Module 5- Continuous Inspection with Jenkins

Code Analysis

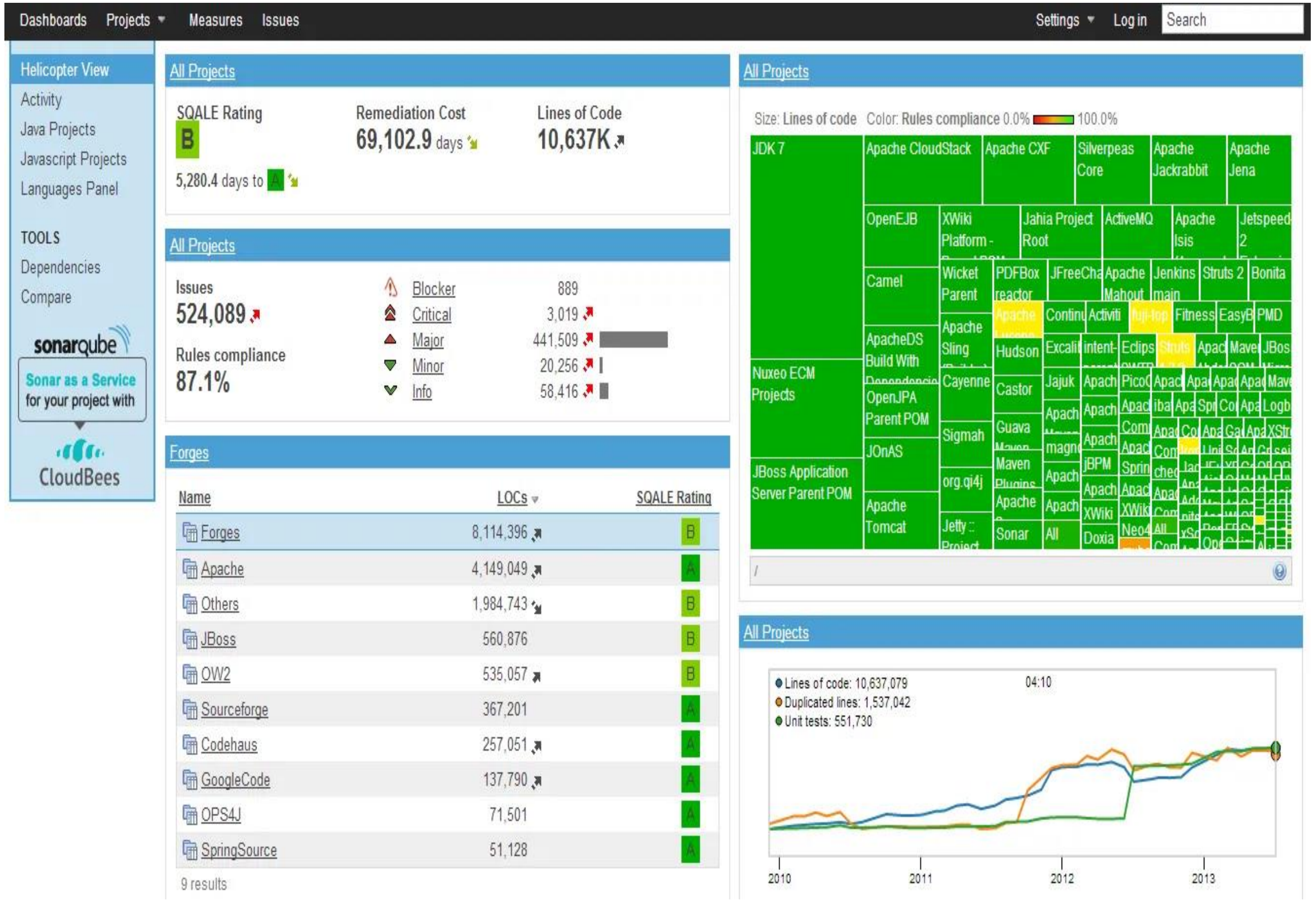
- Jenkins has a host of Code Analysis plugins
- Plugins provides utilities for the static code analysis plugins. Jenkins can parse the results file from various Code Analysis tools such as CheckStyle, FindBugs, PMD etc. For each corresponding code analysis tool, a plugin in Jenkins needs to be installed.
- Additionally the add-on plugin Static Analysis Collector is available that combines the individual results of these plugins into a single trend graph and view.

