Universiti Teknologi MARA

Customization on Software Documentation Standard for 3-Tier Software Architecture for the Web-based System

Azlina Binti Awi

Thesis submitted in fulfillment of the requirements for
Bachelor of Science (Hons) Information System Engineering
Faculty of Information Technology And Quantitative Science

OCTOBER 2004

DECLARATION

I certify that this thesis and the research to which it refers are the product of my own work and that any ideas or quotation from the work of other people, published or otherwise are fully acknowledged in accordance with the standard referring practices of the discipline

OCTOBER 09, 2004

AZLINA BINTI AWI

2002328479

ACKNOWLEDGEMENT

Praise to Allah S.W.T. the Almighty for giving me the strength to complete the research successfully and for showering me a good experienced throughout this study period.

My heartiest gratitude and gratefulness to my supervisor, Puan Zarina binti Zainol whose keenness, patience and trust in the research and wise supervision she benefited me greatly.

I would also like to thank Dr. Noor Habibah binti Hj. Arshad as Thesis Coordinator for giving me valuable insight and guidance that has helped me enormously in the completion of this thesis.

Thanks also to my beloved family who always pray for my success in my studies. Lastly, to all my friends thanks for your helps and supports for me in completing this research project.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

CSCI	-	Computer Software Configuration Item
IEEE	-	Institute of Electrical and Electronics Engineers
IRS	-	Interface Requirement Specification
MIL	-	Military
NASA	-	National Aeronautics and Space Administration
SRS	-	Software Requirement Specification
STD	-	Software Testing Document

ABSTRACT

Software documentation standard is designed to support the documentation of software system development. One type of software documentation standard is the Software Requirement Specification (SRS) that provide as a basis in documenting software requirements during the requirement analysis phase of the software development life cycle. Complete SRS gives a lot of benefits for the developers in developing the software product such as reduce on-going costs, complete specification of the system, avoid later problem during the other development phases and other problems that can cause the big problem to occur and the most important is to avoid the software project to be failed. Three layers that contain in the 3-tier architecture are Presentation Layer, Business Layer and Data Layer that connected directly with the database. The separation of different layers with its specific role and function causes the 3-tier software architecture is the perfect choice to be implemented with the web-based system. Develop a 3-tier web-based system also required the preparation of a complete and detail explanation of the functionality and non-functionality of the 3-tier web-based system in the SRS. There is no specific standard of SRS that specific for the 3-tier web-based system. The main aim of the research is to come out with the SRS standard for 3-tier software architecture for the web-based system. Methodologies used in this research are observation and theory building. From the findings, it can be summarize that SRS plays an important role in the software development and 3-tier web-based system is a perfect choice in developing a web-based system. SRS documentation for the 3-tier web-based system can serve as a guideline to document the requirements in a specific way.

CHAPTER 1

INTRODUCTION

This chapter provides the Research Background, Problem, Objectives, Significance, Approach and Methodology, Limitations and Overview of the research project.

1.1 Research Background

Software documentation standard is designed to support the documentation of the software developed for a big or small project. The main reason for the standard is to provide a guideline or framework for the developer to record the information needed for them in the development of the software and also for the maintenance of the software itself.

The Institute of Electrical and Electronics Engineers (IEEE), Military department (MIL), National Aeronautics and Space Administration (NASA) and other institutions have developed their own standard to be followed in order to produce software system. These standards already followed as a guideline by all developers around the world to develop their own software product. The standards are complete and basically developers only have to follow the guideline and add or remove any items or information that they think suitable for their system.

3-tier software architecture normally known is the new architecture for system development. For web-based system that use database server, 3-tier architecture is a suitable application for the software system. As a new approach, there are no specific standards that have been provided for the development of software system in the kind of 3-tier software architecture. Therefore, from the research, it is hope that one guideline can be produce to be use as a standard for a particular purpose for 3-tier software architecture software system. The guideline is hopefully can be use by the students of Faculty of Information and Technology (FTMSK) in doing their web-based software system project.

