International Journal of Computer Science and Informatics

Volume 2 | Issue 3 Article 10

January 2013

Software Inspection Improves Quality of Software Product

B. H. Barhate
Bhusawal Arts, Science & P.O. Nahata Commerce College, Bhusawal, barhate_1@yahoo.com

Software Inspection Improves Quality of Software Product

B. H. Barhate

Bhusawal Arts, Science & P.O. Nahata Commerce College, Bhusawal
E-mail : barhate_1@yahoo.com

Abstract **-** Quality Software development is needful requirement in each of the software discipline. Software Inspections, Review plays very important role to maintain quality of the software. Customer satisfaction after deliver software fulfill by this concept.

I. INTRODUCTION

This the Software Engineering related work and is contributed for " To Study Software Inspection improves Quality of Software Product" for Software development . This work contains the following research study.

i. Software Engineering Concept

ii. Software Quality Assurance

iii. FTR Quality Management Activity

iv. Verification and validation of Software product

v. Software Inspection Methods.

vi. Reports

II. SOFTWARE ENGINEERING (SE)

Software Engineering (SE) is a profession dedicated to designing, implementing, and modifying software so that it is of higher quality, more affordable, maintainable, and faster to build. It is a "systematic approach to the analysis, design, assessment, implementation, test, maintenance and reengineering of software, that is, the application of engineering to software."

In the modern era of computer society every user of computer system demands prone and error free system. Computers and computer systems have become a significant part of our modern society. It is virtually impossible to conduct many day-to-day activities without the aid of computer systems controlled by software. As more reliance is placed on these software systems it is essential that they operate in a reliable manner. Failure to do so can result in high monetary, property or human loss.

The NASA Software Assurance Standard, NASA-STD-8739.8, defines software reliability as a discipline of software assurance that:

1. Defines the requirements for software controlled system fault/failure detection, isolation, and recovery;
2. Reviews the software development processes and products for software error prevention and/or reduced functionality states; and,
3. Defines the process for measuring and analyzing defects and defines/derives the reliability and maintainability factors.

For prone and error free software it is necessary to adopt some steps inclusion of SQA analysis.

III. SOFTWARE QUALITY (SQ)

The discipline of software quality is a planned and systematic set of activities to ensure quality is built into the software. It consists of software quality assurance, software quality control, and software quality engineering.

The function of software quality that assures that the standards, processes, and procedures are appropriate for the project and are correctly implemented**.**

A formal technical review (FTR) is a software quality assurance activity performed by software engineers with the following objectives:

- To uncover errors in function, logic or implementation of the software ;
- To verify that the software meets its requirements ;
- To esnure that the software has been developed according to the standards ;
- To achieve uniform software ;
- To make projects manageable.

The formal technical review serves to promote backup and continuity because a number of people become familiar with parts of the software that they may not have otherwise seen.

Each FTR is conducted as a meeting and is considered successful only if it is properly planned, controlled and attended.

- **Software Inspection**

Software Inspection is a technique for achieving quality control for written material and for identifying associated process improvements. Its regular application and success in the software industry suggested its introduction to the on-line computing environment. It has first been invented by M. Fagan at IBM in 1976 and has since been improved and adapted by many major software companies. It is part of quality assurance in the SDP, it complements automatic checking tools and is performed by real people.

Software inspections can identify and eliminate approximately 80 percent of all software defects during development. When inspections are combined with normal testing practices, defects in fielded software can be reduced by a factor of 10. By reducing the amount of rework typically experienced in development, inspections increase productivity and reduce costs and schedules. Cost and schedule reductions for typical applications are on the order of 30 percent.

IV. VALIDATION AND VERIFICATION OF SOFTWARE PRODUCT:

In software project management, software testing, and software engineering, Verification and Validation (V&V) is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It is normally part of the software testing process of a project.

Validation checks that the product design satisfies or fits the intended usage (high-level checking) — i.e., you built the right product. This is done through dynamic testing and other forms of review.

According to the Capability Maturity Model (CMMI-SW v1.1),

- Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. [IEEE-STD-610].
- Validation: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. [IEEE-STD-610]

- **Research Problem :**

Software has become a critical technology. It is essential for telephones, aircraft, elevators, medical devices, banking machines, manufacturing, chemical plants, satellites, power plants and many other systems that affect our health, safety, and well-being. Software is also used in the design of many products that do not contain software. Furthermore, software is also critical for today's business operations. There is almost no part of our economy where the quality of the product or service does not depend on the quality of some software.

