

Study of Software Development Using Software Re-Engineering

Manzoor Ahmad Rather, Mr. Vivek Bhatnagar

Research Scholar, Associate Professor

MMICT&BM Applications

Maharishi Markandeshwar University Mullana Ambala

ABSTRACT

Software project planning and management is one of the biggest complex processes in the software project development process. All the steps in the software management process are interlinked with one another and the output of one step becomes the input of the next step. Software project management using re-engineering is the process of using the engineering, scientific knowledge and artifacts to the software project management process. Here in this paper we will describe techniques like reusability methods for the various steps of the project management process. We have to construct the problem according to the method that we are using in it. Example for the requirement gathering we use the reusability method to assign the priority to those requirements that are most important in the project management process.

Keywords:- Software, Re-engineering, Model, Reusability.

I. INTRODUCTION

The software project planning management is the clear need in now a day. To control the software development time, cost, quality is the most important in the development projects. In every aspect of the software development process we have the threat of the software failure. So the goal of the software management process should be to avoid the software failure. Barry Bohem notes “poor management can increase software costs more rapidly than any other factor.” There should be the standard for judging project management performance because it is difficult to manage any activity without any standard method for evaluating that activity. To increase the performance of the newly generated model we perform the case study for that model before the actual execution of the model.

The Re-engineering is the most widely used modeling approach almost in every aspect of the development process to avoid various threats in the development process. In software development process this could be very useful to get rid of time complexity, cost complexity and most important the end product using these approaches will be fruitful for software development process.

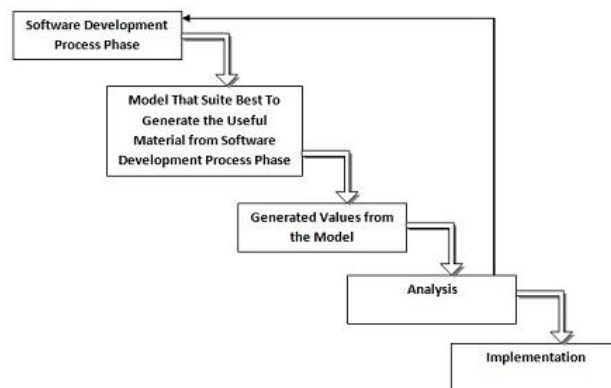


Figure- 1 Software Development Using Models.

II. REQUIREMENT GATHERING PHASE.

Requirement gathering is one of the most important stair of the software development process. All the phases of the software development life cycle are directly or indirectly dependent on requirement gathering phase. So we can say the software complexity is directly

dependent on the requirement gathering. When we gather the requirements of any software we collect some useful and useless data using Re-engineering methods one can easily collect the useful data and all the useless data can be rejected this method uses the priority method i.e. giving high priority to that data which is more important and giving low priority that data which is less important. Only we have to collect the data according to the product that customer actually want. The advantage of this method is that the time and cost is less consumed and the best possible requirements are generated for the development phase.

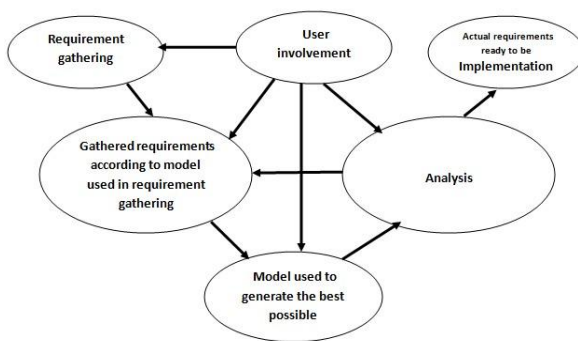


Figure- 2. Software requirement gathering process using modeling approach.

III. OBJECTIVE.

- Make the testing more quick and the simple using Re-engineering.
- Using modeling for requirement gathering so that requirements should be clearer.
- Benefits of re-engineering.

IV. A SOFTWARE RE-ENGINEERING PROCESS MODEL

Re-engineering process model is nothing but is the step by step procedure to carry out the re-engineering process for any particular software project. Re engineering is basically a reconstruction and we can better understand the re-engineering of information system if we consider a parallel activity; the reconstruction of house.

