Software Quality And Management
(Assignment - 1)
moan17@student.bth.se
(970626-8167)

Abstract

This paper helps in understanding and evaluating code quality by conducting code reviews. The aspects and steps that are followed in code review process and challenges, expectations, benefits of code review process can be obtained from this paper. This paper also tries to depict the poor code quality using different quality attributes and its impact on maintainability, structure, performance, security can achieved using code review either by automated tools are manual reviewing. In this paper, we have selected five research articles and studied to know its impact, types and effects on code quality using different code review procedures and tools. In first section of this paper, it consists of overall summary of the selected papers and the second section of the paper consists code quality descriptions in all five paper. Third part of this paper explains about the key findings of this paper. This paper concludes with reflections on our own based on code quality and code review process on each paper.

I. Summary

Paper-1: Pixy: A Static Analysis Tool for Detecting Web Application Vulnerabilities [1]

The Problem domain of this paper tries to address the impact of security vulnerabilities, in rapid evolution of web applications usage over the past years and to find solution in order to avoid the manual code reviews, which are time-consuming and budget oriented. This can be overcomed using static analysis tool. The objective of this paper mainly deals with Identification of vulnerabilities using static analysis tool for web application. SQL and command injection type of vulnerabilities are targeted to detect using pixy and also to identify cross-site scripting vulnerabilities. This paper focuses on usage of automated tool to know its effectivity in producing vulnerabilities. The Main Findings in this paper are using automated tool, they have discovered 15 unknown and 36 known vulnerabilities with 50% of false positive rate, one for each vulnerability in three web applications. Overall results obtained lead to decrease in security audits. The limitations found in this paper are the automated tool pixy, doesn't give effective results for OOP features of PHP. False positive, include keyword, file inclusions, variable scoping cannot be identified by the static analysis tool, which should be eliminated using manual evaluation.

Paper-2: Expectations, Outcomes, and Challenges of Modern Code Review [2]

The Problem domain of this paper mainly focuses on modern code review , which is lightweight process that are adopted by most of the companies in order to avoid decreasing effectivity of inspections. The objective of this paper mainly tries to analyse automated code reviews, motivation, challenges and outcomes. It also deals with modern code review process i.e. known

for tool-based, informal, practised in daily life. The Main Findings in this paper are, they have mainly analysed and obtained main reasons for motivation, expectations and challenges for code reviews. Such as, Identifying the bugs are the main motivation behind this but the unexpected outcomes from this process is that performing reviews for finding defects is expected to be done but in fact code reviews leads to Knowledge sharing, team management and various solutions to vulnerabilities are produced. Another important finding of this paper is to understand the code to make further changes, which is the key aspect in code reviews. The limitations found in this paper consists of qualitative results, evaluating the findings in order to obtained results is difficult. To know the challenges or motivations and expectations they have collected data by interviewing, surveying the manager and developers, for comparing each other results from various sources. They have used Codeflow to evaluate and calculate the results of code reviews but, the frequency of occurrence of same outcomes cannot be predicted for different events that are recorded for knowing the various understanding needs.

Paper-3 : A Reflective Practice of Automated and Manual Code Reviews for a Studio Project [3]

The Problem domain of this paper this paper explores the manual and automated tool effectiveness by using checklist. An reflective practice is produced for code reviews to evaluate the quality attributes such as performance based on significant rules and checklists. The objective of this paper mainly focuses on detecting defects in order to evaluate the quality standards, not only depending on coding conventions but also based on quality attributes such as performance and security. The main findings of this paper are about the more defects can be identified by code reviewing process and also the bugs that are not detected in test stage of PMS development can be obtained. If this bugs can be identified before test phase, it will help in increasing quality and productivity of product. They have also found that, if the project specific checklist can be obtained by automated tool then it will improve the effectiveness of verification after removing defects. The Limitations of this paper deals with code reviews based on checklist, will only help in identifying inter smells of the code. The disadvantage and ineffectiveness of the manual review is produced and a evolution model is recommended in order to prove that the static analysis tools is the optimized code review process.

