

WHITEPAPER

Efficient Testing Strategies with Team Testing and CI/CD in CT

Document Information

Document

Distribution date December 20, 2023

Technical Support

Email help@suresofttech.com

Phone +82-31-606-2000

Legal Notice

© 2023. Suresofttech Inc. All rights reserved.

Proprietary and Confidential This material is proprietary to Suresofttech Inc. It contains trade secret and confidential information which is solely the property of Suresofttech Inc. This material is for client's internal use only. This material shall not be used, reproduced, copied, disclosed, and transmitted, in whole or in part, without the express consent of Suresofttech Inc.

Reproduction and distribution without permission from Suresofttech Inc. is prohibited.

For suggestions or inquiries regarding this document or product, please contact Suresofttech Inc.

Some images included in this document may differ slightly depending on the version. The image corresponding to the function being described is identical to the version specified in the document information.

Table of Contents

1	Overview	4
2	Introduction to the Tool	5
	Introduction to CT	5
3	Introduction to Team Testing	6
	Problem Recognition and Improvement	6
	Maximizing Efficiency	6
4	Software Development in the CI/CD Environment	7
	Definition and Importance of CI/CD	7
	Role of CI/CD in Modern Software Development Processes	7
5	Feature Introduction and Usage	8
	Team Testing	8
	Creating and Sharing Team Projects	8
	Uploading Work to Team Testing Server	10
	Downloading Work Updated by Other Users	11
	Checking Results	12
	Integration with CI/CD Environments	13
	Command Line Interface (CLI) Features	13
	Using CT in a CI/CD Environment	14
6	Conclusion	18

1 Overview

Software development plays a more crucial role today than ever before. However, with this significance comes challenges for development teams, such as increased complexity, rapidly changing technological environments, and heightened user expectations. To overcome these challenges, we must seek faster, more efficient, and more accurate methods of development.

This white paper explores how CT can respond to modern challenges and innovatively transform the software development process. CT's latest team testing and CI/CD integration features play a vital role in meeting complex development needs, enhancing collaboration among teams, and shortening the development cycle.

The team testing functionality assists team members in collaborating efficiently, reducing redundant tasks, and sharing information in real-time. The integration with CI/CD enables automated testing and deployment processes, aiding in the continuous quality assurance of software.

2 Introduction to the Tool

Introduction to CT

CT is an automated unit and integration testing tool for C/C++ code, provided by Suresofttech Inc. It is designed to facilitate software testing and achieve structural code coverage objectives in mission-critical areas where safety and reliability are paramount. CT focuses on enhancing software quality, reducing testing and maintenance costs, and maximizing developer productivity.

CT enhances testing efficiency for large-scale and complex software projects and is certified for reliability by TÜV. It has proven its effectiveness in various industries, including automotive, defense, railway, nuclear, finance, and telecommunications. CT has established itself as an essential tool for high-level requirement testing.

3 Introduction to Team Testing

The software development environment is rapidly changing. As user requirements become more diverse and complex, the scale of software is also increasing. Additionally, to maintain competitiveness in the market, swift updates and deployment are necessary. This requires performing numerous tests in a short period. To address this, the team testing feature has been added to CT. The team testing functionality allows multiple testers to participate simultaneously in a single software project, thereby enhancing the collaborative process of software testing. This feature enables more efficient testing, accommodating the expanding scope and complexity of software development.

Problem Recognition and Improvement

When multiple testers are involved in the same project, sharing the test environment and test cases for dynamic testing is not a trivial issue. Additionally, there are problems such as duplicated efforts among testers, especially in creating test assets like stubs or mock objects. This leads to a waste of time and resources. Moreover, comprehending the overall progress of the tests and merging the results is a complex and cumbersome task.

The team testing feature has been developed to solve these issues. By synchronizing the entire configuration of the test project, including project settings, stubs, and mock object codes, all participants can work with the most up-to-date information. This eliminates redundant work and saves time.

Maximizing Efficiency

Team Testing Server compiles test results on the server and presents them on a dashboard in the form of a web page. This dashboard allows real-time access to various information, such as code coverage and test results. It enables easy tracking of the overall progress of the project and significantly reduces inefficiencies in the distribution and collection of test work.

The team testing feature elevates the collaboration and efficiency of software testing to a new level, enhancing project management and teamwork quality. This feature is a powerful tool that helps companies ensure software quality while reducing development time and costs.

