Integration of Software Development Processes with Enterprise Architecture

Hossein Keramati

Sharif University of Technology
Nov. 2010
Enterprise Unified Process (EUP) and Enterprise Architecture

Goal: Extend RUP to Drive Improvements Across the Entire IT Lifecycle

Introduced by Ambler and Constantine in 2000 as an extended variant of RUP.
- A revised and refactored version was introduced in 2005

Motivated by the belief that RUP suffers from serious drawbacks

Extends RUP by adding two new phases and two new disciplines
- One of disciplines is broken into seven disciplines in 2005 version of the methodology
EUP (contd.)

**Development Disciplines**
- Business Modeling
- Requirements
- Analysis & Design
- Implementation
- Test
- Deployment

**Support Disciplines**
- Configuration & Change Mgmt
- Project Management
- Environment
- Operations & Support

**Enterprise Disciplines**
- Enterprise Business Modeling
- Portfolio Management
- Enterprise Architecture
- Strategic Reuse
- People Management
- Enterprise Administration
- Software Process Improvement

**Phases**
- Inception
- Elaboration
- Construction
- Transition
- Production
- Retirement

**Iterations**
- Initial
- Elab 1
- Elab 2
- Con 1
- Con 2
- Con 3
- Trn 1
- Trn 2
- Production
- Ret 1
- Ret 2
1. Enterprise Business Modeling

- To explore the business structure and processes of the enterprise
- Provides a common understanding of the business activities, customers, and suppliers of the business.
2. Portfolio Management

- Organizations often have suites of applications, called programs
  Can be better managed as a whole than as individual applications.
- This discipline enables you to track and plan your organization's entire software portfolio
  As well as individual programs within your overall portfolio
- Allows you to schedule and implement new requirements in a more strategic fashion
  This also helps avoid implementing the same functionality within different applications
3. Enterprise Architecture

- Addresses the overall architecture issues associated with your organization

- Consists of:
  - Models that define it
  - Prototypes and working models that demonstrate how it works
  - Frameworks that make it easier to use

- Helps ensure consistency across systems and greatly facilitates application architecture efforts
4. Strategic Reuse

- Promotes development and reuse of assets across projects
- To develop higher-quality applications more quickly by reusing assets instead of developing them anew each time
- To use artifacts that have already been tested and proven to work
5. People Management

- People exist within your organization, and you need to manage your staff and mediate the interactions between them and other people.

- The process of organizing, monitoring, coaching, and motivating people

- To ensure they work together well and successfully contribute to projects within the organization
6. Enterprise Administration
   Includes setting up and administering tools, processes, and facilities that are key infrastructure components of your IT organization

7. Software Process Improvement
   - Addresses the need to manage, improve, and support the multiple processes in use across your organization.
   - Remember: one process does not fit all

Now, let see some details of “The Enterprise Business Modeling discipline” for example
EUP: The Enterprise Business Modeling discipline

- Workflow:

  ![Workflow Diagram]

  - Define Enterprise Strategy
  - Periodically

  - Model Business Processes
  - Identify Process Implementation Options
  - Model the Domain
  - Model the Organization
  - Support Project Teams
Define Enterprise Strategy:
Model Business Processes:
Model Business Processes (contd.):

- Potential Business Process Modeling Artifacts:
  - Data flow diagram (DFD)
  - Integrated Computer-Aided Manufacturing Definition (IDEF0) diagram
  - Unified Modeling Language (UML) 2.0 Activity diagram
  - Use case model
  - Value stream map

+ Describing all of these artifacts with their trade-offs (pros./cons.)
EUP: The Enterprise Business Modeling discipline (contd.)

