

OUR CRITERIA AND COMMITMENT

Our program focuses on ecosystem hygiene, security, and open-source best practice. Partnering teams who meet the criteria receive a grant to further strengthen and sustain their impact in the open-source community.

Version End-of-Life

Releases

- Community is notified of a release when it occurs.
 - Should be communicated via all official channels (X/Twitter, GitHub Repo, Website docs, etc).
- Clear definition of what happens when a new version is released.
 - Defines impact of major, minor, or patch release.
 - When a new version is released, the community knows the up-to-date support status of previous releases.
 - Releases that address CVEs are identified as such
 - Inform community as to whether or not backport patches mitigate known vulnerabilities.

Versioning

- It is recommended a project uses an industry standard versioning (eg Semantic Versioning or for PyPi packages, PEP 440).
- If project is not currently following SemVer, versioning structure is predictable, intuitive, and documented for the community.

Announcement Comms and Clarity

- Documentation clearly dictates which versions are end-of-life versus supported.
- Announcing releases
 - Many small project tend to release a new major version with little to no warning, immediately drop support for older majors; Larger projects tend to forecast new release lines in advance and explicitly indicate a long-term support period for the now-previous release line.
 - Announcements for EOL events are made in a timely manner.
 - Benchmark timeframes to consider:
 - Widely adopted foundational libraries/frameworks: 12-18 months
 - Moderately adopted projects (10k+ users): 6-12 months
 - Small and/or niche projects: 3-6 months
 - If applicable, there is a support period for now-previous release lines to allow time for transitioning or migration.
- End of life events are posted to the official channels of the project.

Deprecation

Documentation

- There is a definition of deprecation
 - > How it is treated
 - > If it receives bug fixes / patches
 - > How vulnerabilities affect these versions.
- Tagging is used in all official documents to notate versions that have been deprecated.
- When a new version release occurs that impacts the status of a prior version, there is documentation about which version(s) become deprecated and/or end of life.
- A time frame is specifically mentioned as to when a deprecated version will be removed completely and/or will no longer be supported at all.
- If applicable, project will leverage ecosystem-specific tools to report deprecations.

Support

- There is a recommended timeline for the community to follow for:
 - > Adoption of new features, and
 - > Removal of deprecated features.
- There is structure for support of deprecated features. Example:
 - > Introduced in minor releases
 - > Removed in next major release
 - > Critical vulnerabilities are backported in LTS only.
 - > Deprecations must be non-breaking.

Warnings

- Runtime warnings are added into logs / console while being used.
- Deprecation events show in change logs and release notes.

Support Policies

Process

- ⦿ There is a process in place for the community to report issues.
- ⦿ There is a process in place to respond to reported issues.
- ⦿ There is a process in place to report security violations.

Support Tiers & Documentation

- ⦿ Define version support tiers so that it is clear to the community.
 - Example 1:
 - » Current: Full security vulnerabilities and bug support.
 - » LTS: Security vulnerabilities only.
 - » EOL: If critical, security vulnerabilities.
 - Example 2:
 - » Latest (or Supported)
 - » Unsupported
- ⦿ It is clear via documentation to developers which versions are supported versus unsupported.
- ⦿ Expectations are set as to which versions are recommended for use by developers (supported) versus which are intended as pre-release or development versions.

Encouragement for Support

- ⦿ When a version goes EOL, users are encouraged to either migrate OR establish a support solution of migration is currently not an option.
- ⦿ Note: This may be through HeroDevs services so that migration timelines may be established or identified remaining supported on older versions.