4 Software Development in the CI/CD Environment

Definition and Importance of CI/CD

CI/CD, standing for Continuous Integration and Continuous Deployment/Delivery, is a core element of modern software development. Continuous Integration involves developers regularly merging code changes into a central repository, enabling early detection and resolution of issues in the development phase. Continuous Deployment automates the testing of these changes and their deployment to operational environments, facilitating rapid and efficient product updates.

Adopting CI/CD is crucial for improving software quality, shortening deployment times, and fostering collaboration between development and operations teams. This approach enables agile responses to changing customer needs and continuous service improvement.

Role of CI/CD in Modern Software Development Processes

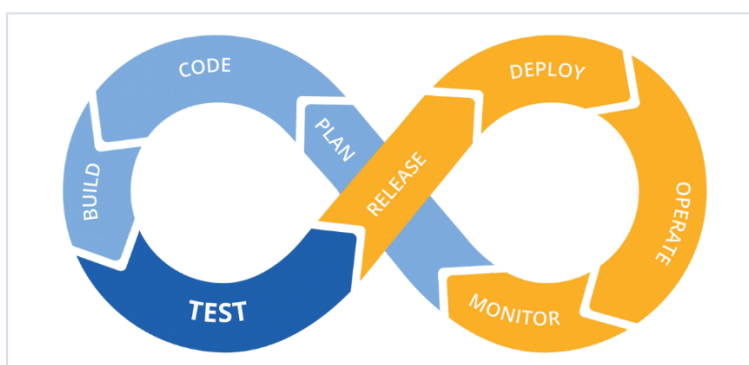


Figure 1 Software Development Process in CI/CD Environment

In Agile and DevOps environments, CI/CD is indispensable. Agile development aims for rapid response to change and continuous improvement, where CI/CD plays a vital role in implementing these Agile principles. In DevOps culture, the barriers between Development and Operations are broken down, pursuing efficient software deployment and management through close cooperation between the two areas. CI/CD acts as a critical means to achieve these DevOps goals.

CI/CD enhances both development speed and quality, providing flexibility to respond quickly to market changes. It plays a significant role in refining software development and deployment processes, contributing to the provision of competitive products and services.

5 Feature Introduction and Usage

Team Testing

The core functionality of team testing includes sharing CT projects, tests, and test assets among testers. This asset-sharing feature follows the usage scenarios of version control systems, familiar to developers and testers.

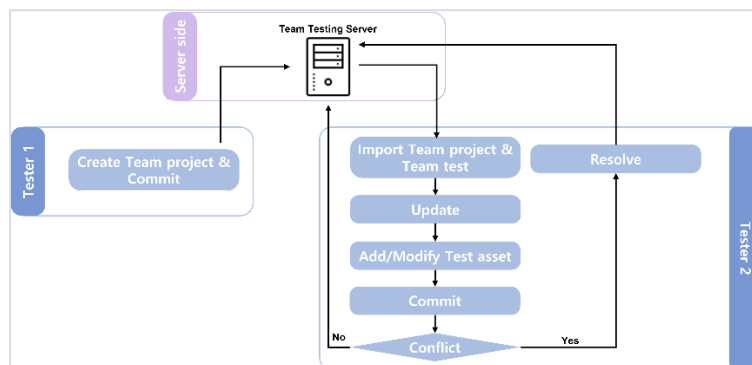


Figure 2 Operation Sequence of Team Testing Functionality

Creating and Sharing Team Projects

Team projects can be created through the [C/C++ Team Project] option added to the project creation menu in CT 2023. To create a team project, it is necessary to first set Team Testing Server information in the settings. When you use this feature to create a team project, it is automatically uploaded to the team project server. (Commit)