- Model the Domain:
... and so on!
The Zachman Framework has several strengths:

- It has been well accepted within the data community who considers it the defacto standard for enterprise architecture.
- The ZF defines the perspectives that your enterprise models should encompass, the implication being that you can apply its guidance within a process such as the EUP.
- The ZF explicitly communicates that enterprise modeling has several stakeholders, not just enterprise architects and developers, whom you should involve in your modeling efforts.
However, the ZF suffers from several weaknesses:

- Documentation-heavy approach. There are 36, each of which could be supported by one or more models.
- Process-heavy approach to development. You can instantly see the opportunity to define a collection of rigorous processes to support the ZF.
- Is not well accepted within the development community. Few developers even seem to have even heard about it.
- Seems to promote a top-down approach to development. This doesn’t have to be the case, you can in fact start in any cell and then iterate from there.
- Appears to be biased towards traditional, data-centric techniques.
## The Zachman Framework

<table>
<thead>
<tr>
<th>Scope (contextual)</th>
<th>Planner</th>
<th>Enterprise Model (conceptual)</th>
<th>Business Owner</th>
<th>System Model (logical)</th>
<th>Designer</th>
<th>Technology Model (physical)</th>
<th>Implementer</th>
<th>Detailed Representation (out-of-context)</th>
<th>Subcontractor</th>
<th>Functioning System</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What (Data)</strong></td>
<td>List of things important to the business</td>
<td>e.g. Semantic Model</td>
<td>e.g. Logical Data Model</td>
<td>e.g. Physical Data Model</td>
<td>e.g. Data Definition</td>
<td>e.g. Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>How (Function)</strong></td>
<td>List of processes that the business performs</td>
<td>e.g. Business Process Model</td>
<td>e.g. Application Architecture</td>
<td>e.g. System Design</td>
<td>e.g. Program</td>
<td>e.g. Function</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Where (Locations)</strong></td>
<td>List of locations in which the business operates</td>
<td>e.g. Business Logistics System</td>
<td>e.g. Distributed System Architecture</td>
<td>e.g. Technology Architecture</td>
<td>e.g. Network Architecture</td>
<td>e.g. Network</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Who (People)</strong></td>
<td>List of organizations important to the business</td>
<td>e.g. Workflow Model</td>
<td>e.g. Human Interface Architecture</td>
<td>e.g. Presentation Architecture</td>
<td>e.g. Security Architecture</td>
<td>e.g. Organization</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>When (Time)</strong></td>
<td>List of events/cycles important to the business</td>
<td>e.g. Master Schedule</td>
<td>e.g. Process Structure</td>
<td>e.g. Control Structure</td>
<td>e.g. Timing Definition</td>
<td>e.g. Schedule</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Why (Motivation)</strong></td>
<td>List of business goals/strategies</td>
<td>e.g. Business Plan</td>
<td>e.g. Business Rule Model</td>
<td>e.g. Rule Design</td>
<td>e.g. Rule Definition</td>
<td>e.g. Strategy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Extending the RUP/EUP with the Zachman

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise Business Modeling</td>
<td>Most significant business concepts (Enterprise glossary)</td>
<td>Enterprise business processes (Process model)</td>
<td>International view of locations (Location map)</td>
<td>Organizational strategy (Organization chart)</td>
<td>Business events and planning</td>
<td>Enterprise vision/mission</td>
<td>Corporate financials</td>
</tr>
<tr>
<td>Portfolio Management</td>
<td>List of systems and inter-relationships</td>
<td>Map business processes to systems</td>
<td>Map project teams to locations</td>
<td>Project team assignments</td>
<td>IT planning</td>
<td>IT vision</td>
<td>Savings from improved management</td>
</tr>
<tr>
<td>Enterprise Architecture</td>
<td>Domain architecture (UML component diagram)</td>
<td>Workflow architecture</td>
<td>Physical network architecture (UML deployment diagram)</td>
<td>Actual and potential interactions</td>
<td>Middleware and scheduling architectures</td>
<td>Enterprise technical requirements</td>
<td>Savings from common architecture(s)</td>
</tr>
<tr>
<td>Strategic Routa</td>
<td>Domain components</td>
<td>Functions (Web services, CICS transactions)</td>
<td>User interface components</td>
<td></td>
<td>Rulebase</td>
<td></td>
<td>Savings from reuse</td>
</tr>
<tr>
<td>People Management</td>
<td>Positions and relationships between positions</td>
<td>Roles played in each location and relationships between roles</td>
<td>Offices and relationships between them</td>
<td>Human resource philosophies and strategies</td>
<td>Annual reviews, project milestones</td>
<td>Career management strategies</td>
<td>Savings from improved team configurations</td>
</tr>
<tr>
<td>Enterprise Administration</td>
<td>Information assets (Corporate data sources, licenses, ...)</td>
<td>Guidance (Standards and guidelines)</td>
<td>Physical assets</td>
<td>Security policy</td>
<td></td>
<td></td>
<td>Savings from common platforms, guidance, and corp. licensing</td>
</tr>
<tr>
<td>Software Process Improvement</td>
<td>Software process definition</td>
<td>Span of the software process (e.g., divisional vs. global)</td>
<td>Software engineering process group (SEP3) mandate</td>
<td></td>
<td>IT department improvement goals</td>
<td></td>
<td>Savings from improved processes</td>
</tr>
</tbody>
</table>
### Extending the RUP/EUP with the Zachman (contd.)

