

Workflow schema

GitHub includes in 2 milestones all features from the Catalogue of user requirements with their catalogue numbers.

Milestones include groups of features (issues) and define the time interval for completion. A feature can be assigned just only to one milestone.

GitHub (Issue tracker)

Development core release milestone

Open features

All features to be developed and bugs

Closed features

All features and fixed bugs ready for testing

Test core release milestone

Open features

Features for testing

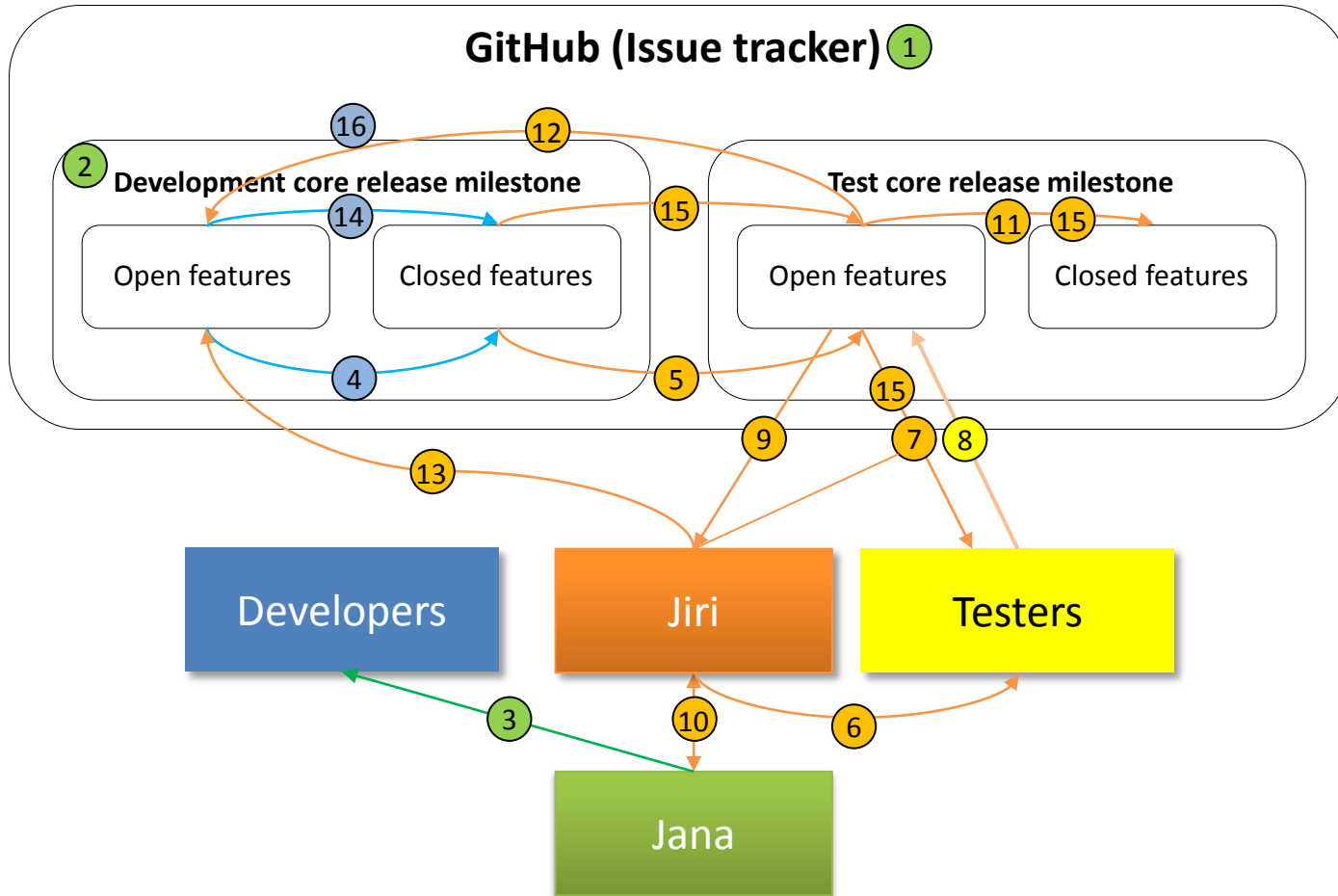
Closed features

Fully functional features including related fixed bugs

Milestone "Development core release" includes all features which are under development or in the process of being fixed.

Milestone "Test core release" includes all features which are ready for testing or completed.

Workflow scheme with testing phases



Explanation: Zero closed issues in “Development core release” milestone means nothing is for testing. Zero open issues in “Test core Release” milestone means all tested features in this milestone are done. All closed issues in “Test core Release” milestone are fully functional. **(Ideally at 1st September)**

Testing phases

1. Jana will set up all features from the Catalogue of user requirements on the GitHub keeping their catalogue numbers. (**Done**)
2. All of these features will be assigned to the milestone “Development core release”. (**Done**)
3. Jana will inform about these GitHub features the developers. (**Done, Techlist**)
4. The developers will close the features in this milestone, which are ready for testing. (**until 29th July, notification Techlist**)
5. Jiri will reopen all **closed** features from the “Development core release” milestone and assigned them to the “Test core Release” milestone. (**1st August**)
6. Jiri will send the informative email to the tester: how the testing process works and what he need from them (this email will also include the Catalogue of user requirements, Testing Workflow with timeline etc.) (**29th July**).
7. Jiri will assign testers to the feature for testing and notify the testers via email with the comment function. (**as soon as closed features are available, 1st August**)
8. Testers have about 2 weeks for testing the features and replying their results and comments via feature email. (**till 12th August**)
9. Jiri will collect all testers’ comments, review them and classify them: bug, no bug. (**15th August**)
10. Jiri will consult with Jana about his results. Jana will decide priority of the bug (minor, normal, critical) and assign responsible developer. Jana will approve feature request by testers (if mentioned) and add them to the catalogue of requirements for later consideration. (**15th August, ongoing**)
11. If the feature is fully functional and all related bugs fixed, it will be closed in milestone “Test core release”, which means that this feature is ready for use and for the public core release.
12. If the feature has a bug, this will be labeled as a bug, classified (minor, normal, critical) and reassigned back to the “Development core release” milestones for the developers. (**integrative process, August**)
13. If several bugs are reported by the tester for one feature, new issues with the describing the bugs and the unique number of the feature (number from the catalog) will be created (**by Jiri**) and assigned to the “Development core release” milestones for the developers.
14. When the developers fixed the bug, they will comment it and close it. (**integrative process, August**)
15. Jiri will reopen this closed issue from the “Development core release” milestone and assigned it to the “Test core Release” milestone. Jiri and Jana will test the feature again. If the bug is fixed and the feature is fully functional, Jiri will close it.
16. There will be a feature development freeze two weeks before the core release and the developers will have to focus on the bugs of already tested features to have a stabile system. The notification will be sent out to the Techlist (**Techlist, mid of August**).