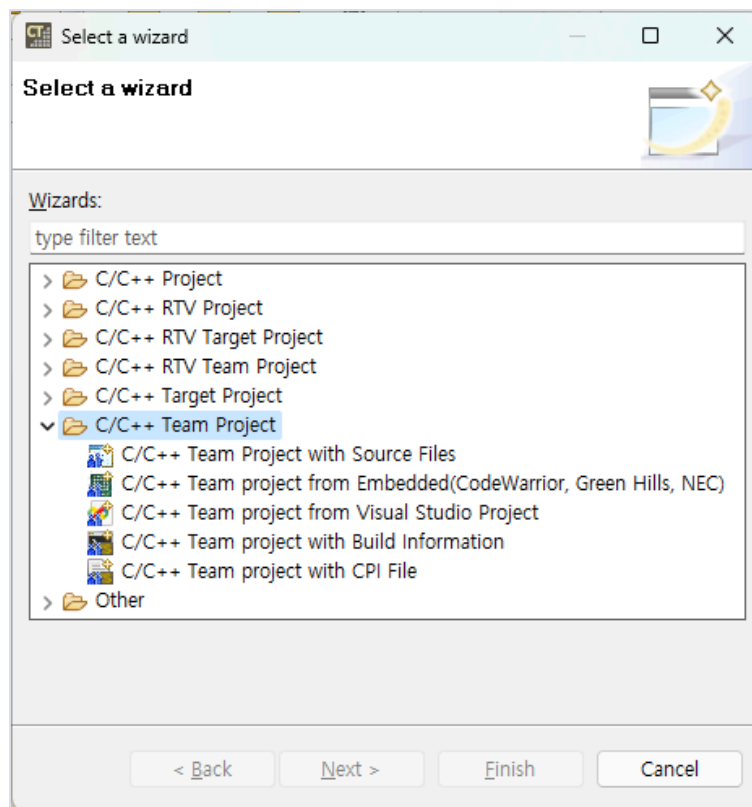


Figure 3 Team Project Creation Menu

To use a team project on another PC, you can import the project registered on the server to the desired PC using the [Import] → [Team Testing] → [Import Team Project] function. (Import)

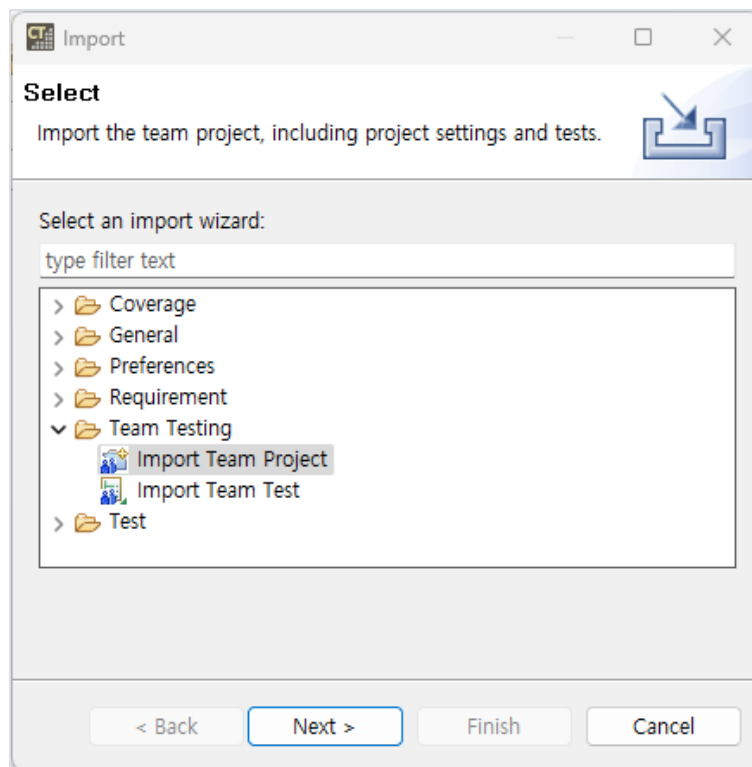


Figure 4 Importing Team Projects Menu

Uploading Work to Team Testing Server

Local work conducted in CT during testing is uploaded to Team Testing Server for sharing with other users. This process, known as committing (Commit), ensures that all participants in a team project have access to the latest updates and developments.

