Issue Tracking

Everything related to the software development should be noted as an issue in GitLab, no exceptions. An issue is stored, has one owner responsible, can be commended by others, linked to Git commits and other issues. That makes it perfect for tracking workflows.

Our Workflow/Use

To map our PM workflow to issues we start with the requirements as *top-level* issues. The next level is the tasks derived from the requirement. To connect both, every task-issue can get a *relation* to its requirement-issue. One another/additional way is to make a checkable TO-DO list in the requirements-list.

Beside the obvious tasks, everything else, every bug, improvement, idea or a comment from a customer should be tracked as an issue. When an issue is solved and can be closed, the final Git commit should be referred by noting the first eight characters of the commit hash (e.g. `a6f374be`) in the issue. GitLab automatically links that to the commit. In the other direction, a Git commit can refer to an issue by writing the issue-number prefixed with a hash (“#”) in the commit (e.g. #42).

To get a better overview about the requirements and tasks, the issue-titles can be prefixed with the requirement-number in square brackets (“[]”).

In general, the issue description should be as comprehensive as possible. Every team member should understand the purpose of the issue.

> > Always start with an issue

Contents

From:
https://dokuwiki.hampel-soft.com/ - **HAMPEL SOFTWARE ENGINEERING**

Permanent link:
https://dokuwiki.hampel-soft.com/kb/bestpractices/issuetracking

Last update: **2022/07/10 13:29**