ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

An Empirical Study on Software Reuse with Reliability

Roli Choudhary
Department of Information Technology
BBA University, Lucknow, UP, India

Dhirendra Pandey
Department of Information Technology
BBA University, Lucknow, UP, India

Abstract: In recent past years, some methods for reuse of software have been designed to provide an end-to- end solution for system development life cycle in the minimum time. Further, the re-use of Software process and cognition is considered as one of the important key of software reliability, maintainability and for the improvement of a software system. In addition, for achieving the quality of the product, software industries required to satisfy customer's requirement in terms of reliability and maintainability. A reuse technology, which allows broad and extensive reuse, could provide the means to achieve reliability of legacy software. In this article, we have proposed a framework for reliability factor of reusing the software. The proposed framework will be illustrated with the case study of the stock management system.

Keywords: Software Reuse, Framework, Reliability, Maintainability.

I. INTRODUCTION

Software reuse engineering has always focused on original product development and its original product's software. Development for reuse is the process of the potentially reuse product. Mainly goals of reuse are productivity, quality, duplication, available reuse assets, time to market, maintenance.[1] Software reuse in software engineering most focused target on original development product but it is now recognized that to create most suitable and better software in minimum time and at low cost, we need to adopt a design to a traditional product that is based on sequentially software reuse. Software reuse achieves to benefit that increased dependability, reduced process risk and effective use of specialist and create to standard compliance. Software reuse based principle on hardware and software. In software, reuse uses the same tool more than one producing the same product more than once, etc. In software reuse, we can implement more technology on the same product having the same domain.

In reuse software engineering the code is the most important part that is a partial or complete computer program written an on time can be should be or is being used in another program written a letter one time. Good software reuses process to increase reliability, productivity, less cost, maintainability and also increases its development to the original productivity. This algorithmic view of Intelligence structures objects, in such a way that they become broadly reusable, interastructural operable and archival. Software engineers may decide to create development generalization so that certain part of their software technology program can be used [2]. Software reuse process improvement is a method to improvement and maintains the software producing organization ability to produce and refinement deliver quality software within a less time consume and less cost. Software reuse is a process of NDP (new development product) which is related to the original domain of software program or program code etc [3].

Software reuse is the orderly practice of systematically improvement system, software from a backup of building blocks, so that similarities in need and/or architecture or structural between apps can be developers to achieve strong benefits in productivity, better quality and business performance and execution of the system.[4] Software is a very better process that creating any product very simply with consume to less time and achieve more benefit from original domain.[5]

The organization of this paper is as follows. Section 2 highlights the reuse-based software development; Section 3 & 4 presents the software reuse landscape and metrics for software reuse. Section 5 elaborates the reliability of software reuse and finally, Section 6 presents the concluding remarks.

II. REUSE-BASED SOFTWARE DEVELOPMENT

Software has been reused in application and software based improvement ever since programming of software started. Reuse-based software development is based on life cycle model with the refinement of each phase and identify to domain software activity or software process and reuse software or application activity.[6]. Following are the content, which is always supported by reuse-based software development:

A. *Application reuse*

The whole of software or applications system can be reused either by incorporating or more improvement it without change to original domain application software into other system application software or by improvement application system software. [8]

B. *Component-based software reuse*

Component-based software reuse is improvement or development of domain component and the component is a totally based on domain modules. Modules have a specific quality to the improvement of component activity in software system improvement. [9]

C. *Object and function reuse*

Small scale software product activity develop to the components function that implements a single well identified and refines object or function may be reused.[10]. After considering the above statements we find that reuse-based software development is most important in terms of application reuse, component-based software reuse, object and function reuse.