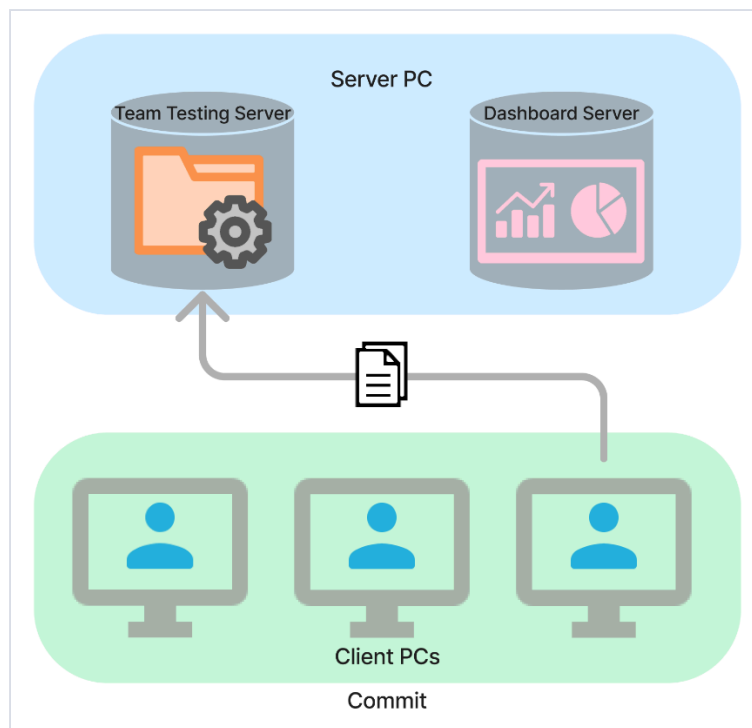


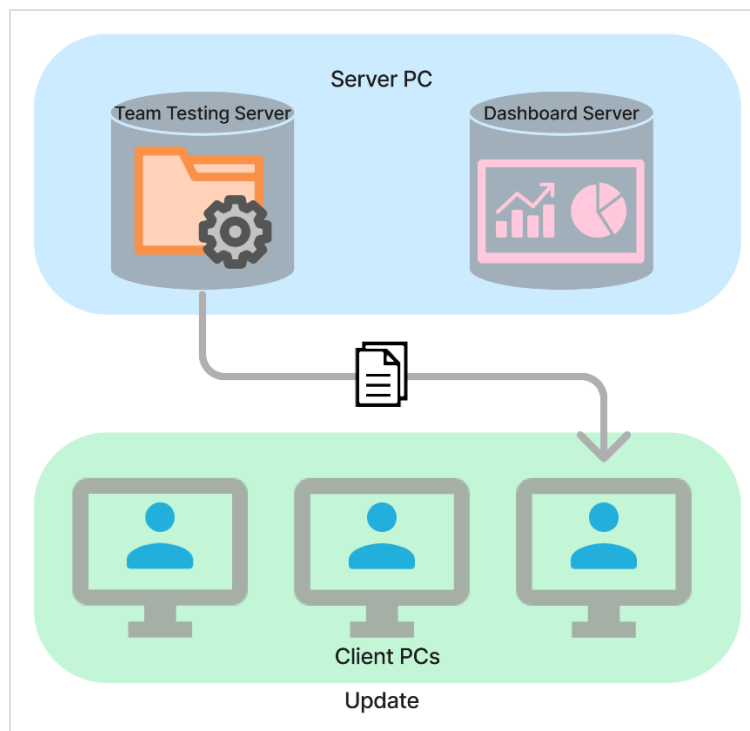
Figure 5 Commit

The work content in the CT Client is committed to Team Testing Server in two cases:

- When executing a test
- When the user manually commits project changes (Manual Commit)

Downloading Work Updated by Other Users

When shared resources or project configuration information changes, it is necessary to download the updates from Team Testing Server to the local project. This function is referred to as updating (Update).

**Figure 6** Update

There are three scenarios for updating:

- Automatically updating according to the user-set update frequency
- Automatically updating when attempting to connect to the server, such as when opening or committing a team project
- Manually performed by the user

When the updating process encounters resources modified in the local CT, it is termed a conflict (Conflict). In cases of conflicts or significant changes requiring user confirmation, an update dialog box appears. Through this dialog box, conflicts can be resolved and the contents committed to Team Testing Server can be reflected in the local project.

There are two ways to resolve conflicts:

- Directly reflecting the server configuration
- Marking as resolved (manually merging changes and then reflecting them on the server)

Checking Results

During the team testing process, information on team project resources, test results, and coverage trends can be viewed on the dashboard.

