Scientific Journal Impact Factor (SJIF): 1.711
International Journal of Modern Trends in Engineering and Research
IJMTER
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

Re-assessment and Refinement of Software Re-engineering Frameworks

Gurdip Singh Sodi
Assistant Professor,A.S.College Srinagar.

I. Introduction

Software Systems are in continuous change for various reasons. Changes that effect the system over time include requirements, technology and business process. All these changes are diverse in nature and may require different treatments according to their impact over the software system. On the other hand, the current software may have to be rebuilt, in order to create a product with added functionality, better performance and reliability, and improved maintainability. On the other hand, if the changes on the business are too profound, a new system may have to be deployed by adapting an already existing legacy system or by building a new one. Therefore. In all these situations, there are processes, artifacts and knowledge that can be taken as a starting point. Traditional reengineering activities include identifying, delineating, and modeling the existent process; analyzing it for deficiencies; proposing new solutions; and implementing the new design in terms of new technical systems and new organizational structures. It is possible to observe that most of the methods proposed for the specification, development or acquisition of software systems already support some of these activities [1].

Software re-engineering projects such as migrating code from one platform to another, or restructuring a monolithic system into a modular architecture are popular maintenance tasks. Usually, projects of this type have to conform to hard and soft quality constraints (or non-functional requirements) such as "the changed system must run as fast as the original", or "the new system should be easily maintainable than the original". Requirements for the migrant system can be encoded using soft-goal interdependency graphs and be associated with specific software transformations that need to be carried out for achieving the target requirement these transformation can be applied as series of iterative and incremental steps that pertain both to the design (architecture) and source code (implementation) levels. An evaluation procedure can be used at each transformation step to determine whether specific goals have been achieved [2].

There are many reengineering definitions. One definition that is frequently cited is:
"Reengineering, also known as both renovation and reclamation, is examination and of an existing system to reconstitute it in a new form and the subsequent implementation of the new form. A broader definition is: " Reengineering is the systematic transformation of an existing system into a new form to realize quality improvements in operation. System capability, functionality, performance, or maintainability and supportability at a lower cost, schedule, or risk to the customer." This definition emphasizes that the focus of reengineering is on improving existing systems with a greater return on investment (ROI) than could be obtained through a new development effort [3].

The reengineering of software was described by Chikofsky and Cross in their 1990 paper, as "The examination and alteration of a system to reconstitute it in a new form". Less formally, reengineering is the modification of a software system that takes place after it has been reverse engineered, generally to add new functionality, or to correct errors.

This entire process is often erroneously referred to as reverse engineering; however, it is more accurate to say that reverse engineering is the initial examination of the system, and reengineering is the subsequent modification. Re-engineering is mostly used in the context where a legacy system is involved. Software systems are evolving on high rate because there more research to make the better so therefore software system in most cases, legacy software needs to operate on a new computing platform.
Reengineering is a set of activities that are carried out to re-structure a legacy system to a new system with better functionalities and conform to the hardware and software quality constraint [4].

Aims & Objectives
Software development is now facing much more challenges than ever before due to the intrinsic high complexity and the increasing demands of the quick-service-ready paradigm.As the developers are now called for more quality software systems from the industries, there is insufficient guidance from the methodologies and standards of software engineering that can provide assistance to the rapid development of qualified business software [11].
Following are the objectives for this research work:
1) To understand the concept of forward engineering, reverse engineering and reengineering.
2) To do critical analysis of current and recent methods and processes used for reengineering the software.
3) To understand the differences between software and data re-engineering and understand why data re-engineering is an expensive and time consuming process.
4) **To understand the activities such as reverse engineering and program restructuring which may be involved in the software re-engineering process**.
5) To develop a general framework and methodology to facilitate the reengineering of software projects.
6) To understand why re-engineering is sometimes a cost-effective option for software systems.
7) To understand the application of reengineering in Business Processes(BPR).
8) To understand pattern-based reengineering.

II. RESEARCH METHODOLOGIES

The research will be based on Quantitative Methodologies, mostly by doing literature and industrial surveys. Comparative analysis of resistance techniques and providing new methods will be main focus of the research. Experiences from the industry on this topic will be collected and summarized with the research.

Study Design
In this work, we discuss the advantages of the pattern-based and other software development framework. We verify the benefits using a pattern-based software framework and a corresponding system design architecture that is intended for the rapid development of software applications. The main purpose of this proposal is to define generic framework in which existing reengineering techniques can be reconciled, adapted and analyzed.

Study Setting
Part of its value added is in identifying, describing, understanding, assessing, and evaluating such factors as
- The scope of reengineering projects.
- Techniques for gaining system understanding.
- Methods and techniques for the reuse of legacy system assets.
- Reengineering process and process models.

- The role of tools and technologies for supporting reengineering.
- Transition and deployment issues and concerns.

III. Data Collection & Data Analysis

The research work will be based on data collected from literature and industry. The research work is verified and validated by using quantitative method. The proposed framework will serve as a reference model for guiding reengineering efforts and as a useful tool for planning, implementing, and assessing reengineering projects.
Through a suitable software reengineering framework, the quality of the product can thus be enhanced, software development time and cost decreased, and software evolution robustness improved [12].

References

1. Gemma Grau and Xavier French: Reef: Defining a Customizable Reengineering Framework. UniversitatPolitecnica de Catalunya (UPC). Barcelona E-08034, Spain
2. Ladantahvildari, Kostas Kontogiannis, John Mylopoulos: Requirements-Driven Software Re-engineering Framework In: Proceeding of WCRE01 The Eight working on Reverse Engineering(WCRE01) organized by IEEE computer society, Washingyon DC, USA.
3. John Bergey, William Hefley, Walter Lamia, Dennis Smith: A Reengineering Process Framework Software Engineering Institute. Carnegie Mellon University Pittsburgh.
4. http://en.wikibooks.org/wiki/Introduction _to_Software_Engineering/Reengineering
5. http://www.artechhouse.com/GetBlob.aspx?strName=Yang_CH3.pdf
6. John Bergey, William Hefley, Walter lamia, Dennis Smith: A Reengineering Process Framework. Software Engineering Institute. Carnegie Mellon University /Pittsburgh.
7. http://www.sdml.infor/library/Bergey’95.pdf
8. Miquel Torres, Tayfor B.vaughn: Software Requirements and its Applications in the Reengineering process. Department of Computer Science, Mississippi state university.
9. John Mylopoulps. Jaelson Castro.: Tropos: A Framework for Requirements-Driven Software Development. University of Toronto Federal University of Pernambuco. Springer-Verlag June 2000
10. Gemma Grau and Xavier French: Reef: Defining a Customizable Reengineering Framework. UniversitatPolitecnica de caralunya(UPC). Barcelona E-08034, Spain.
11. Chih-Hung and Chih-wei Lu and Pao-Ann Hsiungb. Pattern-based framework for modulatized software development and evolution robustness.
12. Chih-Hung and Chih-wei Lu and Pao-Ann Hsiungb. Pattern-based framework for modularized software development and evolution robustness. Department of Information Management, Hsiuping Institute of Technology, No.11, Gongye Rd., Dali City, Taichung Counly, Taiwan.