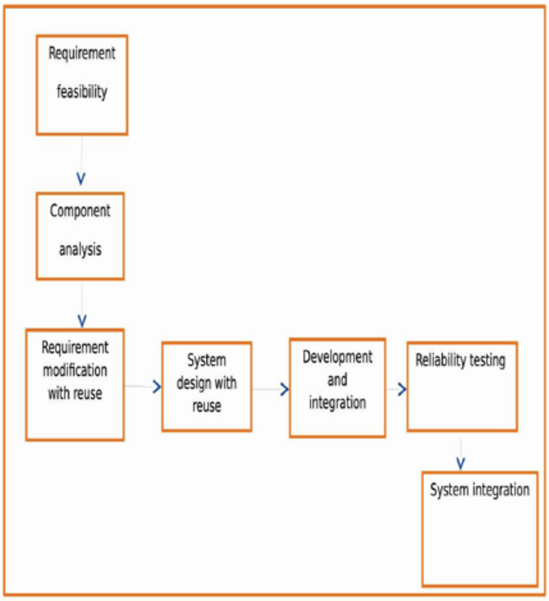

Figure 1 Reuse-Based Software Development

III. THE SOFTWARE REUSES LANDSCAPE

Many types of technologies are used in software reuse has been involved which are increased maintainability and reliability and take minimum time and less consuming work of energy.

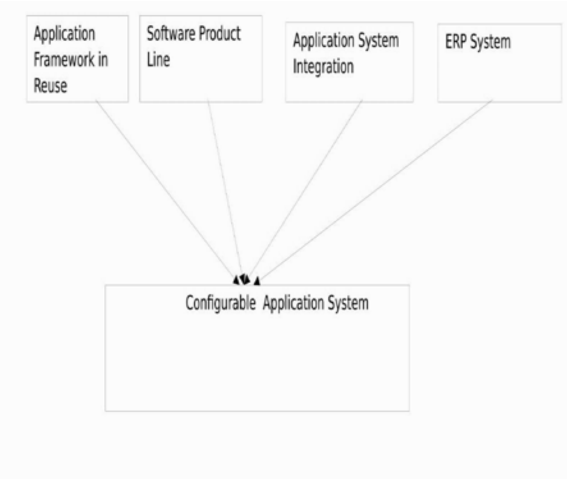
Application Framework in Reuse
Software Product Line
Application System Integration
ERP System
Configurable Application System

Figure 2 Software Reuse Landscape

A. *Application framework in reuse*

Application frameworks in reuse are a promising technology for reifying and identifying proven software designs and implementations and improvement in order to reduce the cost and time and improve the quality of software application system. The basic benefit of application framework from some content which is following they provide by the developer:-

Modularity: Software modularity allows the most efficient reuse techniques by developing application packages interfaces.It is used for consuming common functionality, deploying object oriented concept, instructing code and documentation approach that describes the higher level operation.

Reusability: The stable interfaces given to frameworks introduce reusability by explaining to generic components that can be reused to create new applications software system. Software reuse has been a lofty grand for Software Engineering (SE) explore and practice, as a means to reduced improvement costs and developed software quality. However, non-functional or functional properties (qualities) of a software system are also needed for reliability and maintainability.

Extensibility: A framework introduces extensibility by defining explicit hook method. It allows the applications to extend the previously completed interface. Hook methods systematically deallocate all the stable interfaces in a single application. Hook method decouples the behavior of an application domain by variation recorded in the initialization of application interfaces.

Inversion of control: - The execution-time architecture of a framework is focused by an "inversion of control" and inversion of control give to the each application framework to determine which set of an application-specific technique to invoke in response to window messages or issues send on communication port or arriving from end users.[11]

B. *Software Product line*

Our framework for applications software product line reuse software integrated the main construct of traditional product line engineering, namely the use of programs and the ability to provide huge customization. A applications software product line (SPL) is a set of software-intensive care unit systems that share a common, managed a set of features provide the specific help of an exceptional market section or target and that are an improvement from a common set of core quality in an appointed way[12].

C. *Application system integration*

In application Software reuse and integration has 4.3 Maturity assessment been explained to the process of inventing software systems from the original domain of software which is improved the performances of legacy software and add to new techniques which are integrated to the application software system [13].

D. *ERP system*

Enterprise resource planning software is a most important and integrated system used by system's construction to include, manage and maintain the data necessary for the process. ERP systems are a multi combination of an organization's key process, including the inventing, distribution, related to money, human works, and customer compromise departments, into one application software system. Enterprise resource planning software is a broad system which is used by many types of organizations to combine, organize and maintain the data necessary for operations [14].

