deti universidade de aveiro departamento de eletrónica, telecomunicações e informática

TQS: Quality Assurance manual

Maria Rafaela [107658}*, Marta Oliveira *[107826], Gabriel Teixeira [107876],Fabio Matias [108011]
v2023-04-24

1 Project management

1.1 Team and roles

- **Maria Rafaela Abrunhosa - Team Manager + Developer/Tester**
  As Team Manager, Maria has the responsibility to oversee the project's progress, ensuring that the team fulfills each iteration with the best results , and coordinating between the rest of the team members. Her role as a Developer/Tester means she's also directly involved in coding and quality assurance.

- **Marta Oliveira Inácio - DevOps Master + Developer/Tester**
  Marta's role as a DevOps Master involves managing the CI/CD (Continuous Integration/Continuous Deployment) pipelines, ensuring that software development practices integrate with system operations smoothly. Additionally, her role in development and testing helps her ensure that operational perspectives are considered during the software development phase.

- **Gabriel Melo Teixeira - Software Architect + Developer/Tester**
  As a Software Architect, Gabriel is responsible for the overall design of the software system, selecting appropriate technologies, and ensuring that the system is scalable and maintainable. His role as a Developer/Tester means he's also directly involved in coding and quality assurance.

- **Fábio António Teixeira Matias - Product Owner + Developer/Tester**
  Fábio acts as the Product Owner, which involves defining the product vision, managing the product backlog, and ensuring that the development team delivers value to the business. His role as a Developer/Tester means he's also directly involved in coding and quality assurance.

1.2 Agile backlog management and work assignment

[Description of agile practices defined in the project for backlog management (user stories oriented) and work assignment]

2 Code quality management

2.1 Guidelines for contributors (coding style)

[Definition of coding style adopted. → e.g.: AOS project]

2.2 Code quality metrics and dashboards

[Description of practices defined in the project for *static code analysis* and associated resources.]
[Which quality gates were defined? What was the rationale?]

3 Continuous delivery pipeline (CI/CD)

3.1 Development workflow

[Clarify the workflow adopted [e.g.. gitflow workflow, github 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" for a user story?]

3.2 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*]

3.3 System observability

3.4 Artifacts repository [Optional]

[Description of the practices defined in the project for local management of Maven *artifacts* and associated resources. E.g.: artifactory]

4 Software testing

4.1 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.]

4.2 Functional testing/acceptance

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

4.3 Unit tests

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

4.4 System and integration testing

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

4.5 Performance testing [Optional]

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