



Case: INDiGO

Art Ligthart, Ed Brandt

TestNet, 15 september 2010

INDiGO

- De Immigratie en Naturalisatie Dienst (IND) is bezig met een ingrijpend vernieuwingsprogramma.
- De primaire bedrijfsprocessen worden opnieuw ingericht en de gebruikte legacy-systemen worden in zijn geheel vervangen door een nieuw IT-systeem
- INDiGO is op SOA is gebaseerd, met een centrale plaats voor een rule-engine.
- De afgelopen 2 jaar is de engine gevuld met regels, zodat de IND nu zelf haar regels kan aanpassen.
- Testen van INDiGO bracht speciale uitdagingen met zich mee. De rules, de ontwikkelmethode, de talloze ketenkoppelingen, het standaardpakket Siebel waarin de gegevens worden geadministreerd, de kennisinterface tussen Belnformed en Siebel...
- De IND deelt graag haar ervaringen



De architectuur van INDiGO

Vernieuwingen voor de IND

Voor de organisatie:

- Reorganisatie, generiek proces
- Ondersteuning met regels / kennis
- Vrijheid in processen
- Digitaal dossier

In het IT-systeem:

- Dynamische behandelplannen
- Business services
- Rule-engine
- Generieke Kennis Interface
- Standaard pakketten



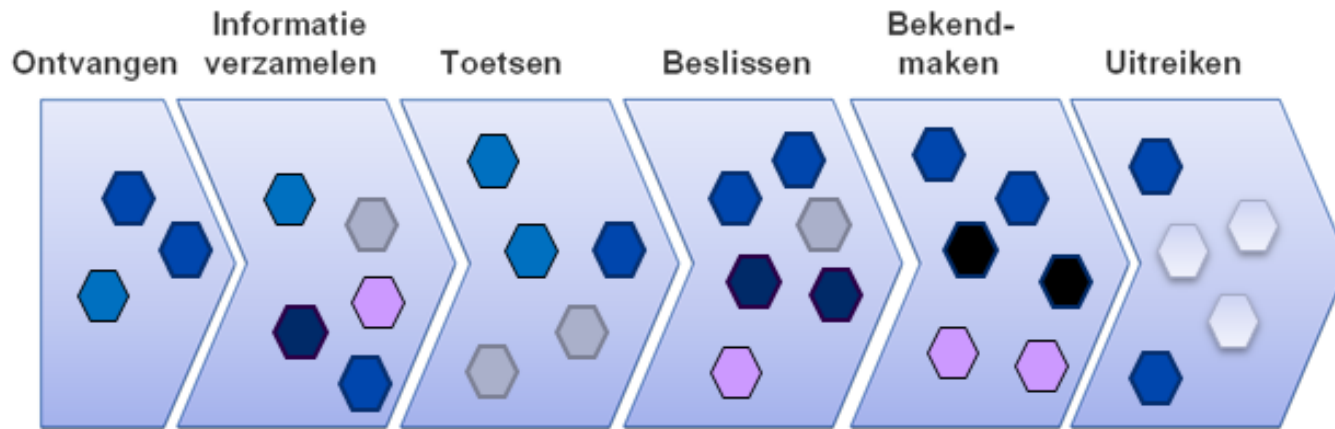
The screenshot shows the website of the Nederlands Architectuur Forum (NAF). The header includes the NAF logo and the tagline 'Nederlands Architectuur Forum voor de digitale wereld'. A navigation menu lists 'Welkom', 'Nieuws', 'Werkgroepen', 'NAF-prijzen', 'NAF Insights', 'LAC', 'NAF Academy', and 'Links'. A list of award winners from 2002 to 2009 is shown, with the 2009 winner, Paul Teeuwen en de IND, highlighted. To the right, the text describes the 2009 award, noting it was given to the INDiGO program for its innovative and secure solution in replacing legacy systems.