A software re-engineering process model is explained in the below figure as;

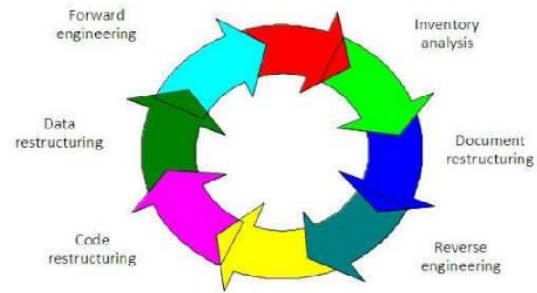


Figure 3 A Software Re-engineering process Model

The re engineering paradigm shown in figure 3. is a cyclic model. This means that the phases of the re engineering model can be re visited according to the need of the re engineering process model.

Below are some phases that are to be carried out in the whole re engineering process.

1. **Inventory analysis**
2. **Document restructuring**
3. **Reverse engineering**
4. **Code restructuring**
5. **Data restructuring.**
6. **Forward engineering**

V. NEED OF RE-ENGINEERING

Re-engineering is the only way to utilize the software fully and solve the problem of software backlogs. Lifespan of software can be increased by the engineers of software system inevitably grow old (becomes complex) and retire but software can be re generated through re engineering in the age of fast changes, process models are changing fast. Re engineering of software improves performance, efficiency, and reliability of the software. Re engineering makes the system new to bit in the new set of requirements. Maintenance of the software starts after the development

of the software correct facts enhance the performance and other attributes of the software system, continuously software maintenance impacts the quality of the software and maintenance cost is too high, we think off re engineering. Following are the reasons for high maintenance cost.

1. Original developers no longer available.
2. Outdated development methods used.
3. Mining of outdated documentation.
4. Future evolution and development is prohibitively expensive.

When maintenance cost is not tolerable software user have to go for re engineering or retiring the software because maintenance cost increases rapidly when software is aged. Job of engineer will be comfortable with re engineering models. Software will be fully utilized and there will not be wastage of resources (costly software). In general following are the objectives of re engineering.

1. Reduce maintenance cost.
2. Functional enhancement.
3. Improves customer satisfaction.
4. Better language or hardware platform.
5. Improve reliability.

VI. MODEL BASED ON SOFTWARE LIFE CYCLE

Maintenance and re engineering are closely related. When maintenance exhausts, re engineering of software is done maintenance is the last stage of SDLC software goes through various stages till it retires.

Software life cycle is defined as a period from software delivery to retirement of the software in this period software is always in operational working state,

software maintenance the last stage of SDLC should be extracted from the cycle and put in SLC, as first stage, re engineering should be added as a second stage in this way SLC will be consisting of two stages, the first maintenance and second re engineering stage, maintenance stage starts just after the delivery of the software.

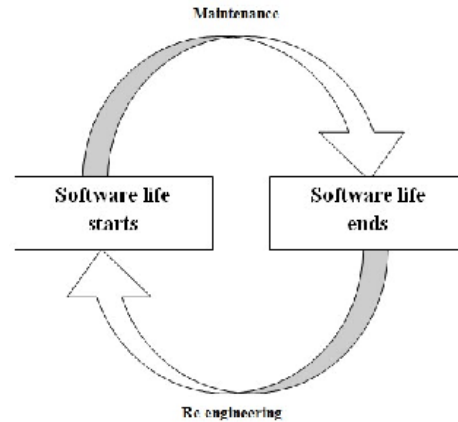


Figure 3.1 software life cycle.

Software remains in the maintenance stage till it gets old. When software gets old re engineering starts. Re engineering cost goes increasing at the maintenance stage. When software is re engineered, software enters in the maintenance stag again and this loop continues until software is retired.

VII. BUSINESS PROCESS OF RE ENGINEERING

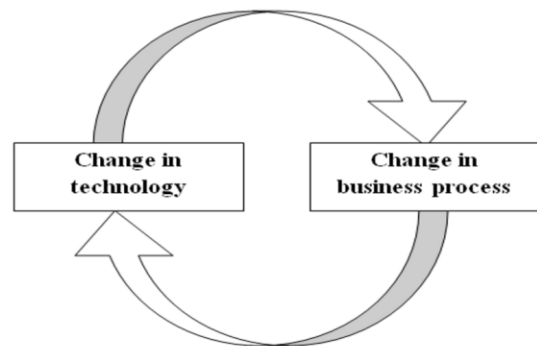


Figure 3.2 business process in re engineering

Legacy software exists everywhere some time maintenance cost becomes unbearable, and then re engineering is the only option to reduce maintenance cost and to avoid software development. Re engineering can reduce the overall investment on the software in the organization.

