

|  |
| --- |
|  |
| RP - DASHBOARD |

| RELATED ARTIFACTS | |
| --- | --- |
| Ref. | Artifact Name |
| [GLO] | [Common Glossary](https://info.epam.com/acronyms.html) |
| [SC] | [Process Description: Software Construction](https://epam.sharepoint.com/sites/policy/SitePages/Policy/Process%20Descriptions/Process%20Description%20Software%20Construction.aspx) |
| [ST] | [Process Description: Software Testing](https://epam.sharepoint.com/sites/policy/SitePages/Policy/Process%20Descriptions/Process%20Description%20Software%20Testing.aspx?path=Delivery%20Process%20Areas) |
| [DT] | [Process Description: Dynamic Testing](https://epam.sharepoint.com/sites/policy/SitePages/Policy/Process%20Descriptions/Process%20Description%20Dynamic%20Testing.aspx) |

|  |  |
| --- | --- |
| Acronyms & Abbreviations | |
| SEPG | Software Engineering Process Group |
| QMS | Quality Management System |
| POC | Proof of Concept |
| MVP | Minimum Viable Product |
| SOW | Statement of Work |
| DM/PM | Delivery Manager/Project Manager |

Contents

[1 Project specific impact to testing 4](#_Toc131519691)

[2 Scope of TESTING 4](#_Toc131519692)

[2.1 In scope 4](#_Toc131519693)

[2.2 Out of scope 4](#_Toc131519694)

[2.3 Third-Party systems 5](#_Toc131519695)

[2.4 System tested by other organizations 5](#_Toc131519696)

[3 Quality and Acceptance Criteria 5](#_Toc131519697)

[4 Test process description 5](#_Toc131519698)

[4.1 Test planning phase 5](#_Toc131519699)

[4.1.1 Entry/exit criteria for each phase 5](#_Toc131519700)

[4.1.2 Defects severity description 6](#_Toc131519701)

[4.2 Test design phase 6](#_Toc131519702)

[4.2.1 Test cases creation rules 6](#_Toc131519703)

[4.2.2 Managing test cases in the test management tool 7](#_Toc131519704)

[4.3 test execution phase 7](#_Toc131519705)

[4.3.1 Defect reports raising and management rules 7](#_Toc131519706)

[4.3.2 Defect lifecycle 7](#_Toc131519707)

[4.4 Test reporting phase 7](#_Toc131519708)

[4.4.1 Test report containment 7](#_Toc131519709)

[4.4.2 List of the metrics to track on a project 7](#_Toc131519710)

[5 Test Strategy 8](#_Toc131519711)

[5.1.1 Definition Of Ready 8](#_Toc131519712)

[5.1.2 Definition of Done 8](#_Toc131519713)

[5.1.3 Automated Testing types 8](#_Toc131519714)

[5.1.4 Automated Testing levels 8](#_Toc131519715)

[6 Risk management 8](#_Toc131519716)

[7 test Team 9](#_Toc131519717)

[7.1 Roles and Responsibilities 9](#_Toc131519718)

[7.2 STAKEHOLDERS 9](#_Toc131519719)

[7.3 Communication plan 9](#_Toc131519720)

[7.4 Escalation plan 9](#_Toc131519721)

[8 Test schedule 10](#_Toc131519722)

[9 TEST Deliverables 10](#_Toc131519723)



# Project specific impact to testing

This test plan has been created to communicate the test approach to automate Dashboard feature of Report Portal to the stakeholders. It includes the objectives, scope, schedule, risks & approach. Report portal is web-based application contains Dashboard feature which is one stop solution for continuous testing. AQA Team is responsible to automate the testing of this feature including both UI and API ensuring that user meets the acceptance criteria.

Project phase: Planning

Technology stack: Java, BDD with Cucumber, Selenium WebDriver, RESTful API, Junit, Azure DevOps, SauceLabs, Browserstack

Constrains and assumptions: Resources, Time, Server availability, UI Slowness

Critical success factors: Customer Satisfaction & feedback, product quality, continuous delivery

Project geography: N/A

# Scope of TESTING

All the below feature of Dashboard which are defined in SRS need to be automated.