1.2 Problem

In developing software, documentation that involved in every step of software development is important to ensure that the software produced satisfied with the requirements needed by the users. Software documentation standard is designed to support the documentation of the software and its goal is to provide a guideline and model for recording the essential information needed throughout the development life cycle and maintenance of software. Software documentation prepared by the developer is important as a tool in gathering information from the users, designing the interface of the software and also in designing the software itself.

Nowadays, 3-tier software architecture becomes famous because of its ability to provide better performance compared to 1-tier and 2-tier software architecture. 3-tier software architecture overcomes the limitations of 1-tier and 2-tier software architecture especially in the web-based system. The limitations include the performance deteriorates if number of users is more than hundred, restricted flexibility, choices of database and also limited functionality in using program

functionality across servers. In the context of 3-tier software architecture for the web-based system, there is no specific documentation standard for the developer in designing their software project. With the supporting documentation standard provided by the Institute of Electrical and Electronics Engineers (IEEE) and Military (MIL), the format for Software Requirement Specification (SRS) will be produced to be followed by the developer in designing 3-tier software architecture for the web-based system.

1.3 Objectives

1.3.1 To identify the characteristics and importance of software documentation standard in the scope of Software Requirement Specification (SRS).

1.3.2 To identify the characteristics and benefits of 3-tier software architecture applied for the web-based system.

1.3.3 To develop a guideline of Software Requirement Specification (SRS) for 3-tier software architecture for the web-based system.

1.4 Significance

1.4.1 Clear understanding about the characteristics and importance of Software Requirement Specification (SRS).

1.4.2 Clear understanding about the characteristics and benefits of 3-tier software architecture for the web-based system.

1.4.3 SRS guideline can be as guidance for the developers especially students in documenting the entire requirement for software system exceptional for 3-tier web-based system.

1.5 Approach and Methodology

In preparing and doing the research, it required additional information gather from various kind of sources. Information will be taken from sources like journals and articles from Internet and also from books. For this research, it also required detail study and information about 3-tier software architecture, web-based system and also current standards from IEEE and MIL. Detail study about 3-tier software architecture and web-based system is important because 3-tier software architecture for the web-based system is different with the other systems like real time system, embedded system and any other systems.

Software Requirement Specification (SRS) produced for 3-tier software architecture for the web-based system will have a bit different with the current SRS for other system. Recommended SRS guidelines from IEEE (IEEE STD-830-1998) and MIL (MIL-STD-498) is only as guidance for the researcher in doing the research. Data collection from reviewing the 3-tier software architecture, web-based system and current standard from IEEE and MIL involved in the research will be analyzed to produce the result, which is the format or guideline for SRS in developing 3-tier software architecture for the web-based system.

1.6 Limitations

Software documentation standard consist of several documentation types like Software Requirement Specification (SRS), Software Testing Document (STD), Interface Requirement Specification (IRS) and else. For this research, software documentation standard will focus on Software Requirement Specification (SRS)

for 3-tier software architecture for the web-based system. Commonly, SRS is also known as user requirement specification or functional specification. An SRS include all the requirement needed in system software development because successfulness of the project mostly depend on a complete requirement specification that will be use as a main reference for the developer to go further in system development.

Requirement gathering involve in the early stage in software development phase. All of the data and information gather must be recorded in proper way as guidance for the next phase and SRS can be referred as the parent document because such documents like design specifications, software testing and also documentation plans are related to SRS. To achieve a high quality of a software product, it is important to ensure that SRS was developed in a systematic and comprehensive way to make sure that the software meets user's requirements.

SRS allows the customers or users to verify the requirement of the system for analyst to understand the problem and produce the software system. SRS later can be as a reference for the architect, system analyst or programmer to develop software. Complete and detail description of a system behavior is essential because from the SRS, system architect can know how to design the system. Thus, whether the software system is big or small, SRS is having a great effect to the system because SRS is a required foundation before the development of the software system.

1.7 Overview

The research final report consists of six chapters inclusive the Chapter 1 for the Introduction. All of the chapters comprise of different contents for the research. All of the six chapters can be described as below:

Chapter 1: Introduction

Describes the Research Background, Problem, Objectives, Significance, Approach and Methodology, Limitations and Overview of the research

Chapter 2: Literature Review

Describes the Literature Review include the Introduction, Software Requirement Specification (SRS), 3-tier architecture or technology, Benefits of 3-tier architecture and 3-tier architecture for web-based system and the Conclusion of the research.

