

A Review to handle Sustainability problems in Software

Razia Falik
Department of Computer Science,
University of Agriculture, Faisalabad, Pakistan

Abstract- One of the major challenge of our society was to achieve sustainability development. Sustainability contains three factor: social, environmental and economics sustainability. For individuals, sustainability was the ability to undergo and the probable for long lasting maintenance. So there was a need to develop a sustainable software with a liable feeding of resources. In software development process, a pattern was a written document that provides a general solution to a design problem that occurs repeatedly in many projects. Requirement engineering was considered one of the most important phases in the development life cycle. Requirement engineering was the crucial activity which can affect the entire life cycle of software development process. The main objective of the requirements elicitation phase was to collect requirements from different views such as requirements from the business, requirements from the customer side, requirements from the user side, and requirements from the security point of view. This research will explore to handle the sustainability problem in design, a new software pattern based on singleton and service locator design pattern will be defined. This research will also focus on requirement engineering techniques that will be affective to overcome these problems. These techniques will be focus groups, interviews and ethnography for eliciting the requirements.

Key Words: Sustainability, Software, Design Patterns

INTRODUCTION

For solving the usually taking place software design problems in software engineering, design pattern was a suitable tool. It was not the easy task to search an appropriate design pattern for solving software design Problem for inexperience developers, therefore it was required to search a design pattern. At present, to regain design Patten, there are several proposed tools for research. Inappropriately, there was a keyword search problem for theses research tools. This problem was solved by using the acquaintance experience. To solve this problem CBR was most suitable model. Furthermore, to refine problem, FCA was useful for maintaining indexes, thus both CBR and FCA will be used consecutively [7].

In the process of software development, Requirement elicitation and analysis are the most important phase. Requirement engineering (RE) includes numerous activities which are requirements elicitation, analysis, negotiation, specification and validation. Frequent studies have shown the importance of requirements engineering process for successful software projects. There was a clear relationship among requirement gathering and analysis and software quality (Tahir and Ahmad, 2010).

For the solution of commonly occurred software problems, design patterns are the best solution that are used by the software developers. These are offered at the domain independent level. In this fashion, consequently design patterns are applicable through several domains and at the multiple domains of the construct. However, there was a difficulty to adopt the design patterns due to generic nature so it was not always easy that how these can be made functional in that domain. In their research, they discourse this problem by concentrating on an approach that was to gather requirements that was specific to that domain [4].

Sustainability was very important for software engineers due to two reasons. One was the need to learn and teach in exact domain that how we can use the resources in an efficient manner. And the other was want to take an advantage of developing markets to attract industrial investors and partners for research collaborations. Therefore, we contend that in software engineering, sustainability was an important future topic. Sustainability was the capability and prospective for long lasting preservation [9].

This research provided a new design pattern that has been based on singleton and service locator design pattern methodology so that to solve the problems in designing the software. This research has focused on requirement engineering techniques that will be affective to overcome these problems. These techniques could be focus groups, interviews and ethnography for eliciting the requirements.

REVIEW OF LITERATURE

[2] mentioned that from several years the typical process for requirements gathering has not been changed for large projects. Functional requirements for the software process can be defined by analysts and software developers. When the requirements are gathered in the natural language so there can be misunderstanding between requirements and the first release of the software. So there are different non-functional requirements that can be defined by different software developers at different spaces of the world.

[13] stated that the software development that was considered debauched and precise, Software design patterns are good proposals. They are defined in terms of interfaces and classes, promising an anticipated functionality. It was necessary to implement a software design pattern at correct pattern level. Conventional testing does not expose mistakes from the desired pattern. The proposed solution was to measure the software design pattern behaviors by verification at run-time that ensure that they meet the specified standards.

[5] mentioned that sustainability was the major issue in software engineering field. They also stated that requirements engineers must add sustainability for achieving quality requirements checklist. It requires the same importance as the other quality requirements. Sustainability was becoming more important in the software projects. So it was necessary to use the tool in the requirements process to achieve sustainability. An environment specialist must cooperate with requirement engineers who will help to mitigate the environmental threats.

