IFRS9 Pull Request Best Practices

We IFRS9 developers would like to apply a new pull request template to improve our pull request processes and make them clear for the reviewers. In this document you will be informed about the below stages;

1. Creating a Pull Request
2. Reviewing a Pull Request
3. Helpful Resources

1. Creating a Pull Request

To create a new pull request, the description should contain the below main points;

Title :
Description :
1. Problem/What Issue the PR is Related To
2. Solution

Is it testable? : Y/N
Test Steps : (optional)
Task/PBI Number : should be linked to the PR

The other suggestions can be applied to create a new pull request;

- As much as possible we should link the task/PBI to the PR.
- We should split the changes into different commits, to make the review of PR easier.
- Commit messages should be clear to understand what it contains/what it fixes.
- Pull Requests should be containing small changes to make it readable and understandable,
  if there is no chance to keep it small, we can also consider using different commits with a clean messages of what it does.

2. Reviewing a Pull Request

The PR owner and the reviewer should consider the below checks before creating/reviewing a PR;

Does the code work? Check whether function and logic are correct.
Are functions, methods, and variables adequately named?
Is it well tested?
Are there unit tests, and they have good quality? Do they test the functionality?
Do tests reach acceptable coverage for your project?
Is the code clear and easy to understand?
Does it meet the team’s guidelines and style guides?
Is there any duplicated code?
Is there any commented or unnecessary code?
Does the code take the most out of frameworks and language? Is there any custom implementation of native or already-existing functions?
Are there code smells, grey areas, or bug-prone?
Is documentation on functions, methods, classes, contexts, and behaviors adequate?
Is the code as modular as possible?
Are the critical spots adequately logged?
Does the code consider failures? Is it just considering the happy path?
Are there better or simpler solutions?
Is there any performance issue?
Are input data sanitized?
Is sensitive information being encoded or encrypted?

3. Helpful Resources

Here are some good guides about Pull Request best practices;

Pull Request Best Practices

Google - Write a Good PR

Solace - Pull Request Best Practices

Best Practices for Code Review and Pull Requests Insights from Developers