

# Subjective Test for Pull Request Quality

Jake Roggenbuck - 02/24/2025 - Criteria for measuring quality of pull requests

In order to ensure that the quality of tasks is not decreasing as speed of work increases, you can use a measure (speed) and a counter measure (STPRQ). Measuring output of an engineering team is notoriously difficult because of the nature of knowledge work. However, if you see an increase in performance and want to verify it isn't because of a lack of quality, STPRQ as a measure will help. Note: these criteria are my subjective opinion and are what I look for as the bare minimum for quality of a pull request.

## Criteria:

1. **5pts** - Compiles / Interprets without error
2. **3pts** - Tests pass
3. **2pts** - No commented out code
4. **2pts** - No dead code except for completed functions planned to be used later
5. **2pts** - Identifier have names that are fit to their purpose
6. **2pts** - Lint, Format, etc. passes
7. **2pts** - Sufficient comments, no hard number
8. **1pt** - Pull request include one task (i.e. multiple tasks should be in multiple PRs)
9. **1pt** - Pull request is named well and uses professional language (commits and code too)

Total of 20 points.

In order for me to merge a pull request, it usually needs 19 or 20 points.

## Tips for getting PRs merged:

- Run the code, testing suite, and linting before you commit
- Add comments to lines that require a "Why" explanation
- If you don't plan to use some code, delete it instead of leaving it
- Use specific identifiers when needed (Good variable names)
- Make a separate pull request for each task you are assigned
- Make the pull request early after you first commit when it is still a work in progress so that I can give feedback and look at the pull request earlier

Derived from an earlier work from 1/30/25 under the same name.

The MIT License (MIT)

Copyright © 2025 Jake Roggenbuck