Section 1 General

1.1 Scope

This document specifies the process used to produce open-source projects within the field of Intelligent Transportation Systems (ITS).

The process follows general practices within the larger open-source community; however, this document:

- provides a step-by-step overview of the process, so that those unfamiliar with open-source processes can better understand the process and become contributors,
- formalizes the process (e.g., by clearly defining what are requirements), and
- tailors the process (e.g., by defining the preferred tools to be used).

The process to approve the resultant product is defined elsewhere (e.g., NTCIP 8001).

The ITS Open-Source Process is based on the practices defined by open-sauced. However, whereas open-sauced is written as an informative guide and describes how systems can work; this document is written as a specification to define how the ITS Open-Source Process will work. While still providing a discussion of the issues; it highlights the requirements and notable options along the way by stating each in its own paragraph, starting the paragraph with the requirement (REQ) or option (OPT) number. Conditional requirements are listed as requirements, but include a condition in the text (e.g., If X, then ...). The remaining text provides further guidance and can include additional options that do not necessitate specific numbering.

We recognize that onboarding to a new project can be challenging, especially if you're new to open source development. Be patient, and don't be discouraged by setbacks or mistakes. You'll become more comfortable and confident in your contributions with persistence and practice.

1.2 References

The following documents are referenced by this document. At the time of publication, the editions indicated were valid.

1.2.1 Normative References

Normative references contain provisions that, through reference in this text, constitute provisions of this document. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standard listed.

- ISO/IEC/IEEE 24765:2017: *Systems and software engineering — Vocabulary, 2017*
- GitHub
- MkDocs
- Materials for MkDocs
- ReqView
- Python

1.2.2 Other References

Other references are included to provide a more complete understanding of this document and its relationship to other documents.

1.2.2.1 Other Resources for Contributors

This document standardizes and tailors certain aspects of the information contained in open-sauced; however, it is not a complete replacement of that material. If you wish to learn more about open-source development, the following materials may be of interest:

- What is open-source?
- Why open-source?
- The Secret Sauce
- Types of Open-Source Contributions
- Open Source Guides
- Introduction to GitHub and Open Source Projects

1.2.2.2 Other Resources for Maintainers

If you wish to learn more about open-source maintenance, the following materials may be of interest:

- Understanding the Role of an Open Source Maintainer
- How to Communicate and Collaborate Effectively
- Building Community
- Maintainer Power Ups
- Building Your Team
- The Power of Open Source Metrics

- Contributor Ladder Template
- Maintainer Community

1.3 General Statements

The remainder of this document is broken into the following chapters:

- **Commenting Process:** Details the process of contributing to open-source projects and provides step-by-step processes for using the preferred tools of the ITS open-source projects.
- **Contribution Process:** Details the process of contributing to open-source projects and provides step-by-step processes for using the preferred tools of the ITS open-source projects.
- **Maintenance Process:** Details the rules that project maintainers are to follow when managing an ITS open-source project. This includes processes for setting up new projects, managing issues and pull requests, maintaining quality, and coordinating with standard development organizations.
- **Approval Process:** Defines the approval stages for ITS open-source projects and the processes required for approval for each stage and subsequent tagging and publication of versions.[1]
- **Documentation Conventions:** Annex B defines the preferred styles, processes, and tools for developing documentation for ITS open-source projects, including projects that are 100% documentation (e.g., the ITS Open-Source Process project).
- **Code Conventions:** Annex C defines the styles, processes, and tools for developing computer code for ITS open-source projects, including Python and ASN.1.
- **Requirements Management:** Defines preferred ways to use requirement management tools to produce content that can be easily integrated into the ITS open-source pojects while providing clear traceability.

1.4 Glossary

For terms not defined here, English words are used in accordance with their definitions by the merriam-webster online dictionary. Electrical and electronic terms not defined in this section or in Webster's New Collegiate Dictionary are used in accordance with their definitions in ISO/IEC/IEEE 24765:2017.

backlog: A backlog is a list of tasks that need to be completed within a

branch: A branch is a separate version of the code that's created for

bug: A bug refers to an error, flaw, or defect in code that adversely affects

clone: Cloning is the process used to copy an existing Git repository into a

code freeze: A code freeze is a period of time where no new code is added to

code review: A code review is when a maintainer or contributor will review

containerization: Containerization is a way of packaging and running

continuous integration (CI): Continuous integration (CI) is a development

continuous deployment (CD): Continuous deployment (CD) is often associated

contributor: A contributor is anyone who makes changes, additions, or

core member: A core member is a contributor who has been granted additional

docs: Docs is an abbreviation for "documentation". It primarily explains how

fork: A fork is a copy of a repository. When you fork a repository, you

GitHub actions: GitHub Actions are a way to automate tasks within your

GitHub discussions: GitHub Discussions are a way to have conversations about

issue: An issue is a problem or bug that needs to be addressed in the code.

linting: Linting is the process of running a program that will analyze code

maintainer: A maintainer is a person or a group of people responsible for

markdown: Markdown is a lightweight markup language commonly used for

merge: Merging is the process of combining changes from one branch into

onboarding: Onboarding documentation helps new team members or collaborators

OSS Projects: OSS stands for "Open Source Software" projects. These are

pull request: A pull request is a request from a contributor to a maintainer

quality assurance: Quality assurance in open source projects involves

release candidate: A release candidate is a beta version of software with the

release notes: Release notes are documents that detail changes, enhancements,

repository: A repository is a central location where code is stored and

style guide: A style guide is a set of rules and conventions that define the

versioning: Versioning is the process of assigning either unique version

[1] TODO: Move the approval process to NTCIP 8001 ↩

October 30, 2024