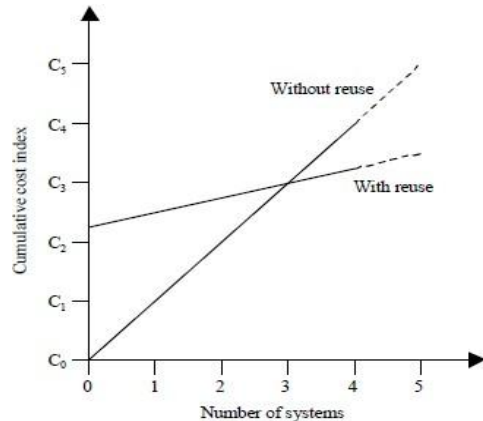


Figure 3.3 cumulative cost of building system with and without reuse.

VIII. BENEFITS OF REUSE

Some of the benefits of reusability of software components are as follows:

- Increased reliability- components exercised in working systems.
- Reduces process risk - less uncertainty in development costs.
- Effective use of specialists - reuse components instead of people.
- Accelerated development- avoid original development and hence speed-up Production.

IX. CONCLUSION

After conducting the study we can easily conclude that the software re engineering is the process through which the project cost, time, and efforts of the project can be easily decreased. In spite of developing the whole project the developers prefer to use some of the already

developed sections of the previous software. It is not mostly used in the small scale companies due to the risk factor although the use to develop whole project instead of reuse the some sections of the project previously developed the main issue of not using of the re usability in the software development is the lack of knowledge. The small scale companies think that the cost of reusability is higher than building the whole project. But using reusability in the large projects there is the less risk and high quality of software can be produced in the time limit. Now a days this technique is used in large companies and the small companies have also started to develop the projects using software reuse. Thus to conclude the software reuse makes the software development at low cost, time, efforts and gives the effective and advanced end product which is the main aim of software reuse.

REFERENCES

- [1] Barry Boehm, "Spiral Development: Experience, Principles, and Refinements", edited by Wilfred J. Hansen, 2000.
- [2] Roger Pressman, Software Engineering: A Practitioner's Approach, Sixth Edition, McGraw-Hill Publication.
- [3] Vivek Bhatnagar and Ashok Kumar Prospective of Software Reusability IJAIEM Volume 3, Issue 1, January 2014.
- [4] Alfonso Fuggetta, "Software Process: A Roadmap", Proc. 22nd Int'l Conf. Software Eng.-Future of Software Eng., 2000, pp. 25-34.
- [5] Frakes, W. B. and Fox, C. J. 16 Questions on Software Reuse. *CACM*, 38, 6 (June 1995), 75-87.
- [6] Kim, Y., Stohr, E. A. Software Reuse: Survey and Research Directions. *Journal of Management Information Systems*, 14, 4 (Spring 1998), 113-147.
- [7] Wikipedia on Software Reuse. Accessed 2008, http://en.wikipedia.org/wiki/Software_reuse
- [8] Kruger, C. Software Reuse. *ACM Computing Surveys*, 24, 2 (June 1992), 131-138.

- [9] Mohagheghi, P. *The Impact of Software Reuse and Incremental Development on the Quality of Large Systems*. PhD Thesis, NTNU, Trondheim, Norway, 2004.
- [10] Baldassarre, M.T., Bianchi, A., Caivano, D., Visaggio, G., An industrial case study on reuse oriented development. In: Proceedings of the 21 International Conference on Software Maintenance (ICSM'05). IEEE Computer Society Press, Budapest, Hungary, 2005, pp. 283-292.
- [11] M. L. Griss. *Software Reuse: From Library to Factory*. IBM Systems Journal, Volume 32, Number 4, Pages 548-566, March 1993.

also written a text book titled “Text book of Computers”.

AUTHORS PROFILE

Manzoor Ahmad Rather did his BCA and MSC-IT from Punjab Technical University Jalandhar Punjab India and

Currently he is pursuing M.Phil-CS from Maharishi Markandeshwar Institute of Computer Technology & Business Management, M. M. University, Mullana (Ambala). India. His current area of interest is Software Engineering and Software Reusability.



Vivek Bhatnagar received his MCA & M. Phil. (Computer Sc. & Technology) degrees and is currently working as Associate Professor at Maharishi Markandeshwar Institute of Computer Technology & Business Management, M. M. University, Mullana (Ambala). His current areas of interest include Software Engineering and Software Reusability. He is an Associate Life



Member of Universal Association of Computer and Electronics Engineers (UACEE), Australia, Member of International

Association of Engineers (IAENG), Hongkong and Member of the Society of Digital Information and Wireless Communication (SDIWC). He has attended four short term courses/workshops. He has