Paper-4 : The Impact of Code Review Coverage and Code Review Participation on Software Quality [4]

The Problem domain of this paper explores the impact of modern code review process on quality aspects. To identify the reasons for huge changes made to code, because of the results obtained from review process that are affecting quality attributes or the results obtained from poor code reviewing participation. The objective of this paper mainly focuses on connection between quality and code coverage, code review involvement. It also tries to track the impact on software quality for poor code reviewing process in large systems through a case study. The main findings of this paper are Insufficient code review participation or low level of code coverage leads to more defects and error-prone, which in turn affects the quality attributes of

large systems. Due to this they have concluded that, in effectiview code review has a negative effect on quality of large and complex systems by using modern automated reviewing tools. The Limitations of this paper deals **I**n evaluating the participation ratio for code review process, they have used 100 lines per hour and also the delay between review process and changes made to display in each release, these overall factors may affect there obtained results.

Paper-5 : Automatic detection of bad smells in code: An experimental assessment [5]

The Problem domain of this paper deals with different types of code smells that indicate poor code structure or design using automated detecting tools. This paper also tries to derive the effectiveness in identifying code smells by various automated tools. The main objective of this paper is to detect the location of code that is most affected by poor code structure and which tools are helpful in detecting and what are the specific places in code that are in need of improvement. The main finding of this paper is to analyse the connection between different code smells and design of the code using code or design patterns, the obtained result is between "feature envy code smell and visitor pattern, where visitor locates the code outside the class where it belongs". The effectiveness of the tool detection of code smells than human to identify the area of impact. The Limitations found in this paper are the identification of better automated tool for detecting the code smells is not obtained with proof. Their own perspective is applied for predicting the needs of future tools rather than experimental analysis.

II. Description of code quality

Paper- [1] **:**
In this paper, they are trying to evaluate the security vulnerabilities i.e system quality of the web applications using different data analysis such as flow and context sensitive and interprocedural data analysis for automated. Alias and literary analysis approaches are used to know whether the obtained results are correct or up to the point. This paper mainly deals with taint-style vulnerabilities such as SQL, cross-site or command injection. Detailed description of factors affecting code quality attribute i,e, security, which is caused by taint-style vulnerabilities. The main goal of the paper is to analyze to identify the possible taint vulnerability also known as sensitive sinks. Factors, harmful properties to describe what are they and how are they defined to identify them are described.

Paper- [2]**:**
In this paper, it mainly describes the reasons or factors for motivating, expectations and challenges to perform code reviews in order to improve the system quality. This paper mainly focuses on quality attribute "understandability" such as, how the results will be obtained for understanding each need and what are steps taken by developer in order to overcome such needs. This paper also tries to classify the motivations of programmers and managers for performing the code review process and also tries to compare the expected and actual outcomes.

Paper- [3]**:**
In this paper, code review process is performed to evaluate the quality attributes such performance and security of PMS development. Here, the detailed descriptions of different types of code review process has been produced such as: Inspections, walkthroughs, code reading, automated reviews. A review method setup is also described about the different rules and checklist that need to be obtained and covered in whole process with flow diagram is explained. A review focus is explained and it contains design and coding standards that are to detected . Another one is quality attributes, performance rule and checklist and security rule and checklist definitions are provide (how the tool identifies based on their definitions and how it is evaluated based on obtained results).

Paper- [4]**:**
In this paper the code coverage and involvement is measured to obtained the system quality level. Whereas less code coverage and low level of participation leads to poor code quality. In order to evaluate, they have studied reviewing policy and traceability systems and mainly focused on gerrit code review tool. They have described and measured by using different quality attribute affecting system quality and calculated using suitable metrics. The following is the classification attributes: product factors such as size, complexity , process metrics defects, churn. Change., coverage metrics reviewed changes, reviewed churn….etc,.

Paper- [5]**:**
In this paper, different code smells are used to evaluate code structure or design to know the maintainability level and overall system quality. In this research paper, every code smell used and detected is clearly defined and measured using six different automated tools are defined with their properties such as version, type, languages, refactoring link to code is written and produced graphical representation for better evaluation. Which code smell is identified by which tool is also described. Finally based on the obtained results, they had compared and contrasted to produce actual outcome and answer the defined research questions.

