**Proper Source Code Management for Software Projects**

Effective source code management is the backbone of any successful software project. It is a set of practices, tools, and strategies that help developers organize, collaborate, and maintain their codebase. The goal is to ensure the project remains efficient, reliable, and scalable over its entire lifecycle. This comprehensive guide delves into the critical components of proper source code management, including Git branching strategies, the importance of an up-to-date README, keeping a detailed changelog file, and the value of tagging branches and releases.

**Git Branching Strategies:**

Git, the most popular version control system, provides the foundation for robust source code management. It offers several branching strategies to organize your codebase effectively:

**Master Branch:** The master branch is often considered the production-ready branch. It should contain only code that has undergone rigorous testing and is ready for deployment.

**Development Branch:** The development branch serves as the integration point for features and bug fixes. It should be stable but may include experimental features.

**Feature Branches:** Each new feature or bug fix should have its feature branch created from the development branch. This isolation helps prevent conflicts with other features. When the feature is complete, it's merged into the development branch.

**Release Branches:** Preparing for a new release? Create a release branch from development to conduct final testing and make last-minute changes. Once ready, it's merged into both master and development.

**Hotfix Branches:** When critical bugs are discovered in the production code, hotfix branches are created from the master. After fixing the issue, the branch is merged into both master and development.

These branching strategies provide a structured approach to managing code changes and ensuring code quality.

**README:**

A well-maintained README is your project's front door. It's where both contributors and users learn about your project. An informative README should include:

* A concise project description.
* Clear installation instructions.
* Useful usage examples and code snippets.
* Configuration details.
* Contribution guidelines.
* Licensing information.
* Contact details.

The README must always remain up-to-date to provide an accurate reference for newcomers and experienced users alike. It serves as a crucial resource for understanding, using, and contributing to your project.

**Changelog File:**

A changelog file documents the historical changes made to your project. A typical changelog entry includes:

* Version number.
* Release date.
* A list of changes, such as new features, bug fixes, and enhancements.
* Names of contributors.
* A standardized format, like Keep a Changelog, can make maintaining the changelog and understanding the project's history easier.

Changelog entries are added when features or bug fixes are merged into the master or development.

An example of a changelog file is below.

# Changelog

All notable changes to this project will be documented in this file.

The format is based on [Keep a Changelog](<https://keepachangelog.com/en/1.1.0/>),

and this project adheres to [Semantic Versioning](<https://semver.org/spec/v2.0.0.html>).

## [Unreleased]

### Added

- v1.1 Italian translation.

- v1.1 Polish translation.

## [1.1.1] - 2023-03-05

### Added

- Arabic translation (#444).

- v1.1 French translation.

- v1.1 Dutch translation (#371).

- v1.1 Russian translation (#410).

- v1.1 Japanese translation (#363).

- v1.1 Norwegian Bokmål translation (#383).

- v1.1 "Inconsistent Changes" Turkish translation (#347).

- Default to the most recent versions available for each language

- Display count of available translations (26 to date!)

- Centralize all links into `/data/links.json` so they can be updated easily

### Fixed

- Improve French translation (#377).

- Improve id-ID translation (#416).

- Improve Persian translation (#457).

- Improve Russian translation (#408).

- Improve Swedish title (#419).

- Improve zh-CN translation (#359).

- Improve French translation (#357).

- Improve zh-TW translation (#360, #355).

- Improve Spanish (es-ES) translation (#362).

- Foldout menu in Dutch translation (#371).

- Missing periods at the end of each change (#451).

- Fix missing logo in 1.1 pages

- Display notice when translation isn't for the most recent version

- Various broken links, page versions, and indentations.

### Changed

- Upgrade dependencies: Ruby 3.2.1, Middleman, etc.

### Removed

- Unused normalize.css file

- Identical links assigned in each translation file

- Duplicate index file for the English version

## [1.1.0] - 2019-02-15

### Added

- Danish translation (#297).

- Georgian translation from (#337).

- Changelog inconsistency section in Bad Practices.

### Fixed

- Italian translation (#332).

- Indonesian translation (#336).

## [1.0.0] - 2017-06-20

### Added

- New visual identity by [@tylerfortune8](<https://github.com/tylerfortune8>).

- Version navigation.

- Links to the latest released version in previous versions.

- "Why keep a changelog?" section.

- "Who needs a changelog?" section.

- "How do I make a changelog?" section.

- "Frequently Asked Questions" section.

- New "Guiding Principles" sub-section to "How do I make a changelog?".

- Simplified and Traditional Chinese translations from [@tianshuo](<https://github.com/tianshuo>).

- German translation from [@mpbzh](<https://github.com/mpbzh>) & [@Art4](<https://github.com/Art4>).

- Italian translation from [@azkidenz](<https://github.com/azkidenz>).

- Swedish translation from [@magol](<https://github.com/magol>).

- Turkish translation from [@emreerkan](<https://github.com/emreerkan>).

- French translation from [@zapashcanon](<https://github.com/zapashcanon>).

- Brazilian Portuguese translation from [@Webysther](<https://github.com/Webysther>).

- Polish translation from [@amielucha](<https://github.com/amielucha>) & [@m-aciek](<https://github.com/m-aciek>).

- Russian translation from [@aishek](<https://github.com/aishek>).

- Czech translation from [@h4vry](<https://github.com/h4vry>).

- Slovak translation from [@jkostolansky](<https://github.com/jkostolansky>).

- Korean translation from [@pierceh89](<https://github.com/pierceh89>).

- Croatian translation from [@porx](<https://github.com/porx>).

- Persian translation from [@Hameds](<https://github.com/Hameds>).

- Ukrainian translation from [@osadchyi-s](<https://github.com/osadchyi-s>).

### Changed

- Start using "changelog" over "change log" since it's the common usage.