NAF-architectuurprijs 2009: de IND

De NAF-architectuurprijs 2009 is toegekend aan het INDiGO-programma van de Immigratie- en Naturalisatiedienst (IND). Binnen dit programma worden de primaire bedrijfsprocessen opnieuw ingericht en worden de gebruikte legacy-systemen in hun geheel vervangen door een innovatieve en gedurfde oplossing. De drijvende krachten achter INDiGO zijn Simone Dobbelaar en Hans Blokpoel in hun rollen van respectievelijk hoofd Beleidsinformatisering en CIO. Zij hebben de kwaliteit van de oplossing laten prevaleren, ook in tijden dat het in de realisatie van het programma tegenzat. De architectuur heeft zijn werk hier gedaan als concrete uitdrukking van het gemeenschappelijke doel.

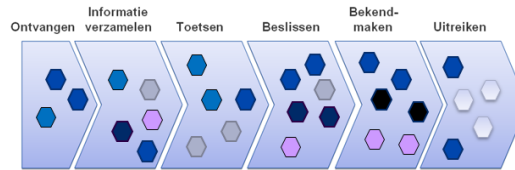
De jury heeft de inzendingen inhoudelijk beoordeeld tegen een aantal criteria. Heeft de architectuur bijgedragen aan het succes van een systeem doordat een verankering in de governance van een veranderprogramma heeft plaatsgevonden? Had de architectuurbeschrijving voldoende diepgang en was deze voldoende breed? Was er sprake van een innovatie in het bestaande landschap van applicaties en infrastructuur? Was er sprake van een brede zichtbaarheid en impact dan alleen in een project? Heeft de architectuur geholpen bij het overwinnen van uitdagingen of moeilijkheden waar een programma mee te maken heeft gehad? Kortom, heeft de architectuur de bijdrage geleverd in de geest van het thema van het LAC 2009?

Een generiek proces met generieke activiteiten




- 150 business services ondersteunen het primaire proces
- Een business service kent dataopslag, schermen, regels, documentopslag
- Op elk willekeurig moment kan aan de rule-engine gevraagd worden welke business services relevant zijn (behandelplan, soort workflow)

De business services zijn gerealiseerd met pakketten



Administratie
en schermen

KDW en Portalen

 Berichtenuitwisseling, Search, Generieke Kennis Interface, Event Handling, Foutafhandeling, Logging

Matching

Regels

Documenten

Correspondentie

De rule-engine

- Vernieuwend, ander modelleringsparadigma
- Inmiddels 30000 concepten, veel meer regels

De communicatie tussen de rule-engine en de gegevensadministratie

- Generieke Kennis Interface
- Kennis = gegevens + regels....dus niet alleen de rule-engine testen!

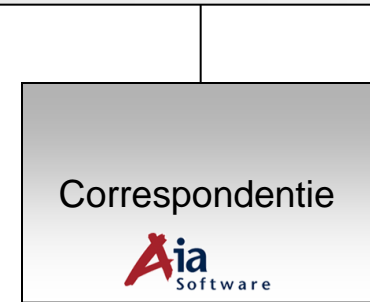
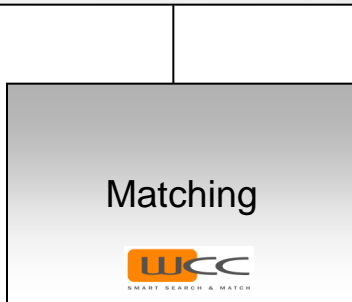
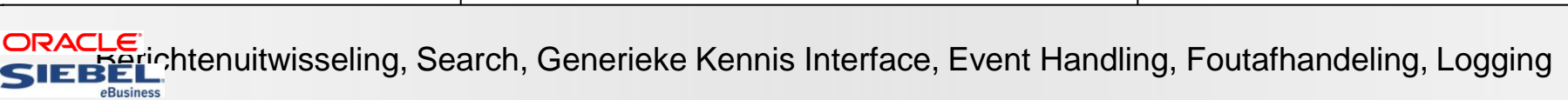
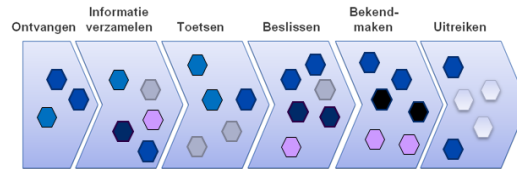
De talloze ketenkoppelingen

- Protocollen, oude codelijsten

De Business Services

- Generieke opzet, regels apart, ...

