

TOGAF usage in outsourcing of software development

Aziz Ahmad Rais¹, Rudolf Pecinovsky¹

¹ Department of Information Technologies, Faculty of Informatics and Statistics,
University of Economics, Prague
W. Churchill Sq. 4, 130 67 Prague 3, Czech Republic
arais@seznam.cz, rudolf@pecinovsky.cz

Abstract: TOGAF is an Enterprise Architecture framework that provides a method for developing Enterprise Architecture called architecture development method (ADM). The purpose of this paper is whether TOGAF ADM can be used for developing software application architecture. Because the software application architecture is one of the disciplines in application development life cycle, it is important to find out how the enterprise architecture development method can support the application architecture development. Having an open standard that can be used in the application architecture development could help in outsourcing of software development. If ADM could be used for software application architecture development, then we could consider its usability in outsourcing of software development.

Keywords: Outsourcing, Enterprise Architecture, TOGAF, Application Development, software application architecture

1 INTRODUCTION

The Enterprise Architecture (EA) framework is one of the standard ways that the organizations use to manage their IT to meet their organization's business strategy. **Enterprise architecture** "is a well-defined practice for conducting enterprise analysis, design, planning, and implementation, using a holistic approach at all times, for the successful development and execution of strategy. Enterprise Architecture applies architecture principles and practices to guide organizations through the business, information, process, and technology changes necessary to execute their strategies." [5]

Therefore it happens that many organizations require that it be possible to use EA development method for developing software application architecture (Application architecture) too. Application software is all the computer software that causes a computer to perform useful tasks beyond the running of the computer itself. A specific instance of such software is called a software application, application program, application or app.

Although there are many EA frameworks available, in this paper only TOGAF will be used for analysis, because TOGAF is one of the most frequently used framework by the organizations and is freely available. Because all EA frameworks should follow a holistic approach at all times, it will not make sense to analyse all of them.

In order to analyse ADM and decide whether ADM could be used for development of application architecture, it is important to define what is application architecture and application architecture method. Application architecture will be understood as defined in online article [11]: Architecture is the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.

There are different methods available on how to build application architecture from different views. For example: 4+1 view architecture method [7], model driven architecture (MDA) from OMG (<http://www.omg.org/mda/>). There are also some goodpractices on how to use design patterns for building application architecture, but they are not important for comparison with ADM. Both the 4+1 and MDA use object-oriented design. Object-oriented design is a method of design encompassing the process of object-oriented decomposition and a notation for depicting both logical and physical as well as static and dynamic models of the system under design [3].

In order to be able to compare both architecture types, it is important to write a few words about the software application development methodologies. Software development methodologies design is a discipline that proposes the architecture of the application software. A software development methodology is a process that divides the software development disciplines e.g. analysis, design, implementation testing, and maintenance into different phases. Some methodologies perform those disciplines in a waterfall way and some more in parallel. It is useful to know that the architecture discipline belongs to design discipline. The details of how a software development methodology performs iteration on the development of an application architecture and in which step of a software application development process the design of an application is performed are beyond the scope of this paper.

The architecture developed in the design phase is called application architecture. Rational Unified process (RUP) software application methodology is architecture-centric and design of architecture is done before coding the application [6]. In agile methodologies architecture design is developed sometimes by first writing a code and by refactoring the code until it meets the business need for example in extreme programming (XP) [2]. Thus RUP is top-down development and XP is bottom-up architecture development methodologies.

The application architecture development (AAD) that is part of the software development life cycle (software development process) is about to describe how the software application should be built. But in order to build AAD, if the target programming language is object oriented then the best practices is to use: object oriented principles, object oriented analysis (OOA) [8], and object oriented design (OOD) [3]**Chyba! Nenalezen zdroj odkazů.**

For analysis in this paper it is important to describe in short the TOGAF EA life cycle. The TOGAF EA life cycle has eleven phases; for this article the following phases are important:

- [1] Phase B: Business Architecture,
- [2] Phase C: Information Systems Architectures - Application Architecture,
- [3] Phase C: Information Systems Architectures - Data Architecture,
- [4] Phase D: Technology Architecture,

Furthermore both types of architecture methods use **business requirements** to achieve their objective. For describing the business both methods use UML diagrams or UML extension diagrams e.g. process diagram. This common practice causes confusion without knowing the details of the UML diagrams.

