4

3

Dynamic Testing

Testing

Static Testing

Functional

Testing

Non- Functional Testing

Review etc.

What is Static Testing?

Sonar Qube

======

(Static Code Analysis Tool)

What is the difference b/w functional Testing and Non-Functional Testing?

test

Ans) Functional testing the will the functionalities of the Project in which the expected results will be matched with actual results eg:: withdraw, deposite, transfer Money and etc..

Non-Functional Features Testing is testing checking overall project performance and monitoring

eg: threads info,processes info, heap dump info and etc.. note:: The end users uses the functional features of the Project where as the maintainance team uses the non-functional features

Static Testing is a type of software testing in which software application is tested without code execution. Manual or automated reviews of code, requirement documents and document design are done in order to find the errors. The main objective of static testing is to improve the quality of software applications by finding errors in early stages of software development process.

Required Tools::

sonarqube,Coverity, pychan and etc.

Here the source code will be testing to check wheather coding following better pratices

or not

Static Testing is also called Code Review Tool

What is Dynamic Testing?

Under Dynamic Testing, a code is executed. It checks for functional behavior of software system, memory/cpu usage and overall performance of the system. Hence the name "Dynamic"

The main objective of this testing is to confirm that the software product works in conformance with the business requirements. This testing is also called an Execution technique or validation testing.

Dynamic testing executes the software and validates the output with the expected outcome. Dynamic testing is performed at all levels of testing and it can be either black or white box testing.

eg: selenium, Junit, Mockito,jcoco JMeter and etc..

note:: SonarQube is useful in static testing done on the source of the application

to generate the review on the source code .. So it is called static code analysis tool

Reasons To Use Static Code Analysis

(Need of Code Reviewing)

Here the the app/code will be executed

for testing to check whether expected results are matching with actual results or not

white box testing ::: Code is visible while doing testing

(Testcases are designed by seeing the source code) :: unit testing (It is also called Developers Testing)

Black Box Testing:: Code is not visible while doing testing

(Testcases are designed with out worrying about source code by

worryingttional document / Requirements Document :: Selenium, Jmeter) (TEster Testing)

Finds errors earlier in development

Detects overcomplexity in code (refactoring/simplification)

Finds Security Errors

Finding errors or bad pratice before running the code helps us to write better code time to time

→if the target method is called through lengthy method chain

then it suggest us to reduce the method chain to simplify the code (one usecase)

if it finds username, password as hardcoded details in the source code then it says there is security Error

(common)

Enforces Best Coding Practices

Automated & Integrates in Jenkins

Checks wheather code is following coding standards or not

use

CI :: Code Integration

CD:: Code Delivery/Deployment

we can Jenkins for CI/CD operation in automated env.. linking SonarQube with Jenkins automates the static code analysis for every code change

Can create proiect specific rules

Technical understanding

Project specific coding standards /profiles can be linked and tested in the execution of the Project.

It directly translates as the implied costs for additional rework that can occur if at an early stage an easy but not efficient solution is chosen. In the future the easy code may restrict scalability.

Developer

Writes code

1011011 1110101

Writes code

Complex Solution

(using best pratices) (With code review)

Easy Solution

(with out using best pratices)

(With out Code Code Review)

Deployment

Deployment

After Some time

should be

recomanded to use

Easy expansion

After Some time

X

Issues! Needs rework

(Give problems in Project maintainance and Expansion)

1

What SonarQube is?

As may you have already guess SonarQube is a static analysis code tool. It

basically goes through developers' code and identifies errors at the early (before compilation)

stage. It is an open-source static testing analysis software. It is used by developers to manage source code quality and consistency. Some of the code quality checks are:

=>SonarQube is Static Code Analysis Tool or Code Review Tool

|----> "null" in ref variable may give NullPointerExeption

• Potential Bugs (the possibility of raising error if the code is compiled or executed) |---> "type casting" in the code may give ClassCastException

• Code defects to design inefficiencies - Identifies the code which is not

compatible with the design structure of the application.

• Code duplication - Code duplications take a lot of memory. The tool can identify those things.

• Lack of Test Coverage — There maybe we are not enough tests written to

•

application. The tool can identify those things.

Excess complexity — Tool can identify a much complex code segments. (Identifies method chaining areas)

Features of SonarQube

• It can work in 25 different languages. (Java, .NET, JavaScript, COBOL,

PHP, Python, C++, Ruby, Kotlin and Scala)

2. Identify tricky issues.

Though it is developed in Java, it can perform static code analysis on 25+ languages code.

Detect Bugs - SonarQube can detect tricky bugs or can raise on pieces of code that it thinks is faulty.

eg:: not initialization final variable at the time of declaration

Code Smells - Code smells are the characteristics of a code that indicates that there might be a problem caused by the code in the future. But smells aren't necessarily bad, sometimes they are how the functionality works and there is nothing that can be done about it. This is something called best practices.