<table>
<thead>
<tr>
<th>Core Development Disciplines</th>
<th>Business Modeling</th>
<th>Requirements</th>
<th>Analysis and Design</th>
<th>Implementation</th>
<th>Test</th>
<th>Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Most significant business concepts (Project glossary)</td>
<td>Domain model (CRC Cards, UML Class Model)</td>
<td>Structural design (UML Class Diagram, Physical Data Model)</td>
<td>Source code and data definition language (DDL)</td>
<td>Test suite</td>
<td>Installation packages</td>
</tr>
<tr>
<td></td>
<td>Project mission, strategies processes (Process model)</td>
<td>Usage of the system (Use cases)</td>
<td>Implementation design of domain classes/services</td>
<td>Source code and DDL</td>
<td>Tests</td>
<td>Installation scripts</td>
</tr>
<tr>
<td></td>
<td>Project view of locations (Location map)</td>
<td>Map of processes to location</td>
<td>Hardware, network, middleware</td>
<td>Hardware, network, middleware</td>
<td>Testing framework</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Affected positions (Organization chart)</td>
<td>User interface design</td>
<td>Implementation of user interface</td>
<td>Implementation of user interface</td>
<td>Test plan</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Business events</td>
<td>Scheduled events</td>
<td>Implementation of system triggers</td>
<td>Implementation of business rules</td>
<td>Test and defect tracking strategy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>System vision/mission</td>
<td>Business rules and “ilities”</td>
<td></td>
<td></td>
<td>Quality goals</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Savings from reengineered business processes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sharif University of Technology
Extending the RUP/EUP with the Zachman (contd.)

<table>
<thead>
<tr>
<th>Supporting Disciplines</th>
<th>Configuration and Change Management</th>
<th>Project Management</th>
<th>Environment</th>
<th>Operations &amp; Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration and Change Management</td>
<td>Configuration builds</td>
<td>Build scripts</td>
<td>CM repository</td>
<td>C&amp;CM plan</td>
</tr>
<tr>
<td>Project Management</td>
<td>Project task list (Gantt chart)</td>
<td>Project schedule (Gantt chart)</td>
<td>Team work area strategy</td>
<td>Staffing plan</td>
</tr>
<tr>
<td>Environment</td>
<td>List of required tools and guidance</td>
<td>Tool installation plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operations &amp; Support</td>
<td>Deployed classes, components, tables, ...</td>
<td>Deployed functions/operations</td>
<td>Deployed hardware, middleware, and software</td>
<td>Deployed user interface (including documentation)</td>
</tr>
</tbody>
</table>

Source:
* http://www.enterpriseunifiedprocess.com/essays/zachmanFramework.html
Changes to ZF in EUP-Zachman

Significant changes to ZF:

1. It replaced the six rows with the sixteen disciplines of the EUP, providing a more finely detailed collection of views.
   - The enterprise management columns are listed first then the project-level disciplines to remain consistent with the ZF’s top-down approach.
   - Unfortunately this approach could exacerbate many of the weaknesses of the ZF.
2. It introduced a cost column.
   The columns of the ZF reflect single word English interrogatives and that if John Zachman had spoken another language then the ZF may have looked different (Graeme Simsion 2004)

3. The question of “what”, is really a structural issue and not a data issue.
   It helps to remove the data-oriented bias of the ZF, making it clear that you have more options available to you than data-oriented artifacts
4. The cells describe the issues to address.
   - Each cell indicates the potential issue(s) to address, if any, as well as suggest a potential artifact to explore those issue(s).
   - Some cells are left blank because the column isn’t applicable to the discipline.
UML, RUP, and the Zachman Framework: Better together