In the other part of both methods use case diagrams are used and not considering the granularity of the use cases in the diagrams is again confusing. Furthermore there are some similarities to other diagrams and disciplines e.g. deployment view and infrastructure view that are used in both architecture development methods.

The data architecture of ADM uses entity relation diagrams that are used also by application architecture method to identify its data domain (sometimes it is called physical entities)

All the above similarities initiate the idea whether ADM could be used for application architecture development. The difference between both methods is in what objective both methods achieve, and whom the outputs of both methods are for.

The table below describes the summary of the process of the development of both types of architecture. The software application architecture development process implies from object oriented analysis and design process from the 4+1 view point of view. The EA development process details are described by TOGAF ADM.

Enterprise architecture	Application architecture
ADM	Object-oriented design
<p>Business architecture:</p> <ol style="list-style-type: none"> 1. Business strategy 2. Business process using UML activity diagram UML extension diagram: business process diagram 3. Organization unit: actors and roles 	<p>Business requirements analysis:</p> <ol style="list-style-type: none"> 1. Business requirement 2. Use UML activity diagram and UML extension diagram: business process diagram. 3. Actors list
<p>Application architecture</p> <ol style="list-style-type: none"> 1. Application Communication diagram 2. Application and User Location diagram 3. Enterprise Manageability diagram 4. Process/Application Realization diagram 5. Application Migration diagram 6. Software Distribution diagram 7. Software Engineering diagram 8. Application Use-Case diagram 	<p>Design:</p> <ol style="list-style-type: none"> 1. Component diagram (logical view) 2. Interaction or sequence diagram (process view) 3. Use case diagram 4. Package diagram, class diagrams (implementation view) 5. Technology view (framework, components, libraries) 6. Integration view 7. Component diagram (Deployment view)
<p>Data architecture</p> <ol style="list-style-type: none"> 1. Data Management 2. Data Migration 3. Data Governance 	<p>Problem domain:</p> <ol style="list-style-type: none"> 1. Business domain 2. Object domain 3. Data domain
<p>Technology architecture</p> <ol style="list-style-type: none"> 1. Environments and Locations diagram 2. Platform Decomposition diagram 3. Processing diagram 4. Networked Computing/Hardware diagram 5. Communications Engineering diagram 	<p>Support /infrastructure view</p> <ol style="list-style-type: none"> 1. Hardware diagram 2. Network diagram

Table 1: Table describes ADM and object oriented design methods steps to develop EA and application architecture. [Source: authors]

2 ANALYSIS OF DIFFERENCES BETWEEN ADM AND AAD

Some differences could be clear from detailed examples or detailed analysis of both methods. But describing all the details would be a lot of duplication work that is clear from the documentation and theory.

2.1 BUSINESS

This chapter will briefly highlight how every method uses the term “business” in the business architecture and business requirements. In the ADM method business is analyzed to develop business architecture. What does business architecture means and how is it different from the result of business analysis in AAD? The difference could be in many places but for this analysis we will concentrate on what objectives are going to be achieved by the specific principle or step of the method.

The objective of business architecture is how the business goal, and business strategy, technology strategy and vision architecture of an enterprise is met. The other objective of it is to develop a roadmap if there is GAP in a vision, a strategy and business architecture. As the strategy and vision architecture is an important discipline for IT governance and according to COBIT [4] this is responsibility of executive management of an organization, so the business architecture is target for executive management of the organization.

The subject of business analysis is to analyze the business process and geographical aspect of organization from organizational perspective. On other hand the objectives of business analysis in AAD is to identify details of business and business process that needs to be implemented by a software application. The objective is to satisfy the stakeholder’s needs and is not an organization, business, or technology strategy. These business requirements are detailed-level information that supports the enterprise business needs.