Chapter 3: Research Approach and Methodology

Describes the approaches and methodologies used in the research and consists of four phases; Research Planning, Data Collection, Data Analysis and Develop Guideline with the Introduction and Conclusion

Chapter 4: Discussion of Findings

Describes the findings from the research with the sub-topics; Characteristics and Importance of Software Requirement Specification (SRS), Characteristics and benefits of 3-tier

software architecture for the web-based system, Software Requirement Specification (SRS) guideline for 3-tier software architecture for the web-based system and Benefits from Findings with the Introduction and Conclusion

Chapter 5: Conclusion and Recommendations

Describes the Conclusion and Recommendations from the research

CHAPTER 2

LITERATURE REVIEW

This chapter contains the literature research and review by the researcher about the topic that has been selected. Sources for the literature review are from the two SRS standards from IEEE and MIL, journals, articles, books and other related sources.

Software documentation standard consists of different types of documentation to be prepared depend on the types of the software system and as a documentation product for certain phases during the development cycle of the software system. The first step in any software developmental effort is to determine exactly what the software system shall do. Software Requirements Specification (SRS) act as a guideline for develop the software product satisfied with the customers perception.

2.1 Software Requirement Specification (SRS)

There has been little if any research examining about Software Requirement Specification (SRS), but mostly researchers and practitioners have commented that SRS having great authority or influence to the development of the software that contribute to the successfulness of the software.

"The SRS is a specification for a particular software product, program, or set of programs that performs certain functions in a specific environment. The SRS may be written by one or more representatives of the supplier, one or more

representatives of the customer, or by both." The SRS plays a specific role in the software development process and also will state the interfaces between the system and its software portion (The Institute of Electrical and Electronics Engineers IEEE, 1998). According to the Military MIL (1999), all of the requirements for a Computer Software Configuration Item (CSCI) and the methods to be used in the software development can be specified by the SRS to ensure that each requirement of the software has been met. Apart of that, SRS also is used as the basis for design and qualification testing of a CSCI. Tuffley (2000) defined that SRS consists of detail explanation about the system to be proposed and enable a development team to use the SRS for the design and development phase. SRS has business and technical considerations which the customer may or may not be able to provide by them. Kim and Sheldon (2004b) stated that software development starts from specifying the requirements and SRS takes the core role as the documentation for development phase in order to avoid problems in later development phase and reduce life cycle costs by preparing a complete specification.

IEEE (1998) noted that SRS should address the *"functionality, external interfaces, performance, attributes and design constraints imposed on an implementation"* of the software product. IEEE also defines the benefits of a good SRS; *"Establish the basis for agreement between the customers and the suppliers on what the software product is to do, reduce the development effort, provide a basis for estimating costs and schedules, provide a baseline for validation and verification, facilitate transfer and Serve as a basis for enhancement"*. Again from the IEEE, *"correct, unambiguous, complete, consistent, ranked for importance and/or stability, verifiable, modifiable and traceable"* defined as the characteristics of a great SRS.

Japenga (2003) commented that IEEE is a good and excellent source for definitions of SRS for the designer as the basis for their SRS in developing a system. MIL-STD-498 measured the requirement specification other than requirements trace ability for the system quality (Ramesh, 1998). IEEE and other standards issuing organizations have identified nine topics that used to be addressed when specifying software requirement specification that are interfaces, functional capabilities, performance levels and data structures/elements that comprise in Engineering Requirements and other five topics in Quality Requirements that are safety, reliability, security/privacy, quality and the last is constraints and limitations (Vie, 2004). Requirements analysis is important phase for the developers in gathering the information to develop new system with clear understanding about the current system work and to develop a new to overcome the problems of the current system. Functional requirements describe the functionality of the system or expected functions that the system must do while non-functional requirements describe aspects that contribute with the good implementation of functional requirements by the system including performance, security and other related information (Bennet et al, 2002).

