Overview and Insight

a practical approach to connect business, applications and technology with Archimate

Marcel van Dijk, Lourens Riemens
NL Tax and Customs Administration
contents

introduction

overview and insight

approach

results

next steps
Tax and Customs

The Tax and Customs Administration is part of the Ministry of Finance.

**Tax Collection**
We collect taxes on behalf of the government.

**Benefits**
We provide benefits in the form of financial support for rent, children and for health insurance.

**Customs**
We monitor goods that come into and leave the country.

**Fighting Fraud**
Whenever necessary we carry out criminal investigations.
Tax and Customs

- 11.2 million tax returns received
- 29 thousand employees
- 3 thousand IT
- 232 billion € tax revenues
- 9 million citizens with benefits
- 12 million containers in the port of Rotterdam
introduction

high level of computerization

complex IT landscape

high costs and not flexible
contents

introduction

overview and insight

approach

results

next steps
overview and insight

overview
Overview of business products and business processes, the applications and data needed to support these business processes, and the technology and infrastructure to realize and operate these applications.

insight
Insight in characteristics of applications and technology in order to determine an application strategy, make improvement plans, etc.
overview and insight

Overview and insight contributes to the following **drivers and goals**: 

**guarantee continuity**
- make overview and insight a permanent part of life cycle management
- refactor or rehost applications to ensure up-to-date-technology

**accommodate change**
- define roadmaps based on consistent and coherent information across domains and layers
- determine the impact of change

**reduce costs**
- spend less money on maintenance and use it for development and enhancements
overview and insight
contents

introduction

overview and insight

approach

results

next steps
incremental approach

a central repository

standardized information on business, application and technology

central governance, local maintenance

integrated with application metrics and other sources
repository structure
application layer concepts

the Application Layer standards and guidelines model contains:

- concepts
- default views
- scripts
application layer concepts

**APPLICATION** with properties for

- identification and long name
- number of users
- metrics: business value and technical value (APW), size (FP) and maintainability score (SIG)

**APPLICATION SERVICE**

- logical relationship between applications (no implementation details)
application layer concepts

APPLICATION DOMAIN
- coherent grouping of APPLICATIONS

APPLICATION TYPE
- classification of APPLICATIONS to distinguish between central / local / external applications

LIFE CYCLE
- life cycle stage of an APPLICATION

OWNER
- relationship between APPLICATION and predefined set of OWNERS
application layer concepts

DEVELOPMENT SERVICE
- relationship between APPLICATION and predefined tool sets used to design, code, test, integrate and deploy

BUILDING BLOCK SERVICE
- relationship between APPLICATION and predefined SBB (Solution Building Block) for business process management, content management, rule management, etc.
application layer concepts

**APPLICATION SERVICE**
- relationship between APPLICATION and BUSINESS process or service

**HOSTING SERVICE**
- relationship between APPLICATION and predefined HOSTING environment(s)
application layer concepts

Part of the Application Layer standards and guidelines model are validation rules to automatically check the quality of the domain models:

For example:
- an APPLICATION must have a unique IDENTIFICATION
- an APPLICATION must belong to one APPLICATION DOMAIN
- an APPLICATION with LIFECYCLE “in use” must have a BUSINESS VALUE
- an APPLICATION must have one or more DEVELOPMENT SERVICES
- an APPLICATION DOMAIN must have an APPLICATION LANDSCAPE view
results

- **guidelines** for business, application and infrastructure layer

- standardized views on application **landscape** and **context**, and domain **dashboards**

- scripts to facilitate the use of existing data and **metrics**

- automated **quality checks**

- self-service **publication** of all models on intranet

- a **supermodel** for analysis
added value

• **consistency** across different business domains and different layers
  – easier sharing and better understanding

• **information** is available for everyone
  – same information is used, less mismatches, less routine questions for architects

• strategic/tactical **analysis** to accommodate change
  – improved quality of decisions, less lead time, lower costs

• **architecture** products are more standardized
  – less time needed for baseline architecture products, improved quality, better start for development
overall IT landscape

a top level view of business units and application domains
portfolio valuation

visualization of the business value and technical value of all applications using the Gartner TIME method
portfolio valuation

a subset with all applications based on end-of-life development tools

driven by a need to prioritize retire, rebuild, refactor or rehost activities for applications
age

