Quality process

Git usages

The conventions of message commits and name's branches are defined in the following document: git.md

Code usage

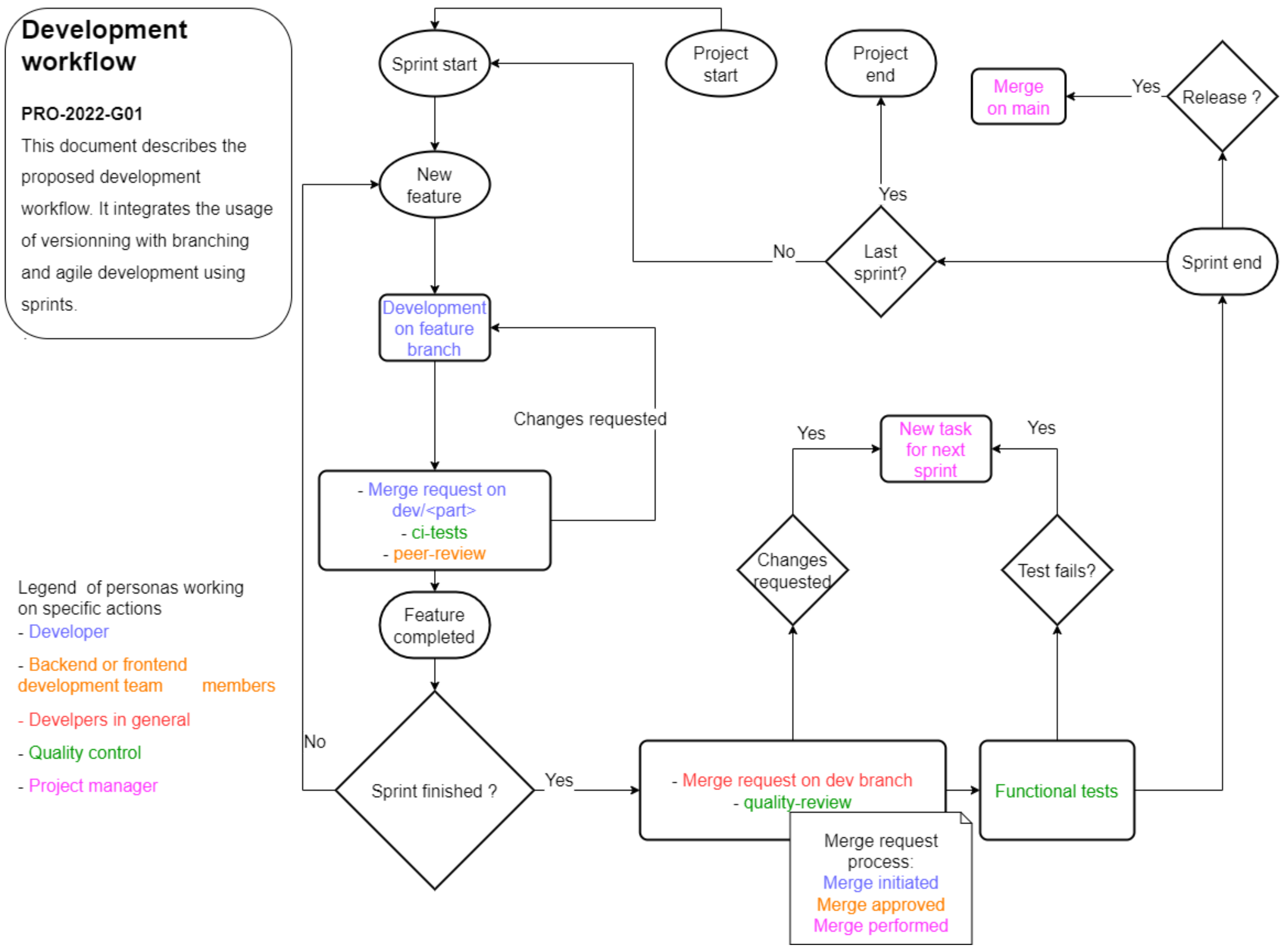
Both of the backend and the frontend use linting script to verify that code convention are respected.

The backend script is based on the PSR12 convention for PHP : https://www.php-fig.org/psr/psr-12/

The frontend script in based on the ESlint rules for TypeScript : https://eslint.org/docs/rules/

Worflow

The workflow will follow the "rules" describes in the following diagram:



As describe in the diagram, at the end of a sprint, the quality manager and the project manager will make functionnals tests and for every bugs detected, a Jira task will be created to fix the bug. The fonctionals tests will follow a document where all tests manipulations are described with the excepted behavior. This document will be modified to add the information of wich test passed and wich did not pass. Every time the functionnals tests are done, all tests must be done even if they passe before.

Merge request

Review

Developpers must make peer-review by adding the other developper of the project (backend or frontend) as Reviewer of the merge request. The reviewer will add comments at the merge request if something seems wrong to him. He can also adds some TODO in the code and commit with a message "Code review". The quality manager can also make code review to alleviate developpers. Moreover, the quality manager will do some code review on the units tests.

Changelog

When creating a merge request, the changes must be spcecified in the changelog.md file.

In the file, the developper must add a lign at the begening of the file. The line must contain the date and the commit message containing the Jira key and the description of the change. For example, if the backend developpers implement a bugfix on the login the 23 april, one of them will write the following line in the changelog.md file:

```
23 April [BUGFIX] {P4-26} Fix backend login
```

Tests

With the CI, all unit tests pass before merge a merge request. If at least one test does not pass, the CI will not allow the merge.

Code coverage

To have a minimum guarantee that the code is tested, we would try to have a minimum code coverage. However, as the backend and the frontend are not in the same language, the code coverage will be defined individually.

For the backend, the minimum code coverage is 90% for the classes and the methods.

For the frontend, the minimum code coverage is 30% for the functions.

The code coverage must be satified for each merge request.

If those percentages are not respected, the CI will not allow the merge.