International Journal of Innovative Research in Computerand Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.194 |

||Volume 8, Issue 4, April 2020||

Implementation of SonarQube tool and Integrate with Jenkins

Tousif Chikkalaki[1], Dr Vijayalakshmi M.N[2]

PG Student, Department of MCA, Rastreeya Vidyalaya College of Engineering, Bengaluru, Karnataka, India[1]

Associate Professor, Department of MCA, Rastreeya Vidyalaya College of Engineering, Bengaluru, Karnataka, India[2]

ABSTRACT: SonarQube is an open source platform or a tool developed by Sonar Source which is used for the Continuous Inspection of Code Quality which analysis the code of the project. SonarQube works on 27 different language projects. There are many alternative versions of SonarQube, the newer version is 8.1. Basically, the projects are scanned in 27 different language the Code Quality is checked or analyzed. SonarQube can be used a guide for the developer to repair the error or correct the error in the project. It is also possible to manage the projects and check the background tasks. Visualization is available in the SonarQube which helps in gaining the deeper insights into the project to look the present status of the project

KEYWORDS: Continuous Inspection; Code Quality; SonarQube; Code Quality; Visualization

I. INTRODUCTION

SonarQube is an open Source platform or tool developed by Sonar Source which is used for Continuous Inspection of Code Quality which analysis the Code Quality of the project. SonarQube is often downloaded because it is open source tool. To run the project, it requires the Sonar Scanner. Most of the corporate or organization may have a problem when they give the project to the client which contains a lot of error or bugs in it or the security issues, vulnerabilities or the duplication of code. To over come this SonarQube was introduced to check the code quality of that project. There are many different languages are used in the SonarQube such as C, C++, Java, Vb.net, C# etc. SonarQube can be integrated with other tools such as Jenkins, Azure, Git, Bitbucket.

Initially the project is scanned using the Sonar Scanner, in SonarQube platform then there is full analysis of the project, which checks the bugs present, then the vulnerabilities, Duplication of code in the project, security hotspots and duplication blocks are deeply analysed. After seeing this the developer can easily understand what changes to be made in the code or in the project. Empowering the development teams of all sizes to solve coding issues within their existing workflows. With over 6000 customers, and the community edition trusted by more then 200000 organization globally, Sonar Source products are de-facto standard for teams and organization to deliver better, safer software.

The main purpose of having the SonarQube is to analyse the whole project and give a proper report to the organization or the end user which shows what are the problems in it and it also ensure were to make the changes in the code of the project. Anything that affects the code base, from minor stying details to critical design errors, is inspected and evaluated buy the SonarQube, which helps application developers to spot the difficulty and its effect.

II. RELATED WORK

The paper mainly focuses on the use of Static analysis tool (ASATs) how it has increased attention in the past five years. This paper contributes with a comprehensive, multi-method usage of the SonarQube, mining 421976 issues from 246 projects in four different instances of SonarQube: 2 are hosted in the open source communities and other two are hosted by Brazilian government institution. The ascension of software development activities over the past decade has increased the main focus on the reliability and quality of software systems, which also incurs in associated costs to confirm the characteristics. The utilization of Automatic Static Analysis Tools (ASATs) could be a prominent approach to enhance internal quality attributes, as they reveal recurrent code violations without having the price of running the program [1].The paper mainly focuses on the way the SonarQube tool, as SonarQube platform may be a widely used open-source tool for continuous code quality management. It provides an API to increase the platform with plug-ins to upload additional data or to counterpoint its functionalities. The Source Meter plug-in for SONARQUBE platform integrates the Source Meter static ASCII text file analyser tool into the SonarQube platform, i.e., uploads the analysis results and extends the GUI to be ready to present the new results. The first version of the plug-in was released in year 2015 and was compatible with the corresponding SonarQube version. However, the platform and important is, its API. Which have evolved plenty since then, therefore the plug-in had to be adapted to the new API [2]. This paper mainly focusses on the Source Meter SonarQube plug-in is truly the graphical computer program of the Source Meter for Java command-line ASCII text file analyser tool, which may perform deep static analysis of the ASCII text file of complex Java systems [3].

