



---

**Review Article**

# Software Inspections and Their Role in Software Quality Assurance

**Abdul Ahad<sup>1</sup>, Zakir Ullah<sup>1</sup>, Luqman Tariq<sup>2</sup>, Saba Niaz<sup>1</sup>**

<sup>1</sup>Department of Computer Science, Virtual University, Lahore, Pakistan

<sup>2</sup>Department of Computer Science, Preston University, Islamabad, Pakistan

**Email address:**

ahad9388@gmail.com (A. Ahad), zakirkind@gmail.com (Z. Ullah), luqmantariq094@gmail.com (L. Tariq),

ms150200232@vu.edu.pk (S. Niaz)

**To cite this article:**

Abdul Ahad, Zakir Ullah, Luqman Tariq, Saba Niaz. Software Inspections and Their Role in Software Quality Assurance. *American Journal of Software Engineering and Applications*. Vol. 6, No. 4, 2017, pp. 105-110. doi: 10.11648/j.ajsea.20170604.11

**Received:** May 12, 2017; **Accepted:** June 1, 2017; **Published:** July 18, 2017

---

**Abstract:** In this paper we will analyse the software inspections related published data of different successful companies. We will compare the impact of software inspections on software quality with other software quality assurance techniques like software tests etc. Our analysis will show that software inspections play the most important role in software quality assurance. We will also briefly discuss what the Software Inspections are; the benefits of Software Inspections; The team structure in a formal inspection; and the activities in an inspection process. We will also discuss the future of Software Inspections.

**Keywords:** Analysis, Walkthroughs, Reviews

---

## 1. Introduction

The objective of writing this paper is to analyse the software inspections related published data of different successful companies. Our analysis will show that software inspections play more important role in software quality assurance than other software quality assurance techniques like software tests etc do. We will move forward towards our objective by giving a brief introduction of inspections, describing the major benefits of inspections, discussing the team structure for performing inspections, and describing the activities performed in inspections. We will see the important role and benefits of Software Inspection by analysing the data provided by different organizations. We will also see the future of Software Inspections.

Organization of the paper as follows. In section 2, the requirement engineering will be discussed. In section 3 gives a brief introduction of related work. Section 4 gives a brief introduction of software inspections, describes some major benefits of inspections, discuss the roles of member taking part in a software inspection process, activities performed during an inspection process, and some guiding principal for performing inspections will have discussed. In

section 5, the role of inspections has been compared with testing. In section 6, which is also the most important section of the paper, inspection data provided by different organization has been analysed to show the importance of software inspections. Section 7 describes the future of Software Inspections. In section 8, the conclusion of the paper has been given.

## 2. Requirement Engineering (Re)

The most difficult phase in system engineering as well as software engineering is that how we ensure that the specified system will need the requirement of the customers. For this purpose, the requirement engineering process is used.

“The process of establishing the services the system should provide; and the constraint under which it must operate is called software engineering”

In the above definition, the two parts of definition are

### 2.1. Requirements

It represents that what the system do. The requirement may be described a system provide or function while the non-functional requirements describes the constraint which

the system should operate.

## 2.2. Engineering

It means that a systematic process is used to derive a definition of the software system. Which is to be developed.

The requirement engineering consists of six steps.

- (1) Requirement Elicitation
- (2) Requirement Analysis
- (3) Requirement Specification
- (4) System Modeling
- (5) Requirement Validation
- (6) Requirement Management

## 3. Related Work

Our work for this paper is to give an analysis of software inspections published data of successful companies and show that software inspections play more important role in software quality assurance than other software quality assurance techniques like software tests etc do. Before giving analysis, we will give a detailed review of software inspections.

## 4. Introduction to Inspections

Whenever a young software engineer or programmer reviews an entire listing of a program, he always finds some ways the program can be improved: repeated logics never used variables, wrong Boolean tests, comments that don't match the code, and a hundred other errors that make him glad he took the time to look into his work. This act can be termed as an *inspection*.

A Software Inspection is a proven technique of inspection which is formally applied by the expert software engineers for reducing the number of defects and achieving a high quality in software product before it goes out the door. If you are in an organization of two or more people, some kind of inspection activity should be a part of your standard software development process. If you work alone, well, at least read the code.

