IOSR Journal of Research & Method in Education (IOSR-JRME)
e-ISSN: 2320–7388,p-ISSN: 2320–737X Volume 6, Issue 5 Ver. VI (Sep. - Oct. 2016), PP 64-67
www.iosrjournals.org

Software Esolvability in Software Industry: A New Quality Factor

Gurdeep S Hura,
Professor, Department of Mathematics and Computer Science, University of Maryland Eastern Shore, Princess Anne, MD 21853

Abstract: *There are many aspects of software quality that needs to be addressed during the modern software design process. One of the most important of those aspects in the software quality is software evolvability. Software evolvability can be defined as level of easiness with which a software system can be modified to meet the future requirements. It may be regarded as the future potential of enhancing the quality of software system. Many studies in number of software industry investigated the relation between particular software metrics and effort on evolving individual classes, but little attention has been given to assessing and measuring evolvability of the entire software systems. This paper mainly focuses on the attributes of software evolvability that can help us to measure the evolvabilty of the software system. A survey template is being developed to find the significance hierarchy of the properties of software evolvability.*

I. Introduction

Evolvability is one of important quality attributes of a software system as it shows its available interfaces for future potential, and cost of ownership. The evolvability of a system may be important for a software client in the process of acquiring the system. Unfortunately clients do not have any mechanism of assessing the software evolvability. It is rather difficult to identify the assessment of evolvable software as many of the software designers have not established the components of software evolvability and also the measuring metrics. There are many few studies on methods for assessing the evolvability of modern software systems; most research is on the area of software evolution that may be related to software evolvability [2]

Recent year have seen an ever growing effort to identify and define the characteristics of software evolvability to achieve goals for modern software system like; measurable attributes, set the priorities of different characteristics of evolvability, knowledge and experience of expert who assess software evolvability etc. A literature review clearly indicates that not enough work has been done in the area of status of evolvability of software system. In order to address these goals, the paper develops a questionnaire and survey that will be sent to software engineers working at different levels at various software industries. The data which has been collected through these questionnaire and survey are assembled, analyzed and evaluated.

In both questionnaire and survey, the following two basic questions determine the attributes of software evolvability

A. What are the primary characteristics which have main importance for the evolvability of system software and what experience software developers suggest for prioritizing these characteristics?

B. What are the minimum qualifications of an individual who will conduct the assessment of the evolvability of software system during expert and critical assessment?

These questions are interlinked with each other as each of these questions supports each other. During the expert assessment of primary characteristics of evolvability of software system on the basis of his qualification, and characteristics of evolvability of software system, the collected data are analyzed and prioritized on the basis of knowledge which an assessment expert may know which verifies the interlinking of the selected two main questions. The first question covers the characteristics of evolvability of system software which are analyzed and prioritized by the assessment expert through his/her knowledge and the knowledge of the expert is encompassed in his/her qualifications are included and discussed in second question.

The paper is organized as follows. Section 2 describes the current status of software evolvability and methods for assessing the software evolvability. Section 3 presents results of the survey conducted for software evolvability. Section 4 concludes and suggests directions for future work.

II. Software Evolvability

In the recent years, Information Technology (IT) and Software Industries have encountered the problem of evolving software system in all their software projects. A software development system should be evolved with new technologies in order to compete with the similar product in the market. To stay on the market, software development systems should inherit new requirements, reusability so that it could adapt to the changes occurring in the environment. However, current marketing trends do not allow industries to work on the newly emerged requirements for a long time period. That trend and practice has focused in adapting evolving software systems and software evolvability has gained significant importance as a software quality attributes [3] and object-based new frameworks [8].

Studies have shown that software maintenance is the most expensive as well as longest phase of the software product life cycle [5]. Once the software has been developed and launched, it requires continuous modifications in order to meet the future requirements. Greater the evolvability of software system, less the cost is required for modification to meet the future requirements.

The factors which may influence evolvability during the software development include: Choice of object classes, Design, Architecture, Components, Encapsulation, Inheritance, Class Libraries, Simplicity, Naming, Comments, and Technical Platform [1]. Until recent years, it looks obvious that little work has been conducted to evaluate and compute the whole software system evolvability, but instead focus has been towards the practice of conducting research in establishing the relationship between software metrics and effort on developing individual classes [7] and reconfigurable software architecture [9]. Thus, based on existing research, it looks quite clear that software evolving received very little attention.

2.1 Structural Measure and Expert Assessments of Software evolvability

Software evolvability can only be measured indirectly [2] and can be grouped into two classes as: structural measures and expert assessments. Using structural measures a great research has been conducted for the assessment of software evolvability but the research mainly emphasized on the evolvability classes. Expert assessments of evolvability are mostly guided or unguided [6]. For instance, for guided assessment, some instructions are provided through brochure and these instructions are used for software system evaluation. On the other hand for unguided assessment, experts utilize their experience and make their assumptions to evaluate the system evolvability. In the case of expert assessment, the major challenge that we have to face is that we are totally dependent on assessment experts. The qualification and knowledge of the assessment expert are those factors upon which these assessments are dependent. In other words, one of the main limitations of the assessment expert is the biasness. Biasness means that an assessment expert may prefer those designs which are more familiar and are easy to them as compared to the others.

III. Software Evolvability Assessment

For finding software evolvability of particular project it is necessary to know relations “equal evolvability as” and “better evolvability than” [4]. A questionnaire and survey are conducted to find the significance hierarchy of the properties of software evolvability. The target population includes professional working in IT and software industries working as software designers, developers and researchers at different levels. The assessment of collected survey is assessed by assessment expert who has e software expert assessment. Software expert is the person who has good knowledge and experience about the design and the development of software. It’s the software expert who assesses evolvability of the software system. Qualification of expert greatly effects the assessment the software evolvability.