III. Description of Main Findings:

Paper- [1]**:**
In this paper, they have derived a process to find taint-style vulnerabilities by automated tool such as SQL and cross-site scripting injection. They have used flow-sensitive data analysis to keep the results correct and up to the point and these measures helped them to overcome degree of false positives. In this process, they have identified 15 unknown and 36 known vulnerabilities by using pixy tool and 50% false positive rate. 14% of false positive rate are defined of global variables in database in inclusion files, which are evaluated using manual reviewing. The ratio between the vulnerabilities and false positive can be increased by attack of tainted-style values injecting into the corresponding files.

Paper- [2]**:**

In this paper, the motivation for performing the code reviews are detecting errors to increase the quality and product productivity. The expectations are in contrary to reality, because by performing code reviews there are external benefits obtained such as, increase in Information sharing and transferring, Team management with coordination and communication improvement, and also achieving multiple solutions to various problems. This can be obtained by clearly understanding the code that to changed and level of understandability is achieved by developer using different techniques according their respective needs. But in disregard this requirement is not met by any of the automated tools.

Paper- [3]:
In this paper , By code reviewing the possible outcomes recorded are: In testing phase of software development, every defect is not identified and evaluated. That is the point code reviews are used to find the hidden bugs and refactore them before test phase, this can help in effective code quality. The static checklist that are used for code review can be produced by static analysis tool but the overall product checklist should be depicted manual. To increase the product productivity and system quality the tools should be able to produce the overall checklist of project. By these feature one can say that static analysis tool is known as optimized code review process than manual reviewing and also more number of rules and checklist are in need to identify more bugs.

Paper- [4]:
In this paper, one can grasp the relation between code coverage and involvement ratio needed in order to improve code quality. Poor coverage and low participation of reviewers leads to bad code quality. If system require more number of changes made that implies two situations: 1). The results obtained from reviewing process prescribes 2). Poor involvement of reviewers in process error-prone quality software is released due to this. The experimental results for relationship between code review coverage or code review participation and post-release defects are moderate and both the properties are influenced by each other.

Paper- [5]:
In this paper, the relationship between code smells and pattern can be depicted. Like visible pattern and feature envy. What are the present automated tools used for detecting the different types of code smells, although the better tool is presented but, which code smells a tool can identify is depicted. Which proportion of regions requires changes to obtain better code quality attribute ‘code structure’. The changes made to large amount of region according to automated tool is more realistic and less error prone than human depictions. The values obtained by 4 tools for six different versions of project by using various metrics, had recorded different ranges in each values produced, so best tool is not identified.

Comparing Code Quality and Main Findings of Each Selected Research Paper

Factors	Paper-1	Paper-2	Paper-3	Paper-4	Paper-5
Quality attribute evaluated	Security	Understandability	Performance and security	Code coverage and team awareness	Code structure or design patterns
Attribute purpose defined	The impact on security caused by different Vulnerabilities in web applications	The changes made to code by understanding it clearly and using different techniques by developers according to their needs.	Code reviews generated are used to measure the performance and security of PMS development by both automated and manual evaluation.	The bulk of changes made to system and covered. The level of involvement by reviewers in code review process.	The code or design problems measured using smells and patterns (design, naming conventions) to classify into bad or good code structure of system.
Measured using	Evaluated using taint-style vulnarbilities values and cross-site data analysis and produce how sensitive sinks affects the security of the system.	Data extracted from interviews and surveys. Data gathered from meetings. Using codeflow tool to interact with rivers or involvers. Using this data they obtained results to achieve their goal.	Suitable code review method is derived and different rules and checklist are used to measure the performance and security quality attributes.	Using different quality attributes such as coverage, process, human factors, participation, product properties and their suitable metrics	Different code smell are used such as feature envy, long method, data class…. Etc,. and automated review tools are used to calculate them.
Key Findings	They have used pixy automated tool, in order	How the motivation, expectations and	If The defects detected before test phase helps	If large amount of changes made to	The relationship between codesmell

	to detect known and unknown vulnerabilities and evaluate security audit.	challenges are lead to code reviews. Benefits adhered by using code review such as knowledge sharing, team support, multiple solutions.	in improving product quality and productivity. Checklist derived by automated tool at project levels helps in time consuming.	system then, one can derive that the defect are obtained from reviewing process or poor code review involvement.	and code patterns are derived and code smells that each tool can detect is depicted. Changes made to specific area depicted by tool are effective than manual review,

