

Defect Management in Agile Software Development

Rida Noor

Department of Software Engineering, University of Engineering & Technology, Taxila, Pakistan
 Email: ridanoor08@yahoo.com

Muhammad Fahad Khan

Department of Software Engineering, University of Engineering & Technology, Taxila, Pakistan
 Email: fahad.khan@uettaxila.edu.pk

Abstract—Agile development reduces the risk of developing low quality software in the first place by minimizing defects. In agile software development formal defect management processes help to build quality software. The core purpose of defect management is to make the software more effective and efficient in order to increase its quality. There are several methods for handling defects like defect prevention, defect discovery and resolution which are used by software developers and testers. Refactoring keeps the system clean by identifying and removing quality defects. To gain the full confidence of the customer defect management should be involved at every stage of development. Agile methodologies focus on delivering the software in form of short iterations. Thus each iteration helps to overcome defects and leads better development and end user satisfaction. This study paints the picture of handling the software defects using agile based Software Development Process.

Index Terms—Defect Detection, Defect Prevention, Refactoring, Software Engineering, Software Process, Agile Software Development.

I. INTRODUCTION

The success of software industry lies in developing a defect free and quality product within reasonable time and budget. Defect in simple terms may be defined as any undesired data in program. Defects may be injected into software at any stage of software development including requirement gathering, design and coding phase. Defects can be introduced due to incorrect testing as well. Among all software development methods Agile practices are most effective and useful as they focus to develop high quality software by minimizing defects. Defect management is a way to manage defects in order to assure better quality. Agile development includes several quality assurance methods that help to discover defects. For example

1. Software Inspections
2. Software Testing
3. Software product metrics
4. Refactoring

Four major techniques to handle and reduce the defects in developments life cycle of the software are as follow:

1. Defect prevention
2. Defect removal
3. Defect tolerance
4. Defect forecasting

Each defect has its own priority so they need to be managed according to its severity level. Severity level of different defects and their priority is clearly shown in following image.

	- Critical - Must be fixed immediately - Requires notification of responsible executive - Requires customer notification and daily follow up until closed
	- Serious - Must be included in the next sprint - Requires notification of a senior manager - Requires customer notification and weekly follow up until closed
	- Moderate - Must be scheduled two or three sprints out
	- Low priority - Schedule when time is available

Fig. 1. Priority Ranking of Quality Defects in Agile

Defects management ensures that defects are handled in some way, either they are fixed or explicitly ignored. It is used to resolve quality issues found from the software. If defects are not handled they lead to customer dissatisfaction. In Agile development the team itself is responsible for the maintenance of the software quality. So defects are handled by development team to ensure quality product.

Agile methods help to build quality software with the continuous feedback from customers within limited time. That is why these methods are more adaptable to changes. Central point in these methods is to keep the system simple and to remove complexity from it. Thus these

methods are more useful and productive than other software development processes. One of the most useful activities to minimize quality defects is refactoring. Restructuring is a controlled method; an existing code is restructured by changing internal structure without changing its external performance. It is difficult to remove all quality defects in one iteration. But this effort makes the code easily readable.

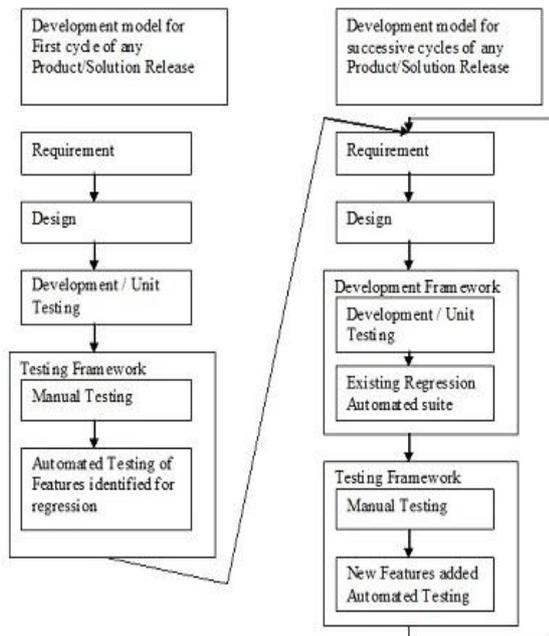


Fig. 2. Testing Process in Agile [11]

This study comprises of four sections. First section provides concise introduction of defect management and the way defects are managed in Agile software development. Literature review is described in second section to consider the related work done for defect management. Third section contains detailed analysis of different techniques applied by different researchers. Conclusion is forth section to provide the results of analysis and research that is conducted in this paper.

