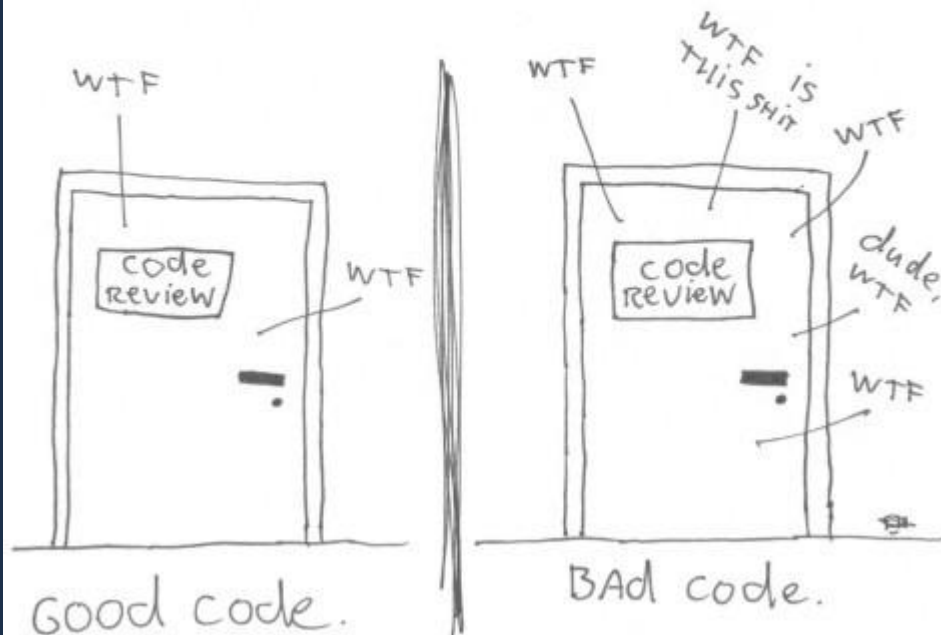


Code quality principles


The ONLY valid measurement
of code quality: WTFs/minute



Maintenance goals:

Readability and Reliability

Hal Abelson and Gerald Sussman in
Structure and Interpretation of Computer Programs
(SICP)



*“Programs must be written for
people to read, and only
incidentally for machines to
execute.”*

A few principles for clean code:

1/ Consistency

2/ Simplicity over cleverness

3/ Quality of tests & testability

4/ Single Responsibility Principle

5/ Don't Repeat Yourself (DRY)

A few more:


Meaningful names

Functions should be small

Functions should have no side effects

Well-documented

Michael Feathers, author of
Working Effectively with Legacy Code



*“Clean code is a code that is
written by someone who
cares.”*

Next:

**Tools to measure code quality,
help enforce code quality**

Code review principles

1/ Have a checklist

- Does the code represent a logical unit of work, meaning does the entire change set under review encapsulate the appropriate amount of work and stay focused to the task it is meant to fulfill?
- Does the code actually solve the problem/business need it is meant to address?
- Are there any glaring syntax errors that were somehow missed?
- Does the code include the appropriate amount of tests to cover the changes both specific to the work done and to the overall application?
- Are there any logical fallacies present and/or edge cases not considered?
- Are language constructs being utilized properly and/or are there opportunities for any optimizations?

- Has there been any premature optimization done that is currently unwarranted?
- If the work is being done in the context of a framework (e.g., Rails) or using a library (e.g., jQuery), is the frameworks/library being leveraged and used properly?
- Is the code organization clean and does it fit within the overall project nicely?
- Are the files that have been modified/added clean and would they pass through a linter without raising any errors or warnings?
- Are there useful comments/documentation where needed, and is the commit message well-formed with succinct details about what the change is?

2/ Train developers how to provide and accept honest feedback and criticism.

3/ Provide a coding style guide

4/ Training in writing proper tests

5/ Use the automated tools: linters, static analysis

6/ Avoid bike-shedding

Poul-Henning Kamp, FreeBSD project

“Parkinson explains that this is because an atomic plant is so vast, so expensive and so complicated that people cannot grasp it, and rather than try, they fall back on the assumption that somebody else checked all the details before it got this far.

*A bike shed on the other hand. Anyone can build one of those over a weekend, and still have time to watch the game on TV. So no matter how well prepared, no matter how reasonable you are with your proposal, somebody will seize the chance to show that he is doing his job, that he is paying attention, that he is *here*.”*

Next:
Pull requests with reviews

Resources:

Clean Code book:

https://www.amazon.com/dp/0132350882/ref=rdr_ext_tmb

Code quality:

<https://medium.com/dev-bits/five-golden-principles-of-a-code-review-ecf7fd977dfd>

<https://blog.goyello.com/2013/01/21/top-9-principles-clean-code/>

<https://dev.to/codemouse92/10-principles-of-a-good-code-review-2eq>

<https://www.perforce.com/blog/sca/what-code-quality-and-how-improve-it>

<https://simpleprogrammer.com/clean-code-principles-better-programmer/>

<https://medium.com/mindorks/how-to-write-clean-code-lessons-learned-from-the-clean-code-robert-c-martin-9ffc7aef870c>

<https://engineering.videoblocks.com/these-four-clean-code-tips-will-dramatically-improve-your-engineering-teams-productivity-b5bd121dd150>

Linting:

<https://eslint.org/docs/rules/>

<https://codeengineered.com/blog/2014/golint/>

<https://www.atlassian.com/git/tutorials/git-hooks>

Code Climate:

<https://docs.codeclimate.com/docs/maintainability>

<https://docs.codeclimate.com/docs/cognitive-complexity>

Resources:

Pull requests:

<https://github.com/18F/culper/pull/1628>

<https://github.com/18F/culper/pull/1834>

<https://github.com/18F/culper/pull/1688>

<https://github.com/18F/culper/pull/1646>

<https://github.com/18F/culper/pull/1628>

<https://github.com/18F/culper/pull/1769>

Projects:

<https://github.com/18F/federalist>

<https://github.com/18F/cms-hitech-apd>

<https://github.com/18F/culper>

Miscellaneous:

<http://phk.freebsd.dk/sagas/bikeshed>