To summarize, the business architecture describes business on enterprise level for strategic needs of enterprise. The enterprise strategic goals and enterprise architecture are supported by detailed business processes that are automated and implemented by software application.

2.2 DESIGN

According to the ADM development life cycle, application architecture and data architecture compose system architecture. The application architecture has to follow the objective of system architecture. The objective of system architecture is similar to business architecture. In short targeted users of this architecture are also executive management of an enterprise.

The **Application Architecture** provides a blueprint for the individual application systems to be deployed, their interactions, and their relationships to the core business processes of the organization. [9] As mentioned in the introduction chapter, application architecture is one of the phases of ADM life cycle. Like every other phase in ADM this phase also has some activities that should be performed in order to build the application architecture in EA. These activities are: defining objectives, describing an approach, defining necessary inputs and outputs and steps to take for generation of outputs. For

comparison and analysis of ADM and AAD is enough to analyse the activities inputs, outputs and steps of application architecture in ADM.

Every aspect of the Application architecture development has many items. Here only some aspects will be described to make clear the differences between ADM and AAD.

2.2.1 INPUTS

The inputs are draft architecture requirements specification that contains a gap analysis results (from Business Architecture and Data Architecture, if available) and relevant technical requirements that will apply to this phase. Another input is business and data architecture components of an Architecture Roadmap. Besides others, Organizational Model for Enterprise Architecture describes budget requirements and governance and support strategy, Scope of organizations impacted.

The AAD inputs can be detailed business requirements, business process and their models and list of actors and stakeholders. Furthermore, to compare the above inputs with one another the application architecture development method requires low level business details and the Enterprise architecture high level and strategic requirements. This means that the subject of both architecture is different and AAD could be seen a subset of ADM.

2.2.2 OUTPUTS

In the ADM one of the outputs of the application architecture process is draft architecture definition document that includes baseline application architecture, target application architecture. The target application architecture contains the following models:

1. Process systems model
2. Place systems model
3. Time systems model
4. People systems model

Other output is an update of draft architecture requirements specification and diagram mentioned in the table 1.

In the AAD the output of the application architecture development is mostly only one document - that is, architecture document - that contains the details how to implement each business process, use case and domain model. This output is used by the developers to know how to implement the software application. Comparing that result with ADM application architecture is quite different and the output of ADM would be very difficult for a developer to follow and to know what the application software to implement and how to implement.

But in the ADM application development output there is good information about what services an application provides, what business process an application provides without details (e.g. how the process is implemented). There is only information that makes executive to judge how the strategy is met and what GAP in the business exist.

2.2.3 STEPS

Bellow is the list of steps according to AMD method that need to be taken to produce the output.

1. Select Reference Models, Viewpoints, and Tools
2. Develop Baseline Application Architecture Description
3. Develop Target Application Architecture Description
4. Perform Gap Analysis
5. Define Candidate Roadmap Components
6. Resolve Impacts Across the Architecture Landscape
7. Conduct Formal Stakeholder Review
8. Finalize the Application Architecture
9. Create Architecture Definition Document

In AAD there is a need to identify use cases and use case description. After identifying the use cases there is need for identification of the logical components and relation between them. The next step is that the components would be grouped in the packages and it can be determined how to deploy them.

Comparing steps of both methods, we can see that they are too far from each other, but some of the steps in ADM can be useful like GAP analysis. The GAP analysis is good when there is a need to build software application on third party solution or packages, in which case it will be required to do a GAP analysis of the components. But the GAP analysis in ADM is about to find out GAP between strategy, objective and software application service. In such case even though some step can be used but the subject and objective of the steps are different.

2.3 DATA

The data architecture process of ADM is similar to the application architecture step of ADM, because both architectures belong to the same step of the ADM life cycle.

So from the input, output and step perspective the results of comparing data architecture to the problem domain of AAD would be the same as comparing the application architecture development process of ADM with the design of application architecture in AAD. Here other aspect of data architecture and compared with design in AAD would be analyzed.

