agile manier – of eigenlijk op welke manier dan ook – te kunnen werken. Een substantieel deel van de totale inspanning wordt geïnvesteerd in het opzetten en onderhouden van geautomatiseerde build, test en deploy procedures. Tools voor ondermeer test (functioneel & load), source code control en versiebeheer, incidenten management en de Wiki voor het delen van instructies, best practices, de service administratie en connectie-gegevens voor alle omgevingen zijn ook van groot belang voor productiviteit, samenwerking en kwaliteit. Eind maart 2011 ging i-bridge een nieuwe fase in: nog meer rollen en medewerkers sloten aan bij de agile werkwijze en diverse projecten voor verdere modernisering van ondermeer payroll services en digitaal declareren gingen van start, op de nieuwste versie (11gRI Patch Set 3) van vrijwel alle componenten van Fusion Middleware.

Patrick Stevens is Team Manager bij i-bridge, het IT bedrijf van de Randstad Groep Nederland, en drijvende kracht achter de modernisering.

Lucas Jellema is als Solution Architect vanuit AMIS Services werkzaam bij i-bridge.