### 4.1. Related Inspection Types

#### 4.1.1. Code Review

It's a special kind of inspection in which a team examines a piece of code and then correct all the defects found in it. In code review, defect is a piece of code which are not properly applied its requirements, it doesn't function as the programmer proposed, or it is not incorrect but it can be improved. In other hand it is very helpful to the team to fix and find bugs. Code review is very useful for both, its helping junior's developers in new programming techniques learning and helping coder review for cross training programmers.

#### 4.1.2. Peer Reviews

Peer reviews are measured in industries for best practice as detecting software defects in early stages and learned about artefacts software. Peer reviews are collected of software

inspections and software walkthroughs. The collection of skills, behaviors facilitates and coordinated knowledge the greatest possible practice of peer reviews. Peer reviews elements includes, typical of brilliance product checklists, organized review process, defined roles of members, and the reports and forms.

Software inspections are the utmost hard form of peer reviews and completely use all these elements in detecting defects. Software walkthroughs draw selectively upon the fundamentals in support the creator to obtain earnest understanding of an artefact and reaching a harmony among participants.

### 4.2. Benefits of Inspections

- (1) In this section we will discuss the benefits of Software Inspection. Let us see them briefly first.
- (2) Inspections minimize the chances of defects reported by users
- (3) Customer satisfaction is increased by inspections
- (4) Amount of productivity is increased by inspections
- (5) Inspections help in in-time delivery of the software projects
- (6) Inspections help in meeting the committed schedules efficiently
- (7) Inspections increase the experience and speed up the cross-training of team members on new products;
- (8) Inspections continuously improve the development process model
- (9) Inspections provide team building environment.
- (10) Inspections, in some cases, eliminate the need of unit testing of code.

After studying any software engineering book, we will see that the cost of removing a defect increases to a dramatically high factor, when the defect is found at the last stages in software development lifecycle. Software inspection helps us in detecting and removing the defects as soon as possible, hence saving money and time for us in the long run.

In general, the Software testing is limited to be applied on the code of a program. It is not a wise act. Defects exist in all the artifacts of a software product. Software inspections can be applied to all human-readable artifacts that are produced during the development lifecycle of a program i.e. requirements, design, code, documentation, testing etc.

The greatest benefit of software inspection comes when we apply inspection on software requirements, because removing a defect at this early stage save money and time for us. Software inspection helps us here to get a good customer-satisfied document that can be used in next stages of software development process.

Delivery time of a software product is shortened by software inspections. It is so because inspections reduce the time that is spent in the integration and system testing and debugging. Inspections pass a cleaner product to the later stages. Inspections produce better quality software that saves maintenance time.

Inspections are not cost free. Incorporating inspections in the software engineering process adds additional cost to the

budget. Inspection can consume 5 to 15% of the budget. However, many successful companies have concluded that the project budget saved by performing a good inspection is much more than the budget invested in performing inspections.

Mature organization use the data, collected from inspections, to improve the inspection process further. Most common and costly types of bugs can be identified by analysing the inspection reports summary. The root causes of bugs can then be identified. Using this knowledge, the defect prevention ability can be enhanced in the development of future projects.

There are also some side benefits of performing inspections which are not much obvious in general. In fact, participants of inspections learn always something new from every inspection. Participants learn the working style and approach of other team members. Knowledge in different aspects of software development process is exchanged. Programming language feature, commenting and coding style, project architecture, design notations, requirement documentation skills and all other practices of software development process are effortlessly and automatically exchanged among team members.

#### 4.3. Team Structure in Inspections

Inspection is a fully established and proven activity. All participants in an Inspection activity have their special roles. We are discussing these roles here briefly. These roles will also help us know the importance and benefits of inspection in software quality assurance, which is the objective of our paper. The participants in a software inspection activity have the following roles.

- (1) *Moderator*: This person is the leader of inspection activity. Moderator schedules and conducts the meeting. He discusses the results of inspection, and conducts the rework activities and issues. Moderator is an experienced person who knows how to keep the participants technically strong and keep the participants away from social skills of killing each other.
- (2) *Author*: This person creates and maintains the work product that is to be inspected. The author may answer the questions asked about the product during the inspection, and he also knows for defects.
- (3) *Reader*: This person reads the work product to the team as they proceed through the inspection. He tries to explain the functionality of each section to the participants as more clearly as possible by using different words other than words used in the work product.
- (4) *Recorder*: This person records the defects and issues that were raised during the inspection activity. These defects and issues are then classified by the recorder for better understanding.
- (5) *Inspector*: This person tries to find errors in the work product as the reader reads the work product. Inspector is a skilled person in finding errors as he knows the critiques and corners where the defects might exist.
- (6) After that we have discussed the roles of the

participants in the inspection activities, we can easily understand now how important role inspections play in software quality assurance.