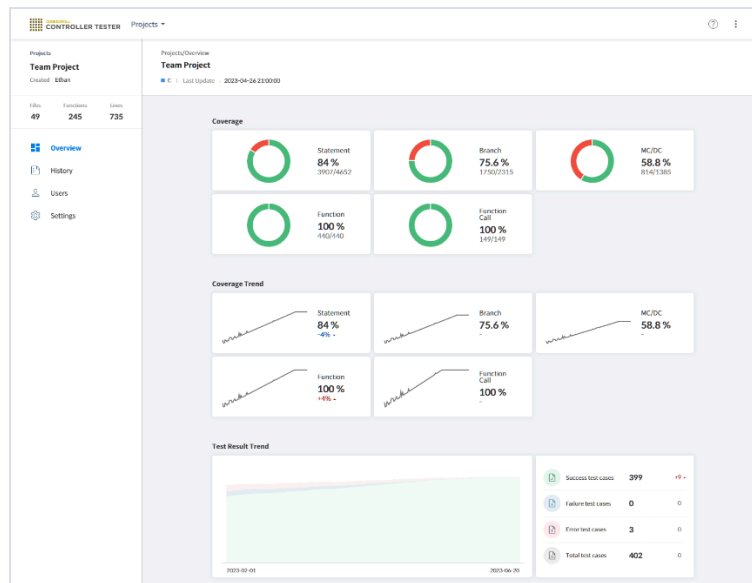


Figure 7 Team Dashboard

Integration with CI/CD Environments

CT provides a Command Line Interface (CLI) to perform tests in CI/CD environments. Integration can be carried out by setting the CLI to be called in the CI/CD tool settings. Additionally, CT offers a dedicated plugin for Jenkins, which is widely used among CI/CD tools.

Command Line Interface (CLI) Features

The following functions can be performed through the command line interface provided by CT:

- Creating projects
- Executing tests
 - Uploading team project test execution results to the Team Testing Server
- Importing projects
 - Importing general projects
 - Importing team projects from Team Testing Server
- Exporting projects
- Deleting projects
- Generating reports

```

C:\Program Files\Suresoft\CT 2023>csc -e -p "d:\CT\workspace\CLI" --ini -O "D:\CT\2023\12\WhitePaper\cli\testProject_CLI.ini"
[MAIN-INFO] parse the arguments...
[MAIN-INFO] set workspace
[MAIN-INFO] install Location: C:\Program Files\Suresoft\CT 2023
[MAIN-INFO] PRODUCT_VERSION: 2023.12.012
[MAIN-INFO] workspace: D:\CT\workspace\CLI
[MAIN-INFO] initialize()
[MAIN-INFO] global database loaded.
[MAIN-INFO] execution job name: Execute CT CLI
cshome path= C:\Program Files\Suresoft\CT 2023\plugins\com.codesroll.sp.core.1.0.2.201202152119#
cshome path= C:\Program Files\Suresoft\CT 2023\plugins\com.codesroll.ut.2023.12.0#
cshome path= C:\Program Files\Suresoft\CT 2023\plugins\com.codesroll.ut.2023.12.0\bin#
cshome path= C:\Users\Sure\AppData\Roaming\CodeScroll\1\1#

[Load Information Success]

[TestProjectProject CLI start!!!]

[BEGIN : Project operation]
[END : Project operation]

[BEGIN : Analysis operation]
[START : PA]
##PersistentProperty: analyzed
[MAIN-INFO] Analyze
[MAIN-PROGRESS] 100%
[END : PA]
[START : ckdb]
[VALUE1]
[VALUE100]
[END : ckdb]
[END : Analysis operation]

[BEGIN : ExecuteTest operation]
##PersistentProperty: analyzed
[START : PA]
##PersistentProperty: analyzed
[MAIN-INFO] Analyze
[END : PA]
[START : sync]
[END : sync]
[START : tfg]
Serialization query: 69
[END : tfg]
[START : bid]
[VALUE1]
[VALUE24351]
[VALUE5]
[VALUE24353]
[VALUE45]
[VALUE24355]
[VALUE50]
[END : bid]
[START : ecn]
[VALUE5]
[VALUE24451]
[VALUE24452]<cs#delim>{(0/50)<cs#delim>function_0006_test0<cs#delim>|
[VALUE55]
[VALUE55]
[VALUE53]
[VALUE57]
[VALUE71]
[VALUE74]
[VALUE24452]<cs#delim>{(30/50)<cs#delim>function_0009_test0<cs#delim>|
[VALUE76]
[VALUE52]
[VALUE55]
[VALUE90]
[VALUE24454]
[VALUE5]
[END : ecn]
[END : ExecuteTest operation]

[MAIN-INFO] result: 'Execute CT CLI' succeeded.
[MAIN-INFO] result: succeeded.
C:\Program Files\Suresoft\CT 2023>

```

Figure 8 Executing Tests using Command Line Interface (CLI)

Using CT in a CI/CD Environment

In a CI/CD environment, software deployment typically follows these steps:

1. Updating (syncing) the source code
2. Building the source code
3. Testing
4. Deployment

During the testing phase, to perform tests using CT, it is necessary to set up the environment for CT's Command Line Interface (CLI) on the PC or server where the test is conducted. The Jenkins Plugin facilitates this setup process. Particularly for team testing projects, no additional CLI-related preparatory work is required, and tests can be easily conducted with just the team testing information and project name.