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.194 |

||Volume 8, Issue 4, April 2020||

The paper mainly focuses on "Java Quality Assurance by Detecting Code Smells" that was published 10 years ago at WCRE. The work presents an approach for the automated detection and visualization of code smells and discusses how this approach may be employed in the planning of a software inspection tool [4].

This paper deals with the popularity of tools for software quality analysis has increased over the years, with special attention to tools that calculate technical debt supported violations of a collection of rules. SonarQube is one among the foremost used tools and provides an estimation of the time needed to remediate technical debt. SonarQube checks code compliance against a group of coding rules. If the code has any error or violation t any of the classified rules, SonarQube considers it as a violation or a Technical Debt (TD) item. SonarQube calls the estimated effort needed to refactor the violated rule as remediation time and considers it as TD [5].

This paper exposes the search, classify and analysis of aspects associated with technical debt within the code, also showed the employment of SonarQube combined with static analysis as a method to investigate the software characteristics from the ASCII text file to work out possible issues within the those that develop software [6].

This research paper deals with the Technical debt, which is known to impact maintainability of software. As source code files grow in large size, maintainability becomes more challenging for this. Therefore, it is expected that the density of technical debt in larger files would be reduced for the sake of maintainability [7].

In this paper, there are three main contributions: (i) a quality model combining heterogeneous data sources; (ii) the implementation of the quality model based on Big Data technologies, including frameworks for collecting and analysing the data.(iii) The preliminary evaluation of the quality model indicatesthat the assessed metrics and the implemented factors are understandable [8].

This paper deals with the evaluation approach, experimental set up and results from the study. Through this experiments and analysis of data, it has been identified a set of influential factors affecting software reliability [9].

III. CONTINUOUS INSPECTION

Continuous Inspection is a new available pattern in code quality management which is designed to make internal software quality an integral part of the software development life cycle. It's a holistic and fully realized process which increases both inner software quality of a project and also the visibility of software quality for all stakeholders. The key concept of Continuous inspection is finding problems early-when fixing them remains cheap and straightforward

Continuous Inspection provides continuous code quality management in the SonarQube. The main aim of the Continuous Inspection is to find the problems early, while fixing them is very easy and is cheap. Under this model, the automated code audits are performed on daily basis and made available to the organization Along with this the code audits analyses a project code multiple maintainability axes, tests its bugs, compare the code and also check the duplication of the code. Audits are completed by tools that detect those issues directly in the developer environment, much like spell checker. Team member is alerted if there are many new issues are found so they can be checked quickly as possible-while the code is still fresh in the developer's mind.

Principles of Continuous Inspection

- Managing Software quality should be everyone's concern from the start of the development
- Development team is responsible of managing the code quality
- Software Quality must be a part of the event process, meaning that meeting quality standards is one amongst the hard requirements to be able to declare development complete
- Software quality should be subjective, it should not fail
- Software quality requirements must be common to all software products
- Data for software quality should be up to date
- All the new issues and the existing one should be assigned a clear path

IV. BACKGROUND

Technical Debt-Technical debt is also known as TD. It was introduced by the OOPSLA by the Ward Cunningham in the year 1992, it is related with the failure in the software quality produced by bad practices in the development process.

Technical debt is not related with the code but it is related with the structural or architectural options.

In technical Debt there are two choices or options such as

[1] Leave the software application in a condition without the changes made to it

[2] Consider reducing the debt to fix the specific problem

SonarQube-it is a platform to manage and control the code quality of the projects in seven axes through SQALE method. SonarQube is basically a mixture of 2 main static tools such as Check style and PMD

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.194 |

||Volume 8, Issue 4, April 2020||

Check Style is tool of code quality used to validate and building standards and also find potential problems in the source code. PMD is a source code analyser, through this an end user is able to find the programming the defects

Static Analysis-Static analysis is a technique which analyses the code such as clustering, static, and dynamic analysis have a great impact on the quality attributes like maintainability, flexibility, portability, reusability, interoperability, which is to understand the whole structure of the application.
Static analysis allows to identify the attributes such as files, classes, hierarchical classes, methods, variables, names, packages diagrams, patterns, and the relationship among them, also some of architectural styles, and coding styles.

