The Relationship Between AIC and The Quality Product

Lina khalid Ahmed

Software Engineering Researcher, Amman, Jordan
lenaalhafed@gmail.com

Abstract - *one of the main goals of software architecture is to build a software product with high quality and then improve this product according to the quality and the business goals of the product. Architecture influence cycle (AIC) has the impact on building any product with high quality by introducing the factors that influence software architecture and those that are influenced by software architecture then presenting the overall cycle that affects the goal of the product. Architects need to know the basis, the nature as well as the priority of these influences as early in the cycle as possible.*

Keywords: *Software Architecture, Quality Attribute, Business Goal, Architecture Influence Cycle, Architecture Significant Requirements, Architectural Design Decisions.*

1 Introduction

Software architecture is a set of structures that are built for a reason, which comprise software elements, relations, and properties of both. All software systems are built to satisfy the goals of organizations (business goals), so the software architecture is described as a bridge between those business goals and the final resulting system. The good thing is that software architecture is built upon known techniques that ensure achieving these business goals [1].

Business goals and qualities are the basis on which software architecture is built, so the main goal of the architect is to identify the requirements that affect the quality of the business goals which ultimately affect the overall structure of the system. We must classify quality attributes as the system properties, and they are separate from the functionality of the system.

This paper introduces the relationship between the architecture influence cycle and building a product with high quality by introducing factors that affect the architecture as well as factors affected by it. It also describes the meaning of Architecture Significant Requirements (ASR) and their role in this cycle.

This paper is classified as follows: Section 2 describes the related works and includes all the authors who have worked in this area. Section 3 defines AIC with all the details of the factors that influence and others that are influenced by software architecture. Section 4 defines relationships between software architecture, business goals and qualities. Section 5 describes the results which present how AIC affects software architecture in building high quality products. Here, Architectural Significant Requirement (ASR) is defined and its role in the cycle is presented.

2 Related works

Many researchers work on Architectural design and explain how to achieve high quality with a variety of techniques.
Many papers also apply the effects of certain factors on case studies. [2] Presents both practical and theoretical benefits from a case study using Architectural Business Cycle (ABC) to understand how to manage software architecture in automotive manufacturing. It represents the role of ABC in modifying the environment and defining the context of the interviewer. So both the theoretical framework and the Interview methodology that used in this case should be possible to be used for studies at other organizations.
A comparison between software architectures of five industries is done in [3] and from the comparison, authors extracted a general software architecture design approach. Moreover, this paper finds an ideal pattern and from this pattern, the author derives an evaluation that can be used for further method comparisons.
Enterprise Architecture (EA) is one type of software architecture which describes the business structures and it is the most important technology that is being used today in business organizations. [4] Defines and describes the fundamental Business Rule Life Cycle (BRLC) by integrating Enterprise Architecture with Enterprise Decision Management (EDM). It starts by defining the business rule and its qualities then shows its function. Another study is [5], it presents how architectural design decisions affect achieving the goal that the software is built upon. This is done through a design fragment concept and the quality control on these fragments.
This paper presents the relationship between AIC and producing a system with high quality.

3 Architecture Influence Cycle

There are factors that influence building architectures and these factors have an effect on the architect and ultimately on the entire system. This is called an Architecture Influence Cycle [1]. From this definition we conclude that the architecture itself has some factors that influence it as well as other factors that are influenced by.

3.1 Factors influencing software architectures

The architect is influenced by several factors when building the architecture of the system, including:

- **Stakeholders**: a stakeholder is anyone who has a stake in the success of the system. The problem is that each stakeholder has his own goal and concern when building the system. Documenting the requirements and capturing the system's qualities is beneficial for the stakeholder. Here, the role of the architect is to capture all the requirements that do not explicitly appear and referee all the conflicts that frequently appear. Early engagement of stakeholders allows the architect to understand the constraints of the task, manage expectations, negotiate priorities, and make tradeoffs. So the most valuable advice that should be given to the architect is: *know your stakeholder* [1].

- **The development of business environment**: the architecture of the software can be formed by the business/mission concerns such as: time to market, use of legacy system, cost, plans for long-term infrastructure and many other factors that influence the organization, so the architect must know these factors [6].

- **The technical environment:** here, the technical environment means all the technical parts that exist at the time the architecture is to be designed. These technical parts are what determine the new technology to be assembled and added to the infrastructure and that is why this particular factor is always changing. Technical parts include: patterns and styles, social networking, and aspect–oriented programming [6].

- **Professional background and experience of the architecture:** architects always make choices according to their past experience that is why architects must have certain knowledge and skills and this is why an architect's choices might be influenced by experience and training [1, 6].