[6] said that one of the main reasons of the software project failure was incomplete and incorrect requirements so in the software development process, requirements engineering was considered the most important activity and software project success depend on it. One of the main intentions of requirement engineering process was to describe requirements that exactly meet user's need. Software Many researchers and practitioners highlighted the need for choosing the appropriate techniques and models during the software process. For effective requirements analysis, it was essential to apply the proper technique for a given problem. On the base of project characteristics, it was also necessary to use blend of RE techniques. Advantage of using theses blend of techniques was that other techniques can be used with a special technique that was used for problem solving. By doing this high quality requirements will be achieved in software developments.

[14] mentioned that when errors are detected at the implementation or testing phases, it was extremely difficult and expensive to remove them, so requirements engineering phases are time costing. In the past two decades, several problems are found in order to gather requirements in software engineering situations so experts have discovered a large number of substitute approaches such as interviews, group-meetings, goals, questionnaires, viewpoints, aspects and scenarios. Requirements engineering was basically an interdisciplinary issue that are large number of contextual, Social, Psychological, human, economic, political, functional and educational factors that may generally employ more or less impacts on the requirements elicitation and management processes. Project managers and requirements analysts face some common problems during requirements engineering activities in the software development phase.

[10] mentioned that by using agents, any task or sub task can be carried out, so the goal of each agent was obviously placed in any software application. They define that an agent was a portion of code that achieve a definite goal; whether this goal was a single task or collection of tasks. Requirements engineering was the field of software engineering in which the software to be developed achieved the desired objectives for a particular software. RE was an important phase within software engineering, meanwhile success of software systems can be measured by satisfying their requirements. Most of the present software engineering approaches have

focused on the design of the software system and then paid less consideration to requirements engineering that was the major cause of software failure.

[12] stated RE was the key factor that measures the importance of product. A well-defined requirement elicitation reduces the time and cost of development and increases the quality of the software system. Therefore, to degree and additionally identify the current methods and difficulties that are challenged by the software developers, it was most significant to design well-defined RE methods. So it was considered a positive thing for software developers to define good requirements engineering.

[1] stated that RE was the process of gathering, analyzing, trimming, documenting, and authenticating the needs, and requirements of the stakeholders for the desired system. They also stated that improper RE can lead to system failure, by using appropriate RE approach, software quality can be improved. There are some features of ES that are same as in the requirements of basic projects that are application domain, types of requirements engineers, resources of information, involvement of users and requirements possessions. In this research, their main goal was to enhance the quality of ES by offering appropriate RE approaches that are used in the development process of ES. The use of these appropriate approaches can reduce the cost, time and scope and avoid the effort to rework in the development process of ES.

[4] discussed that from software architectural design patterns, domain specific software architectures can be build. For building the software that uses the specific domain from architectural patterns was the main purpose of this research. By taking the advantage of software design patterns, this methodology increases the quality of DRE software architecture. To plot a design pattern by using a single domain specific feature and to discourse a wide variety of architectures, this approach was a flexible. Moreover, the engineers' time was saved by using executable design pattern templates when building software architectures, for performing design time validation on the software architecture produced they also provide the foundation using this approach.

[8] stated that in the software development process, requirements engineering was the most important phase. Its objective was to gather quality requirements from numerous stakeholders by using suitable approaches. It's one of the objective was to gather quality requirements from numerous sources for the whole software development process. Requirement analysis was achieved by information and system developers in the early stage before the designing, development and delivery process irrespective of the technique that was used. Contextual inquiry was one of the most beneficial approach for information requirements analysis which was established in the Digital Equipment Cooperation as a research technique that was used to gather and examine the user requirements for design of the product in detail.

[9] mentioned that they presented a concept sustainability into software engineering. In the typical quality characteristic, Sustainability was one of the important quality attribute as identical to correctness and efficiency. They said that there are two reasons why sustainability was important for software engineering educators: First, need to teach and learn how we can use the resources in an efficient manner in a specific domain. Second, need to attract the students, industrial investors and partners to work collaboratively. Therefore we can say that sustainability was an important topic in the field of software engineering.

[15] mentioned that for designing object oriented systems, use of design patterns was best practice. Many experienced developers used design patterns for their design problem and consider it as a solution of the problem. Many developers focuses on the identification and documentation of patterns instead of their experiences about using the patterns because design patterns are generated from the experiences of the software developers. Design patterns provide a good tool to develop design for specific problem, so it was essential for design pattern to be used by unexperienced developers.