The need to understand new features and different about such system become important including web based system. The development of the web system required the developers to develop appropriate strategies to ensure that the systems are developed in a complete and professional manner. These include the preparation of complete requirements for the new system. It is important to ensure that SRS is developed with detail explanation and should describe all of the required functions of the software to be addressed. Gellersen and Gaedke (1999) stated that important aspect of web-based development projects is requirement analysis to design the structure of the web-based system. The chances for software failure in a software development project can be reduced by preparing a complete requirement specification document. Spending enough time

for requirement phase and follow the guidelines of requirement specification document can help developers to produce a good and complete requirement specification document. (Wakeham, 2000).Documenting a complete and reliable requirements specification is important as to avoid problem in the data development phase and to reduce life-cycle costs. It is also to ensure that majority of software errors can be introduced during the requirement phase. (Kim & Sheldon, 2002a).

A Standish Group Report identified that incomplete requirements is the number-one factor contribute to cancelled projects and clear requirements as the third most important factor for successful project development. (Crowston & Kammerer, 1998).The completeness of specification and system behavior for the system is important for the implementation of the system and also to have a complete specification of the system as guidance in the development of the system (Leveson, 1995). Conflict about software requirement and what software must produce happened between users and software engineers can cause the project being cancelled before it completes (Kreitzberg, 1997). Heimdahl, Leveson and Nancy (1996) noted that consistency in requirements specification means that the specification is free from conflicting requirements and undesired no determinism.

SRS consists all of the requirements required for the system include functional and non-functional specification of the system. Complete SRS can give the users and the developers better understanding about the system, avoid problem during the other phases, reduce costs and the most important thing is complete documentation of SRS can avoid the software to be failed. Thus, it is important for the developers to know all the characteristics of the SRS and how to prepare a complete and detail SRS to be followed when designing a system.

2.2 The 3-Tier Architecture or Technology

The three tier (3-tier) software architecture emerged in 1990s to overcome the limitations of the two tier architecture. The third tier that is middle tier was added between the user interface (client) and the data management (server) components. 3-tier architecture application overcomes the limitations of 2-tier architecture, and most of the practitioners think that it is worthwhile to develop a system with 3-tier architecture because of its usability and importance.

The basic concept of a tiered architecture involves breaking up an application into logical tiers that each of tiers is assigned general or specific roles and most common approach used for web application today is 3-tier architecture because for simple web applications, 3-tier is really sufficient (Petersenfam, 2003). 3-tier software architecture comprised of code that can be broken down into 3 distinct areas with Presentation logic which contains User Interface, displaying data to the user and accepting input from the user. Second is Business logic which has Data Validation, ensuring the data is valid before being added to the database and last is Data Access Logic contain Database Communication, accessing all of the information needed by the system.(Marston, 2002).

"The 3-tier architecture is considered to be a second generation of client server architecture because it is extended from traditional 2-tier architecture. The 3-tier architecture adds an application server as a middleware tier between client and database servers. The middleware tier is a separate piece of software, typically running on a separate piece of hardware that performs most of the application logic such as enforcing business rules and performing complex processing. When an application is broken into user interface, application logic and database management, it is 3-tier architecture" (Fong & Hui, 1999). In 3-

tier architecture, most of the processing is done on an application server in conjunction with a web server. The application server downloads relevant data or information from a back-end database and publishes in a web server (Willet, 1996a). Leon (1997) described that the completion of 3-tier architecture required application server as a middle tier that holds the business and application logic to support complex web applications, back end with database server and browser or thin client as front end tier. Basic common architecture for 3-tier software architecture shown by the Figure 2.1 below:

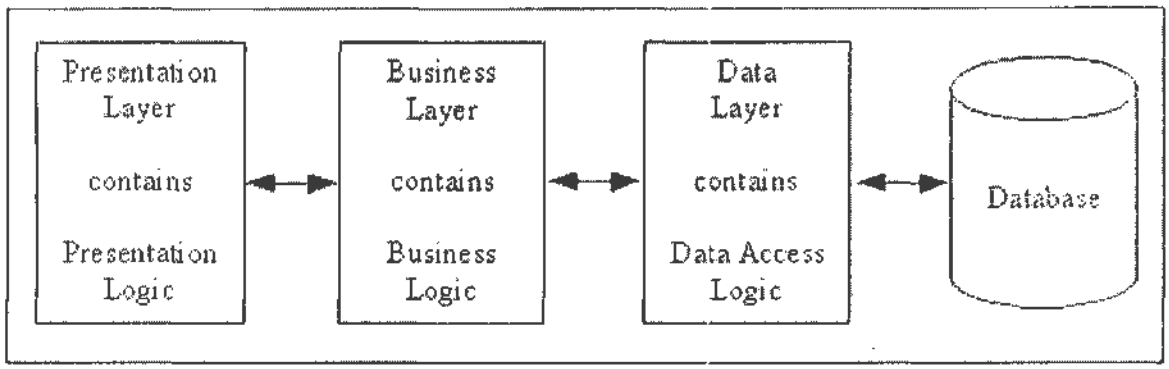

Figure 2.1: 3-Tier Software Architecture (Adapted from Marston, 2002)

2.3 Benefits of 3-Tier Architecture

Many researches have been done by the researchers and practitioners to describe the benefits of applying the 3-tier architecture. Other than to solve the problem occurred by the 2-tier architecture, 3-tier architecture addressed many benefits not only for the developers but also for the users.

2-tier architecture is easy to maintain, simple and performs well. But two-tiered design involve with security exposures. 3-tier mitigates the problem by moving the business logic tier to a middle tier behind another firewall and cause the system to be more secured (Chiu, 2001). Lithium (1996) described that application server added to client/server created a 3-tier architecture cause the end result for architecture with significantly more scalable than traditional two-tier client/server and also provides central application control. Orfali (1998) described that 3-tier architecture provides developers with more choices for enhancement of application scalability, performance, reliability, and security because it allows application components to run on middle-tier servers, independent of presentation tier servers and database servers.

The use of different sources of database can be done without modification of application logic and business rules and many types of database can be use instead of only certain type of database by using 3-tier architecture because this architecture can work best in a heterogeneous database environment. This is because implementing 3-tier application using a database gateway as a middleware server can cause the application logic and business rules achieve database independence (Fong & Hui, 1999). To support scalability and concurrent generation of multiple data analysis applications in a heterogeneous and distributed computing environment, system can be integrated with 3-tier client/server architecture as each tier has their own functions (Penix, 2000).

Benefits of separating applications into distinct pieces can be stated that it enables parallel development of the different tiers of the application, allows for easier maintenance and support, since it's easier to change and upgrade a single specific component than to make changes for the whole application, and to produce more throughputs from the system (Press, 2001). In 3-tier architecture,

middle tier or application server can centralized process logic that makes administration and change management easier by localizing system functionality. Any changes can be done once and placed on middle tier server and available throughout the systems compared with other architectural designs (Eckerson & Wayne, 1995).

Platform independence, universal access to data, ease of application development and deployment and also cost effectiveness are the main strengths and benefits of the 3-tier architecture (Wang & Apicella, 1998). Biggs (1999) stated that tangible benefits of multi-tier application architecture include making application creation, easier system modification and maintenance, improve business process and also by moving business logic to middle tier cause less expensive cost. Levin (1998) noted that 3-tier software architecture promised to lower IT costs, made more easy application development and also speed deployments by centralized business logic on a server.

3-tier architecture possible to transfer a legacy system to another system and can be done with low risk and cost effective. Developer can maintain the old database and process management rules so that the old and new systems will run side by side until each application and data element is moved to the new design. Main advantages of 3-tier architecture compared to the two tier architecture are; increased performance, flexibility, maintainability, reusability, and scalability while hiding the complexity of deploying and supporting underlying services and network communications (Sadoski & Dorda, 2000). Huskisson & Mulcahy (1999) had founded upon the changed of the Computerized Maintenance Management System (CMMS) to a three tier system gave the benefits like improvement of integration with other systems, easy to maintain programming code and not have to be re-written, more convenience to users and also flexible