Unfortunately, even after more than 30 years of research and development, software remains an unmastered technology. In spite of many advances in our understanding, we still find that most software is delivered to customers with serious faults. Where most products carry a guarantee, packaged software has a disclaimer.

Experienced software engineers and software project managers recognize that the earlier errors are detected in the software lifecycle, the less costly it will be to resolve the error. As a result, software engineers and project managers plan verification and validation activities into the development lifecycle to detect errors as early as possible in the lifecycle. One of the most well known verification and validation activities is the software inspection. Many more software Inspection techniques are available for improve the quality of the software. By using various software inspection methods I can resolve this big problem of occurring defects or Bugs in developed software. Testing strategy is applicable for developed software. This research problem is very vast but I am selecting appropriate which is applicable for developed software.

- **Hypothesis**

After the study of this research it will conclude that every user or customer need not necessary to beg for help frequently. More prone and error free code will be generated after software inspection . Required Software Configuration Items will reduced drastically after release of new version of software. Maintenance cost of multiple version of same software will be reduced.

V. THE SOFTWARE INSPECTION PROCESS

To eliminate defects, many organizations use an inspection process with at least three steps: Preparation, Collection, and Repair. First, each member of a team of reviewers reads the artifact separately, detecting as many defects as possible. Next, these newly discovered defects are collected and discussed, usually at a team meeting.

Then the author repairs them. Under some conditions an artifact can be inspected one or more times. The several variations on this process are detailed in the following taxonomy of inspection methods(Adam Porter,95)

- **Software Inspection Methods :**

Software inspections and technical reviews detect errors early in the software development cycle, when those errors are least expensive to correct. Furthermore, by participating in software inspections and technical reviews, developers improve their own skills, thus reducing the occurrence of errors in the future. Leading software companies have found that a properly implemented program of inspections and technical reviews drastically reduces the time required for testing, debugging, and rework, and dramatically improves the quality of the resulting product. This workshop, Software Inspection And Review Techniques, provides participants with the opportunity to learn about, and experience using, these powerful software quality tools.

The goal of this paper is to present software engineers and software project managers with a survey of available inspection techniques with an empirical evaluation of their effectiveness to allow for software engineers and project managers to not only identify inspection as a necessary activity in a verification and validation plan

- **Fagan Inspection**

The Fagan Inspection is the most commonly cited software inspection technique in software engineering literature. In a sense, the Fagan Inspection is a standard software inspection technique as its essence is included in the IEEE Standard for Software Reviews (1997). Developed by M. E. Fagan in 1972 at IBM, the Fagan Inspection consists of six steps: (1) Planning, (2) Overview, (3) Preparation, (4) Inspection, (5) Rework, and (6) Follow-Up. Compared with walkthroughs, which are informal and consist only of two steps, Preparation and Walkthrough, Fagan Inspections are a more formal alternative requiring an inspection team, with each member playing a role: moderator, designer, coder, or tester.(kendrickhang 2009).

- **Active Reviews**

Active Review technique involves: inspectors working on a single technical area, inspectors working alone, authors of the artifact supplying questionnaires to the inspectors to check comprehension, all of which results in individual discussions between each inspector and the author to arrive at agreed-upon feedback that results in product rework.

- **Phased Inspection**

The Phased Inspection technique involves conducting an Inspection as series of tightly focused steps(phases) instead of in one inspection meeting.

- **Objective of the study :**

1. Testing is widely employed by industry and formal verification widely advocated by the research community as methods of improving software quality. Inspection falls somewhere in between testing and formal verification.
2. It also does not require the time and formula manipulation ability that verification of typical programs would require.
3. Testing and formal verification help to detect errors and determine mathematical correctness, respectively, but it is possible to have error free (mathematically correct) code that is hard to understand and difficult to maintain. In addition to finding errors in code and related software documents, inspection can also help to determine if coding style guidelines are followed, comments in the code are relevant and of appropriate length, naming conventions are clear and consistent, the code can be easily maintained, etc.
4. Cost control is possible in software development cost.
5. Customer get satisfied when he got fulfill and error free software.
6. You can inspect the code itself, not just abstract models of it.
7. Inspection doesn't require as substantial a training investment as verification.

- **Benefits of Software Inspection (Fagan,2001)**

Reduction in user reported defects;

- Increased customer satisfaction;
- Increased development productivity, which materializes in shipping more function
- in a given time or reduction in time to delivery;
- Improvement in meeting committed schedules;

- Rapid cross-training of developers and maintainers on new products;
- Continuous process improvement through removal of systemic defects (which are the cause of defects in the product);
- Authors rapidly learned to avoid creating defects through participating in inspections that find defects in their own work products and in the work products of others;
- Team building; and • Inspections have, in some cases, eliminated the need to unit test code.