E. *3.5 Configurable application system*

Software reuse configurable application system has most important part of the systemically process for improvement in system behaviors, which can easily handle the configurable application that may also provide reliability and maintainability in reuse software system [15].

IV. SOFTWARE REUSE METRICS

In software reuse engineering, many reuse metrics and models have been improved to make the short code in the systematic term (important codes are only used) reuse and reusability.

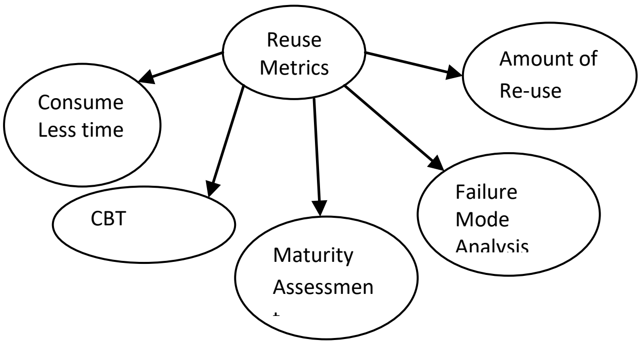

Figure 3 Software Reuse Metics

A. Consume less time

In software reuse metrics take help with original software product domain and improve or an enhance that product help by domain of software and thus consume less time for making the software product.

B. Cost benefit analysis

Cost-benefit analysis models include financial cost-profile project and quality and productivity payoff practices. Margono and Rhoads [1993] applied the cost of development model to assess the economic benefits of a reuse effort on measure a large-scale Adam project (the United States Federal Aviation Administration's Advanced Automation System (FAA/AAS) the main benefit is at low cost provided good productivity.

C. Maturity assessment

Maturity assessment models define reuse performances by how advanced they are in implementing orderly reuse. Total Amount of reuse metrics is used to measure and handle a reuse improvement or development effort by tracking portion of software reuse for life cycle target.

D. Failure mode

Failure modes analysis give a technique to measure and developing a reuse metrics process based on a model of the ways a reuse process can give to an incorrect result.

E. Amount of reuse

A number of reuse metrics are generally used to measure and handle a reuse development effort by tracking percentages of reuse of life cycle target over time. The amount of reuse is mainly used for improving the product of application software and give to high-level performance. [16]

V. RELIABILITY WITH SOFTWARE REUSE

Better system reliability of good quality is one of the aims of software reuse with reliability and maintainability. Software reuse, the use of existing software architect or knowledge to make a new application of software systems, is increased to realize profit such as improved software quality, productivity, or reliability "Reliability is the nature of a device performance and execution of its operation. The purpose of reliability performs adequately for the limited period of execution time intended under the operating conditions gathered". The definitions of error, failure, and fault used in reliability engineering are from the IEEE Standard Glossary that can be taken from Software Engineering Terminology-

Error- An error is value among a computed, observed, or measured value or the condition and the correct value.

Fault – Fault is a physical defect, incorrectness or defect that occurs in software application system.

Failure- Failure is which performance that generally achieved by a developer from some activity of software [17]. Software reliability achieves to more significance activity for improve to the system behavior and satisfied to the customer requirements.

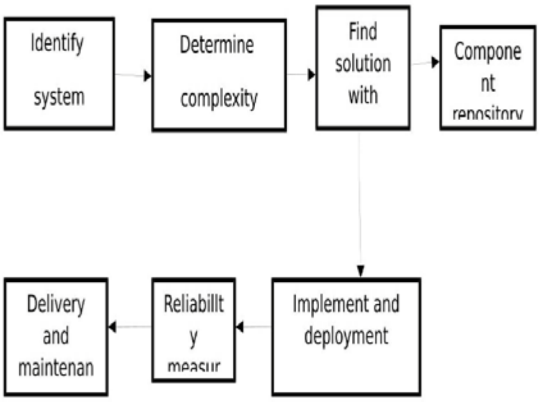

Figure 4 Software Reliability with Reuse

Software Reliability can be achieved with the help of Software Reuse. Software Reuse Terminology introduces the technique of cost reduction in software Development. Today, Software Development Process Uses the Agile Methodology that can help in achieving the reliability with the parallel development process. Software Reliability can be increased due to the development of software with a compatible component of Software that are using in Software Reuse procedure. Software Reliability may be increased or not in Software Reuse Terminology because of the objective of software reliability based on the compatibility of the previous introduced Software Component.[18]

