

Chapter 1

Introduction

Chapter 1 Introduction

1.1 Problem Domain

For the past of many decades, software developers and software designers have been taught to build systems based on statements of the technical system requirements. In the practice, requirements produce the design and eventually beget the system. Design of the system is originated from product technical requirements.

A modern software system is characterized by its changing nature. The ever-changing environment includes computation that may be distributed over a network of heterogeneous machines and components, where tasks can migrate at runtime; connections between software components that can evolve in time and space; hardware platforms offer vastly different functionalities and performance; software environments provide applications with changing services and et cetera. This calls for software architecture to adapt to these changes.

Architecture has emerged as a vital part of the design process. There are some considerations of software architecture to become the realization of early design decisions based on decomposing system into sub parts. Software architecture serves as an important communication, reasoning, analysis and tool for systems.

Software architecture deals with the structure of large software systems. It is a result of business, technical and social influences. Thus, the existence of software architecture affects the business, technical and social environments that eventually influence future architectures. Hence, the architecture becomes engraved in the structure of the development project.

In business context, project management has become the part and parcel in the development of a software system. In the past, the projects involved are straightforward.

However, in today world, the projects become increasingly complex. It is rapidly become a special need in management and a standard way of doing business.

Project management is relatively becoming modern. It is characterised by new methods of restructuring management and adapting special management techniques in order to obtain better control and utilization of existing resources. The emphasis is on development of an integrated project management that focuses all project effort toward the strategic plan of the organization and reinforces mastery of both the project management tools or techniques and interpersonal skills necessary to orchestrate successful project completion.

In other words, the design world of a system development is separated from the on-going organization's strategic goal. The gap that occurs between these contexts may hinder the success of a project as whole. Software architecture is limited to apply to the design of the system where its usage does not contribute to the organization objective.

1.2 Problematic Issue in Definition

Software architecture is a young discipline in software engineering where by there has been no accepted definition of the term. In practice, this lack of an engraved definition will not hinder us from utilizing the concept of software architecture.

The study of software architecture has evolved by observation in terms of design principles during when real systems are developed. In other words, it is an attempt to abstract the commonalties inherent in system design where it must account for other wide range activities such as concepts, methods, approaches and results. A few most often-heard definitions are as follow [2]:

- ❑ ***Architecture is components, connectors and constraints.*** Constraints are defined as the behavior of the components, which are consistent. However, the definition is still lacking of notion of externally visible properties, where it is essential in producing a useful architectural description.
- ❑ ***Architecture is components and connectors.*** Connectors imply a runtime mechanism for transferring control and data within a system. It concentrate on the runtime architectural structures and does not map well to the non-runtime structures.
- ❑ ***Architecture is the overall structure of the system.*** This common architecture implies that systems have more than one structure. It is usually some structure that we mean reflect the overall of the system that is developed.
- ❑ ***Architecture is high-level design.*** There are other tasks associated with early design that are not architectural. It is known that design and architecture are not interchangeable. For instance, the interface to those data structures is decidedly an architectural interface but their actual interface does not reflect the architectural design.
- ❑ ***Architecture is the structure of the components of a program or system.*** The interrelationships, principles and guidelines are governing their design and evolution over time. This is one of a number of process-centered definitions that consists ancillary information. Most people demand that architecture include a statement of stakeholder needs and a rationale for explaining how those needs are met. Any system has an architecture that can be discovered and analyzed independently from any discovery of process by which the architecture was designed, evolved or intended to satisfy requirements stated.

1.3 Objectives

The primary objective of this project paper is to review the possibility of merging the design world in terms of software architecture into the business world of the project management. This includes the understanding of notation of software architecture in predicting the allocation of existing resources in a project.

Project management visual notation is designed to support the architectural design for the conceptual view in project management. The objective here is by using the architectural design of software engineering, an architectural-assisted project management notation will be conceived prior to creating the work breakdown structure. This visual notation will be the fundamental guideline to project manager in terms of planning the schedules, tasks, dependencies and constraints.

In addition, the objective also includes the impact of risk and cost on delivering the excellent result of a successful project. By forecasting the risk involved, the affect of the risk can be identified and thus predict its impact on the project. In terms of cost control or budgetary control, the planning and control system that is selected must be able to satisfy management's needs and requirements in order to accurately project the status toward objective realization.

Finally, from the research conducted, an architectural-assisted project management will be drafted out in order to demonstrate the planning of project resources based on the structure of software architecture that goes with the organization goal.

1.4 Contents of Report

Chapter 1

This chapter draws the introduction of this project paper, the problem domains persist in this area of research and the objectives of the project.

Chapter 2

Under this chapter is the Literature Review where studies had been carried out to recognize the traditional approach in software development, how the software architecture brings the software development into another dimension and the different styles of software architecture. Besides, in the project management discussion, it also explains different approaches in project management, project life cycle, risk identification, quantification and management, cost estimation and finally life cycle costing.

Chapter 3

This chapter will bring readers to the architectural analysis of a software system. It will cover the architectural requirements, the functionality and quality attributes that contribute to the design of software architecture. Software architecture analysis method is used to carry out the analysis method based on the scenario development method. Later, the software architecture will be assessed.

Chapter 4

Upon completion of the architectural analysis, architectural design will follow. This chapter describes the process of designing, the design method, which will be based on functionality and architecture transformation. Later, it will demonstrate different visual notation, which are derived from Unified Modeling Language (UML) at different views (conceptual, module, execution, code).

Chapter 5

This chapter will implement the visual notation, which are described in previous chapters to be incorporated during the project initialization and implementation. Software architecture will come into the traditional flow of project management. This chapter explains how software architecture brings changes to the current project management and how it is implemented by the architectural design of visual notation. In addition to that, it also describes how software architecture is utilized to analyze risk involved in estimating the milestone of the project.

Chapter 6

This chapter is the final chapter of the research paper. The conclusion includes the future of software architecture, project management and the future role of software architect and project manager. Besides, there is also a future enhancement provided and comments from the industry on this research. An overall conclusion will describe how the software architecture contributes to the current advantage of software architecture to the software engineering area.