Plugins

The plugins can provide information such as:

- The total number of warnings in a job
- A showing of the new and fixed warnings of a build
- Trend Reports showing the number of warnings per build
- Overview of the found warnings per module, package, category, or type
- Detailed reports of the found warnings optionally filtered by severity (or new and fixed)

SonarQube



All Projects

SQALE Rating	Remediation Cost	Lines of Code
B	69,102.9 days	10,637K
5,280.4 days to A		

All Projects

Issues
524,089

Rules compliance
87.1%

Blocker	889
Critical	3,019
Major	441,509
Minor	20,256
Info	58,416

Forges

Name	LOCs	SQALE Rating
Forges	8,114,396	B
Apache	4,149,049	A
Others	1,984,743	B
JBoss	560,876	B
OW2	535,057	B
Sourceforge	367,201	A
Codehaus	257,051	A
GoogleCode	137,790	A
OPS4J	71,501	A
SpringSource	51,128	A

9 results

All Projects

Size: Lines of code Color: Rules compliance 0.0% 100.0%

All Projects

- Lines of code: 10,637,079
- Duplicated lines: 1,537,042
- Unit tests: 551,730

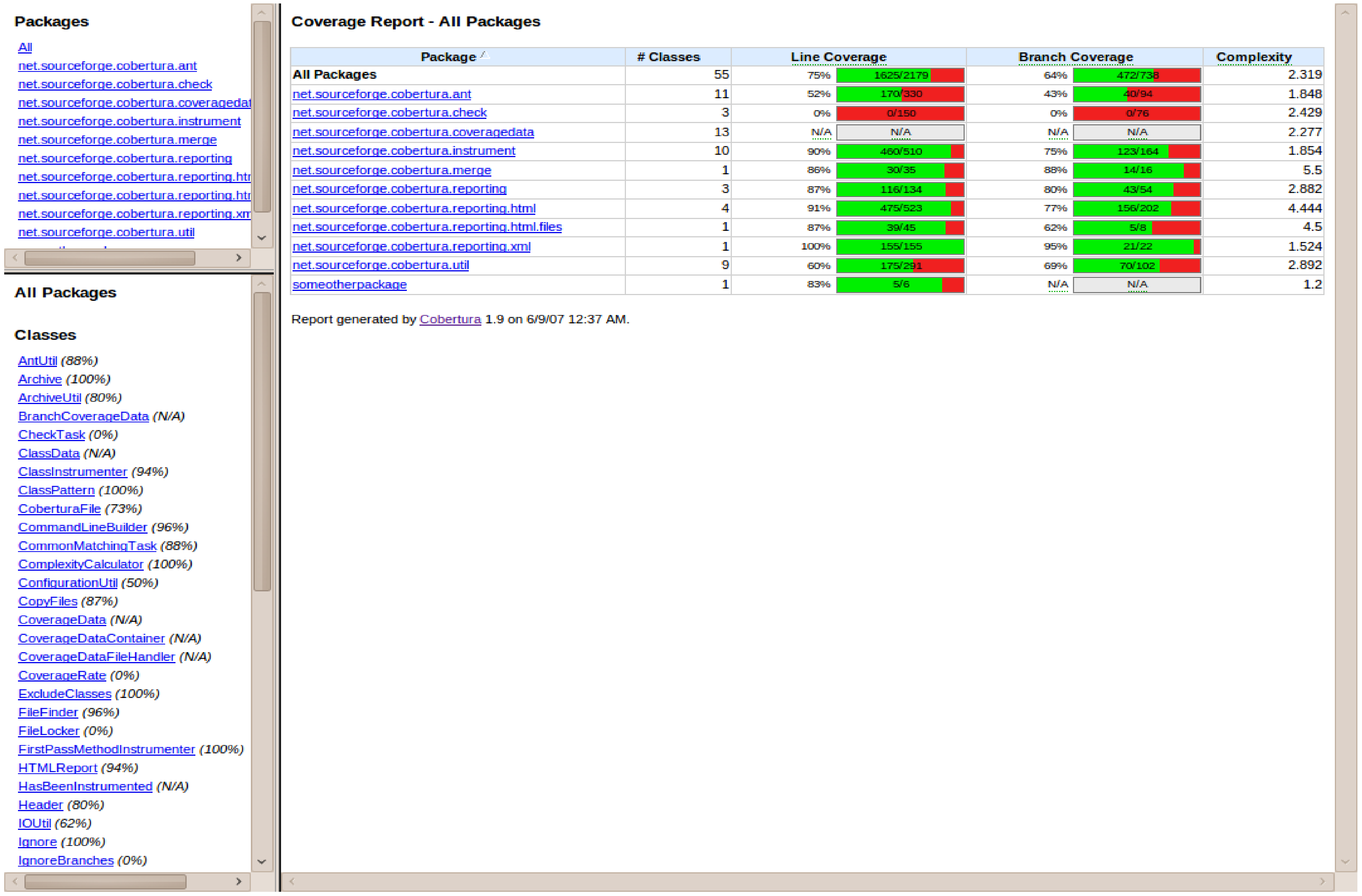
Code Coverage

- Code coverage is an indication of how much of your application code is executed during your tests—it can be a useful tool for finding areas of code that have not been tested by your test suites