Furthermore, the plugin enables monitoring of test result trends for each test in the CI/CD environment, with more detailed information available on the dashboard provided by Team Testing Server.

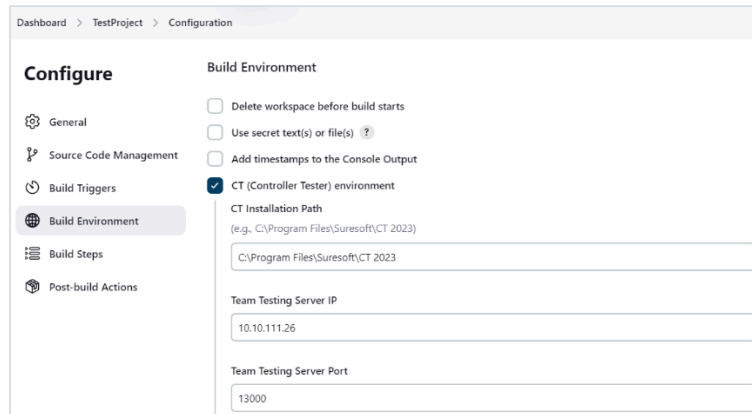


Figure 9 Build Environment Settings in Jenkins Plugin

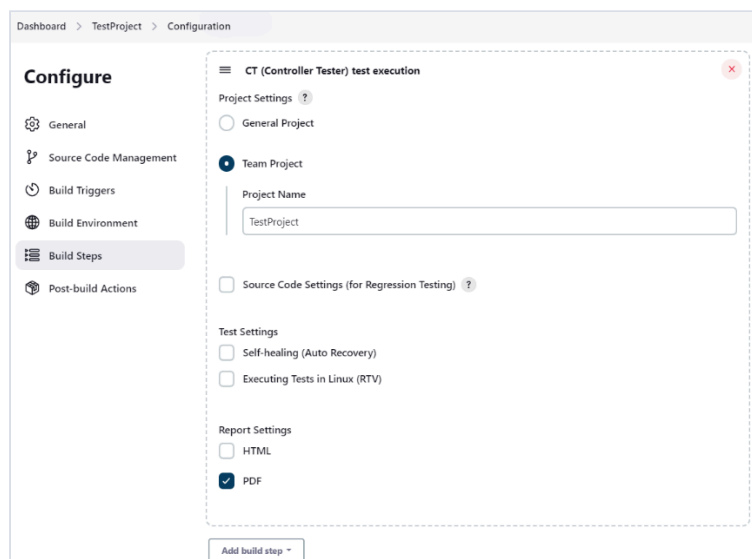


Figure 10 CT Project Test Execution Settings in Jenkins Plugin

```

***** EXECUTE TEST *****

[CT 2023] $ cmd.exe /C csc.exe -g "C:\ProgramData\Jenkins\jenkins\workspace\TestProject\ct\global" -e -w
"C:\ProgramData\Jenkins\jenkins\workspace\TestProject\ct\workspace" --ini -O
"C:\ProgramData\Jenkins\jenkins\workspace\TestProject\ct\ini\sample_cli.ini"
[MAIN-INFO] parse the arguments...
[MAIN-INFO] set workspace
[MAIN-INFO] Install Location: C:\Program Files\Suresoft\CT 2023
[MAIN-INFO] PRODUCT_VERSION: 2023.12
[MAIN-INFO] workspace: C:\ProgramData\Jenkins\jenkins\workspace\TestProject\ct\workspace
[MAIN-INFO] initialize()
[MAIN-INFO] global database loaded.
[MAIN-INFO] execution job name: Execute CT CLI
cshome path= C:\Program Files\Suresoft\CT 2023\plugins\com.codescroll.gp.core_1.0.2.201202152119\
csuthome path= C:\Program Files\Suresoft\CT 2023\plugins\com.codescroll.ut_2023.12.0\
csuthomebin path= C:\Program Files\Suresoft\CT 2023\plugins\com.codescroll.ut_2023.12.0\bin\
csglhome path= C:\ProgramData\Jenkins\jenkins\workspace\TestProject\ct\global\CodeScroll\1.1\
[TestProjectProject CLI start!!!]

[BEGIN : Project operation]
[END : Project operation]

[BEGIN : Analysis operation]
[START : PA]
##PersistentProperty: analyzed
[MAIN-INFO] Analyze
[MAIN-PROGRESS] 100%
[END : PA]
[START : ckdb]
    Setting environment for using Microsoft Visual Studio 2010 x86 tools.
    VALUE1
    VALUE100
[END : ckdb]
[END : Analysis operation]

[BEGIN : ExecuteTest operation]
##PersistentProperty: analyzed
[START : PA]
##PersistentProperty: analyzed
[MAIN-INFO] Analyze
[END : PA]
[START : sync]
    Setting environment for using Microsoft Visual Studio 2010 x86 tools.
[END : sync]
[START : tfg]
Serialization query: 536
[END : tfg]
[START : bld]
    Setting environment for using Microsoft Visual Studio 2010 x86 tools.
    VALUE1
    PHASE24351
    VALUE6
    PHASE24353
    VALUE45
    PHASE24356
    VALUE50
[END : bld]
[START : ecm]