#### 4.4. Activities in a Formal Inspection

- (1) In the previous section, we have discussed the roles of participants in Software Inspections. Now we are going to discuss the activities that must be performed during the inspections. Studies have shown that inspections performed through these activities were most successful. The activities are as follows.
- (2) *Planning*: In this activity, the moderator chooses team members that will take part in the inspection and collects the products to be inspected. Moderator, in advance, sends the products and other documents to the inspection team members.
- (3) *Overview Meeting*: This is an informal meeting, before the actual inspection meeting, in which author informs inspection team about major features of the product to be inspected.
- (4) *Preparation*: In this activity, every inspection team member reviews its assigned work product. He notes down all the defects found in the work product and all the issues that will be highlighted by him in the inspection meeting. Almost 75% of all product defects are identified in this activity.
- (5) *Inspection Meeting*: This activity is led by moderator and reader. All the inspection team members highlight and raise issues regarding defects found. All the highlighted issued are recorded by the recorder.
- (6) *Casual Analysis*: In this activity, we try to get the long – term benefits from the inspection program. Casual Analysis provides us guidelines that help us in improving the Software Quality Assurance program.
- (7) *Rework*: In this activity, the author tries to settle all issues highlighted during the inspection. If some issues are not settled, then author takes some decision on those issues.
- (8) *Follow-up*: This is the final activity in inspection process. Moderator coordinate with author and verifies whether all defects found have been removed and issued highlighted resolved. If done, then the inspection activity is declared to be completed.

#### 4.5. Guiding Principles for Software Inspections

In this section, we will discuss some guiding principal that must be followed to execute a successful inspection.

- (1) Don't bring egos into the inspections process. It is difficult to expose our carefully created work to the co-workers for criticism. It is easy to defend our work about the prospective bugs from criticism. Egos adversely affect the inspection spirit.
- (2) Make a criticism to the product but never make a criticism to the producer of the product. While performing the inspection, our focus should be only to make the product error-free.

- (3) During the inspection activity, focus should be to identify the errors, not to fix the errors.
- (4) The duration for inspection should not be greater than two hours at one time. After two hours, it is natural that our enthusiasm and spirit dies down and will not be contributing or helping towards the inspections objective outputs.
- (5) Don't pay much focus to the style issues unless they disturb the performance or understandability. Admit the fact that everyone has his own way of writing, designing and programming.
- (6) Perform inspections as early as possible and as frequently as possible. Perform inspections formally and informally.

## 5. Inspections vs. Testing

In this section, we will discuss the role and importance of Software Inspections in software quality assurance. There are many formal methods that are used to discover the defects in a software product. Out of them, Software inspections and software testing are two prominent techniques. Both inspections and testing can discover defects that, then, can be removed to have an improved software quality. However, the inspections cover greater range of development phases, and thus playing more important role in software quality assurance. Figure 1 and Figure 2 show the situations; when the inspections performed, and performed in a development lifecycle.

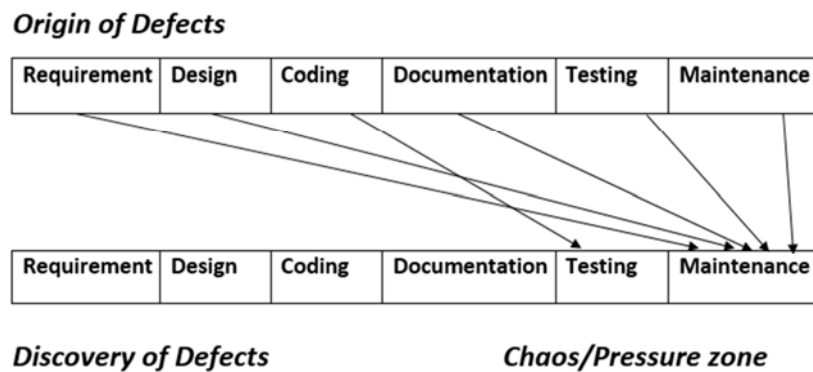


Figure 1. Defect discovery when inspections not performed.