II. LITERATURE REVIEW

Defect analysis is very important activity in software development. Its helps to trim down the time, cost, resources and effort required for rework. If defects are detected at early stage then they are not transferred to next phase. Thus defect detection and prevention are two stages of defect management which helps to improve the quality of software.

This section is concerned with the background and related work in agile software engineering and quality defects.

A. Effective Defect Prevention Approach in Software Process for Achieving Better Quality Levels (N. Gopalakrishnanm, 2008)

This paper describes that defect management is a vital

factor to engineer a good quality product. N. Gopalakrishnanm in his paper has made analysis of defect detection and prevention techniques which are employed in Agile development. For this analysis data has been obtained from five projects of leading software development companies. On an average 13 % to 15% of inspection and 25% - 30% of testing out of entire project effort time is required for 99% - 99.75% of defect elimination N. Gopalakrishnanm, 2008 [1]. The results have been compared so that the specific company can position itself with right balancing ratio of inspection testing.

B. Defect Management in an Agile Development Environment(O. Don, 2003)

This paper is detailed study of how agile development environment involves defect detection and its prevention once a defect is detected. It is thought that Agile practices lack effective defect management but actually agile developments reduce defects in first place. O. Don has discussed two wide categories of defect management in his paper: requirements defects and implementation defects [2]. These categories include finding defects in all types of requirements and technical implementation of a project.

C. Handling of Software Quality Defects in Agile Software Development (R. JÖRG, 2005)

Refactoring is a significant phase to improve the software systems continuously. Refactoring focuses to remove the quality defects. But all quality defects are not removed in one iteration. This paper has been written to discuss the repeated and sustainable discovery process, handling, and treatment of quality defects in software systems R. JÖRG [3]. Information about quality defects found in source code has been stored using an automation language. Automation language also represents the defect and treatment history of small parts of the software products. The process and annotation language is helpful for quality defect discovery processes, testing and inspection processes.

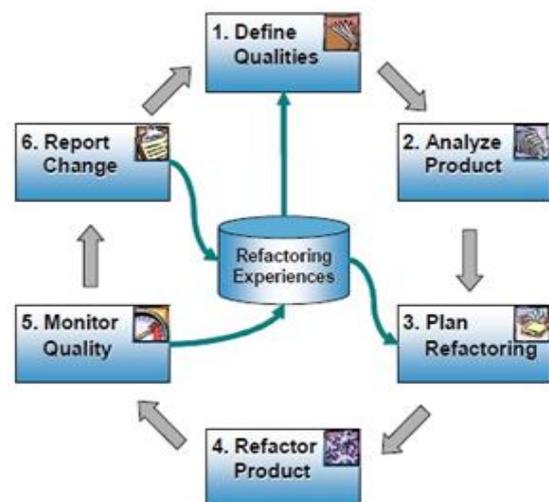


Fig. 3. Quality-driven refactoring by R. JORG [3]

D. Defect Prevention Technique in Test Case of Software Process for Quality Improvement (S. Abhiraja et al, 2012)

In software engineering test case is used to ensure whether the system is working as intended or not. This test case consists of some conditions. In this paper quality defects have been detected using test case. If the software system or process is not working correctly then defect is found. Some preventive actions have been employed to avoid the defects like defects classification and discovering the root causes of the defects. First level of Orthogonal Defect Classification (ODC) has been applied to organize the defects [4]. Thus the use of the test case and preventive ideas would help to improve the quality of software process.

E. Defect Management Strategies in Software Development (V. Suma, 2011)

Software is an important part of every field. Software process like Agile helps to develop defect free software. Any undesirable behavior that is presented by software product comes under defect. Due to the importance of the software, defect management is core need of software industry. Study of this paper describes that inspection is significant to discover the static defect close to the origin V. Suma [5]. Defect-free software can be developed by applying defect managements techniques at every stage of the development process.