- Start versioning based on the current English version at 0.3.0 to help

translation authors keep things up-to-date.

- Rewrite the "What makes unicorns cry?" section.

- Rewrite the "Ignoring Deprecations" sub-section to clarify the ideal

scenario.

- Improve the "Commit log diffs" sub-section to further argument against

them.

- Merge "Why can't people just use a git log diff?" with "Commit log

diffs".

- Fix typos in Simplified Chinese and Traditional Chinese translations.

- Fix typos in Brazilian Portuguese translation.

- Fix typos in Turkish translation.

- Fix typos in Czech translation.

- Fix typos in Swedish translation.

- Improve phrasing in French translation.

- Fix phrasing and spelling in German translation.

### Removed

- Section about "changelog" vs "CHANGELOG".

## [0.3.0] - 2015-12-03

### Added

- RU translation from [@aishek](<https://github.com/aishek>).

- pt-BR translation from [@tallesl](<https://github.com/tallesl>).

- es-ES translation from [@ZeliosAriex](<https://github.com/ZeliosAriex>).

## [0.2.0] - 2015-10-06

### Changed

- Remove exclusionary mentions of "open source" since this project can

benefit both "open" and "closed" source projects equally.

## [0.1.0] - 2015-10-06

### Added

- Answer "Should you ever rewrite a change log?".

### Changed

- Improve argument against commit logs.

- Start following [SemVer]([https://semver.org](https://semver.org/)) properly.

## [0.0.8] - 2015-02-17

### Changed

- Update year to match in every README example.

- Reluctantly stop making fun of Brits only, since most of the world

writes dates in a strange way.

### Fixed

- Fix typos in recent README changes.

- Update outdated unreleased diff link.

## [0.0.7] - 2015-02-16

### Added

- Link, and make it evident that the date format is ISO 8601.

### Changed

- Clarified the section on "Is there a standard change log format?".

### Fixed

- Fix Markdown links to tag comparison URL with footnote-style links.

## [0.0.6] - 2014-12-12

### Added

- README section on "yanked" releases.

## [0.0.5] - 2014-08-09

### Added

- Markdown links to version tags on release headings.

- Unreleased section to gather unreleased changes and encourage note

-keeping prior to releases.

## [0.0.4] - 2014-08-09

### Added

- Better explanation of the difference between the file ("CHANGELOG")

and its function, "the change log".

### Changed

- Refer to a "change log" instead of a "CHANGELOG" throughout the site

to differentiate between the file and the purpose of the file — the

logging of changes.

### Removed

- Remove empty sections from CHANGELOG, they occupy too much space and

create too much noise in the file. People will have to assume that the

missing sections were intentionally left out because they contained no

notable changes.

## [0.0.3] - 2014-08-09

### Added

- "Why should I care?" section mentioning The Changelog podcast.

## [0.0.2] - 2014-07-10

### Added

- Explanation of the recommended reverse chronological release ordering.

## [0.0.1] - 2014-05-31

### Added

- This CHANGELOG file to hopefully serve as an evolving example of a

standardized open-source project CHANGELOG.

- The CNAME file enables the custom domain of GitHub Pages.

- README now contains answers to common questions about CHANGELOGs.

- Good examples and basic guidelines, including proper date formatting.

- Counter-examples: "What makes unicorns cry?".

[unreleased]:<https://github.com/olivierlacan/keep-a-changelog/compare/v1.1.1...HEAD>

[1.1.1]:<https://github.com/olivierlacan/keep-a-changelog/compare/v1.1.0...v1.1.1>

[1.1.0]:<https://github.com/olivierlacan/keep-a-changelog/compare/v1.0.0...v1.1.0>

[1.0.0]:<https://github.com/olivierlacan/keep-a-changelog/compare/v0.3.0...v1.0.0>

[0.3.0]:<https://github.com/olivierlacan/keep-a-changelog/compare/v0.2.0...v0.3.0>

[0.2.0]:<https://github.com/olivierlacan/keep-a-changelog/compare/v0.1.0...v0.2.0>

[0.1.0]:<https://github.com/olivierlacan/keep-a-changelog/compare/v0.0.8...v0.1.0>

[0.0.8]:<https://github.com/olivierlacan/keep-a-changelog/compare/v0.0.7...v0.0.8>

[0.0.7]:<https://github.com/olivierlacan/keep-a-changelog/compare/v0.0.6...v0.0.7>

[0.0.6]:<https://github.com/olivierlacan/keep-a-changelog/compare/v0.0.5...v0.0.6>

[0.0.5]:<https://github.com/olivierlacan/keep-a-changelog/compare/v0.0.4...v0.0.5>

[0.0.4]:<https://github.com/olivierlacan/keep-a-changelog/compare/v0.0.3...v0.0.4>

[0.0.3]:<https://github.com/olivierlacan/keep-a-changelog/compare/v0.0.2...v0.0.3>

[0.0.2]:<https://github.com/olivierlacan/keep-a-changelog/compare/v0.0.1...v0.0.2>

[0.0.1]:<https://github.com/olivierlacan/keep-a-changelog/releases/tag/v0.0.1>



**Tagging:**

Tagging is a valuable version control technique for labelling specific points in your Git history. Tags are frequently used for version releases. For instance, when you're ready to release version 1.0 of your software, you create a tag named v1.0. These tags serve as stable reference points for the code at specific moments, making it easy to revert to a particular version when needed.

In summary, proper source code management is the cornerstone of a successful software project. It encompasses effective Git branching strategies, a well-maintained README, a comprehensive changelog, and tagging practices. These practices enhance code quality, improve user satisfaction, and boost developer productivity. When integrated into your project, they ensure that your codebase is organized, collaborative, and maintainable from its inception to its growth and evolution.