RESEARCH ANALYSIS OF EFFECTIVE OF SOFTWARE REENGINEERING TECHNIQUES AND TOOLS

M.Anjalin Rosy[1], M. Gomathi[2] & K.Suganthi[3]

Abstract

Software Testing is a process of finding errors while executing a program so that we get a zero defect software. It is aimed at evaluating the capability or usability of a program. Software testing is an important means of accessing quality of software. Though a lot of advancements have been done in formal methods and verification techniques, still we need software to be fully tested before it could be handled to the customer side Automatic testing tools are widely utilized to provide testing coverage metrics in order to gauge the quality of software, but these tools may have some shortcomings such as the difference among the values of code coverage metric of a given program using different code coverage tools. the ability of detect and predict poor quality of software is of major important to software engineering and reengineering, manger, quality assurance organization. Poor quality software leads to increase development cost and expensive maintenance. With so much attention on exacerbated budgetary constraints, a viable alternative is necessary. Software quality metrics designed for this purpose. Metric measure certain aspect of code or PDL representations, and can collected and use through life cycle. Automated software quality measures are important for easy integration into the software development process.

Keywords: Software testing, Testing Coverage, Testing Coverage Tools, Software Re-engineering.

1. INTRODUCTION

Software Testing is an activity that is performed for evaluating software quality and also for improving it (Guide to the Software Engineering Body of Knowledge, Steenbok – A project of the IEEE Computer Society Professional Practices Committee, 2004). Software testing is a process to detect defects and mitigate their associated effects, which is used to indicate the quality of software. Software testing plays a significant role in the software development process.

Indeed, most of the costs and resources of testing coverage shows which portion of code, for a given program, is touched by at least one test case. Moreover, code testing coverage is considered by developers as an indicator of confidence level in their software. In order to facilitate the analysis of code coverage, there is a need to automate this process. Therefore, there are many code testing coverage tools, which attempt to help researchers, practitioners, and end-users understand the software testing process. Therefore, many researchers have studied differing aspects of testing coverage analysis process.

Software analysis generally extracts arbitrary properties of software source code. General or custom analyses of software can be implemented using DMS. Software metrics are a special kind of analysis focused on the structure of the source code. Classic software metrics range in variety from the very simple Source Lines of Code (SLOC) to more complex measures such as Cyclamate Complexity measurements. More effectively focus its attention on the lower-quality portions of their portfolio[1].

Metrics tools can be used for several tasks

- ✓ Estimating projects (i.e. estimating time, effort, cost)
- ✓ Managing change of scope

- ✓ Measuring productivity
- ✓ Communicating functional requirements
- ✓ Benchmarking

Many companies try to reduce their software engineering costs by reducing the test costs. There exist several different approaches to do so. They range from simply testing less "in less important areas" [2] to save time, to the use of pro- grams or tools which shall decrease the expenses of the development process [3].

2.LITERATURE REVIEW

The ability of detect and predict poor quality of software is of major important to software engineering and reengineering, manger, quality assurance organization. Poor quality software leads to increase development cost and expensive maintenance once the broad outlines of the new system and the transition path have been charted, authority can more effectively be decentralized for local implementation and optimization. The Matrix highlights interactions and complementary practices. An example of a collection of critical complements includes the use of flexible machinery, short production runs, and low inventories (Dudley and Lasers; Mailgram and Roberts). Emphasizing one such practice increases returns to its complementary practices. Likewise, doing less of a given complement reduces returns to its operating dependents[2].

Blanket and Flare (1995) introduced the following defining elements of reengineering: Radical change – a focus on radical change of processes and systems. Constant reevaluation – including the radical abandonment of dysfunctional ways of working and thinking; the abandonment of old assumptions and rules. Innovation – an emphasis on creativity and production, to go where no one has gone before. Dramatic improvement – the primary command is to make many "leaps" in the set outcomes that have dramatic results, not gradual improvement in processes and systems alone. Effect genesis – a birth of a new beginning, a time of creativity. Multifunctionality – a synergistic process that affects multiple functions and limits of the current state[3].

3.EFFIECTIVE OF SOFTWARE ENGINEERING

Another distinction is that intuitive maintain that rules of thumb and general data regarding what has been drained different firms is beneficial, however that formal bench marking isn't. They are saying that bench marking constrains the BPR team from finding really innovative solutions. Methodologists, against this, feel that bench marking will bring breakthrough concepts (when bench marking outside the organization's own industry) yet as inject reality into the method.

