

Linters

Use [rubocop](#).

Commit linter configuration to project repo for the team to share.

Ensure that linter configuration is [integrated into your IDE](#).

Code Quality : Static Analyzers

[CodeClimate](#) Static analysis of code quality, SaaS

[SonarQube](#) Comprehensive static analysis of code quality, can run locally

[Ruby Critic](#) Ruby static analysis gem

[Rubocop](#) Code style analysis & feedback, can integrate with IDE

[Semgrep](#) Static analysis for lots of languages

[rails_best_practices](#) Check the quality of your Ruby on Rails code

Pull Requests

PRs should be as small and incremental as possible

Include screenshots / animations when appropriate

All CI tests should pass prior to assigning reviewers

Don't take feedback personally! Receiving and learning from criticism is one of the best ways to become a better software developer.

Security

Always stay up to date on the [OWASP Top Ten security risks](#).

Integrate static analysis security tools into CI. Dependency auditing is an easy place to start. Enable [Github's Dependabot](#).

Use the [Qualys SSL Server Test](#) to ensure that SSL is correctly configured.

Always require SSL. [Let's Encrypt](#) provides free, secure certificates.

Use [bundler-audit](#) and [brakeman](#)

Testing

Unit Tests : test the output of a single given state

Functional Tests : test the output of changes to state

Integration Tests : test the results of cross-service interactions from the user's perspective. Use [Cypress](#).

Implement **continuous integration**, which runs all tests, performs static analysis, and automates security audits.

Ensure that **test coverage** is over 95%.

Performance

Identify and eliminate N+1 queries. [bullet](#) is helpful.

An RPM like [New Relic](#), [Scout](#), or [DataDog](#) is great for identifying performance problems.

Ensure that GZIP / HTTP Deflate are enabled and properly configured.

Use [Rails caching](#). It includes easy-to-implement patterns for lots of different types of caching.

Documentation

`README.md` must include these sections:

Summary, Prerequisites, Infrastructure, ENV Variables, Cron, Gotchas

Use [semantic versioning](#).

`CHANGELOG.md` should follow <https://keep-achangelog.com/en/1.0.0/>

Provide well-formatted and sensible **seed data**.

Diagram **complex processes**. [Whimsical](#) is great for this.

APIs should have comprehensive [Swagger](#) documentation.

Frontends should have **wireframes** prior to writing code.

Pivotal / Jira stories document application evolution.

Always consider the perspective of a developer who has never touched this project, but needs to rescue it because you're on vacation.



By **Jack Collier**
(imnotquitejack)

cheatography.com/imnotquitejack/

Not published yet.

Last updated 10th June, 2021.

Page 1 of 1.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>