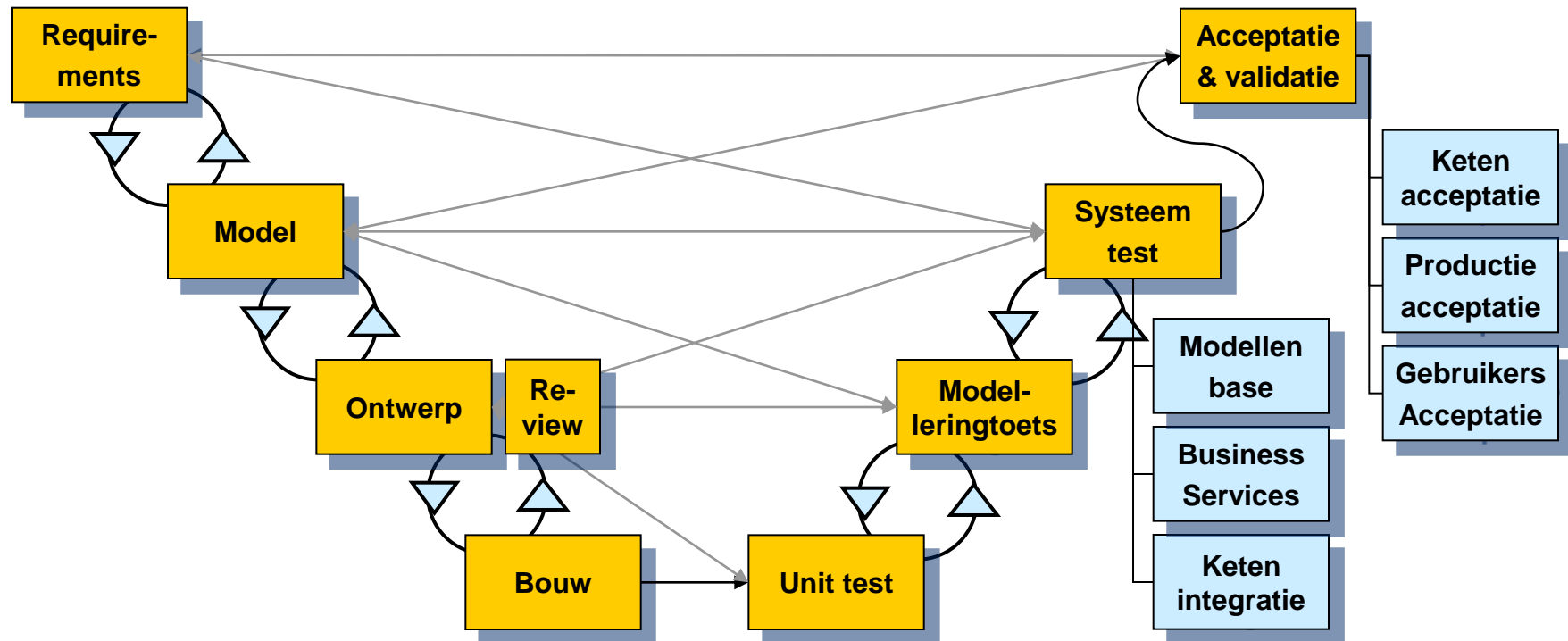
Werking van een business service





De testuitdagingen

Mix van “oude vertrouwde V-model” en incrementele ontwikkeling



Huidige systeemtestset

Circa 1000 scripts tbv Business Services (handmatig)

Circa 2500 scripts tbv BelInformed Modelbase (geautomatiseerd)

Circa 1000 scripts tbv Keten integratie (handmatig)

Uitdagingen

De service architectuur

- Vele mogelijke scenario's en 'volgorde' waarin services worden doorlopen
- Berichtenverkeer ESB

De modellenbase

- Rule engine vs inhoud

Koppelingen

- Extern (GBA, BVV, CJIB, etc)
- Intern met legacy systemen

Ontwikkelmethodiek

- Hoge wijzigingsgraad vraagt veel herhaling van testen (regressie)

Discussiepunten

Is Siebel een pakket of een ontwikkelomgeving?

Hoe test je de inhoud van een BOS/expertsysteem?

Hoe “herhaalbaar” en “overdraagbaar” moet de testware zijn?

Waar ligt het break-even-point bij inzet van geautomatiseerd testen?

