ASSIGNMENT-1

**CODE SMELLS**:

code smells are the problems that increase bugs in the code which makes the code to mal-function leading to unsatisfied results.A software code may be in different languages namely c,c++,java,python etc.Code smells varies on the different aspects like software developer,programming language they have used and also the developers way of methodology.Code smells are the indicators of the many issues that are present in the code which make the quality of the source and preventing the code not to be further developed.They effect the quality, maintainenance, security, readability and reliability.They may occur in source code containing Switch statements,Message chains,Data clumps,long method, large class,Middle chains,primitive obsession,Misplaced class etc.Kent beck was the one who introduced code smells and martin fowler was the one who introduced improving the existing code by removing the code smells.There are various tools in to existance for the users to detect the code smells and to resolve them or removing them.We can check a code quality using the tests like as follows:

1)It should not contain duplication in the code

2)It should run all the test cases successfully.

3)It should be able to implement the thoughts of the users.

4)There should be minimal classes and methods as far as possible.

5)There should be case-sensitivity to be considered in some of the source code languages.

**CODE REVIEWS:**

Code review is the process of finding the code smells which effect the software quality in the software code.Code review is of analyising the code using the tools to eliminate the defects and improve the quality of the code.It is used for identifing the defects and tracking them in order to modify those defects. This process can be applied to each and every source code in order to deliver a good final product to the customer.If code review is done in early stages we can easily re-correct the identified bugs very easily.We review this process earlier because the cost for fixing the defects is much higher if they are done later in the software development life cycle.If code smells are found in design phase then it will cost us less for fixing the code smell when comparatively if they are identified after the stages like development, testing and production.

The steps for reviewing the code will include the following activities:

1)We should know the changes of the code to be made for the required conditions mentioned initially.

2)It is better to know the full scope of the problem or issue with the code carefully and implement testing.

3)Recognising those defects which are to be changed in the defect tracking system.

4)Action items on the identified code smells are to be noted in clear.

5)Prepare a clear report after close review of the code.

**PERFORMING CODE REVIEWS:**

The code reviews are done in a typical peer review.After systematic examination using peer review of the code we can find the mistakes by overlooking in the intial phase of development to improve the over all quality of the software.Peer reviews are to be done when a new member joins the team or if there new methodology used in the source code.Peer review need to be done carefully taking consideration of each and every aspect.

we can also use pair programming for code review where two programmers are involved for for code review.while one person is observer and the other is driver.They can switch roles at each time.

The ways of performing code reviews are static and dynamic.But generally static analysis is done where the tool itself is executed.Static analysis includes usage of the static tools,review the complex method and classes in code and to check the factors like redability,design etc

**DEFECTS IN CODE REVIEW:**

Defects in the code review are of various types based on their occurences in the source code.some of the well known defects are:

1)Documentation defects(Language defects,textual defects)

2)Interface defects

3)Functional defects

4)Structural defects(solution approach defects,organization defects)

5)Resource defects

6)check defects

7)logic defects

8)larger defects

**IMPACT OF CODE REVIEW ON SOFTWARE QUALITY :**

Code smells have effect on the properties of the code like quality,readability,reliability and maintainability.Hence code review is necessary in order to maintain good quality of the code.Impact of code review on software quality are as follows:

1)Improves the readability of the code.

2)If once code review is done we can gain knowledge and have a better experience on code.

3)code review helps in delivering the product to the stake holder with good quality.

4)we can have better securable applications after the security review.

5)Consistenty is also one of the impact by peer review to main the consistency in the software design.

6)code review is cheap if done in the early periods of design phase in software development life cycle.

7)Code reviews discussions in team make them closer and provide them with better ideas.

8)code reviews improve the performance of the code.

9)Code reviews are necessary for bug fixing issues which increase the quality.

**TOOLS SUPPORT FOR MAINTAINENCE ASPECTS:**

The tools that are used for code reviewvhelps in reviewing the code . There are various tools for the maintenance of code

1. Check style
2. Collaborator
3. Gerrit
4. Code striker
5. Code brag
6. Review assistant
7. Find bugs
8. Visual code gripper(VCG)
9. Phabricator

The following are the five articles I have considered

ARTICLE1:[11]

In this article author initially focoused on defect counts rather than defect types.He considered various codes of 9 industrial codes of c/c++ and 23 java codes detecting 388 and 371 defects.He says by reviewing the code which did not effect functionality but improved the evolvability.Secondly he compared the functional defect classifications that are useful for review roles,creating checklists and building software engineering tools.By this author concludes that functional defects,code reviews offer additional benefits over execution based quality methods which lack in finding evolvability defects.Hence it suggests for software products with longer life cycle need evolvability defects than short kife cycle systems.

ARTICLE 2:[12]

In this article author says in modern software engineering code review is a standard part.It requires involvement of the people like discussions.With the experience at Microsoft and the given data author found that code reviews should be performed by the experts.The costs,code reviewing topics must be well understood and should be applied systematically to the software engineering work flow with more perfection.This is a best practice comparatively than just doing the code reviews normally by finding the defects only.