One in every of the difficulties in assessing the relative price and effectiveness of intuitive and method approaches to BPR(Business Process Re-Engineering) is that the majority of the according case studies area unit automatically intuitive, as a result of no methodologies existed at the time the comets were done. In fact, the projects were recognized as business method re-engineering solely once the fact: at the time, they were conceptualized as one thing else. A second problem exists in assessing the relative price and effectiveness of various BPR methodologies. Consulting companies, most of that promote a technique, will offer references Associate in Nursing samples of sure-fire comes exploitation their most well-liked ways (as will companies that use an intuitive approach); if they might not, they might be out of business your text here and click on "Next" to observe this text redactor do it's issue[2].

4.LATEST RESEARCH AND DEVELOPMENT IN SOFTWARE

RE-ENGINEERING

With the advancement in research and development on component system testing, analysis techniques and form- modeling in embedded software and the software credibility, new results and important issues on software testing keep emerging in last recent years.

The latest results of software re-engineering.,

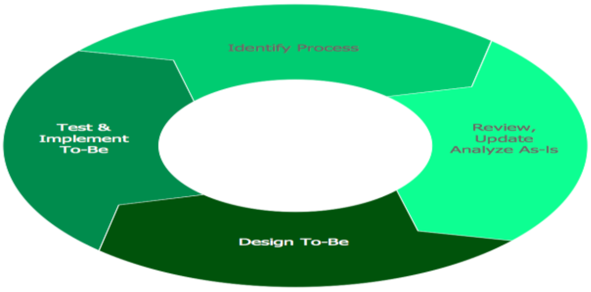
- ✓ Test-driven development
- ✓ Iterative and incremental testing
- ✓ GUI automation test
- ✓ Testability of

Fig.1 Business Process Reengineering Tools[6].

- ✓ component software
- ✓ Embedded Software Simulation Test

5. BRP TECHNIQUES

- ✓ Process Visualization
- ✓ Process Mapping / Operational Method Study



- ✓ Change Management
- ✓ Bench Marking
- ✓ Process and Customer Focus

6.RE-ENGINEERING TOOLS

BPR tools are used more frequently on methodology-based BPR projects than on intuitive ones (unless the user counts a piece of paper, a flow chart template, and a pencil as tools). In fact, some methodologies are based on the use of specific tools. For example, Gemini Consulting Construct reengineering methodology incorporates an object-oriented toolset developed by Park Place Systems.

Coopers & Lybrand's Break Point BPR methodology uses a proprietary process modeling and simulation tool called SPARKS. By using tools, the BPR practitioner expects to improve productivity, finish projects faster, produce higher-quality results, and eliminate tedious housekeeping work in order to concentrate on value-added work. To produce these benefits, BPR tools should be usable by businesspeople (managers and professionals), not technicians.

7. SIX CATEGORIES OF BPR TOOLS

- ✓ Project Management.
- ✓ Coordination.
- ✓ Modeling.
- ✓ Business Process Analysis.
- ✓ Human Resources Analysis and Design.

- ✓ Systems Development.

8.ADVANTAGE OF RE-ENGINEERING

- ✓ Lower costs
- ✓ Lower risks
- ✓ Better use of existing staff.
- ✓ Revelation of business rules
- ✓ Incremental development

9.DISADVANTAGE OF RE-ENGINEERING

- ✓ Practical limits to the extent that a system can be improved by re-engineering

- ✓ Major architecture changes can't be carried out automatically. So, involve high additional costs.

- ✓ Although Re-engineering system can improve maintainability, the Re-engineered system will probably not be as maintainable as new system developed using modern software engineering methods.

10.CONCLUSION

First Static analysis tools which analyze automatically the properties of program without executing it. Second group Function point analysis tools are a reliable method for measuring the size of software and there is also third group consisting of metrics tools that cannot be split in to either of these and we take one examples for each type of metrics tool which we mention above to discuss it in detail. In addition we mention the use of FPA in countries in all over the world and at end we make comparative study between difference between static analysis tools and function point, for that we take LOC tools as type of static analysis to compare it with software development[4].

11.REFERENCES

1. AikaI braheem KreedyMetrics Tools for Reengineering Software Projects. INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 3, ISSUE 9, SEPTEMBER 2014

2. Khalid Alemerien1 and Kenneth Magel2 Computer Science Department North. The Effectiveness of Testing Coverage Tools: An Empirical Study .International Journal of Software Engineering and Its Applications.

3. Mesfin Abebe and Chloe-Jung Yoo Chonbuk National University, Republic of Korea 567 Baekje-daero, A Systematic Literature Review.

4. International Journal of Software Engineering and Its Applications
   Balaam, F. Tip and R. Fuhrer, "Refactoring Support for Class Library Migration", ACM, (2005).

5. http://www.conceptdraw.com/How-To-Guide/bpr-diagram