TQS: Quality Assurance manual

***Ana Alexandra Antunes [876543]*, *Bruno Bernardes [876543]***

v2023-04-22

[1 Project management 1](#_Toc132723568)

[1.1 Team and roles 1](#_Toc132723569)

[1.2 Agile backlog management and work assignment 1](#_Toc132723570)

[2 Code quality management 2](#_Toc132723571)

[2.1 Guidelines for contributors (coding style) 2](#_Toc132723572)

[2.2 Code quality metrics 2](#_Toc132723573)

[3 Continuous delivery pipeline (CI/CD) 2](#_Toc132723574)

[3.1 Development workflow 2](#_Toc132723575)

[3.2 CI/CD pipeline and tools 2](#_Toc132723576)

[3.3 System observability 2](#_Toc132723577)

[3.4 Artifacts repository [Optional] 2](#_Toc132723578)

[4 Software testing 2](#_Toc132723579)

[4.1 Overall strategy for testing 2](#_Toc132723580)

[4.2 Functional testing/acceptance 3](#_Toc132723581)

[4.3 Unit tests 3](#_Toc132723582)

[4.4 System and integration testing 3](#_Toc132723583)

[4.5 Performance testing [Optional] 3](#_Toc132723584)

# Project management

## Team and roles[Description of the roles assigned in the team and the responsibilities of each member]

**Team Leader** -> João Teles

Ensure that there is a fair distribution of tasks and that members work according to the plan. Actively promote the best collaboration in the team and take the initiative to address problems that may arise. Ensure that the requested project outcomes are delivered in time.

**QA Engineer** -> Marcus Peterson

Responsible, in articulation with other roles, to promote the quality assurance practices and put in practice instruments to measure que quality of the deployment. Monitors that team follows agreed QA practices

**DevOps Master** -> Bruno Moura

Responsible for the (development and production) infrastructure and required configurations. Ensures that the development framework works properly. Leads the preparing the deployment machine(s)/containers, git repository, cloud infrastructure, databases operations, etc.

**Product Owner** -> Victor Melo

Represents the interests of the stakeholders. Has a deep understand of the product and the application domain; the team will turn to the Product Owner to clarify the questions about expected product features. Should be involved in accepting the solution increments.

## Agile backlog management and work assignment

[Description of agile practices defined in the project for backlog management (user stories oriented) and job assignment, and links to associated resources. cfr. [PivotalTracker workflow](https://www.pivotaltracker.com/help/articles/workflow_overview/) ]

…

[Jira: <https://amanacu.atlassian.net/jira/software/projects/AT/boards/1>]

# Code quality management

## Guidelines for contributors (coding style)

[Definition of coding style adopted. → e.g.: [AOS project](https://source.android.com/source/code-style.html)]

## Code quality metrics

[Description of practices defined in the project for *static code analysis* and associated resources.]

[Which quality gates were defined? What was the r[ationale?]

# Continuous delivery pipeline (CI/CD)

## Development workflow

[Clarify the workflow adopted [e.g.. [gitflow](https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow) workflow, [github flow](https://guides.github.com/introduction/flow/) . How do they map to the user stories?]

[Description of the practices defined in the project for *code review* and associated resources.]

[What is your team “[Definition of done](https://medium.com/@anca_51481/user-story-definition-of-done-dod-in-agile-software-development-and-the-technical-debt-a3abf6821ef2)” for a user story?]

## CI/CD pipeline and tools

[Description of the practices defined in the project for the continuous integration of increments and associated resources. Provide details on the tools setup and config.]

[Description of practices for continuous delivery, likely to be based on *containers*]

## System observability

## Artifacts repository [Optional]

[Description of the practices defined in the project for local management of Maven *artifacts* and associated resources. E.g.: a[rtifactory](https://www.jfrog.com/artifactory/)]

# Software testing

## Overall strategy for testing

[what was the overall test development strategy? E.g.: did you do TDD? Did you choose to use Cucumber and BDD? Did you mix different testing tools, like REST-Assured and Cucumber?...]

[it is not to write here the contents of the tests, but to explain the policies/practices adopted and generate evidence that the test results are being considered in the IC process.]

## Functional testing/acceptance

[Project policy for writing functional tests (closed box, user perspective) and associated resources.]

## Unit tests

[Project policy for writing unit tests (open box, developer perspective) and associated resources.]

## System and integration testing

[Project policy for writing integration tests (open or closed box, developer perspective) and associated resources.]

API testing

## Performance testing [Optional]

[Project policy for writing performance tests and associated resources.]