IV. Reflections based on Selected Research Papers

In Paper [1], one can understand and evaluate the different types of vulnerabilities found in web applications and how they impact the security of the system. Data flow analysis techniques helps users to evaluate the constraints, control flow graph depictions and straightforward approaches. By using this, we can obtain the taint -style issues in the front end that are mostly based on sql and cross-site scripting errors. Detection of unknown vulnerabilities helps in identifying hidden bugs and we can analyse, the effectiveness of the tool by identifying the error-prone ratio of false positive rate for each vulnerability. We can also, know the urgent detection of tools for web applications, where more number of automated tools should be invented to help large systems and complex system and measure various quality and maintainability aspects such as formatting, structure, styling of code, architecture , design, patterns...etc,.

In paper [2]**,** By studying this research paper we can obtain the reasons for performing the modern code review, how efficient it is in uncovering the reduced code quality, what challenges one might face based on the adoption process of code review. One truly grasp the benefits of code review process such information sharing, team support, multiple solutions in depth. The key-aspect of code review survives on quality attribute that is understandability of the changes made to code by understanding it briefly and using various techniques for knowing the source code well. How surveys and interviews helps in gathering the information, which comes by experience and well hand eldidy even in a tight situation, helps the upcoming programmers to evaluate and learn about code review process by automation tools in order to increase the code quality.

In Paper [3]**,** This paper helps in knowing how the defects affect quality and not only based on coding standards, but performance and security affected. Code review process helps in

detecting bugs that are hidden in testing phase. THe static checklist is generated by automated tool sufficiently. Results obtained by comparing different types of code reviews helps in choosing suitable technique according to the situation such as inspections, walkthroughs, code readings, automated reviews. Different rules and checklist for each of performance and security helps in measuring the precise point to know the defect and refactore them. How manual tool is less effective than automated tool can be understood. How the unused local variables and functions affect the system performance and sql query statement input from user affect the security of system can understood.

In Paper [4], This helps in exploring the code coverage importance and relationship between participation of reviewers in code review process. How different attributes of code quality and their metric helps in predicting the quality of system. For code coverage “the reviewed changes and churn metric” for every 200 line per hour is calculated to test the code coverage of the system. The “self approved, hastily changed and changed without discussions” affect the involvement of reviewers in code review process. One can also know the effect of post-release defect on coverage and participation level, which is found negative in this case study for 4 components. The main aspect that can be learned from this paper is poor code review leads to poor code quality.

In Paper [5]**,** This research paper helps in identifying code smells by automated tools to evaluate the code structure. The increase in size of the product leads to problematic for manual code review, thats where automated tool comes in hand. Different types of code smells present, various automated tools with its properties and which tools can produce the specific code smell list helps in understanding lad smells and how effective the automated tool work. By this mentioned list that is describe in this paper helps users for future purpose more accurately. The relationship between the code smells and design patterns helps in understanding coding standards and its effectiveness on code quality. The places where humans can't think of detection bugs are the more error prone regions, which can be discovered by automated tools. By this one can evaluate the benefits of automated tool over manual code reviews.

V. References:

[1] Jovanovic, Nenad, Christopher Kruegel, and Engin Kirda. *Pixy: A static analysis tool for detecting web application vulnerabilities (short paper)*. IEEE, 2006.

[2] Bird, Christian, and Alberto Bacchelli. "Expectations, outcomes, and challenges of modern code review." (2013).

[3] Oh, Jun-Suk, and Ho-Jin Choi. "A reflective practice of automated and manual code reviews for a studio project." *Computer and Information Science, 2005. Fourth Annual ACIS International Conference on*. IEEE, 2005.

[4] McIntosh, Shane, et al. "The impact of code review coverage and code review participation on software quality: A case study of the qt, vtk, and itk projects." Proceedings of the 11th Working Conference on Mining Software Repositories. ACM, 2014.

[5] Fontana, Francesca Arcelli, Pietro Braione, and Marco Zanoni. "Automatic detection of bad smells in code: An experimental assessment." *Journal of Object Technology* 11.2 (2012): 5-1.