3.2 Factors influenced by software architectures

Once the architecture is completed and the system is built, both will affect the technical environment, the business goals and social experience.

The architecture also affects the requirements of the stakeholder by giving the customer the chance to receive a more reliable system and certainly with a fewer defects so the requirement must have a way of being negotiated rather than just being an order.

All in all, it must be noted that the architect is affected by the architecture he built. If the architect built the architecture for any system and it worked, he will repeat it again on the future. On the other hand, he will avoid it if it failed. This shows that the architecture will affect the experience and knowledge of the architect and in some cases it will affect the technical environment that is used to improve the system.

That is what is called the cycle of influence. All the factors that affect the architecture when building the system get affected by the architecture when the system is finally built. Figure 1 represents the Architecture Influence Cycle (AIC).

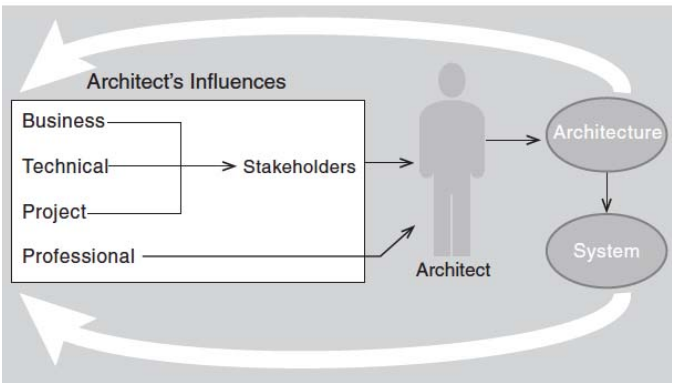

Figure 1. Architecture Influence Cycle as defined by Kazman et al. [1]

To summarize, architecture is more than the technical environment and the functional or non functional requirements. All factors that were described earlier must work as a unit that affects the architecture and the system architect must be aware of that in order to be a competent architect.

The two principles in building any software architecture are [7]:

- The quality attributes that drive the software architecture
- Architecture centric activities, a method of software architectural design. The main goal of it is helping the software development team to build the architecture of the entire system in an iterative way through its life cycle.

4 Software architecture, Business Goal and Requirements

Software architecture or any solution of it cannot be designed well without the understanding of the business goals, qualities and the relationship between them, which makes the relationship a very critical part in designing any system.

4.1 Software architecture and requirements

The requirements for any software system come in different forms, but all these forms encompass the following types [1]:

- Functional requirements: these requirements state what the system should do, and how to react at runtime events.
- Quality attribute requirements: this type of requirements determines the qualification of the functional requirements of the overall system.
- Constraints, which are the architect's design decisions such as the specific programming language that is used through building the system.

According to these types of requirements, the architecture has its response to each type of them as follows:

- Functional requirements are satisfied by assigning responsibilities throughout the design or to specific architectural elements. Assigning responsibilities means building the early design decisions.

- Quality attribute requirements are satisfied by various structures designed to the system, and the behaviors and interactions that populate that structure.

- The constraints are satisfied by accepting the design decisions.

The architectural design is affected by three types of requirements described in figure 2.

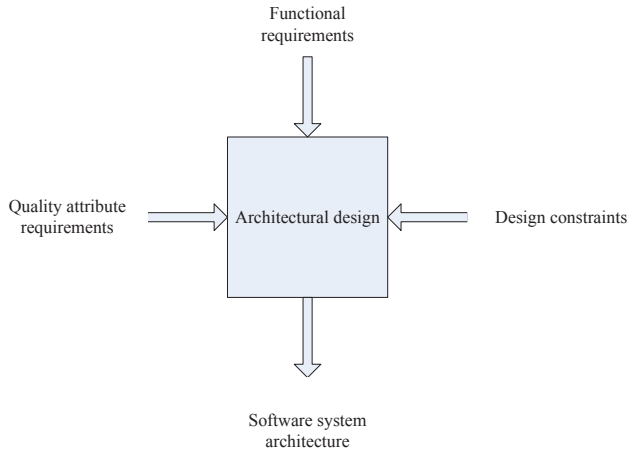

Figure 2. Influences of requirements on Architectural design.

The main point is, structuring the architecture needs these three types of requirements, not just the functionality requirements.