Security Vulnerability - SonarQube can detect security issues that code may face. As an example If a developer forgets to close an open a SQL database OR If important details like username and password have been directly written in the code. Then SonarQube can identify these things. Because leaving SQL database open can cause issues in the source code and you definitely do not want to write username and password directly in the code. You should inject them.

Activate Rules Needed - You can create and maintain different sets of rules that are specific to particular projects, these are known as Quality Profiles. This means a team or project should follow specific rules. Then we can create a Quality profile in SonarQube.

Execution Path - Whenever there is Data flow in your program, and there is a lot of involvement between the different Modules. SonarQube can figure out if there are any tricky bugs in these execution paths. When a company works on an application there obviously have a code pipeline a data flow in the program. SonarQube when it integrated to Jenkins or any deployment tool it works by itself it keeps looking on errors and bugs. Sometimes SonarQube identifies these tricky bugs in these pathways. Suppose an error that depends on Module that is way back in the code pipeline or way back in the data flow in the program then can figure out the integration error that happens between these. (solution for the above problem)

T

Enhanced Workflow (Ensure Better CI/CD)

Automated Code Analysis - Keep working in the background from the development phase itself, monitoring and identify errors. SopnarQube can be automated by integrating with the deployment tool or integration tool and

it will keep working on the background and it finds all the errors, the Code Smells, Technical Dept by itself.

Get access through Webhooks and API - To initiate tests do not need to come to SonarQube directly, we can do that through an API call. You do not need to install SonarQube directly. You can just use APIs and call them.

Integrate GitHub - It can be directly integrated with your choice of version control software. You can find errors as well as the version of the code you are using.

Analyze branches and Decorate pull requests - It gives us a branch Level analysis. As an example, it does not just analyze the master branch it also analyzes the other branches, identifying any errors.

- Built-in methodology

Discovery Memory Leaks - It can show the memory leaks in your application if the application has a tendency to fail or go out of memory. This generally will happen slowly happen over a period of time.

Good Visualizer - It has a good way visualizing, it gives simple overviews of the overall health of the code. After the code has been developed a proper record of how the core is been performing created by SonarQube and it will be presenting on the Dashboard. So the team Lead or the Developer himself can go through it.

Enforces a quality gate - It can enforce a quality gate, you can tell SonarQube based on your requirements and practices what code is wrong and what is correct.

eg:: using deprecated features with out replacing them with alternates

eg:: do not hardcode Db user name, password

in the Source Code .. prefer getting them from other place like properties file, xml file in the form of encrypted content

Project specific code review rules which can be pre-defined rules or user-defined rules or mix of both

CI/CD ===

CI :: Continuos Integration

CD:: Continuos Deployement Delivery

Readymade Restful webservices/APIs are given

to take source code as the input for static code analysis

呆

(.java file)

(sonar qube)

-->code development ---> static code analysis

Developer

Gives the detailed report on the static code analysis

(java compiler)

Code compilation

Code execution

(output generation)

↓

Dynamic Code testing

(unit testing & other testings)

Digs into issues - If it shows that there is a problem SonarQube allows you

to go and directly check it out from the summary report or from one code

file to another. In the SonarQube summary dashboard, you can see

furthermore details of the errors by just clicking on the error.

Plugins for IDEs - It has a plugin called "SonarLint" which helps SonarQube

to integrate itself with an IDE. Which means there is no need to install the whole SonarQube package.

To use sonar qube in eclipse

through, the sonarLint should

be installed as the plugin eclipse IDE

DevOps VS DevSecOps

Development

IT Operations

Parameters

(Build operations)

Development

IT Operations

(Build operations)

Application Delivery

Security

Application Delivery

DevSecOps DevOps + Security

DevOps

(This security will be applied in all phases of SDLC)

|-->Software Development Life cycle

DevSecOps

Culture

Promotes a work culture or shared responsibility for development and operations.

Emphasizes security, extending the culture of shared responsibility.

Treatment of security

Usually implements security at Introduces security into the continuous integration and continuous

the end of the SDLC.

Combines modern DevOps Security tools pipelines with traditional security methods.

Efficiency

Often results in security bottlenecks and technical debt due to slow feedback loops.

development (CI/CD) pipeline.

Requires teams to embrace new security tools and techniques. The DevOps process must include security tools and controls from the beginning, adapting security to the CI/CD workflow.

Reduces vulnerabilities in production, minimizing the costs of addressing security issues and bugs. It enables scalability without compromising security, placing secure code as a core DevOps

objective.

Automation

Automates development processes but relies on a human team to handle security

Brings security to all stages of the development and delivery process, embracing automation to accelerate security tasks.

note:: DevOps/DevSecOps are enhanced Agile SDLC methology