Writing commits

Tom Rochette <tom.rochette@coreteks.org>

January 4, 2026 — 5fe32278

Nowadays I do not write useful commit messages anymore.
The main reason is that most repositories I work in professionally use squash merges, meaning that all commits done in a branch are squashed into a single commit when merging the branch into the main branch. The commit message of the squash commit is usually the title of the pull request, meaning that the individual commit messages are not very useful anymore.
The commit message ends up containing the PR description, which gives the what and why of the changes, which is generally enough.

With the rise of LLMs I have never found myself looking at the commit history and the commit messages to understand why things changed.
If something isn't working, I simply get the LLM to help me fix the problem.

1 Pre-LLMs era

- One liner describing what changed (not period terminated)
- A few lines describing in more details why things changed
- GPG signed commit

2 How to Write a Git Commit Message

- Separate subject from body with a blank line
- Limit the subject line to 50 characters
- Capitalize the subject line
- Do not end the subject line with a period
- Use the imperative mood in the subject line
- Wrap the body at 72 characters
- Use the body to explain what and why vs. how

3 References

- https://chris.beams.io/posts/git-commit/