- Cobertura is a tool that calculates the percentage of code accessed by tests.

- It can be used to identify which parts of your Java program are lacking test coverage. It is based on JCoverage.

Cobertura – Code Coverage Results



Packages

All
net.sourceforge.cobertura.ant
net.sourceforge.cobertura.check
net.sourceforge.cobertura.coveragedat
net.sourceforge.cobertura.instrument
net.sourceforge.cobertura.merge
net.sourceforge.cobertura.reporting
net.sourceforge.cobertura.reporting.htr
net.sourceforge.cobertura.reporting.htr
net.sourceforge.cobertura.reporting.xm
net.sourceforge.cobertura.util

All Packages

Classes

AntUtil *(88%)*
Archive *(100%)*
ArchiveUtil *(80%)*
BranchCoverageData *(N/A)*
CheckTask *(0%)*
ClassData *(N/A)*
ClassInstrumenter *(94%)*
ClassPattern *(100%)*
CoberturaFile *(73%)*
CommandLineBuilder *(96%)*
CommonMatchingTask *(88%)*
ComplexityCalculator *(100%)*
ConfigurationUtil *(50%)*
CopyFiles *(87%)*
CoverageData *(N/A)*
CoverageDataContainer *(N/A)*
CoverageDataFileHandler *(N/A)*
CoverageRate *(0%)*
ExcludeClasses *(100%)*
FileFinder *(96%)*
FileLocker *(0%)*
FirstPassMethodInstrumenter *(100%)*
HTMLReport *(94%)*
HasBeenInstrumented *(N/A)*
Header *(80%)*
IOUtil *(62%)*
Ignore *(100%)*
IgnoreBranches *(0%)*