## In scope

|  |  |  |  |
| --- | --- | --- | --- |
| System/component/interface under test | Description | Responsible side | Reference |
| ALL DASHBOARDS | This web page contains details of all the new & existing dashboards. | AQA Team |  |
| ADD NEW DASHBOARD | This dashboard feature enables to create new dashboard | AQA Team |  |
| UPDATE DASHBOARD | This dashboard feature used to update the existing dashboard. | AQA Team |  |
| DELETE DASHBOARD | This dashboard feature allows to delete the existing dashboard. | AQA Team |  |
| DASHBOARD CONTROLLER - API | This dashboard controller gives the access to create dashboards, get the details of dashboard, update it, delete the dashboards. | AQA Team |  |
| ADD/REMOVE WIDGET | This dashboard feature allows you to add & remove widget. | AQA Team |  |

## Out of scope

|  |  |  |
| --- | --- | --- |
| System/component/interface under test | Description | Reference |
| Website Security and Performance testing | Security & Performance testing will be covered by dedicated team. |  |

## Third-Party systems

|  |  |  |  |
| --- | --- | --- | --- |
| System/component/interface under test | Description | Responsible side | Reference |
| Azure DevOps | Integration with Azure DevOps | AQA Team |  |
| SauceLabs | Integration with SauceLabs | AQA Team |  |
| Browserstack | Integration with Browserstack | AQA Team |  |
| MS Teams | Integration with MS Teams | AQA Team |  |

## System tested by other organizations

|  |  |  |  |
| --- | --- | --- | --- |
| System/component/interface under test | Description | Responsible side | Reference |
| NA | NA | NA | NA |

# Quality and Acceptance Criteria

The test objectives are to verify the functionality of the dashboard, the project should focus to automate all the testing operations such as add/update/delete etc. to guarantee all these operations can work normally in the real business environment.

# Test process description

API testing Approach:

AQA engineers perform automated API testing using a testing tool that drives the API. To be able to plan API tests, team need to determine testing boundaries and requirements which include feature & function of API, other API & API interacts with, the problem we are testing for, priority in testing, desired output and definition of pass and fail. Once the functional scope is determined, team will set up the API test environment. Before proceeding to automate API, team need to make sure that API is operational. Once it is confirmed, prepare the input parameters & create the API test scripts using REST Assured. Team will write the automated script using java to validate all the combinations, positive and negative scenarios.

UI Testing Approach:

UI Automation Testing will perform using Selenium Webdriver with Java and TestNg testing framework. Team will cover all in-scope functional scenarios with data driven framework. AQA team will automate the script for all the negative and positive scenarios and once test script is completed, it needs to be sent for review & then merge. Team will run the CI/CD pipeline and verify the build if it pass or fail. Team will perform the RCA for failed test scripts.

## Test planning phase

### Entry/exit criteria for each phase

Entry Criteria

All test hardware platforms must have been successfully installed, configured, and functioning properly.

All the standard software tools including the testing tools must have been successfully installed and functioning properly.

Proper test data is available.

The test environment such as, lab, hardware, software, and system administration support should be ready.

AQA resources have completely understood the requirements.

AQA resources have sound knowledge of functionality.

Reviewed test scenarios, test cases and RTM.

Exit Criteria

A certain level of requirements coverage has been achieved.

No high priority or severe bugs are left outstanding.

All high-risk areas have been fully tested, with none or only minor residual risks left outstanding.

The schedule has been achieved.

### Defects severity description

|  |  |  |
| --- | --- | --- |
| Severity | Meaning | Examples |
| *Blocker* | *Behavior causes the business/consumer to stop working. Application under test crashes or inoperable* | * *the business users are unable to continue a successful business operation in the system unless the issue is resolved;* * *loss or partial loss of key functionality;* * *system crash;* * *massive performance degradation.* |
| *Critical* | *Behavior causes the business/consumer to operate in a limited way. No workaround available.* | * *loss or partial loss of key functionality;* * *operational error;* * *data integrity;* * *some performance degradation.* |
| *Major* | *Behavior allows business/consumer to continue working in a limited way with work around.* | * *partial loss of functionality of the software, but allows the user to continue proceeding normal business operations;* * *usability/UI issues causing confusion due to inconsistency or ambiguity.* |
| *Minor* | *Behavior has no functional impact on business/consumer and can also be attributed to UI aesthetics.* | * *the business user decides that the issue around this functionality is not vital to his use of the system;* * *a minor feature that is not functional in one module but the same task is easily doable from another module.* |
| *Trivial* | *The defect does not affect functionality or data, it is merely an inconvenience.* | * *cosmetics (font face/font size/text alignment;* * *misspelling.* |

## Test design phase

### Test cases creation rules

Automated Test Case needs to be created, reviewed by team lead. Once reviewed and merged code test script should be run in the pipeline.