ARTICLE 3:[13]

In this article author wants to explain about survey he has conducted on the peer code review.peercode review has benefits like sharing knowledge,sharing expertise,sharing development techniques.By conducting survey on members of OSS communitieswho are involved in peer code review.A questionaire is made from Psychology,Information science and organizational behavior.By representing a subset of survey results the four aspects of peer impression formation.Therefore trust,reliablility,perception of expertise and friend ship.The results indicate there is a high level of these four aspect among the members.Therefore code review helped most in building a perception of expertise between code review partners.

ARTICLE 4:[14]

In this article author want to explain regarding code review.He compared that reviews today are less formal and more light weight compared to the code reviews in 1970s and 1980s.Now-a-days our code reviewers are based on the outcomes of the tools.Here a case study is done on the Microsoft teams in which the defects are only obtained by the code review tools while there is no knowledge transfer and team discussions.This is recommended by the author for practitioners and researchers.

ARTICLE 5:[15]

In this paper author defines code review quality.code review is to improve the quality of code by fixing the bugs before they are commited in the repository.sometimes bugs are introduced at the same time.So an investiogation on code review quality for MOZILLA, a large open source project.from the relationships between the code inspections and set of factors has affect the quality of such inspections.Author applied SZZ algorithm to detect bugs from code which is inspected.There are 54% of the reviewed changes introduced in the code.Hence from the findings metrics,workload,experience and participation members are having affect with the quality.

**REFERENCES**:

[1] A. Edmundson, B. Holtkamp, E. Rivera, M. Finifter, A. Mettler, and D. Wagner, “An Empirical Study on the Effectiveness of Security Code Review,” in Engineering Secure Software and Systems, 2013, pp. 197–212.

[2] B. C. Anda, A. Yamashita, D. I. Sjoberg, A. Mockus, and T. Dyba, “Quantifying the Effect of Code Smells on Maintenance Effort,” IEEE Transactions on Software Engineering, vol. 39, no. 8, pp. 1144–1156, 2013

[3] S. McIntosh, Y. Kamei, B. Adams, and A. E. Hassan, “The Impact of Code Review Coverage and Code Review Participation on Software Quality: A Case Study of the Qt, VTK, and ITK Projects,” in Proceedings of the 11th Working Conference on Mining Software Repositories, New York, NY, USA, 2014, pp. 192–201.

[4] [What Types of Defects Are Really Discovered in Code Reviews?](https://ieeexplore.ieee.org/document/4604671/)Mika V. Mäntylä ; Casper Lassenius,[IEEE Transactions on Software Engineering](https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=32),Year: 2009.

[5][Exploring the usability and effectiveness of interactive annotation and code reviewfor the detection of security vulnerabilities](https://ieeexplore.ieee.org/document/7357234/),Tyler Thomas,[2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)](https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7347691)

Year: 2015

[6][A Security Perspective on Code Review: The Case of Chromium](https://ieeexplore.ieee.org/document/7781793/) Marco di Biase ; Magiel Bruntink ; Alberto Bacchelli,[2016 IEEE 16th International Working Conference on Source Code Analysis and Manipulation (SCAM)](https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7781136)

[7] M. V. Mäntylä and C. Lassenius, “What Types of Defects Are Really Discovered in Code Reviews?,” IEEE Transactions on Software Engineering, vol. 35, no. 3, pp. 430–448, May 2009.

[8] C. F. Kemerer and M. C. Paulk, “The Impact of Design and Code Reviews on Software Quality: An Empirical Study Based on PSP Data,” IEEE Transactions on Software Engineering, vol. 35, no. 4, pp. 534–550, Jul. 2009.

[9] F. Palomba, G. Bavota, M. D. Penta, R. Oliveto, D. Poshyvanyk, and A. D. Lucia, “Mining Version Histories for Detecting Code Smells,” IEEE Transactions on Software Engineering, vol. 41, no. 5, pp. 462–489, May 2015.

[10] S. Vidal, H. Vazquez, J. A. Diaz-Pace, C. Marcos, A. Garcia, and W. Oizumi, “JSpIRIT: a flexible tool for the analysis of code smells,” in 2015 34th International Conference of the Chilean Computer Science Society (SCCC), 2015, pp. 1–6.

[11] Mika V. Mantyla, Casper Lassenius, "What Types of Defects Are Really Discovered in Code Reviews?", IEEE Trans. Software Eng., vol. 35, no. 3, pp. 430-448, May 2009

[12] [Code Review Do Not Find Bugs. How the Current Code Review Best Practice Slows Us Down](https://ieeexplore.ieee.org/document/7202946/)**,j**acek Czerwonka ; Michaela Greiler ; Jack Tilford**,**[2015 IEEE/ACM 37th IEEE International Conference on Software Engineering](https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7174815)

[13] Amiangshu Bosu, Jeffrey Carver, "Impact of Peer Code Review on Peer Impression Formation: A Survey", Proceedings of the 7th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM) 2013, pp. 133-142, 2013.

[14] Alberto Bacchelli, Christian Bird, "Expectations outcomes and challenges of modern code review", Proceedings of the 2013 International Conference on Software Engineering (ICSE ' ‘13). IEEE Press, pp. 712-721, 2013.

[15][Investigating code review quality: Do people and participation matter?](https://ieeexplore.ieee.org/document/7332457/)

Oleksii Kononenko ; Olga Baysal ; Latifa Guerrouj ; Yaxin Cao ; Michael W. Godfrey

[2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)](https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7321954)