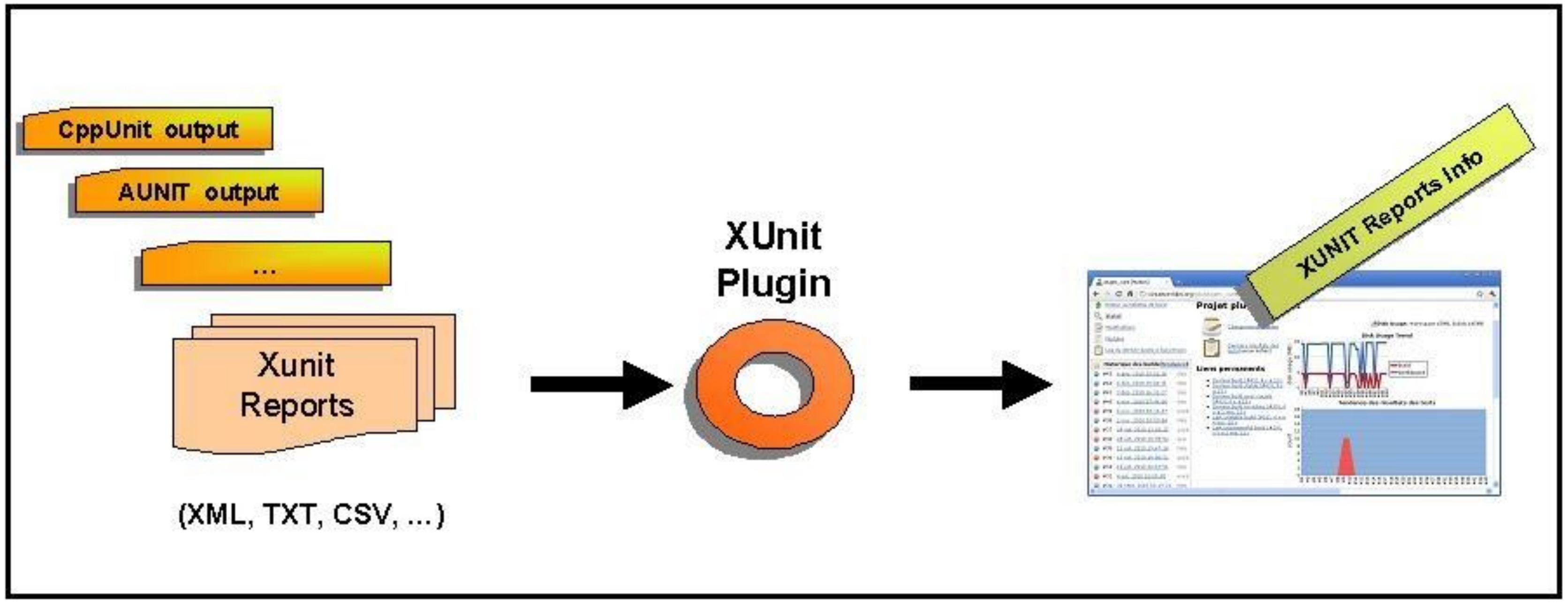
Coverage Report - All Packages

Package	# Classes	Line Coverage		Branch Coverage		Complexity
All Packages	55	75%	1625/2179	64%	472/738	2.319
net.sourceforge.cobertura.ant	11	52%	170/330	43%	40/94	1.848
net.sourceforge.cobertura.check	3	0%	0/150	0%	0/76	2.429
net.sourceforge.cobertura.coveragedata	13	N/A	N/A	N/A	N/A	2.277
net.sourceforge.cobertura.instrument	10	90%	460/510	75%	123/164	1.854
net.sourceforge.cobertura.merge	1	86%	30/35	88%	14/16	5.5
net.sourceforge.cobertura.reporting	3	87%	116/134	80%	43/54	2.882
net.sourceforge.cobertura.reporting.html	4	91%	475/523	77%	156/202	4.444
net.sourceforge.cobertura.reporting.html.files	1	87%	39/45	62%	5/8	4.5
net.sourceforge.cobertura.reporting.xml	1	100%	155/155	95%	21/22	1.524
net.sourceforge.cobertura.util	9	60%	175/291	69%	70/102	2.892
someotherpackage	1	83%	5/6	N/A	N/A	1.2

Report generated by Cobertura 1.9 on 6/9/07 12:37 AM.

