Avancier Methods (AM)
Enterprise Architecture

Analyse baseline apps

It is illegal to copy, share or show this document
(or other document published at http://avancier.co.uk)
without the written permission of the copyright holder
What is the apps architects’ domain?

Environment

Business layer

Application layer contains digital information systems

Technology layer

Business applications with a focus on rationalisation integration applications road map
Motivation: the need for application rationalisation

▶ “The typical IT environment today is overwrought with complexity that impedes service quality, stifles dynamism, and is expensive to maintain.”

▶ “Commonly, solution architects in the design division are driven to meet the immediate requirements of individual business units, with the result that only tactical stand-alone solutions are developed and implemented.

▶ “The end result of this approach is a highly fragmented and disparate IT estate, with duplication and waste of resources.”

▶ IT Business Edge 2010
Motivation: the need for apps portfolio management

“Organizations spend > 70% of technology budgets in maintaining an existing inventory of applications

- many redundant and/or have exceeded their useful life, consuming resources without returning value.

… organizations need a clear inventory of applications, their interdependencies and related business processes.

Organizations can use enterprise architecture and portfolio management approaches to get the required knowledge [to]

- streamline and rationalize the apps portfolio
- reduce redundancy, consolidate IT capabilities
- define sound IT governance policies.”

it-whitepapers@groups.ittoolbox.com;
on behalf of; Toolbox for IT Research Alert [ITtoolboxAlerts@ITtoolbox.com]
Re-engineering an enterprise’s apps portfolio

- An idealised transformation process for reengineering an apps portfolio
- You can rarely follow this process completely, or separately from consideration of other architecture domains.
Strategic apps architecture

Initiate

Govern

Manage

Architect
- Understand the baseline
- Review initiation outputs
- Review NFRs
- Design the target

Plan
Analyse baseline apps

1. Scope the business data processing context
2. List applications in scope
3. Map apps to business activities
4. Document inter-app data flows
5. Map apps to data stores
6. Document use cases
7. Map use cases to data entities
8. Document automated services
9. Catalogue, classify and score apps (see following presentation).
1. Scope the business data processing context

- Identify the business context for data processing
  - What business functions are or will be supported?
  - What business processes are or will be supported?

Core business functions:
- Marketing
- Sales
- Delivery
- Customer Service

Support business functions:
- HR
- IT
- Accounts
- Legal
2. Lists apps in scope

- Accounts Payable
- Accounts Receivable
- Activity Management
- Benefits
- BI Warehouse
- Billing
- Bills of Material
- Capacity
- Cash Management
- Claim Processing
- Commission Calculation
- Commissions
- Cost Management
- Costing
- Customer Contact & Call Center support
- Engineering
- Fixed Assets
- General Ledger
- Human Resources

- Inspection of goods
- Inventory
- Manufacturing Flow
- Manufacturing Process
- Manufacturing Projects
- Order Entry
- Payroll
- Product Configurator
- Purchasing
- Quality Control
- Rostering
- Sales & Marketing
- Scheduling
- Service
- Supplier Scheduling
- Supply Chain Planning
- Time & Attendance
- Time & Expense
- Training
- Workflow Management

NCC monitor use of “packages” under these headings

- ERP
- CRM, Call Centre & Marketing
- Data Warehousing, Business Intelligence & CPM
- Document Management, Content Management & BPM
- Accounting & Financial Reporting
- HR & Payroll
- Project Management & PSA
3. Map apps to business activities (coarse-grained analysis)

► Applications are often mapped to 2\textsuperscript{nd} or 3\textsuperscript{rd} level functions in a business function/capability hierarchy

![Diagram showing mapping of apps to business activities]
Mapping NCC apps to generic business functions

Vision and Strategy
- Business Intelligence
- Data Warehousing

Product/Service Design

Sales and Marketing
- CPM
- CRM
- Marketing

Customer Service
- Call Centre

External Relationships
- Outsourcing (BPO)