The key concern of data architecture in ADM is data management, data migration, data governance. Data management concern is how and where data entities are used, created, transported or reported in business architecture. How the data could be used by customers and suppliers.

In the data migration, this concern happens only if information system architecture is going to be changed. The data governance is the concern of the executive of an organization and it is regarded to resources, skills, structure in the organization. The data governance provides a management system to manage the data governance aspects.

The concerns of the problem domain in AAD are to identify the business entities of business domain and to find a proper object model that can meet business domain requirements. Another concern of the

problem domain is to identify physical entities to store and manage business data based on an object domain and a business domain.

The concerns of AAD are again related only to the problem domain, but data architecture solves and provides strategies for the problem domain of the whole information system in an organization.

2.4 TECHNOLOGY

Based on the above analysis the objective and concerns of technology architecture in ADM is high level and it solves and provides a platform picture of whole infrastructure of IT in the organization. But the application level infrastructure view is only to identify where physically the software application runs. It does not solve IT infrastructure issues of the organization.

3 CONCLUSION

The conclusion is whether TOGAF ADM can be used to as method developing application architecture of software. The answer is no, it cannot be used as a method for developing application architecture of a software. Hence TOGAF ADM cannot be used as a method for software architecture development in outsourcing software development.

This implies also that EA is architecture for capturing the IT picture on an organizational level and using any EA framework for developing application software architecture is not suitable, because of the level of details that are captured by EA. We can conclude that software application architecture development is one of the pillars of EA.

Besides ADM, The TOGAF framework provides, reference model, capability framework content framework, ADM Guidelines and Techniques, Enterprise Continuum and Tools. Furthermore, the content framework of enterprise architecture framework (TOGAF) provides very useful information that can play crucial role in outsourcing. TOGAF reference model can be used to reference the application architecture developed with outsourcing.

Agile development methodologies are by nature iterative. Solving strategic issues in outsourcing of software development project iterations would not be so effective and would delay the software delivery. That is why all strategic and base line architecture, architecture foundation and common architecture best practices of TOGAF can help in increasing the quality of the software application architecture proposed by supplier. Some strategies can be even used as an acceptance criterion and could be used during QA processes. Thus, TOGAF or any other EA framework is useful in outsourcing.

4 REFERENCES

- [1] ArchiMate 2.1 [Online]. *The Open Group*. available on: <http://www.opengroup.org/subjectareas/enterprise/archimate>
- [2] BECK, Kent. *Extreme Programming Explained*, First Edition, 1999, ISBN: 0201616416.

- [3] BOOCH Grady, et al. *Object-Oriented Analysis and Design with Applications*. Third Edition. April 2007, Addison-Wesley. ISBN 0-201-89551-X.
- [4] COBIT 4.1. *IT Governance Institute*, ISBN 1-933284-72-2
- [5] Federation of EA Professional Organizations. Common Perspectives on Enterprise Architecture. *Architecture and Governance Magazine* 9 (4), November 2013.
- [6] KRUCHTEN, Philippe. *Rational Unified Process: An Introduction*, Third Edition, Addison Wesley, 2003, ISBN: 0-321-19770-4.
- [7] KRUCHTEN, Philippe. Architectural Blueprints—The “4+1” View Model of Software Architecture. Paper published in *IEEE Software* 12 (6) November 1995, pp. 42-50
- [8] McLAUGHLIN, Brett, POLLICE Gary, WEST David. *Object-Oriented Analysis and Design*, O'Reilly Media, 2006, ISBN: 978-0-596-00867-3.
- [9] *Open Group Standard TOGAF Version 9.1*, Document Number: G116, ISBN: 9789087536794.
- [10] STEHLÍK, Hynek. Vizuální jazyk Archimate pro modelování Enterprise architektury na příkladu integrace ekosystému tabletu [online]. *Acta Informatica Pragensia* 2 (1), pp. 57-69. [2013-12-12]. available on: <http://aip.vse.cz/index.php/aip/article/view/29/18>
- [11] What is a software architecture? [online]. *IBM*. [2013-11-15]. available on: <http://www.ibm.com/developerworks/rational/library/feb06/eeles/>