4.2 Business goals and software architecture

Business goals shape the architecture. The architect makes sure that the system architecture has high performance, reliability and security which are suitable to the business goals that are built upon them [8].
The architect needs to know the system's business goals because these goals are what determine the quality attribute requirements of the system and that helps architect to design the architectural decisions of the system. This will be defined later on in this paper. Knowledge of business goals enables the architects to know the tradeoffs and discard the requirements that are useless for the system.
Each organization has its own business goals for the system under development. Ideally, the system will satisfy the business goals and it is the responsibility of the architect to design a system that is able to do that.
Building a design based on business goals is done through a method that is used to generate, prioritize and refine the quality attribute scenarios before the software architecture is completed. This is what is called QAW (Quality Attribute Workshop). The scenarios in this method help us to better describe quality attributes [1].
The stakeholders are connected to the QAW early in the life cycle in order for the stakeholders to discover quality attributes. This method increases the communication between stakeholders, clarifies quality attributes and decides the early design decisions which are applied to the system. Design decision concepts play an important role in designing any software architecture and they are applied on the architecture design processes. For instance, the main decisions are: the main approach for structuring the system, the strategy that is going to be used to control the system, the style that is going to be used to build the software architecture, how to evaluate the architecture, and how it should be documented. [6].

4.3 Business goals and qualities

Business goals are quality attributes the system is expected to achieve such as cost, schedule, and time-to-market.
For example, time-to-market can be achieved by reusing some elements or deploying a subset of the system, while the goal cost can be achieved through having a budget for the development effort which must not be exceeded.
The idea is that different architectures will need different development costs. For instance, an architecture that needs technology that does not exist in the organization will be more costly than one that takes advantage of assets already in house. An architecture that is highly flexible will typically be more costly to build than one that is strict [9].

5 RESULT

The factors that affect the entire system and its quality are decrbied in previous section. This section describes the relationship between AIC and quality attribute (which is the result of this work). Figure 3 concludes this relation; it describes how the architect receives these factors and operates on them to achieve high quality products.

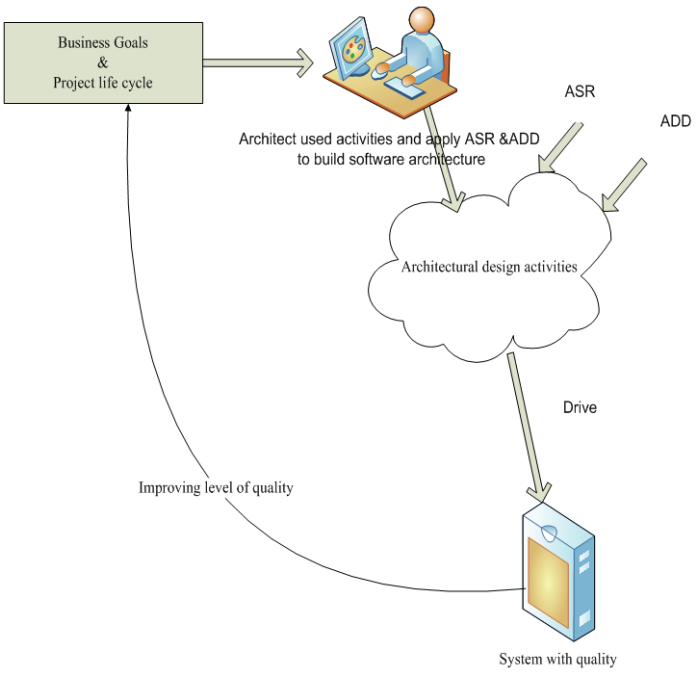

Figure 3. The relationship between AIC and quality attributes

Regardless of the methodology he uses to build the system, the architect should know the basic goals of the system, understand ASR (Architecture Significant Requirement), and choose the important design decisions to put them into the process of building the architecture.

Important definitions that appear in this cycle are:

ASR and Architecture Design Decisions (ADD).

ASR can be defined as a set of requirements for a software system which drive the architectural design; that are why they are significant. This means that these requirements have a deep effect on the architecture of the system. There are many techniques for gathering requirements from the stakeholders; for example, object oriented analysis uses use-case and its scenarios [10, 1]. Furthermore, ASR can be gathered by understanding the business goals because quality attribute requirements can often be derived from business goals

Any technique used to extract ASR should be helpful to record them in one place so that that can be reviewed, referenced, used to justify design decisions, and revisited over time if changes are to be made on the system [1].

ADD can be defined as a set of decisions and considered alternatives that directly influence the design of software architecture. The important thing is that these decisions are achieved through set of reputations which ultimately affect building a system of high quality.

ADD is positioned in the first step of architectural activities, but these decisions are modified through these activities until they are appropriate.

These two important concepts are used by the architect through architectural activities. These activities, which are shown in figure 4, include:

- Architectural analysis

This is the first step of the cycle which defines the problem of the system and models the user's requirements for the system to do.

- Architectural synthesis

This step is the core of the cycle; it processes the ASR and finds new requirements that influence the architecture. The main advantage of this step is moving the requirements from the problem space to the solution space.

