ERP Course: Planning, Design, and Implementation of ERP
Readings: Chapter 3 Mary Sumner

Peter Dolog
dolog [at] cs [dot] aau [dot] dk
5.2.47
Information Systems
October 3, 2007
ERP Implementation

Phases stay:
- Planning
- Requirements analysis
- Design
- Detailed design
- Implementation
- Maintainance

Focus changes
- To fit the existing software (ERP) package to an organization
Planning (Business justification)

Inventory cost reductions
- Ability to use timely operational data

IT cost reductions
- Ability to integrate systems instead of maintaining many separate systems

Personnel cost reductions
- Ability to enhance systems without incurring the time and cost of custom development and modifications

Increased profitability
- Ability to introduce new features

Productivity improvement
- Access online to real time data

Better cash management
- Reduction in cost and time of systems development and maintenance
Requirements Analysis

Analyzing business process (how company works)
Analyzing how those process are already supported
Specifying the processes to be supported in addition or change of the current support
Should fit with organization’s goals and competitive strategy
Analysis of technical infrastructure
Specification of technical infrastructure which should enable the change
Selecting an ERP System

Create a vision
Create a feature list
Create a software candidate list
Narrow the field to 4 - 6 candidates
Create RFP
Select 2 - 3 finalists
Select a winner
Justify the investment
Negotiate the contract
Run a pre-implementation pilot
Validate justification
Technology Factors

Cost of technology (start-up and recurring)
Installation (support, time, and cost)
User interfaces
Upgradability
Computing environment
Personnel requirements (to use and to design)
Design

Re-engineering vs. Customization

Re-engineering

• Analyse possibilities to change processes and organizational structures
• Design changes to fit ERP best practices

Customization

• Analyse current processes
• Suggest an ERP system change to fit it to existing processes
Re-Engineering vs. Customizing

**Customizing**
- Supports unique business processes
- Strategic processes are maintained
- Difficulty to introduce some changes
- Difficulty with upgrades

**Re-Engineering**
- Features and processes supported by ERP
- Based on best practices
- Does not support strategic or unique business processes
- Resistance to organizational change
Re-engineering and customization factors

<table>
<thead>
<tr>
<th></th>
<th>Re-engineering</th>
<th>Customizing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Re-engineering</td>
<td>Software system best practices</td>
<td>Independent of tools being implemented</td>
</tr>
<tr>
<td>business processes</td>
<td>works well with minimal changes but can disturb the</td>
<td>may disrupt organization less</td>
</tr>
<tr>
<td></td>
<td>organization if extensive changes are required</td>
<td>because the software is designed to the processes</td>
</tr>
<tr>
<td>Organizational fit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evolution</td>
<td>depends on vendor</td>
<td>evolution can support unique requirement but</td>
</tr>
<tr>
<td></td>
<td></td>
<td>create difficulties when a vendor upgrades</td>
</tr>
<tr>
<td></td>
<td></td>
<td>features you have changed</td>
</tr>
<tr>
<td></td>
<td>Re-engineering</td>
<td>Customizing</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>Implementation is cost effective</td>
<td>may involve extensive costs of custom implementation</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td>boundaries set by business process models and best practices</td>
<td>more flexibility for custom requirements</td>
</tr>
<tr>
<td><strong>Competitiveness</strong></td>
<td>other firms have the same settings</td>
<td>do not have to use the software which other companies in industry adopted</td>
</tr>
<tr>
<td><strong>Fit</strong></td>
<td>Need to fit to requirements drawn by the ERP</td>
<td>Unique requirements has to be supported by a customization</td>
</tr>
<tr>
<td><strong>External consulting</strong></td>
<td>Needed to consult business process change</td>
<td>Needed to consult system implementation change</td>
</tr>
</tbody>
</table>
ERP Implementation Alternatives

Vanilla implementation
Single vendor with customization
In-house with supplementary ERP modules
ASP
Detailed Design

Select applicable business processes
Discard inaplicable business processes
Reorganize and document new processes
Identify areas not covered by the best practices which require customization and development
Models Used

Component model – show major functions
Organization model – breakdown of organization structure
Data model – information needed by a company
Interaction model – information flow between organizational units
Implementation

Dialog customization
Dialog connection customization
Processing functions customization
Data model customization
Reports customization
Integration with the office systems
Protocols/Activities

SearchRoom → ConfirmReservation → Pay
Collaboration/Interaction Diagrams

1: SearchRoom() →
2: ConfirmReservation() →
3: Pay() →

1.1: GetRoomOffer() →
1.2: AcceptRoomOffer() →
2.1: ConfirmRes ←

TourismWebApplication

ReservationBroker

Top Package::User

PaymentBroker

3.1: SendCCNo() ←
Dialog Sequencing – classify product

User

Categories
- Connect(CategoryID, ProductID)
- Show()

Modify Product
- updateCategory(categoryOID)
- Show()

ProductDetails
- fetch()

Product
- fetch()

Category
Dialog Sequencing – user interaction
Template: Business Process Management

- **Contract Selector**
  - *Date
  - *Instrument
  - +primaryparties
  - +counterparties

- **Contract Filter**
  - +isIncluded
  - +selectContracts(in Collection)

- **Complex Filter**
  - HardCodedFilter
  - *+filter
  - 1..1
  - BooleanMethod

- **Operational Filter**
  - SetFilter
  - *+filter
  - 1..1
  - SetOperation
  - {documentation = Instances: Union, Intersection, Negation}

- **Contract**
  - Value(scenario)

- **Contract Portfolio**
  - «derived»
  - {All instances on contract on which self.filter is true}
Client Customization

4G languages – forms
- Adding, modifying attributes, control boxes, …
- Adding/modifying client function
- Adding/modifying connection to database
- Adding/modifying menus, control flows, …

Client APIs
- Externilize dialogs to functions
- Allow to instantiate and embed client dialogs and functions in external programs
Forms Development
Visual Basic for Applications
Message Queues (e.g. Navision)

```csharp
public Form1() { //
    // Required for Windows Form Designer support //

    InitializeComponent();

    // // TODO: Add any constructor code after InitializeComponent call //
    mqFromNavision.Formatter = new
    System.Messaging.XmlMessageFormatter(new Type[]{typeof(String)});

    private void mqFromNavision_ReceiveCompleted(object sender,
    System.Messaging.ReceiveCompletedEventArgs e) {
        System.Messaging.Message m =
        mqFromNavision.EndReceive(e.AsyncResult);
        txtReceive.Text = (string)m.Body;
    }
```
Customer calling

Call center software identifies the number

The procedure at the call center site associated with sales telephones runs a procedure where a navision objects are embeded

A function for opening and looking up particular contact is implemented
Business And Database Tier Customizations

Set of business functions and rules
Language to create them, e.g. Enterprise Java Beans, Oracle Application Server procedures, Oracle stored procedures
APIs/SDKs to access database and business functions on the server
Automation technology to embed and use it in external programming environments and applications
Adding attributes/tables/triggers