V. ARCHITECTURE OF SOURCE METER PLUG-IN

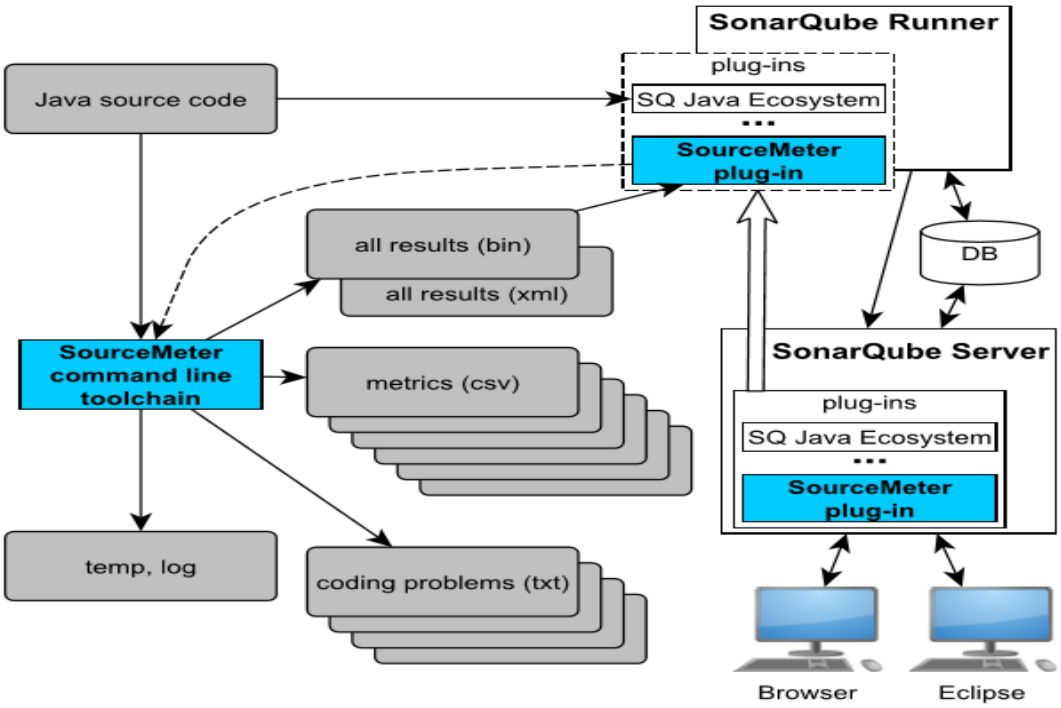

Fig 1.1 Architecture of Source Meter plug-in

This is the architecture of SonarQube with the Source Meter plug-in. The basic thing is to install the Source Meter plug-in later first thing to do is to set the properties then the end user can do analysis.
SonarQube Runner is used to set the properties of the java project then the user can start doing analysis. SonarQube Runner first initiates the connections to the SonarQube Server and then establishes connection with the database later it downloads the required plug-in from the server.

VI. SONARQUBE WITH DOCKER

The SonarQube is a platform which is an open source were the user can make use of Docker
There are few steps to follow

- Install Docker: Download and install the Docker as it is best way to test the tool
- Install the SonarQube community and start it
  There are 2 ways to run it
  1.Docker pull SonarQube
  2.Docker run -d –name -p 9000:9000 -p 9092:9092 SonarQube
  Now browse http://localhost:9000 after this there is a SonarQube GUI
- Login as Admin here, most of the people use the java to analyze the code or the project. Suppose there is a php files then php code should be scanning, for this CLI should be downloaded.
- Now add the Sonar Scanner bin directory path
  Export \ PATH=$PATH:/opt/sonar-scanner-3.1.0.1141/bin

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.194 |

||Volume 8, Issue 4, April 2020||

- To verify everything is been done correctly open a new shell and then execute a command sonar-scanner.bat -h
- Now scanning can be done
- After the scanning is completed the docker container is checked using docker PS command
- The code overview is given in the form of visualization which is in the form of bubble.
- The bubble which are closer or is red then it represents the more severe bugs, bubble size indicates bug volume

