



keep a changelog

Don't let your friends dump git logs into changelogs.

Version 1.0.0

```
### 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 "What makes unicorns cry?" section.
- Rewrite "Ignoring Deprecations" sub-section to clarify the ideal scenario.
- Improve "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.
```

What is a changelog?

A changelog is a file which contains a curated, chronologically ordered list of notable changes for each version of a project.

Why keep a changelog?

To make it easier for users and contributors to see precisely what notable changes have been made between each release (or version) of the project.

Who needs a changelog?

People do. Whether consumers or developers, the end users of software are human beings who care about what's in the software. When the software changes, people want to know why and how.

How do I make a good changelog?

Guiding Principles

- Changelogs are *for humans*, not machines.
- There should be an entry for every single version.
- The same types of changes should be grouped.
- Versions and sections should be linkable.
- The latest version comes first.
- The release date of each version is displayed.
- Mention whether you follow [Semantic Versioning](#).

Types of changes

- `Added` for new features.
- `Changed` for changes in existing functionality.
- `Deprecated` for soon-to-be removed features.
- `Removed` for now removed features.
- `Fixed` for any bug fixes.
- `Security` in case of vulnerabilities.

How can I reduce the effort required to maintain a changelog?

Keep an `Unreleased` section at the top to track upcoming changes.

This serves two purposes:

- People can see what changes they might expect in upcoming releases
- At release time, you can move the `Unreleased` section changes into a new release version section.

Can changelogs be bad?

Yes. Here are a few ways they can be less than useful.

Commit log diffs

Using commit log diffs as changelogs is a bad idea: they're full of noise. Things like merge commits, commits with obscure titles, documentation changes, etc.

The purpose of a commit is to document a step in the evolution of the source code. Some projects clean up commits, some don't.

The purpose of a changelog entry is to document the noteworthy difference, often across multiple commits, to communicate them clearly to end users.

Ignoring Deprecations

When people upgrade from one version to another, it should be painfully clear when something will break. It should be possible to upgrade to a version that lists deprecations, remove what's deprecated, then upgrade to the version where the deprecations become removals.

If you do nothing else, list deprecations, removals, and any breaking changes in your changelog.

Confusing Dates

Regional date formats vary throughout the world and it's often difficult to find a human-friendly date format that feels intuitive to everyone. The advantage of dates formatted like `2017-07-17` is that they follow the order of largest to smallest units: year, month, and

day. This format also doesn't overlap in ambiguous ways with other date formats, unlike some regional formats that switch the position of month and day numbers. These reasons, and the fact this date format is an [ISO standard](#), are why it is the recommended date format for changelog entries.

There's more. Help me collect these antipatterns by [opening an issue](#) or a pull request.

Frequently Asked Questions

Is there a standard changelog format?	V
What should the changelog file be named?	V
What about GitHub Releases?	V
Can changelogs be automatically parsed?	V
What about yanked releases?	V
Should you ever rewrite a changelog?	V
How can I contribute?	V

Conversations

I went on [The Changelog podcast](#) to talk about why maintainers and contributors should care about changelogs, and also about the motivations behind this project.



This project is MIT Licensed // Created & maintained by Olivier Lacan // Designed by Tyler Fortune