Supply chain
- ERP

Manufacture
- ERP

Product service delivery
- BPM

IT
- EAI Middleware
- IT Service Management
- Network Management
- Server Management

Finance
- Financial Reporting
- Accounting

Human Resources
- HR
- Payroll

Knowledge Improvement and Change
- Project Management
- Content Management
- Doc Management

Copyright Avancier Limited
Mapping ERP apps to generic business functions

Product/Service Design
- Costing
- Engineering

Sales and Marketing
- Sales and Marketing
- Order Entry
- Billing
- Product Configurator
- Commission Calculation
- Commissions

Customer Service
- Claim Processing
- Customer Contact
- Service
- Call Center support

Supply chain
- Inventory
- Supply Chain Planning
- Inspection of goods
- Supplier Scheduling
- Purchasing

Manufacture
- Manufacturing Projects
- Manufacturing Process
- Manufacturing Flow
- Bills of Material
- Cost Management

Product service delivery
- Activity Management
- Workflow Management
- Scheduling
- Capacity
- Quality Control
- Time and Expenses

Finance
- General Ledger
- Fixed Assets
- Cash Management
- Accounts Payable
- Accounts Receivable

Human Resources
- Payroll
- Human Resources
- Time and Attendance
- Rostering
- Benefits

Knowledge Improvement and Change
- BI Warehouse
- Training
Managing very large application portfolios

► What if the application portfolio is too large to manage?

► Focus attention on apps that:
  ■ are mission critical
  ■ used by many actors

► Or

► Map “system families” to business functions
► And document each system family separately
4. Document inter-app data flows

► What data flows support business activities and flow between applications?
► Given a small system family, or enough time, you can later document the relationship via the relevant use cases.
Draw apps communication diagram(s)

One system family
Draw apps communication diagram(s) – shock and awe level
Draw apps communication diagram(s)

► One system family

This document is part of:
SVP 200 Volume 1

Page 1 of 2

Links to Site Level Application Integration Drawings
SSF CCP1 HFF NIMO

GTI PROP & RQC Architecture – Drawing number ARCH_6600_1
PROP / RQC Corporate Systems Application Integration
Version 3.6, Aug 18, 2012 – Drawn by FileNet – gPOMS Interface, changed FileNet Authentication to AD.

Application Architect

The signature above means that corporate applications and integrations are accurately reflected in this Drawing.

Quality Assurance

The signature above means that this drawing has been reviewed and approved per quality policies and procedures.

Printed Name
Signature
Date
Define data flows - 2

Data Flow Diagram Labware LIMS - Empower

![Data Flow Diagram]

GTI PROP Architecture – Drawing number ARCH_6655
Labware LIMS – Empower Data Flow Diagram
Version 1.2 August 25, 2010 – Added border, changed object behavior, change title for consistency, standardized dwg..

Empower Tech Lead
The signature above means that the integrations with Empower are accurately reflected in this drawing.

Labware LIMS Tech. Lead
The signature above means that the integrations with Labware LIMS are accurately reflected in this drawing.

Quality Assurance
The signature above means that this drawing has been reviewed and approved per quality policies and procedures.

System Owner
The signature above means that the application interactions shown in this drawing are consistent with the validated functionality of this application.

Printed Name
Signature
Date
Define data flows - 3

Data Flow Diagram MCS - POMS

NIMO MCS

1. Recipe Creation Request – (Recipe Name, Recipe Version ADDED BY SVC), Item #, Lot #
2. Request Material Status – Item #, Lot #
3. Material Status – Item #, Lot #, Status response
4. Report Production – Item #, Lot #, Qty
5. Report Consumption – Item #, Lot #, Qty
6. Unit Procedure Status – Complete or cancelled, Item #, Lot #
7. Lot Status Change, Item #, Lot #, new state, expiration date

POMS

GTI PROP Architecture – Drawing number ARCH_6710
DFD Campaign Studio – POMS

MCS Tech Lead
The signature above means that the integrations with MCS are accurately reflected in this drawing.