F. Software Defect Origins And Removal Methods (J. Capers et al.,2012)

Most luxurious component in the history of the software is to repair the bugs. Defect repair costs are not measured at the beginning. If some companies measure these costs they use metrics which do not measure the value of software quality. Use of poor measurement techniques has led to the fact that most of the companies cannot achieve high level of quality. Amalgamation of defect prevention, pre-test defect removal and formal testing using mathematical strategies all need to be part of the quality technology stack J. Capers, 2012 [6].

G. Supporting Agile Transformation with Defect Management in Large Distributed Software Development Organization (K. Kirsi, 2012)

The main goal of this study was to explore the impact of agile methods when organizations are making transformation to agile software development. To adopt these, agile methods is beneficial for a company. Agile transformation is a journey K. Kirsi, 2012 [7]. This journey involves making improvements on the basis of what have been learnt through work and issues that encountered.

H. Arranging Defect Management in an Agile Development Process (K. Matti, 2004)

Techniques that have been used to manage the defects in this paper are defect tracking and defect analysis. Entire team is considered to be responsible for managing the quality. Dedicated testing team is not involved. Scrum

as a reference model has been used because agile development requires managing the defects for each feature separately. K. Matti states first version of defect tracking agreement for the case company is presented and then evaluated on the basis of interview of company's personal [8]. Next version of defect tracking arrangement is recommended on the basis of evaluation.

I. Defect Analysis and Prevention Techniques for Improving Software Quality (Rajni et al, 2013)

Basically the emphasis of the study is to use defect tracking and defect prevention for the improvement of the quality. Large quality control projects have defined quality control process. These quality control processes consist of many tasks which are performed by different groups. Database is the storage of the defects found which helps to improve the process by tracking and prioritizing defects. Testing is done when the software is developed and defects found are removed using defect prevention. Defect Tracking System still needs improvement in it and a lot of research is required to mature the Defect Tracking Systems Rajni, 2013 [9].

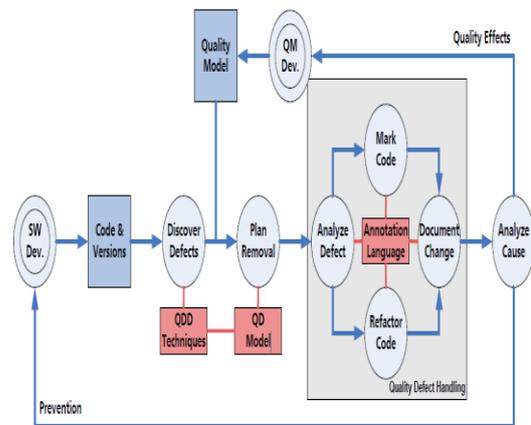


Fig. 4. Defect Handling Process Model [9]

J. Agile- Why the fear?(Sydney et al.,2009)

In this paper the author has used a different technique that is The Defect Management Meeting. In this meeting team members communicate face to face. The meeting is time-boxed to review and prioritize all new defects found. Time-boxing is particularly very helpful when request for change arises late in project and risk of defects increases due to changing requirement. Main goal of this meeting is to review existing defects. This time-boxed meeting is arranged to avoid the game of "defect ping-pong" Sydney et al. [10]. Defect ping-pong means team members or individuals try to understand the nature of the defects. They consider the impact of the identified defects and who is responsible for fixing defects. Defects are prioritized on the basis of severity level in the given time.

K. Establishing a Defect Management Process Model for Software Quality Improvement (K. Ansar, 2013)

Major goal of defect management is to develop high quality software with minimum defects. But defects are

part of software; they remain in its entire life because software is built by humans. This study proposes defect detection and analysis to discover the root causes of potential defects and prevention technique to remove defects. In this paper defect management process model has been proposed by K. Ansar [11] to produce quality products. Proposed model have been used in a case organization and it proved very valuable to handle harmful defects.

L. Towards a Defect Prevention Based Process Improvement Approach (K. Macros, T. Guilherme, 2008)

The study represents a unique concept of Defect causal analysis (DCA) to recover software development process and to reduce amount of potential defects. Defect causal analysis (DCA) approach involves detects identification and classification, discovery of systematic errors and root causes of the errors. This tactic requires defect data to analyze defects in order to mitigate them. Defect causal analysis (DCA) is a source to figure out the causes of defects and to take meaningful preventive action to avoid them T. Guilherme [12].