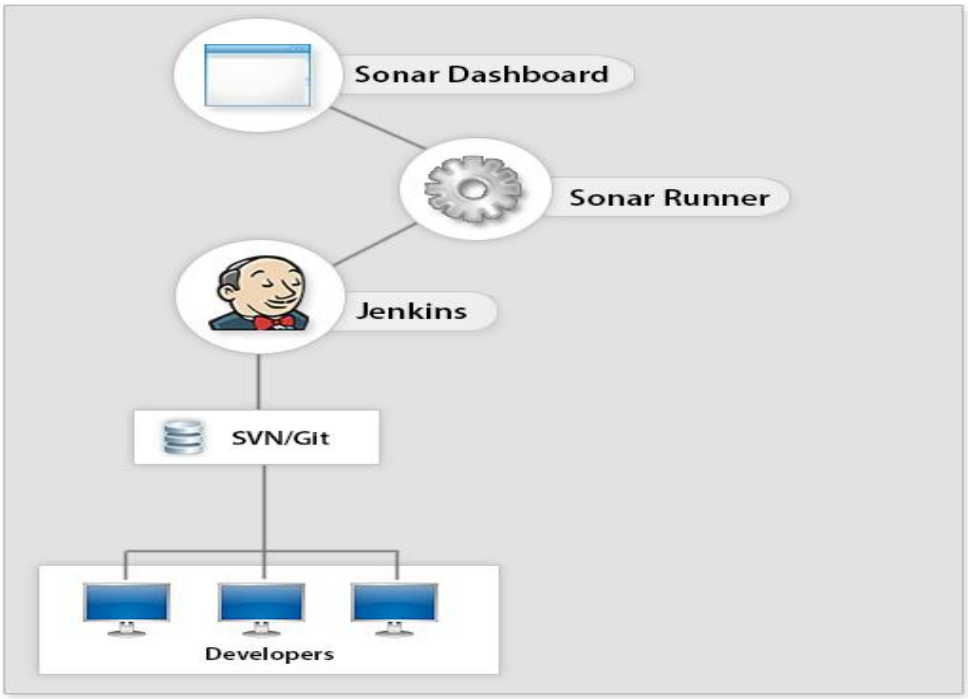

Fig1.2 Integration of SonarQube with Jenkins

VII. SONARQUBE PLUG-IN

The SonarQube platform is most widely used platform or a tool to continuous code quality management. SonarQube also provides an API to extend the features or platform with the plug-ins to upload the extra data which is in generally used to extend the functionalities. To build plug-in the user should have is java 8 and maven 3.1 or more. There are more than 60 community and commercial plug-ins which are present in the SonarQube. Some of the plug-in which are available in market place are SonarABAP, Source Meter, Sonar Apex, SonarC#, SonarC/C++, Sonar Flex, Sonar Java, SonarJS, GitLab plug-in, Sonar Graph, Azure etc.

Source Meter is a plugin for the SonarQube which integrates the Source Meter static code analyser as this plug-in has java code analysis engine of the SonarQube. This plug-in has given the features such as GUI to the SonarQube dashboard so that to make a good user interface for the SonarQube and make it more comfortable and work with tool for more productive.

The source code of the program is usually its up-to-date. As SonarQube is an open source platform to manage the code quality, it also gives many features packaged in the installation and it also has free plugins

FrontENDART-SourceMeter for JAVA SonarQube plug-in is specially developed for SonarQube users, who really want to boost the built-in features of the SonarQube and increase the productivity. Frontend ART developed Source Meter based on the Columbus technology which was researched and developed at the department of software engineering.

Using the results based on the analysis, the quality of the analysed source code can be improved and can be developed in very short time or longer period of time which is based on the static analysis

Source Meter SonarQube plug-in is nothing but a Graphical user interface of the Source Meter for Java command-line source code analyser tool, which can perform the deep static analysis for the complex java source code

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.194 |

||Volume 8, Issue 4, April 2020||

VIII. BENEFITS OF SONARQUBE

- SonarQube helps the Developer, as Software Developer is responsible for code quality
- SonarQube helps the Devops to make sure whether the software is built in right way
- SonarQube solutions decreases the risk and improve team productivity in the field of executives