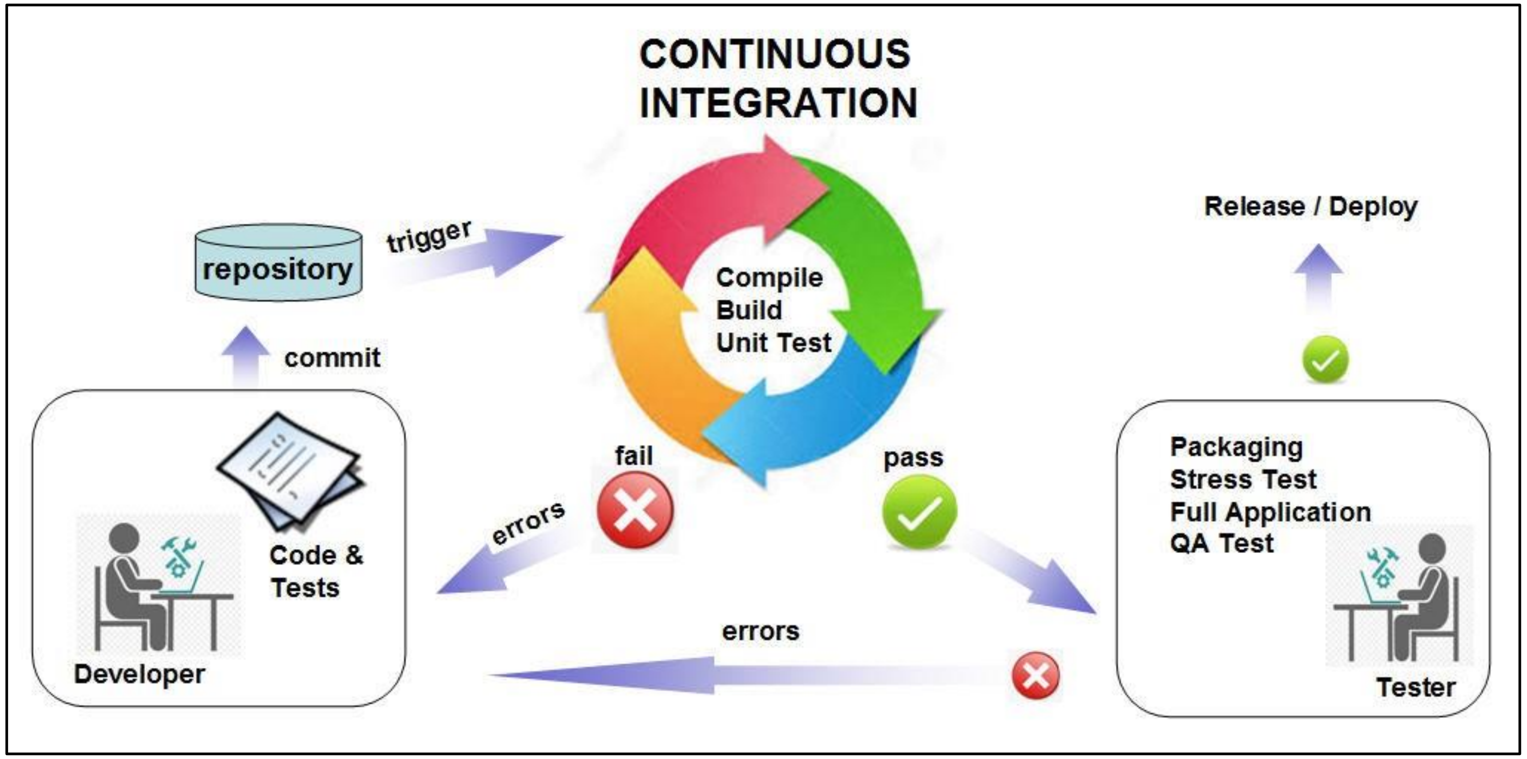
Unit Testing with Jenkins

- Jenkins provides an out of box functionality for Junit, and provides a host of plugins for unit testing for other technologies.
- List of Unit Testing plugins -https://wiki.jenkins-ci.org/display/JENKINS/xUnit+Plugin
- This plugin makes it possible to publish the test results of an execution of a testing tool in Jenkins.



Automated Testing with Jenkins

Writing automated tests for Jenkins and its plugins is important to ensure that everything works as expected while helping to prevent regressions from being introduced in later releases.



Ref: https://www.jenkins.io/doc/developer/testing/

Advanced Notifications

In Jenkins notification endpoints can be defined. Each endpoint can be configured with:

- "Format" : notification payload format, JSON or XML.
- "Protocol": protocol to use for sending notification messages, HTTP, TCP or UDP.
- "Event": Job events that trigger notifications: Job Started, Job Completed, Job Finalized or All Events (the default option).
- "URL": URL to send notifications to. It takes the form of "http://host" for HTTP protocol, and "host:port" for TCP and UDP protocols.
- "Timeout": Timeout in milliseconds for sending notification request, 30 seconds by default.

Ref: https://plugins.jenkins.io/notification/

THANKS