III. ANALYSIS

Defect detection and prevention are defect management techniques which are used in agile development. Inspection and testing is done to detect and prevent the defects in order to make the quality of software better. Among all the software development practices Agile is considered to be more effective that manages defect in first place. Refactoring is well known technique that removes quality defects in each iteration. Test cases are used to identify the defects and then preventive measures for example Orthogonal Defect Classification (ODC) are applied to prevent the defects. Transformation to Agile methodologies proves to be good

for most of the organizations because these methodologies are more focused to manage the defects as they proceed in iterative manner. Defect management can be established by tracking the defects and to analyze them. Whole team working on the project must be responsible for managing the quality. Defect management meeting helps to review and prioritize the defects that are found. Inspection, Test case, defect detection and prevention, defect tracking and analysis, Orthogonal Defect Classification (ODC) and Refactoring etc are different techniques of defect management. Defect always costs something so it should be removed as early as possible. Cost of removing defect at early stage is less than to repair defect once the product is delivered to the customer. Selection of the test cases is made by keeping in mind which test case will run in regression testing. In SDLC test cases are usually added to automation suite. This helps to continually increase the regression automation bed. New suites which are added to automation bed are established after multiple runs in that sprint. After the completion of the sprint the automated regression suite is sure to be used without any hitches. Utilization of agile mode of development means the functionality of previously developed sprint would be combined with the fresh functionality developed. This information helps to make changes in units settled in former sprints, which can also cause bugs in the previously developed functionality. Thus agile models are indications of bugs in cyclic manner. They involve reporting, fixing and re-verification of bugs.

Following Table 1 gives quantitative measure of defects identified in different phases of Agile development life cycle. These results have been collected from various projects for analysis. According to this table most of the defects are found in testing and implementation phase.

Table 1. Phase Wise Defects [8]

Sr. No.	Agile Development Phase	Number of Defects	Identified Phase
1	Requirement	980	Testing and Implementation
2	Designing	3442	Testing and Implementation
3	Coding	7724	Testing and Implementation
4	Testing	78	Implementation
5	Implementation	58	Implementation

Agile provides a number of techniques to repair defects and to avoid them that do occur during the development of software products. Table 2 summarizes different methodologies that have been discussed by some

researchers in their study. This table analyzes the scope and strength of defect management methodologies to make clear importance of each technique.

Table 2. Critical Analysis of Defect Management Methodologies

Ref. #	Methodology	Strength	Scope
1	Defect Detection and Defect Prevention	Defect Detection and Defect Prevention provide better results for the improvement of the software process.	The study lacks recommendation measures to reduce the level of defects.
2	Requirements Defects and Implementation Defects	The strength of this study is that in this paper different defect management approaches have been identified.	In this paper comparison of different techniques has not been discussed.
3	Refactoring	Refactoring is innovative method for defect anticipation using has been suggested in the study.	The study highlights merely one area.
4	Test Case and Orthogonal Defect Classification (ODC)	The contribution of the study is that it describes the significant of the Test Case and Orthogonal Defect Classification (ODC) in Agile development of software.	The study did not clarify the advantages and disadvantages of the Orthogonal Defect Classification (ODC).
5	Inspection	The study focuses on the use of Inspection method to manage the defects for the development of the software.	The paper did not discuss the software inspection tools.
6	Inspections and Static Analysis	The study discusses that Inspections and Static Analysis are commonly used techniques to manage the defects in Agile Life Cycle.	The study lacks description of tools for handling defects and is conducted truly in generic form.
7	Case study	The study highlights the major problems in defect controlling process which can help the organization to from escape defects in the product.	The study merely discusses the case study for the defect prevention.
8	Defect Tracking and Defect Analysis	The study describes Defect Tracking and Defect Analysis techniques improve the defect management process and enhance the level of quality.	The paper lacks discussing complexity level defect management techniques.
9	Defect Tracking System	By using Defect Tracking System defect management activities have been improved.	The study has limited scope as it did not discuss the managerial aspects of defect management.
10	Time-boxed defect meeting	Approach used is this paper very effective. Entire team is involved to detect and prevent the defects.	This paper did not describe any suggestion for handling defects.
11	Defect Management Model	Defect Management Model introduced in this paper strengthens defect management process of organization.	The study comprises complex work nature and scope is limited.
12	Defect causal analysis (DCA)	Promising practice Defect causal analysis (DCA) has been introduced to identify defects. DCA works using cause-effect method.	The study of paper does not clearly represent the comparison of different techniques.