### Managing test cases in the test management tool

Azure DevOps will be used to manage the testing. Continuous integration performed by team in Azure devOps, Team will push their scripts in Azure Repo, once PR is created, reviewed and merged in the main branch, new build will get generate and all the new and existing tests will get executed. After every test execution test case status & tags will be updated to Automated in Test Plan section.

## test execution phase

### Defect reports raising and management rules

Azure DevOps will be used to manage the defect. Defect should be raised with all the important details e.g.,Defect Summary & Description, Steps to reproduce, Expected Behavior, Actual Behavior, Priority, Severity, Reported By, Environment.

### Defect lifecycle

New - Potential defect that is raised and yet to be validated.

Assigned - Assigned against a development team to address it but not yet resolved.

Active - The Defect is being addressed by the developer and investigation is under progress. At this stage there are two possible outcomes; viz - Deferred or Rejected.

Ready to Test - The Defect is fixed and ready for testing.

Verified - The Defect that is retested and the test has been verified by ÀQA.

Closed - The final state of the defect that can be closed after the AQA retesting or can be closed if the defect is duplicate or considered as NOT a defect.

Reopened - When the defect is NOT fixed, AQA reopens/reactivates the defect.

Deferred - When a defect cannot be addressed in that particular cycle it is deferred to future release.

Rejected - A defect can be rejected for any of the 3 reasons: viz - duplicate defect, NOT a Defect, Non-Reproducible.

## Test reporting phase

### Test report containment

Below items need to be added in the test report.:

* Test Cycle – System Test/Integration Test/Unit Test/E2E test
* Test modules, Description, % TCs Executed, % TCs Passed, % TCs Failed, %TCs Blocked, Priority, Defect, Remark
* List of the tasks with priority and status.
* Metrics to calculate for the current iteration with trendline for the last 5 releases.

### List of the metrics to track on a project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Metric name | Formula | Responsible role | Frequency | Way to provide |
| DC – defect containment | DC = (Number of non-rejected defects registered during by end-users on production environment)/(Total number of non-rejected defects registered)> | Testing Team Lead | after each release to production |  |

# Test Strategy

Test Strategy describes how the target-of-test will be tested:

### Definition Of Ready

When a user story is ready to be taken into a sprint and team can start automating the script. All the pre-requisite & set up for the test automation is done.

### Definition of Done

When team completed the test automation of the story and all the scripts have reviewed and merged in the repository. All the script executed successfully in CI/CD pipeline.

### Automated Testing types

UI Testing, API Testing, Functional Testing, Sanity Testing, Smoke Testing, Regression Testing

### Automated Testing levels

Unit Testing, Integration Testing, E2E Testing

# Risk management

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Risk name | Probability | Severity | Likelihood (Severity \* Probability) | Responsible person | Symptom | Mitigation plan | Contingency plan |
| **Product-specific risks** | | | | | | | |
| *[Put the name of the risk identified here]* | *[Set the probability in range [1;10] where:*  *1 – the risk is about to happen*  *10 - the probability is the lowest]* | *[Set the severity in range [1;5] where:*  *1 – highest*  *5 – lowest]* | *[Multiplication of Probability and Severity.*  *Range the table against this value, smallest value shows the most important risk]* | *[Put the name of the person responsible for managing this risk]* | *[Put the artifacts/facts which indicates the risk exists. In other words, why did you decide that is risk is possible to happen?]* | *[the actionable points what to do in order to prevent risks happening.*  *Sometimes some risks may be transferred from Test Team to another one]* | *[the actionable points what to do in case the risk happened. What are the next steps.]* |
| *Docker Issues* | *10* | *1* | *10* | *DevOps* | *There are the cases in the past that Application was not running due- to server unavailability.* | *Need to transfer to DevOps team* |  |
| **Project risks** | | | | | | | |
| Delay in Login to RP | 2 | 1 | 2 | Technical/Support team | Every time after login, it takes time to login and gives error message about server issue. | Need to transfer to Support team | Need to wait |