Figure 1 shows the defect origins and defect discovery stages when inspections are not incorporated in the software development process. However, software testing activity is performed here. We can see that testing can discover the coding defects only. No other defects are caught by software testing. All the defects in requirements, design, documentation,

testing, and maintenance are discovered at maintenance stage. This will create chaos at the delivery time. The software quality will also suffer a lot due to this chaos zone. The cost of defect repair will also increase drastically.

Software inspections save us and our product from all the above-mentioned problems. See the figure 2.

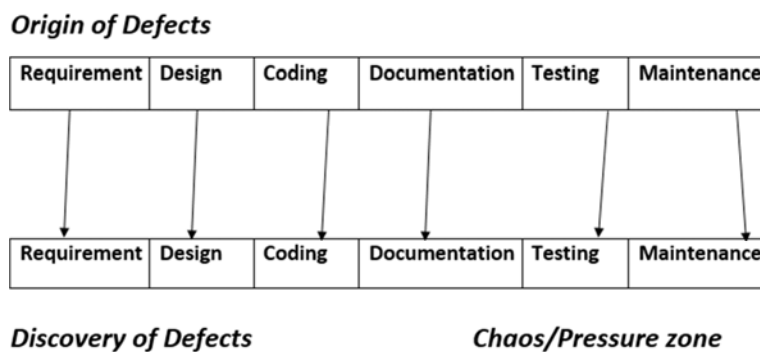


Figure 2. Defect discovery when inspections performed.

In Figure 2 we can see that when inspections are applied, the defects of a stage are discovered at the same stage and then removed by the end of the same stage. This will save us from the chaos zone and also minimized the defect removal cost to a great extent.

Software inspections also play more effective role in discovering the different types of bugs than that of software testing does. This fact can be seen in the Table 1

Table 1. Finding Bugs By Inspection Or Testing.

Nature of Error	Inspection	Testing
Interface defects at module level	✓	
Complexity of code	✓	
Extra features	✓	
Product usability defects		✓
Errors of performance	✓	✓
Poor code structure	✓	
Missing or wrong features	✓	✓
Defects at boundary test	✓	✓

Table 1 shows in its *error type* column, the different types of errors that could be present in a software product. Next two columns *inspection* and *testing* show that which type of error will be discovered by inspection or testing or both of them. Table shows that inspections are more effective than that of testing in order to detect the errors.

Let us now look at the role of inspections from another point of view. Here we are giving a comparison between inspections cost and test cost. The published data from studies of different companies has been given in the Table 2. As the studies were independent, so they present the data in different units. However, in each entry or pattern, we can see that the cost of inspections is much less than the cost of testing.

**Table 2.** Comparison Between Inspections Cost And Testing Cost.

Company	Inspections Cost	Test Cost
AT&T	1.40 hrs/defect	8.50 hrs/defect
JPL	\$110/defect	\$1715/defect
IBM	01 unit of cost	09 times more cost
IBM	\$47/defect	\$60-\$1230/defect
AT&T	01 unit of cost	19 units of cost
ICL	1.2~1.6 hrs/defect	8.39 hrs/defect
Shell	01 unit of cost	29 units of cost
Thorn EMI	01 unit of cost	7~25 units of cost
Appicon, Inc.	01 hour	--
Infosys	01 unit of cost	3 – 6 units of cost

By observing the published data from companies in Table 2, we can see that it is cheaper to detect and remove defects using

software inspections than doing it using software testing both with respect to cost and time.