[11] stated that software systems have the sound effects on our environment and its sustainability that we need to talk. We estimate that there are two basic motives in the process of dominant software's development Firstly was that when we take sustainability in the process of software develop, there must be an intelligent blockade that was overwhelmed that was a difficult task. Secondly, this difficulty can increase the cost of the software development process, particularly when we want to hire experts for the

measurement of sustainability. For inspiring software developers to measuring sustainability, we proposed a pattern for the requirements that are to be sustainable. These patterns offer direction to measure the requirements for achieving sustainability. These requirements consists material on the conditions in which we should use these patterns taking as a preliminary edge for the development of sustainable requirements and it also contains data that was required to development of these requirements.

[3] mentioned that many organizations faced problems for achieving sustainability in their business. Although, these companies need extra energy to incorporate sustainability factor in their industry. Because in the process of old business and also in the old software development processes finding sustainability factor was a difficult task. In the traditional business processes, there was a no clear definition of sustainability. Because it was not considered sustainability an important factor in all the phases of software life cycle. So, if we want to consider sustainability a most important factor we have to understand sustainability in the software development process.

REFERENCES:

- [1] Ang, J.K., Leong, S. B., Lee, C. F. and Yusof, U.K. 2011. Requirement Engineering Techniques in Developing Expert Systems. *IEEE sunoisium on Computers & Informatics (ISCI)*,1(1):640-645.
- [2] Berenbach, B and M. Gall. 2006. Toward a Unified Model for Requirements Engineering. *IEEE International Conference on Global Software Engineering (ICGSE'06)*,1(1):237-238.
- [3] Betz, S and T. Caporale. 2014 Sustainable Software System Engineering. *IEEE Fourth International Conference on Big Data and Cloud Computing*,1(1):612-619.
- [4] Fant, J. S., 2011. Building Domain specific Software Architectures from Software Architectural Design Patterns. *Proceeding of the 33rd International Conference on Computers & Informatics (ICSE)*, 1(1):1152-1154.
- [5] Mahaux, M., P. Heymans and G. Saval. 2011. Discovering Sustainability Requirements: An Experience Report. *In 17th International Working Conference on Requirements Engineering: Foundation for Software Quality*,6606(1):19-33.
- [6] Mishra, D., A. Mishra and A. Yazici. 2008. Successful Requirement Elicitation by Combining Requirement Engineering Techniques. *In proceeding of the First International Conference on the application of Digital Information and Web Technologies*,1(1):258-263.
- [7] Muangon, W and S. Intakosum . 2009. Adaptation of Design Pattern Retrieval Using CBR and FCA. *IEEE Fourth International Conference on Computer Sciences and Convergence Information Technology*,1(1):1196-1200.
- [8] Pandey, D. ,U.Suman and K. A. Ramani, 2011. An Approach to Information Requirement Engineering. *IEEE International Conference on Information Science and Application (ICISA)*,1(1):1-4.
- [9] Penzenstadler, B and A.Fleischmann. 2011. Teach Sustainability in Software Engineering?. *In 24th IEEE-CS Conference on Software Engineering Education and Training (CSEET)*,1(1):454-458.
- [10] Ranganathan, P and K. Magel. 2010. Understanding Requirement Engineering (REQ) from a Software Agent Modeling Perspective. *IEEE International Conference on Software Engineering and Service Science (ICSESS)*,1(1):83-85.
- [11] Roher, K and D. Richardson. 2013. Sustainability Requirement Patterns. *IEEE third International Workshop on Requirement Patterns (RePa)*,1(1):8-11.
- [12] Tahir, A and R. Ahmad. 2010. Requirement Engineering Practices – an Empirical Study. *IEEE International Conference on Computational Intelligence and Software Engineering (CiSE)*,1(1):1-5.
- [13] Teplitsky, M and I. Exman. 2006. Measuring Behavioral Software Design Patterns. *IEEE 24th Convention of Electrical and Electronics Engineers*,1(1):8-11.

- [14] Yang, Y., F. Xia., W. Zhang., X. Xiao., Y. Li and X. Li. 2008. Towards Semantic Requirement Engineering. *IEEE International Workshop on Semantic Computing and Systems*,1(1):67-71.
- [15] Zhang, C and D. Budgen. 2012. What Do You Know about the Effectiveness of Software Design Patterns?. *IEEE Transactions on Software Engineering*,38(5):1213-1231

IJERGS