# test Team

## Roles and Responsibilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Project Role | Name | Location | Responsibilities |
| 1 | Automation Tester | Suwarna Wagh | Krakow, Poland | Automation Testing |

## STAKEHOLDERS

|  |  |  |
| --- | --- | --- |
| # | Project Role | Name, e-mail, location |
| Internal (inside EPAM) | | |
|  | <Project Manager> | Andrey Kozlov, [Andrey\_Kozlov@epam.com](mailto:Andrey_Kozlov@epam.com), Belarus |
|  | <Project Coordinator, Key Developer> | Artsiom Baradazin, Artsiom\_Baradazin@epam.com,Spain |
|  | <Test Leader> | Ulyana Yeryksonava, [Ulyana\_Yeryksonava@epam.com](mailto:Ulyana_Yeryksonava@epam.com), Serbia |
|  |  |  |
| External (from customer side) | | |
|  | <Product Owner> |  |
|  | <Business Analyst> |  |
|  |  |  |

## Communication plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Issue | Responsible person/people | Contact point | Communication plan |
|  | Inform the team about critical defect | All test team members | All project team members | Via project chat |

## Escalation plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Issue | Responsible person/people | Contact point | Communication plan |
|  | Test team is not in a schedule | Test Team lead | Delivery Manager | Via e-mail |
|  |  |  |  |  |

# Test schedule

*For Waterfall projects or projects with fixed dates use the template:*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | Activity | Begin Time | End Time | Assignment | Location | Work content |
|  | <Test plan creation> |  |  | <resource name> |  |  |
|  | <Test cases creation> |  |  | <resource name> |  |  |
|  | <Build installation> |  |  | <resource name> |  |  |
|  | <Smoke Test execution> |  |  | <resource name> |  |  |
|  | <Critical path Test execution> |  |  | <resource name> |  |  |

*All dates and values are examples. If more than one person executes an activity, please specify the activity and each person in separate table row.*

*For Agile projects use the following template:*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | Sprint day 1 | Sprint day 2 | … | … | … | … | … |
| Test Plan creation |  |  |  |  |  |  |  |
| Test cases creation |  |  |  |  |  |  |  |
| Smoke test execution |  |  |  |  |  |  |  |
| TAF design |  |  |  |  |  |  |  |
| Script Development (UI & API) |  |  |  |  |  |  |  |
| CICD Pipeline |  |  |  |  |  |  |  |
| Integration with third party tool |  |  |  |  |  |  |  |

# TEST Deliverables

*[List all test documentation and deliverables (like bug reports etc.) here]*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | Artifact to be provided | Target audience | Author/ Responsible Person(s) | Frequency (delivery time) | Method of delivery | Link to the template if exists |
|  | RP-Dashboard Test Plans | Customer, AQA Team,  Testing Team | Suwarna Wagh | Once before the testing start | via e-mail, confirm from Management is needed |  |
|  | RP-Dashboard Test 4gy | Customer, AQA Team,  Testing Team | Suwarna Wagh | Once before the testing start | via e-mail, confirm from Management is needed |  |
|  | Dashboard Test Cases | Customer, AQA Team,  Testing Team | Suwarna Wagh | Once before the testing start | via e-mail, confirm from Management is needed |  |
|  | Bug reports | AQA Team,  Testing Team | Suwarna Wagh | After Testing started | via e-mail, confirm from Management is needed |  |
|  | Test Result Reports | Customer | Suwarna Wagh | After Testing Completed | via e-mail, confirm from Management is needed |  |

| REVISION HISTORY | | | | | |
| --- | --- | --- | --- | --- | --- |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Date |
| V0.1 | Test Plan | Suwarna Wagh | 04-Apr-2023 |  |  |
|  |  |  |  |  |  |