3.1 Assessment using Expert Assessment

A questionnaire and survey are designed in the light of 11 characteristics that affect software system evolvability. The survey form has three parts. The first part deals with personal information of the participant of survey. It includes name, education and experience etc. In part II, the 11 characteristics of evolvability of the software system are prioritized by the participant of survey. Finally in III, the qualification of the expert required for the assessment of the evolvability is defined by the participant of the survey. A sample of survey looks like (shown below):

Part II: Characteristics of software evolvability

Choice of Classes	1	2	3	4	5	6	7	8	9	10	11
Design Adopted to System	1	2	3	4	5	6	7	8	9	10	11
Three Layer Architecture	1	2	3	4	5	6	7	8	9	10	11
Good Use of Components	1	2	3	4	5	6	7	8	9	10	11
Encapsulation	1	2	3	4	5	6	7	8	9	10	11
Inheritance	1	2	3	4	5	6	7	8	9	10	11
Good Use of class Libraries	1	2	3	4	5	6	7	8	9	10	11
Simplicity	1	2	3	4	5	6	7	8	9	10	11
Naming	1	2	3	4	5	6	7	8	9	10	11
Comments	1	2	3	4	5	6	7	8	9	10	11
Appropriate Technical Platform	1	2	3	4	5	6	7	8	9	10	11

We conducted this survey and there were total 30 participants in the survey, out of which 25 responded and rest of the 5 did not respond. The following table shows the survey statistics of the software evolvability characteristics and their priority hierarchy as shown in the table design of the software system has got the highest priority while technical platform is at the bottom.

Table 1 Survey Results for evolvability characteristics

Characteristics	Choice Of Classes	Design	Architecture	Components	Encapsulation	Inheritance	Class Libraries	Simplicity	Naming	Comments	Technical Platform
Highest No. of Votes	11	15	10	9	6	12	7	11	10	11	8
Priority Number	4	1	2	5	3	7	6	8	9	10	11

Secondly it was found that what the requirements are regarding the qualification of an expert who assesses the evolvability of the software system. As one important method of assessing the evolvability of software system is an expert method so assessment of software evolvability is highly dependent upon an expert. Expert qualification and experience are also very important for avoiding the biasness in expert assessment. Majority of the participants of survey have the same opinion regarding the expert qualification and experience. According to them an expert should have educational qualification of 16 years, technical education of 4 years, technical training of 1 year, and development experience of 3 years, technical experience of 2 years, 2 years of, maintenance experience of 1 year, testing experience of 1 year and quality assurance/management experience 1 year.

IV. Conclusion And Future Work

The ideal method for accessing the evolvability of software system would provide as result a score of evolvability on at least an ordinal scale ranging for example from very low evolvability to a very high evolvability. This paper has given an overview of existing methods for evolvability assessment and according to the results of my research it is recommended that Software evolvability should be given importance in software development because even though it is a nonfunctional requirement but it is a quality attribute that can affect the cost of software development significantly.

Higher importance should be given to the design of software system because if design is good then less cost will spend on future adaptations. The priority level of characteristics of evolvability of software system is only the direction for software developer, it does not indicate that those characteristics could be ignored which are at the bottom of hierarchy.

In this paper the main focus is on software evolvability and its effect on software development life cycle. However, it is suggested that there are some areas in software evolvability which need to be paying attention and taken as future work. To avoid the limitation of expert assessment and structural measure and to avail their benefits it is recommended that combined approach should be used.

References

[1]. Anda, B. 2008. Assessment of software system evolvability. In Ninth international Workshop on Principles of Software Evolution: in Conjunction with the 6th ESEC/FSE Joint Meeting (Dubrovnik, Croatia, September 03 - 04, 2007).IWPSE '07.ACM, New York, NY, 71-74.

[2]. Anda, B. 2007. "Assessing Software System Maintainability using Structural Measures and Expert Assessments," Software Maintenance, 2007. ICSM 2007. IEEE International Conference on, vol., no., pp.204-213, 2-5 Oct. 2007.

[3]. Beck K. and M. Fowler "Bad Smells in Code," in Refactoring: Improving the Design of Existing Code, G. Booch, I. Jacobson, and J. Rumbaugh, Eds., 1st ed: Addison-Wesley Object Technology Series, 1999, pp. 75--88.

[4]. Jorgensen, M. 2007. Estimation of Software Development Work Effort: Evidence on Expert Judgments and Formal Models, International Journal of Forecasting, pp.449-462, 2007.

[5]. Luer, C., D. S. Rosenblum and V. D. Hoek, 2001. The evolution of software evolvability. In Proceedings of the 4th international Workshop on Principles of Software Evolution (Vienna, Austria, September 10 - 11, 2001). IWPSE '01. ACM, New York, NY, 134-137.

[6]. Mantyla, M. V. and C. Lassenius 2006. Subjective evaluation of software evolvability using code smells: An empirical study. Empirical Software. Engg. 11, 3 (Sep. 2006), 395 431.

[7]. Tom M., M. Wermelinger, S. Ducasse, S. Demeyer, R. Hirschfeld, M. Jazayeri, "Challenges in Software Evolution," Principles of Software Evolution, International Workshop on, vol. 0, no. 0, pp. 13-22, Eighth International Workshop on Principles of Software Evolution (IWPSE'05), 2005.

[8]. Gurdeep S Hura: An Object-First approach to teaching software development: A novel instructional framework", International journal of MTMI, 2013, 29-33

[9]. Gurdeep S Hura, Dynamic Reconfigurable Software architecture: A Novel intelligent framework, Proc MTMI, Virginia Beach, VA, Sept 11-12, 2015