

Notation for versioning of collaborative codes

Brendon Bradley
Jan 2018

Context:

- For highly-collaborative and integrated research we need a commonly-understood language when talking about software products that are being used
- For all our software releases, we aim to have the development, usage and features documented for each release.

Approach:

- a. To make versioning easy to understand the prefix of the versioning should be the year.month (e.g. 18.1 for Jan 2018).
- b. Minor iterations (e.g. bug fixes etc) should be denoted with a suffix increase (e.g. a bug fix means to go from 18.1.1 to 18.1.2).
- c. All changes in major and minor releases of software products developed shall have their feature changes noted.
- d. Every major project will have a separate GitHub repository and a file called 'CHANGELOG.md' in the root. The format of which is prescribed by the standard at <http://keepachangelog.com/en/1.0.0>
 - a. Any changes that is in a branch that is currently being used should be documented in the changelog in the [unreleased] section
 - b. The reviewer of the pull-request shall also make sure that the changelog contains the relevant information.
- e. All deprecations, removals, and any breaking changes will also be present in the changelog.
 - a. In the case of workflow alterations additional documentation may be linked to in the changelog.
- f. Descriptions should be high-level, human readable and avoid direct references to code.

Implications for backward compatibility

- a. Somewhat by definition, backward compatibility should be maintained in all minor iterations (18.1.1 to 18.1.2)
- b. Backward compatibility may not be maintained for major versioning changes (e.g. 18.1.2 to 18.3.1).

Implications for planning:

- a. Minor iterations (18.1.1 to 18.1.2) are expected to happen on the fly
- b. Major revisions (e.g. to release a new version 18.3.X) should be explicitly planned in advance using Gantt charting and elicit input from multiple people in order to make strategic decisions around desired functionality and backward-compatibility.

This policy shall apply to the following software projects:

- EMOD3D
- gm_sim_pkg
- gm_sim_workflow
- groundfailure (liquefaction/landslide impact analysis)
- Post-processing
- QCore library
- Seisfinder
- vs30 mapping