Staatdiagram

year 2000
don’t know
outdated
**insights**

<table>
<thead>
<tr>
<th>Object</th>
<th>concept</th>
<th>aantal</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>APPLICATIE</td>
<td>392</td>
</tr>
<tr>
<td>object</td>
<td>APPLICATIE SERVICE (PROCES)</td>
<td>268</td>
</tr>
<tr>
<td>object</td>
<td>APPLICATIE SERVICE (KOPPELVLAK)</td>
<td>451</td>
</tr>
<tr>
<td>object</td>
<td>ONTWIKKEL SERVICE</td>
<td>9</td>
</tr>
<tr>
<td>object</td>
<td>HOSTING SERVICE</td>
<td>16</td>
</tr>
<tr>
<td>object</td>
<td>BOUWBLOK SERVICE</td>
<td>5</td>
</tr>
<tr>
<td>object</td>
<td>EIGENAAR</td>
<td>7</td>
</tr>
<tr>
<td>object</td>
<td>APPLICATIE DOMEIN</td>
<td>11</td>
</tr>
</tbody>
</table>

451 application services published

The most-connected application has connections with 106 other applications

5% of all applications is involved with 80% of all connections

175 application services are not used (or not defined yet)

16 hosting services, but 41 unique combinations of hosting services
insights

radar diagrams for business value (left) and technical value (right) can be used to see on what aspects an application should be improved.

![Radar diagram for business value](left)

![Radar diagram for technical value](right)
application landscape (LOA)  
domain taxes

a similar landscape map of all locally developed applications in the taxes domain
visualization of the **business value** and **technical value** of all applications in the taxes domain taxes using the Gartner TIME method
The number of applications that have been built with a development service:

Most applications in this domain are Java (12), followed by COBOL (9) and EUC (7).

Groupware (1), Gen (1) and PBS (6) are development services that will retire in 1 or 2 years, so a rebuild, refactor or retire of these applications is needed.
the size of each application

function points are used as a measure for the size of an application

ABS is by far the biggest application, followed by SES en VBN

but this domain also has many small applications
application dashboard

APK diagram FPOS 0.1

gebruikte ontwikkelservices

domain taxes

the total size of applications for each development service in function points COBOL is the most important development service. PBS is also a big part of this domain, so there is a lot of work to do rebuild, refactor or retire before this development service can retire.
Application dashboard

There are different norms depending on the lifecycle of the application and the development service used.

This overview can be used to plan improvements.

domain taxes

The maintainability score for each application.
The life cycle of each application

The oldest application in this domain is 30 years old (VBN)

A lot of applications will be retired or replaced in the next two years

For most applications there is no end-of-life known
a part of the application landscape of the taxes domain, i.e. all applications involved with corporate taxes

20 applications are maintained within this domain

30 more applications from 8 other domains are needed to support the whole (not shown)
these tables show similar information as the previous diagrams
corporate tax, the business:

- 700,000 companies
- 14 billion euro revenues in 2015
- first VBN-release in 1985
- > 3 releases each year with law changes, process improvements and maintenance activities
application context

Since 1985

corporate tax system (VBN)

corporate tax, the application:

• VBN
• first released in 1985
• supporting 25 business processes
• 1.400 end users in 2016
• COBOL, JCL and CICS/DB2 technology
• 5.830 function points
• 1.713.931 lines of code in 2008
• 1,9 maintainability score
• 1.500 function point changes per year
• > 22 connections with other applications

it’s old ... it’s big ... it’s complex ... it’s never finished
- our legacy -
application context
corporate tax system (VBN)

Building cathedrales

VBN, the application
- supporting 25 business processes
- 1,400 end users in 2016
- COBOL, JCL and CICS/DB2 technology
- 1,713,931 lines of code in 2008
- **1,113,917 lines of code in 2015**
- 1,9 maintainability score
- 5,830 function points
- 1,500 function point changes per year
- > 22 connections
- productivity < 12 h per fp

it’s old ... it’s big ... it’s complex ... it’s never finished
- our legacy -
application context

corporate tax system (VBN)
application context

corporate tax system (VBN)
exploitation corporate tax system (VBN)
next steps

• add standards and guidelines for architecture at project/system level

• combine meta models for business, application and infrastructure into one

• extend the meta models with concepts for information and data

• add perspectives for target architecture

• ...
challenges

• modeling of **scope** and **time**-aspects for baseline, transformation and target views

• **track changes** and progress in our models

• deal with **different levels of details** of architecture
  – enterprise level, domain (segment) level, project (system) level

• define and keep track of different **versions** of (technology) concepts

• support for **stereotypes, properties, validation rules** and **consistency checks**

• use of **repository** in a large scale IT environment
  – with 700+ model packages and 500+ architects and designers
lessons learned

• iterate in small steps
• use a standardized meta model
• reuse existing data / metrics
• automate quality checks
• centralize standards, guidelines and support
• decentralize domain models
• visualize and publish
• have a sponsor (CIO)
contact information

**Marcel van Dijk**, IT Architect
m.w.van.dijk@belastingdienst.nl

**Lourens Riemens**, IT Architect
lj.riemens@belastingdienst.nl

**Wim Scholtes**, Program Manager
wj.scholtes@belastingdienst.nl

www.belastingdienst.nl