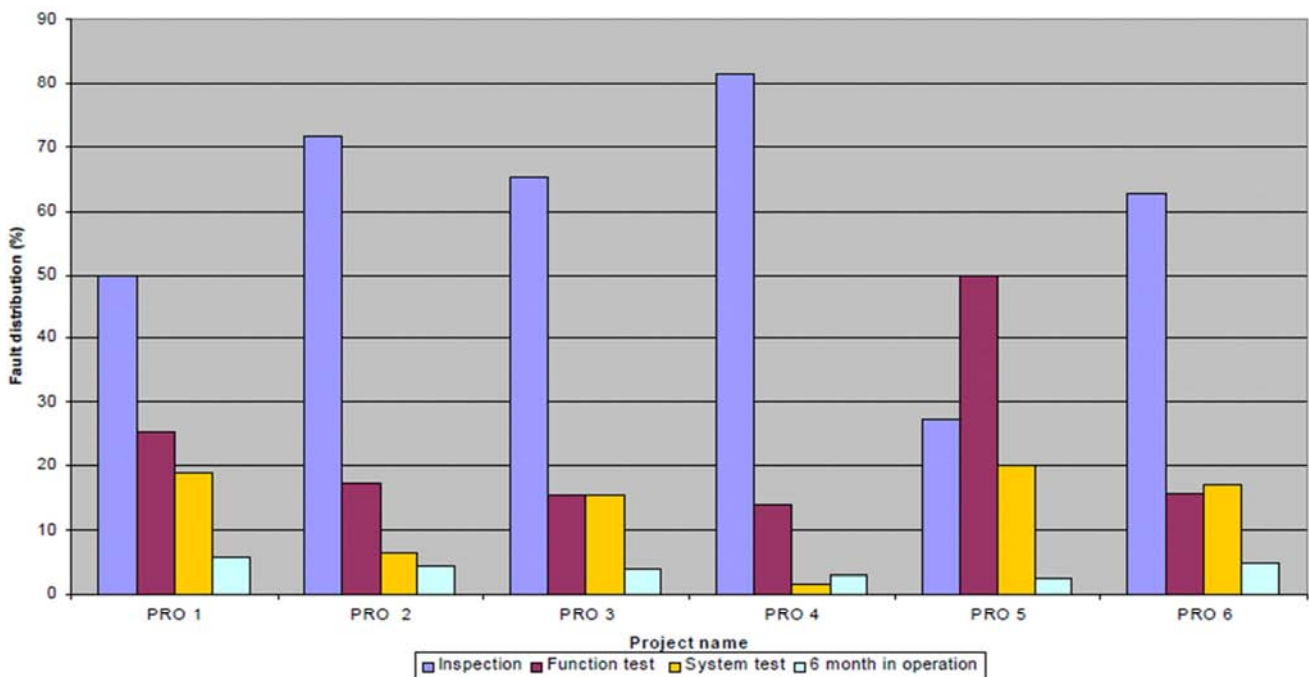
In this section, we have seen that the software inspections play very important and effective role in software quality assurance. Software inspections play more cost efficient, time efficient, excessive pressure efficient and effective role than all other software defect detection and defect removal techniques.

## 6. Analysis of Data Collected from Different Software Development Projects

In this section, we will analyze the data collected from different software development projects. This analysis will show us the importance and role of software inspections in software quality assurance. Any tool which is more affective in discovering faults will have more contribution in the quality of a software project.

Software inspections and software testing data has been collected from different organizations. Software Inspections data has been collected during the document design phases. Software testing data has been collected during function test, system test and 6 months of operation period. These data have been analyzed, compared and shown in the following 2 figures Figure 3 and Figure 4.

Figure 3 shows the fault distribution on different projects.



**Figure 3.** Fault distribution per project.

Figure 3 shows that in most of cases, inspections play the most affective role in discovering the defects in any project. However, in some cases, when inspections do not discover a considerable amount of defects, the function test and the system test discover the defects.

Figure 4 shows the analysis of per quarter data of two years of 1997 and 1998. It is a cumulative view of data.