IV. CONCLUSION

Today the Agile development approach has become a business need. It is a right process to deliver required

functionality and manage risks. It allows taking feedback from one iteration and using it to make decision in next iteration. Thus defects found in one iteration help to prevent the defect in next iteration and thus resulting in

better and desired quality. Agile model is best suited for defect management. Defects are common to software products and they must be given attention so that they can be prevented. If they are controlled and fixed at earlier phases of software development, they save much time and budget. In this paper different defect management methodologies have been discussed. A Critical Analysis has been presented using the strengths and scope of each methodology. The study basically focuses the effectiveness of defect management to assure the software quality in Agile development. Inspection is the straightforward and common approach for the prevention of defects. However, new approaches like Orthogonal Defect Classification (ODC) are getting popularity.

FUTURE SCOPE

More work is required to suggest a structured scheme to handle potential defects so that they can be diminished at early stages of development to produce quality software within specified time and cost.

REFERENCES

- [1] N. Gopalakrishnan, "Effective Defect Prevention Approach in Software Process for Achieving Better Quality Levels", World Academy of Science, Engineering and Technology 42, 2008.
- [2] O. Don, "Defect Management in an Agile Development Environment", The Journal of Defense Software Engineering, 2003. K. Elissa, "Title of paper if known," unpublished.
- [3] R. JÖRG, "Handling of Software Quality Defects in Agile Software Development", Fraunhofer Institute for Experimental Software Engineering (IESE), 2005.
- [4] S. Abhiraja et al., "Defect Prevention Technique in Test Case of Software Process for Quality Improvement", Int. J. Comp. Tech. Appl, Vol 3 (1), 56-61, 2012.
- [5] V. Suma, "Defect Management Strategies in Software Development", Wseas Transactions on Computer, 2011.
- [6] J. Capers et al, "Software Defect Origins and Removal Methods" Namcook Analytics LLC, December 2012.
- [7] K. Kirsi, "Supporting Agile Transformation with Defect Management in Large Distributed Software Development Organization ", ISBN, 2012.
- [8] K. Matti, "Arranging defect management in an agile development process", Seminar in Software Engineering, Spring 2004.
- [9] Rajni et al., "Defect Analysis and Prevention Techniques for Improving Software Quality", International Journal of Advanced Research in Computer Science and Software Engineering, 2013.
- [10] Sydney et al., "Agile-Why the fear", Planit Software Testing, 2009.
- [11] K. Ansar, "Establishing a Defect Management Process Model for Software Quality Improvement", International Journal of Future Computer and Communication, Vol. 2, No. 6, December 2013.
- [12] K. Macros, T. Guilherme, "Towards a Defect Prevention Based Process Improvement Approach", 34th Euromicro Conference Software Engineering and Advanced Applications, IEEE, 2008, DOI 10.1109/SEAA.2008.47.
- [13] https://www.ibm.com/developerworks/community/blogs/siddharthsaraya/entry/a_process_for_reducing_defect_risksfr
- [14] M. Chhavi, "Agile Testing with Scrum-A Survey", International Journal of Advanced Research in Computer Science and Software Engineering, March 2013.

Authors' Profiles

Engr. Rida Noor is MS scholar in Software Engineering Department at University of Engineering and Technology, Taxila, Pakistan. She graduated from Fatima Jinnah Women University in Software Engineering in July 2012. Her focused research areas are Software Design and Architecture, Software Requirement Engineering, Object Oriented Programming, Computer Communication and Networks, Desktop and Web Application Development.

Engr. Muhammad Fahad Khan is serving as an Assistant Professor in the Department of Software Engineering at UET Taxila, Pakistan. He is also a PhD Scholar in Department of Computer Engineering at UET Taxila, Pakistan. He has received his MS degree in Computer Engineering from UET Taxila, Pakistan in February, 2010. He has received B.Sc. degree in Software Engineering from UET Taxila Pakistan in September 2007. His focused areas of research are Software Design and Architecture, Video Summarization, Mobile Application Development, Software Requirement Engineering.