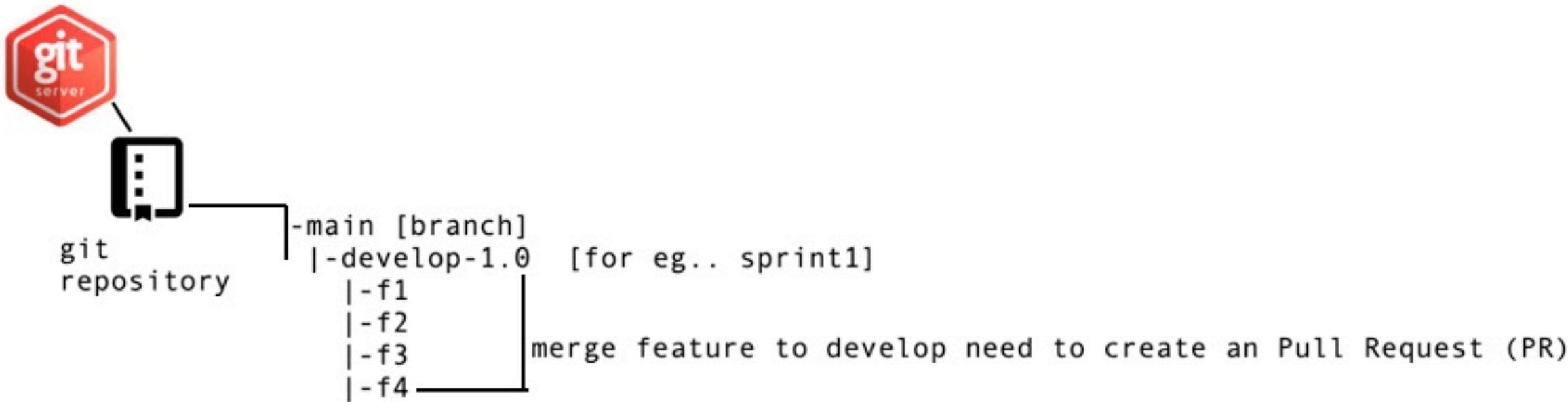Feature-based release

In Feature-based release, the application will be delivered to the customers/endusers based on release plan



```
git
repository
            -main [branch]
            |-develop-1.0  [for eg.. sprint1]
               |-f1
               |-f2
               |-f3        merge feature to develop need to create an Pull Request (PR)
               |-f4
```

1. When we push the code of a Feature branch into Central Repository
Then there are 2 ways we can enforce the quality gating process:

1.1 If the project is existing project and we are migrating into agile methodology/devops process:
In an existing application that is migrating/adopting the Agile/Devops process we cannot enforce quality gating from day-one of the migration. Rather developers are being encouraged and provided sufficient time in implementation unit-test and integration test automations and enable code quality tools like sonarqube, jacoco etc

So, when the developer pushes the code changes into central repository, he will be allowed to push the changes without any quality check rather look for all the unit tests that are in-place are passed or not. Wont enforce code coverage or code review checks to clear for merging/pushing the code locally on the developers machine.


A Jenkins pipeline will be setup which would be triggerred either
1. periodically
2. or for a new push into the repository
which will execute the unit-tests captures the test results, code coverage results and code review and publish the report to the whole team enabling them to understand the status of the repo

1.2 if the project is being built-up from scratch using Agile/Devops process:
Then from day#1 the code quality gating will be enabled/enforced like
    1. automation unit testcases
    2. code review tools
    3. code coverage tools
etc
unless the code passes all these checks your code will not be pushed into central repository.