POMS Tech Lead
The signature above means that the integrations with POMS are accurately reflected in this drawing.

Quality Assurance
The signature above means that this drawing has been reviewed and approved per quality policies and procedures.

System Owner
The signature above means that the application interactions shown in this drawing are consistent with the validated functionality of this application.

Printed Name
Signature
Date
Define I/O data flows

► Catalogue data flows
  ■ provided via use cases to business activities
  ■ provided by applications to each other

► Data flows can include:
  ■ Serial files
  ■ Messages
  ■ User interfaces
  ■ Reports
Manage complexity by composition and decomposition

- Where architecture diagrams grow too complex, then
  - Create higher level diagrams for system families/applications
  - Elaborate in lower level diagrams for applications/components

- Similar patterns and principles must be applicable to applications large and small; else architects would have no repeatable methodology.
5. Map apps to data stores

Apps access data stores to monitor and control the state of entities and activities represented by that data.

<table>
<thead>
<tr>
<th>Application</th>
<th>CRM</th>
<th>ERP</th>
<th>Billing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Product</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Accounts</td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>
6. Document use cases (more fine-grained analysis)

Given a small system family, and enough time, you can document the relationship via the data flows and/or use cases the application offers.

Look for overlaps between use cases provided by different apps
Draw top-level use case diagrams – without sub processes

- Use case diagrams scope what processes a single app supports
Describe use cases as services (without process flows)

- Look for overlap between use cases provided by different apps

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Service Type</th>
<th>Use Case</th>
<th>999</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signature</td>
<td>Name</td>
<td>Order Goods</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Input</td>
<td>Order and Payment details</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Output</td>
<td>Order confirmation messages</td>
<td></td>
</tr>
<tr>
<td>Semantics or rules</td>
<td>Preconditions</td>
<td>Payment card is verifiable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Post conditions</td>
<td>Money transferred, Goods ordered</td>
<td></td>
</tr>
<tr>
<td>Measures</td>
<td>Response time</td>
<td>5 minutes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Throughput</td>
<td>20 per hour</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Availability</td>
<td>99% 06.30 to 03.30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Security level</td>
<td>Etc.</td>
<td></td>
</tr>
</tbody>
</table>
7. Map use cases to data entities

Apps store data to monitor and control the state of entities and activities represented by that data:
- Customer (Cust Id, Name, Postal Address, Email Address)
- Product Type (SKU, Description, Unit Price, Warranty Period, Delivery Fee, Installation Fee)
- Product Instance (SKU, Instance Number, Supplier Id, Purchaser Id, Delivery Date).

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Order</th>
<th>Deliver</th>
<th>Collect payment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer</td>
<td>Create/Use</td>
<td>Use</td>
<td>Use</td>
</tr>
<tr>
<td>Product Type</td>
<td>Use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Order</td>
<td>Create</td>
<td>Use</td>
<td>Use</td>
</tr>
</tbody>
</table>

Motivation
- Business Activity
- Services
- Use Case
- I/O Data Flow
- Data Flow
- Data Entity
- Data Store
- Objects
- Components
- App

Copyright Avancier Limited
8. Document automated services

Coarse-grained

Fine-grained
9. Catalogue, classify and score apps (see following)

- Catalogue and classify apps to understand them
  - By user type: Public / Technical
  - By generality: Universal, Business-specific
  - By user base: Single-user / Departmental / Enterprise
  - By usage style: OLTP / Business Intelligence
  - Whatever >>
Further classify and score apps (see separate presentation)

Classify Apps by
- Size
- Complexity
- Value
- Cost
- Business dependency
- Usability
- Business fitness
- Technical fitness

Devise a weighted scoring system
Form an enterprise application road map.
1. Scope the business data processing context
2. List applications in scope
3. Map apps to business activities
4. Document inter-app data flows
5. Map apps to data stores
6. Document use cases
7. Map use cases to data entities
8. Document automated services
9. Catalogue, classify and score apps (see following presentation).