VI. CONCLUSION

The reliability terminology increases the efficiency of the Software Reuse process. It also enhances the whole process of compatible issues with software reuse component because the component of a software should be supportive and compatible the other component of other software. Today, the scenario of software reuse is highly progressive. Software reuse terminology can be more extended in a progressive manner in future. Software reuse directly impacts on the factor of cost and time. The third factor reliability can be introduced in the software reuse terminology. In the article, we have tried to get the possible issues of reliability with the help of software reuse terminology. Finally, it is observed that the software reuse process can be beneficial for achieving the reliable and quality software products.

VII. REFERENCES

[1] Michael Leyton, “Mathematical Theory of Reusability”, 2007 IEEE International Conference on Information Reuse and Integration, 13-15 Aug. 2007

[2] Ferose Babu TA“SOFTWARE REUSE”,2007Department of Computer Science, Cochin University of Science and Technology

[3] Sven A. Carlsson, Mikael Schönström” SOFTWARE PROCESS IMPROVEMENT THROUGH KNOWLEDGE MANAGEMENT” Aaen, I., J. Arent, L Mathiassen and O. Ngwenyama (2001). “A conceptual MAP of software process improvement”. Scandinavian Journal of Information Systems, 13, 81-101

[4] Noor Hasrina Bakar, Zarinah M. Kasirun Norsaremah Salleh, Hamid A. Jalab Extracting features from online software reviews to aid requirements to reuse”Science DIRECT Volume 49, December 2016, Pages 1297–1315

[5] VeronikaBauer, AntonioVetro“Comparing reuse practices in two large software-producing companies SCIENCE DIRECT Volume117, July 2016, Pages 545-582

[6] Germán Vazquez, J.Andres Diaz Pace, Marcel Campo Reusing design experiences to materialize software architectures into object- oriented designs science direct Volume 259, 20 February 2014, Pages 396411

[7] Ina Schaefer, IoannisStamelos Softwae reuse science direct 11 November 2016

[8] Youngsul shin, Yunja choi, woojin Lee Integration testing through reusing representative unit test cases for high confidence medcal software science direct volume 43, Issue,1 june 2013, page 434-443

[9] MELO, W.L., BRIAND, L.C., BASILI, V.R., MEASURING THE IMPACT OF REUSE ON QUALITY AND PRODUCTIVITY ON OBJECT-ORIENTED SYSTEMS, TECHNICAL REPORT CS-TR-3395, University of Maryland, 1995, 16 pages.

[10] Ivan Aaen ,Jesper arent, Lars Mathiassen, Ojelanki Ngwenyama A Conceptual MAP of software process improvement Department of computer science, Aalborg University Aalborg, Denmark

[11] Kyo C. Kang (Pohang University of Science and Technology), Sholom G. Cohen,□Robert R. Holibaugh, James M. Perry, A. Spencer Peterson□ A Reuse-Based Software Development Methodology Software engineering institute January 1992

[12] WILLIAMFRAKES, CAROL TERRY Software Reuse: Metrics and Models ACM Computing Surveys, Vol. 28, No. 2, June 1996

[13] J. Roland Ortt and Jan P.L. Schoormans□ The pattern of development and diffusion of breakthrough communication technologies□ Garcia and Calantone (2002) define different types of innovations rather than technologies.

[14] William B Frakes, Carol Terry□ Software Reuse: Metrics and Models.□ Research Gate June 1996

[15] Bouabana-Tebibel, Thouraya,□Rubin, Stuart Harvey Integration of Reusable Systems□ Springer 2014

[16] R.L. Glass a, T.Y. Chen b,□ An assessment of systems and software engineering scholars and institutions □ELSEVIER 1 MAY 2003

[17] DR. P. K. SURI1, NEERAJ GARG Software Reuse Metrics: Measuring Component Independence and its applicability in Software Reuse IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.5, May 2009

[18] WILLIAM FRAKES, CAROL TERRY, Software Reuse: Metrics and Models ACM Computing Surveys, Vol. 28, No. 2, June 1996