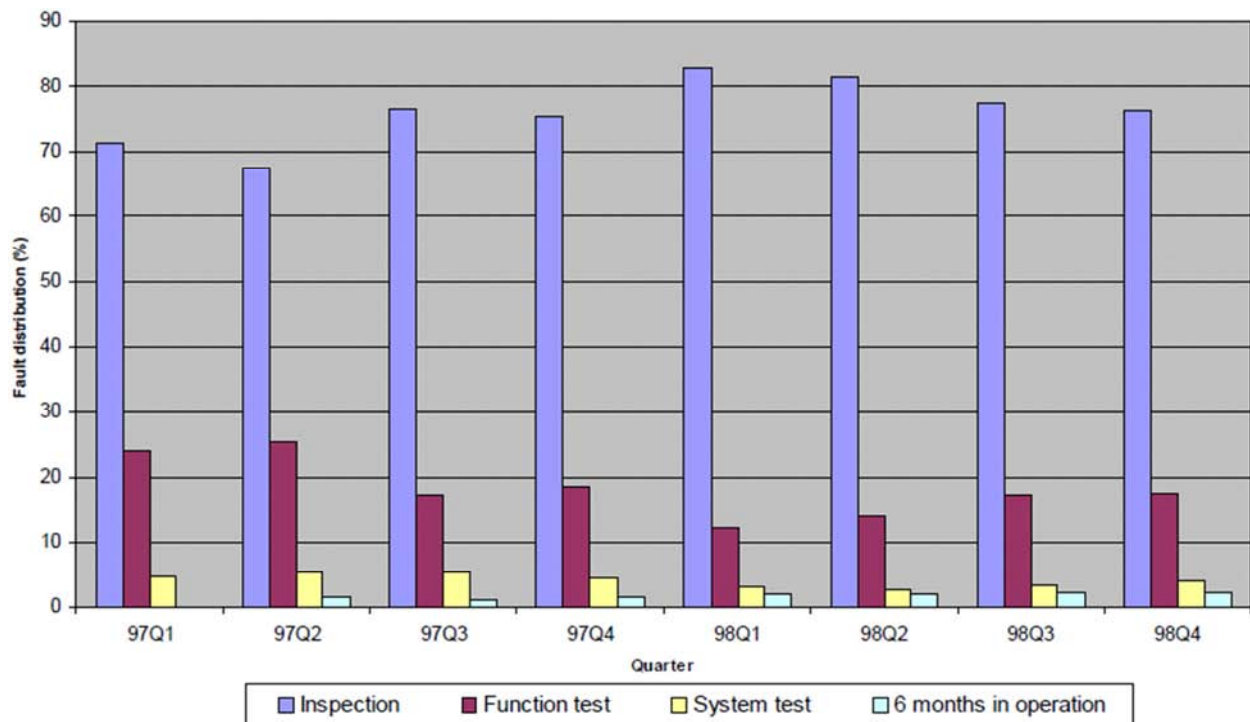


Figure 4. Fault distribution --- cumulative data.

Here in Figure 4 we can see that the product quality increases when more inspections are performed.

## 7. Software Inspections' Future

As the time goes on, computer and software industry is spanning out. Organizations are growing in size. Developed countries are showing more interest in Space projects. Due to all these increasing requirements, the sizes of software applications are also growing up. Small software development companies were paying less attention towards software inspections as they were developing smaller size application. But as the businesses are increasing in number and size, larger software applications are needed by organizations, and therefore, small software development companies also having large software applications to be developed. For ensuring the quality of a large size application software inspections are a must tool. Therefore, the future of software inspection will always be brighter.

## 8. Conclusion

We have seen that demand of large software applications is increasing as the time passes. Software quality is becoming a big challenge. We have also seen that software inspections play the most important role in achieving a high quality of software products. We have given some analysis on data from successful software development companies that showed that software inspections play the most important role in achieving the high quality of larger software applications. We concluded that the future and scope of software inspections is very vast. It is a must activity for those organizations that wish to independently

survive in the software development industry.

## References

- [1] Software Inspections by Ronald A. Radice, Tata-McGH, 2003.
- [2] Karl E. Weigers, "Improving Quality Through Software Inspections", Software Development, in 1995.
- [3] David L. Parnas, University of Limerick, Mark Lawford, McMaster University "Inspection's Role in Software Quality Assurance".
- [4] M. E. Fagan, "Design and code inspections to reduce errors in program development," *IBM Systems Journal*, vol. 15, no. 3, pp. 182-211, 1976.
- [5] Anderson, P.; Reps, T.; Teitelbaum, T. 2003. Design and implementation of a fine-grained software inspection tool. *IEEE Transactions on Software Engineering*, 29 (8), pp. 721-733.
- [6] Roger S. Pressman, *Ph.D.* Software Engineering. A Practitioner's Approach.
- [7] Sneed H. M., and Merey A. (1985) Automated Software Quality Assurance! *IEEE Transactions on Software Engineering*. 909-916.
- [8] Owe S. W., and Yaacob M. H. (1996) A Survey on Software Quality Assurance - A Malaysian Perspective. *Proceedings of the 1996 IEEE*.
- [9] Computer Society Information Systems Conference of New Zealand (ISCNZ'96) pp 154-163.
- [10] Oliver Laitenberger, A Survey of Software Inspection Technologies, Fraunhofer Institute for Experimental Software Engineering (IESE) Sauerwiesen 6 D-67661 Kaiserslautern, Germany.