Source:
The Zachman Framework weaknesses:
- In many organizations, it literally acts as a great poster that everyone admires, but no one uses!
- Zachman does not address how artifact *intake* should be handled
- Lack of a standard *notation*

RUP weakness:
- Main strength of being a *software development* process
  - people from other disciplines simply do not buy it as their core methodology
  - Project managers prefer: Project Management Book of Knowledge (PMBOK)
  - Enterprise architects prefer: Zachman

UML limitations:
- Process and framework independence of UML, limits its usefulness
- Many organizations use other structured methods and notations such as
  - Entity Relationship Diagram (ERD)
  - Business Process Modeling Notation (BPMN)
They have been created to address different problems within the same domain (information systems architecture).

The objectives set in front of them imply virtually no functional overlap.

UML, Zachman, and RUP can be used jointly to deliver better overall enterprise value.
RUP already makes use of UML

The Zachman Framework incorporates neither a notation nor a process, nor does it impose limitations on their selection

Situations where the two technologies can best be used are also different
- It would be questionable to use Zachman in a development environment by virtue of its inherent unsuitability
- Using RUP as the foundation for an enterprise information architecture would be equally unjustified

There are similarities between their respective lifecycle definitions
- Both RUP and Zachman are artifact-driven
- Inherit their structure from principles of architecture-centric design

Similarities imply that the two processes may be mutually enriching given the ever-growing complexity of enterprise systems and projects
**Application 1:** Consider UML for the role of common enterprise notation

**Context:**
- A management "call for action," to drive a "to-be" model for enterprise IT systems ➔ Full fill Zachman Requirements
- Architect: Pull together pieces of system knowledge scattered across corporate repositories
- Using existing artifacts: But mismatch raises

**Problem:**
- Artifacts are hybrids of at least several different aspects
- the artifacts are often created using **different notations**

**Solution:**
- UML may have a role to play in resolving this difficulty
Application 2: Use UML to bond Zachman and RUP

Context:
- The lack of a standard notation is quite understandable
  - Systems might have existed for a very long time, UML, for example, has been around for a mere decade
  - Shortage of the necessary skill sets

Problem:
- The need for UML in the organization
  - A RUP project
  - An enterprise architecture modernization effort

Solution:
- After UML has gotten traction in the organization through RUP-driven systems delivery projects, its use can be extended to enterprise architecture based on Zachman without running a risk of diminishing its appeal
Application 3: When starting from scratch, start with UML

- Efforts to introduce either RUP or Zachman to an organization are more likely to succeed when UML is already established.
- Can act as a point of departure into both RUP and Zachman
- Even if you later decide not to use RUP or Zachman, UML remains a helpful and well-understood visual language
Application 4: Learning RUP alongside Zachman (and vice versa) may be easier

- Understanding Zachman is, arguably, comparatively easier than understanding RUP and UML
  - Zachman deals with static views of enterprise systems architecture rather than dynamic models and processes
- The path toward learning Zachman may strongly benefit from the application of some core RUP principles such as
  - Requirements-driven
  - Architecture-centric
  - Model-driven
  - Iterative
Application 5: After the RUP project is over, Zachman may take over

- RUP Deployment discipline is rich in activities supporting migration of a developed system into production, but it scarcely covers activities surrounding the transition of architecture models created during a project.
- RUP lifecycle does not include the enterprise architecture discipline
- Zachman has a role to play here: as a framework for externalizing project architecture work
Artifact traceability between RUP to Zachman
Application 6: Zachman as an aid during RUP tailoring

Application 7: When planning a RUP project, look at the Zachman structure

Application 8: Use RUP with Zachman to help bridge the gap between enterprise and project architecture

...
UML, RUP, and the Zachman: Cross-Function Applicability (contd.)
RUP for other EA Frameworks and other Works
RUP for other EA Frameworks and other Works

- **TOGAF:**
  - “TOGAF or not TOGAF: Extending Enterprise Architecture beyond RUP”
    - Vitalie Temnenco, 2007

- **FEAF:**
    - A technical discussion of FEAF using RUP and UML
    - An article from IBM Software Group, 2003

- **Agile Methodologies**
  - Lots of works!
  - Another Presentation is on this topic

- **MDA and EA**
References and Resources