    Setting environment for using Microsoft Visual Studio 2010 x86 tools.
    VALUE51
    PHASE24451
    codescrolllib_testrun_log
    codescrolllib_testrun_log
    PHASE24452|<cs#delim>|(0/50)|<cs#delim>|function_0001_test1|<cs#delim>|1
    VALUE55
    VALUE59
    VALUE63
    VALUE67
    VALUE71
    VALUE74
    PHASE24452|<cs#delim>|(30/50)|<cs#delim>|function_0004_test1|<cs#delim>|1
    VALUE78
    VALUE82
    VALUE86
    VALUE90
    PHASE24454
    ut:ecm::Engine::runTestUsingThread: 1.115
[END : ecm]
[END : ExecuteTest operation]
[MAIN-INFO] result: 'Execute CT CLI' succeeded.
[MAIN-INFO] result: succeeded.

```

Figure 11 Team Project Regression Test Results

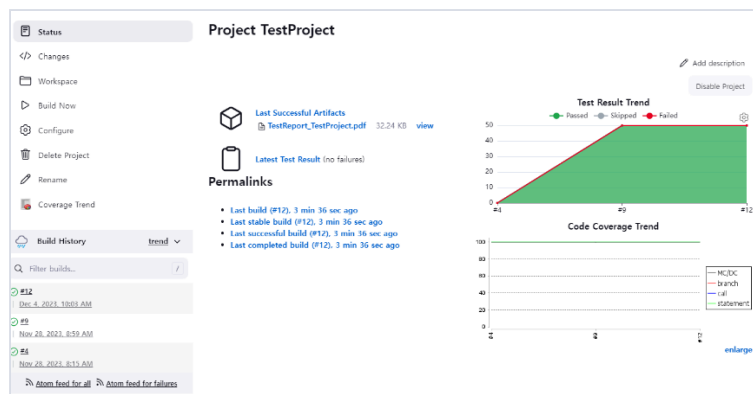


Figure 12 Software Deployment Management Outcomes via Jenkins Plugin

6 Conclusion

The team testing and CI/CD integration features of CT support developers and testers in working faster, more accurately, and more reliably. These features play a vital role in reducing the complexity of the development process, shortening time to market, and maximizing software quality, thereby enhancing a company's competitiveness.

Team testing and CI/CD integration greatly enhance software development efficiency and collaboration. Team testing allows multiple developers and testers to share common test assets in real-time, minimizing redundant tasks and boosting productivity. It also provides functionalities that facilitate the sharing and reuse of test assets. These features improve project management quality, reduce development time and costs, and ensure the quality of the final product.

The CI/CD integration feature enables developers to create, execute, import, export, and delete projects through the Command Line Interface (CLI), smoothly supporting continuous integration and continuous deployment environments. This is essential for automating the entire development cycle, including updating source code, building, executing tests, and deploying. Furthermore, it offers methods for integration with various CI/CD tools, including Jenkins, allowing developers and testers to easily set up and manage CI/CD pipelines without an in-depth understanding of the system.

Our development team plans to continue research and development to save time on testing and integrate CT more seamlessly into CI/CD environments in the future.