- **Software Inspection Process**

1. Work Product

In what state of development is the source code being inspected.

2. Participants

Who are the participants in the inspection?

3. Preparing for Inspection

How much lead time should be granted to the inspectors?

4. Checklist

Preparation of the checklist.

5. Meeting.

What are the entry criteria? I.E., what must be completed before the meeting can occur? Complete Preparation Log and marked source code listing from each inspector.

Who attends? Who does not attend?

What is the purpose?

What is the time limit for the meeting?

What happens during the meeting? Describe the sequence of activities.

What roles does each participant play during the meeting?

What documentation is created and what data is gathered during the meeting? How is the documentation and data gathered and reported? To whom? When?

6. Rework

What are the acceptance criteria for the work product?

What are the procedures for carrying out rework?

7. Follow up

Who is responsible for verifying that rework has been completed satisfactorily?

Moderator will determine if defects have been corrected in an acceptable manner.

How does the verification happen?

Author e-mails moderator the revised code. Moderator checks against defect Log to verify acceptability. Moderator may call a new inspection if necessary.

How is the verification recorded?

Inspection Report is amended to shows which defects have been corrected. Report inspection status is changed from fail to pass.

What are the exit criteria? I.E. what must be finished before the inspection is completed?

Completed source code ready for unit test. Amended inspection report

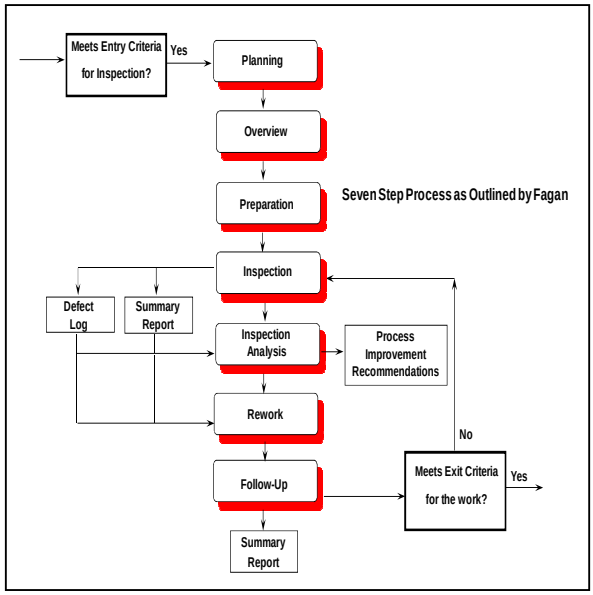

Fig. 1 : Fagans Software Inspection process

REFERENCES

[1] Adam Porter_, Harvey Siy Lawrence Votta, A Review of Software Inspections

[2] Barry Boehm, University of Southern California, Software Defect Reduction Top 10 List

[3] David L. Parnas, Senior Member, IEEE, and Mark Lawford, Member, IEEE , The Role of Inspection in Software Quality Assurance David,IEEE TRANSACTIONS ON

SOFTWARE ENGINEERING, VOL. 29, NO. 8, AUGUST 2003

[4] David L. Parnas, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 29, NO. 8, AUGUST 2003

[5] Michael Fagan ,President, Michael Fagan Associates,Palo Alto, California, USA .A History of Software Inspections .

[6] NASA-STD-2202-93, SOFTWARE FORMAL INSPECTIONS STANDARD.

[7] Stefan Wagner ,Institut f¨ur Informatik Technische Universit¨at M¨unchen. Towards Software Quality Economics for Defect-Detection Techniques

[8] Stefan Wagner and Tilman Seifert.Technische Universit ¨at M¨unchenBoltzmannstr. 85748 Garching, Germany. Software Quality Economics for DefectDetection Techniques Using Failure Prediction∗

[9] http://en.wikipedia.org/wiki/ oftware_engineering

[10] http://orise.orau.gov/emi/scapa/software-quality-assurance/activities.htm

[11] http://en.wikipedia.org/wiki/Verification_and_Validation_(software)

[12] http://www.cas.mcmaster.ca/sqrl/sqrl.html

[13] A Review of Software Inspections Adam Porter_, Harvey Siy Lawrence Votta October 18, 1995

❑❑❑