IX. CONCLUSION

SonarQube is an open source platform where a developer checks the project or analyses the code it also checks code inspection. Continuous inspection is a new model for Software Quality one that incorporates and then sends feedback to developers and it also check the code duplication, code smells, many other issues.Continuous Inspection is a holistic, fully-realized process designed to make internal code quality an integral part of the software development life cycle. This paper details the key challenges in code quality management. It then introduces the Continuous Inspection paradigm and illustrates how it addresses those challenges, supporting thousands of enterprises in improving their software quality. The paper also introduced the plug-in such as Source Meter Which eliminates many weaknesses presents in the SonarQube. To understand the accuracy of the Technical Debt remediation effort that SonarQube as associates to TD items.The most important thing is that SonarQube platform is most easy to use and very easy for Developers to understand because of its Visualization

REFERENCES

1. Daniel Gauman, Pablo Alegendro, Luis Barba, Paola Carbrera, Liliana Enciso"SonarQube as a tool to Identify Metrics and Technical Debt in the source code through static analysis" Proceeding of 2017 the 7th international workshop on computer science and engineering. Beijing, 25-27 June, 2017,pp.171-175ISBN 978-981-11-3671-9
2. Nyytisaarimaki, Maria Teresa, Valentina Lenarduzzi, Simone Romano"On the Accuracy of SonarQube Technical Debt Remediation Time " 2019 45thEuromicro on Software Engineering and Advanced Application. 978-1-7281-3421-5/19/$31.00 ©2019 IEEE DOI 10.1109/SEAA.2019.0005
3. Evan van Emden, Leon Moonen "Assuring Software Quality by Code Smell Detection"
4. Qirat Ashfaq, Rimsha Khan, SehrishFarooq "A Comparative Analysis of Static Tools to Check Java Code Adherence to Java Coding Standards" 2019 2nd International Conference on Communication , Computing and Digital Systems (C-CODE) 2016 IEEE 8th International Workshop on Managing Technical Deb. 978-1-5090-3854-1/16 $31.00 © 2016 IEEE DOI 10.1109/MTD.2016.7
5. Rudolf Ferenc, Laszlo Lango, Istvan Siket, Tibor Gyimothy, Tibor Bakota "SourceMeter SonarQube plug-in"201414th International Working Conference on Source code Analysis and Manipulation. -0-7695-5304-7/14 $31.00 © 2014 IEEE DOI 10.1109/SCAM.2014.31
6. Diego Marcilio, Rodrigo Bonifacio, EduradoMonterio, Edna Canedo, Welder Luz and Gustavo Pinto"Are Static Analysis Violations Really Fixed? A Closer Look at Realistic Usage of SonarQube" 2019 IEEE/ACM 27th International Conference on program Comprehension (ICPC) 978-1-7281-1519-1/19/$31.00 ©2019 IEEE DOI 10.1109/ICPC.2019.00040
7. Md Abdullah Al Mamun, Antonio Martini, MiroslawStaton, Christian Berger, Jorjen Hansson " Evolution of Technical Debt: An Exploratory Study" Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
8. Silverio Martínez-Fernández, Andreas Jedlitschka, Liliana Guzmán, Anna Maria Vollmer" A Quality Model for Actionable Analytics in Rapid Software Development " e Q-Rapids H2020 European project (no. 732253), and the ERCIM Fellowship programme. IEEE copyright notice (© 2018 IEEE
9. Sanjay L. Joshi, Bharat Deshpande, and Sasikumar Punnekkat "Experimental Analysis of Dependency Factors of Software Product Reliability using SonarQube" Copyright © 2019 for this paper by its authors. 130 Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
10. S Javier Garcia-Munoz, Marisol Garcia-valls, Julio Escribano-Barreno "Improved Metrics Handling in SonarQube for Software Quality Monitoring" Distributed Computing and Artificial Intelligence, 13th International conference pp463-470
11. VSilverio Martinez, Andreas Jedlischka, Liliana Guzman, Anna Maria Vollmer "A Quality Model for Actionable Analytics in Rapid Software Development" 2018IEEE 44thEuromicro conference on Software Engineering and Advanced Application (SEAA)2018
12. NAndriy Shapochka, BorysOmelayenko "Practical Technical Dept Discovery by Matching patters in Assessment Graph" 2016 IEEE 8th International on Managing
13. https://en.wikipedia.org/wiki/SonarSource
14. https://en.wikipedia.org/wiki/SonarQube