Semantic versioning

There are 2 stages of app development

1. 0 — Pre-MVP/Alpha
2. Major.Minor.Patch — Alpha/Beta

0 — Pre-MVP/Alpha

At this stage, you're developing your app and still building towards an MVP.

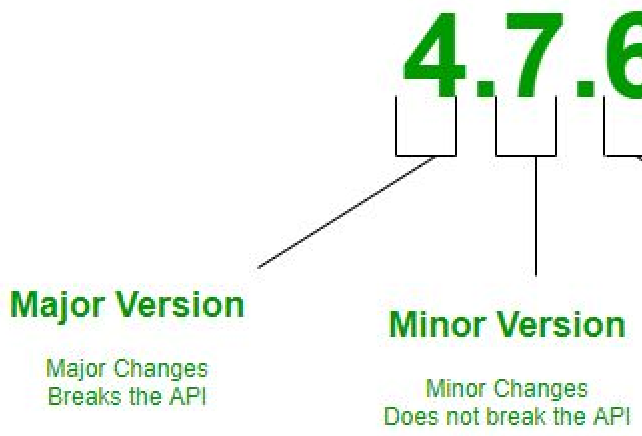
Major.Minor.Patch — Alpha/Beta

1. Once you reach your **MVP,** you might iterate over this and create **Alpha/Beta** releases for users **to test and supply feedback on**.

2. When creating your first release, you'll want to start versioning.

3. **A pre-release version may be denoted by appending a hyphen and a series of dot separated identifiers immediately following the patch version**.A pre-release version indicates that the version is unstable and might not satisfy the intended compatibility requirements as denoted by its associated normal version. Examples: 1.0.0-alpha, 1.0.0-alpha.1, 1.0.0-0.3.7

Semantic versioning specifications



Major Release — Increment the first digit by 1 if the new features break backwards compatibility/current features

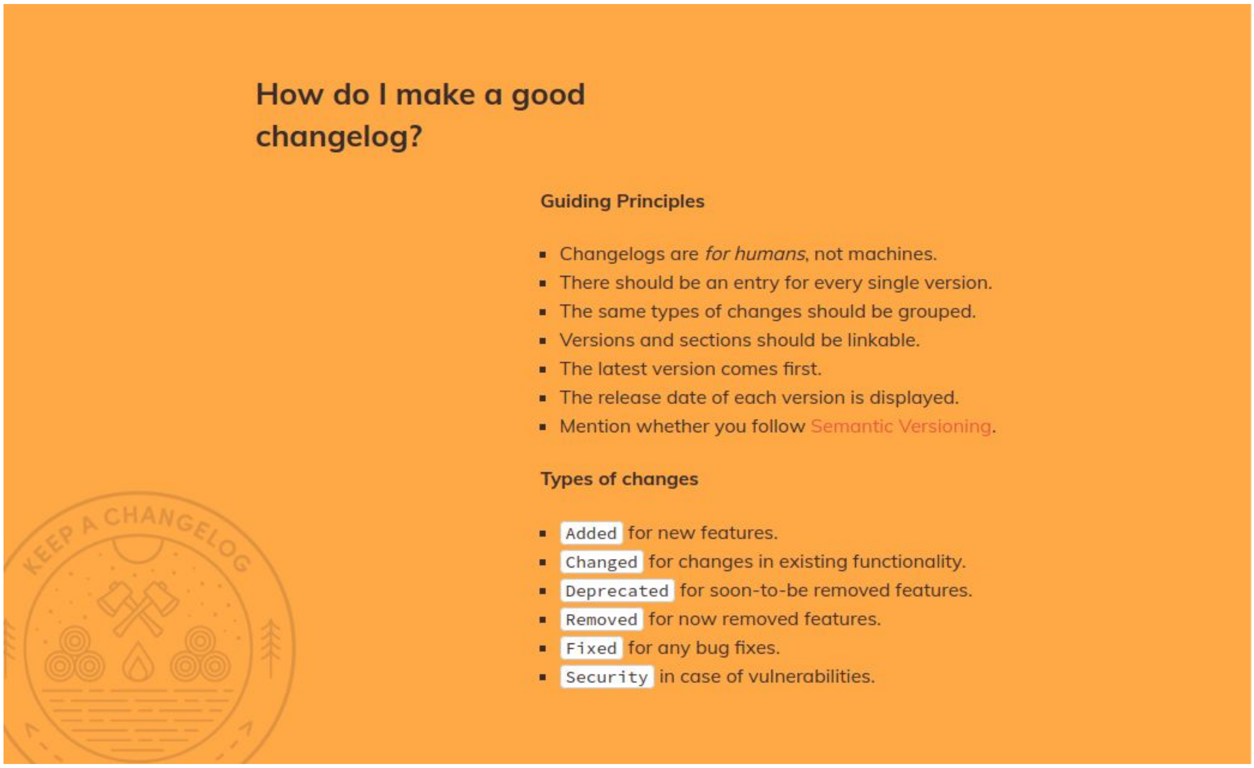
Minor Release — Increment the middle digit by 1 if the new features don't break any existing features and are compatible with the app in it's current state.

Patch Release — Increment the last digit by 1 if you're publishing bug/patch fixes to your app.

Key points to remember

1. **The first version starts at 0.1.0** and not at 0.0.1, **as no bug fixes have taken** place, rather we start off with a set of features as first draft of the project.

2. **Before 1.0.0 is only the Development Phase**, where you focus on getting stuff done.
3. **The first stable version is v1.0.0.**
4. It's best to keep track of all these changes in a **CHANGELOG.** This way you can keep track of what's been fixed and any upcoming features that are being released into the next version. ***Ref*** : https://keepachangelog.com/en/1.0.0/



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.