**Author**

Dorokhin Andrey

**Date**

17.04.2024

**Version**

1.0

**Test Plan** TP-01AD-1

**• Introduction**

The purpose of compiling this Test Plan is to describe the process of testing the “**TeamCity**” software. The document allows you to get an idea of the program testing activities.

**TeamCity** software is server software from JetBrains, written in Java, a build server to ensure continuous integration. Used mainly by developers and DevOps engineers as a powerful continuous integration and deployment tool.

**• Test Goals:**

• checking the correct operation of critical software functionality;

• achieve 100% coverage of all critical scenarios at the API and UI level;

• provide automatic monitoring and control of test metrics;

• the time spent on autoregression should not exceed 10 minutes for all platforms in total;

• UI auto testing should occur for the Chrome browser;

**• autotests should be implemented taking into account best practices:**

• generating test data;

• parameterization;

• scalability

**• Test items:**

• functionality;

• availability;

• stability;

• performance.

**• Features to be tested:**

• creating a user;

• user logging;

• creating basic build configurations;

• build execution;

• obtaining the correct result of the completed build.

**• Approach**

**• In the process of testing the TeamCity software, the following types of testing will be used:**

• Black Box;

• Positive;

• Negative;

• Performance;

• Functional.

**• Next test data will be used:**

• credentials for several users;

• several variations of commands that create configuration builds;

• data on the expected results of executing prepared build configurations.

**• Item pass/fail criteria:**

• requirement coverage for this program is 100%;

• 90% of tests passed with a pass result;

• the test report was compiled by AQA Lead and approved by the product owner.

**• Suspension criteria and resumption requirements:**

• the criterion for stopping testing is failure to pass the smoke test;

• the requirement to resume testing is to re-pass the smoke test.

**• Test deliverables:**

• auto tests;

• test report.

**• Environmental needs**

• IntelliJ IDEA;

• Chrome browser v.122 or higher;

• Java 11;

• Docker.

**• Strategy**

**• Automation technologies:**

**• API:** Java + Rest Assured

**• UI:** Java + Selenide

**• Table of responsibilities and deadlines:**

**Task**

**Task Responsible**

**Task Deadline**

Test framework development

AQA Team Lead - Andrey Dorokhin

05.05.2024

Setting up a test stage

DevOps Engineer - Andrey Dorokhin

05.05.2024

Creation of additional endpoints for testing functionality

Backend developer - Andrey Dorokhin

05.05.2024

Creating Test Data

AQA Team Lead - Andrey Dorokhin

06.05.2024

Setting up metrics

AQA Team Lead - Andrey Dorokhin

10.05.2024

Creating a user

AQA Team Lead - Andrey Dorokhin

15.05.2024

User logging

AQA Team Lead - Andrey Dorokhin

21.05.2024

Creating basic build configurations

DevOps Engineer - Andrey Dorokhin

01.06.2024

Build Execution

AQA Team Lead - Andrey Dorokhin

07.06.2024

Getting the correct result of the completed build

AQA Team Lead - Andrey Dorokhin

13.06.2024

Monitoring test metrics

AQA Team Lead - Andrey Dorokhin

15.06.2024

Generating a testing report

AQA Team Lead - Andrey Dorokhin

16.06.2024

**• Schedule**

• Completion of work – 17.06.2024.

**• Risks and contingencies:**

• testability;

• functionality;

• availability;

• stability.

**• Approvals:**

Product owner

Alex Pshe

AQA Team Lead

Andrey Dorokhin

Ok

DevOps Engineer

Andrey Dorokhin

Ok

Backend developer

Andrey Dorokhin

Ok