- Architecture evaluation

This activity ensures that the architecture is according to the ASR that are proposed and that it certainly achieves the specific goal the system is built upon. This must be the basic part of every development methodology because it has many advantages of which the most important is to validate the functional and quality attribute requirements. Another important advantage is to improve the architecture. One important technique that is used to evaluate the architecture is ATAM (Architecture Tradeoff Analysis Method); it is the best known scenario-based evaluation method. The main purpose of this technique is to assess the consequences of architectural decisions according to the business goals and quality attribute requirements. This paper did not go in details with this technique; first because the evaluation is not the core and second because there are variety of techniques for performing architecture evaluation, and each has a different cost and provides different information. However, according to this paper, ATAM is the most appropriate technique.

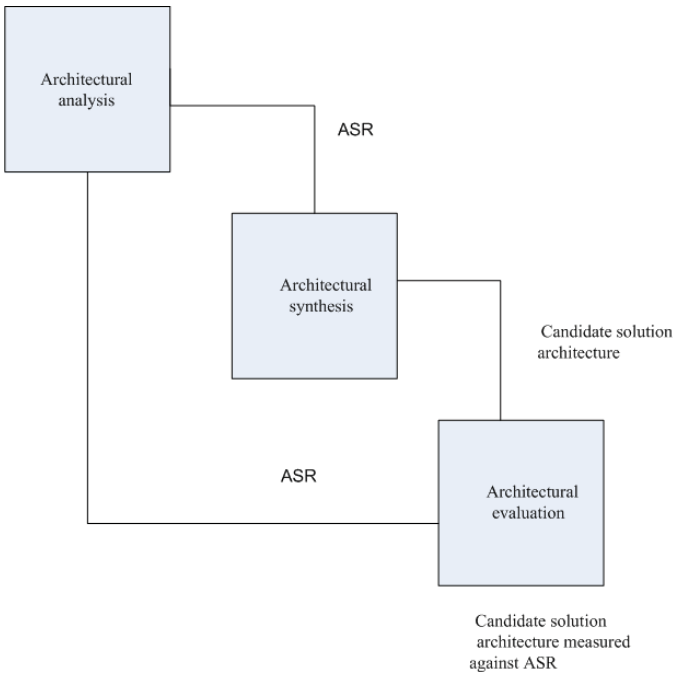

Figure 4. Architectural design activities

Figure 4 shows that the ASRs go through architectural evaluation in which they are evaluated against candidate solution architectures.

After completing the architectural process, the architect can build a system according to its architecture and this system will certainly be a high quality product. If the architect need to develop the capabilities of the system, he can go back to the business goal and the methodology and in some cases he judges the technical environment to have a new system environment.

The relationship between AIC and the quality attributes appears through building the architecture and through improving the capabilities of the system.

6 Conclusion

The idea of this paper is building a software system with high qualities. This work shows that the factors that influence the architect and their relationship with producing systems with high qualities needs to introduced and defined. Moreover, it shows that sometimes we need to go back through cycle to improve the quality and that achieves the goal of the system. That is the reason of changing the environment of technical support or negotiating with stakeholders about the requirements of the system.

7 References

[1] Bass L., Clements P., Kazman R., "Software Architecture in practice", 3rd edition, Addison-Wesley, 2013.

[2] Eklund U,Olsson C," A Case Study of the Architecture Business Cycle for an In-Vehicle Software Architecture,"IEEE,2009.

[3] Christine H., Philippe K. b, Robert L., Nord, Henk O, Alexander R., Pierre A., "A general model of software architecture design derived from five industrial approaches," The Journal of Systems and Software 80, PP: 106–126, 2007.

[4] Tortolero A," Enterprise Architecture and the Business Rules Life Cycle," Innovations Software Technology, 2008.

[5] Khaled L., Achieving Goals through Architectural Design Decisions, and Journal of Computer Science (JCS) .ISSN:1549-3636, PP: 142-1429, 2010.

[6] Software Architecture: Principles and practices. An on line training, SEI, Carnegie Mellon university, 2016.

[7] Nortrop L.,"Architecture Business Cycle Ensuring Product Qualities,"SEI,Carnige Mellon University,2004.

[8] Khaled L., Driving Architectural Design through Business Goals, International Journal of Computer Science and Information Technology.Vol.8. No.3. ISSN: 1947-5500,2010, PP:68-71.

[9] http://etutorials.org/Programming/Softwarearchitecture in practise 2nd edition.

[10] Khaled L., Architectural Design Activities for JAS, International Journal of Computer Science and Information Technology, Vol.6. No.2, ISSN: 1947-5500, 2009, PP: 194-198.