2016-03 Taverna Graduation Maturity Assessment

Outline completed 4/4/2016. Ready for content.

Goal: Make self-evaluation comments objective and demonstrable.

Overview

The purpose of this plan is to document the Apache Taverna podling's progress toward graduation from the Apache incubator. Initially, it is a roadmap for the Taverna PMC, showing completed items as well as those that have yet to be accomplished, and will help us focus our efforts to achieve project maturity. As graduation nears, the plan will be a measurable means for mentors, community members, the Incubator PMC, and the ASF Board of directors to assess Taverna's readiness to graduate.

This Maturity Assessment Plan is based on the Apache Project Maturity Model and benefits from the experience of the Groovy Podling.

Status

We are adding content to the plan.

Assessment Summary

When completed, this section will summarize the readiness of the Taverna podling to graduate.

Maturity Model Assessment

The Goals

A mature Apache project will meet the following goals:

- Code: open, discoverable, fully public history, documented provenance
- Quality: security, backwards compatibility, etc
- Contributions: welcome from anyone based on technical quality
- License: Apache License, dependencies must not put additional restrictions
- Community: inclusive, meritocratic, no dictators, clear documented path to entry
- Discussions and decisions: asynchronous, in a single central place, archived
- Decision making: consensus, votes if needed, technical vetoes in the worst case
- Independence: from any corporate or organizational influence
- Releases: source code only, notices, long-lived release format

These goals are from a January 6, 2015 message from Bertrand Delacretaz on the Apache community-dev mailing list. They provide a clear and concise introduction to the maturity model.

Assessment Categories

The Apache Project Maturity Model assesses whether or not a project complies with each element in the following seven categories: code, licenses and copyright, releases, quality, community, consensus building, and independence. There are no intermediate levels: a project either complies with an element or it does not. The following sections address Taverna's status regarding each element of the model.

NOTE: Numbers in brackets ([ ]) refer to footnotes on the Apache Project Maturity page, and are reproduced at the bottom of this document.

Code

- **CD10:** *The project produces Open Source software, for distribution to the public at no charge. [1]*
    - Apache Taverna's source code is fully licensed as Apache License 2.0 and available with no charge.
- **CD20:** *The project's code is easily discoverable and publicly accessible.*
    - Apache Taverna's source repositories (git) are available from https://taverna.incubator.apache.org/download/code/ and mirrored to GitHub. Source code archives are available for each released component, e.g. https://taverna.incubator.apache.org/download/engine/#source-code. (See also https://archive.apache.org/dist/incubator/taverna/source/).
- **CD30:** *The code can be built in a reproducible way using widely available standard tools.*

- Apache Taverna's source code can be built using build automation tools such as Apache Maven, Gradle (taverna-mobile) and Rake (taverna-databundle-viewer), resulting in reproducible builds. Each repository has a README file, e.g. https://github.com/apache/incubator-taverna-engine#building, that describes how to build the repository. There are also build instructions available on the website. Furthermore, builds are routinely verified by Jenkins (e.g., https://builds.apache.org/user/stain/my-views/view/taverna/). (But note that the *-workbench repositories are ***TBD***.)

- **CD40:** *The full history of the project's code is available via a source code control system, in a way that allows any released version to be recreated.*

  - Apache Taverna's source code is hosted on http://git.apache.org/ in multiple git repositories. (See https://git-wip-us.apache.org/repos/asf?a=project_list&s=taverna&btnS=Search and https://taverna.incubator.apache.org/download/code/.) Each release has a tag (based on release candidate VOTE identifier), e.g. https://github.com/apache/incubator-taverna-engine/releases/tag/3.1.0-incubating. Note: pre-Apache releases are tagged as old/*.

- **CD50:** *The provenance of each line of code is established via the source code control system, in a reliable way based on strong authentication of the committer. When third-party contributions are committed, commit messages provide reliable information about the code provenance. [2]*

  - Apache Taverna uses Git for version control. Every commit has a unique id. Commits are only allowed using ssh keys. All third part commits are checked for compliance with the Apache way before being merged.

Licenses and Copyright

- **LC10:** *The code is released under the Apache License, version 2.0.*

  - Apache Taverna code is released under the Apache License, version 2.0
  - See also License review

- **LC20:** *Libraries that are mandatory dependencies of the project's code do not create more restrictions than the Apache License does. [3, 4]*

  - Apache Taverna does not contain any restrictive dependencies.

- **LC30:** *The libraries mentioned in LC20 are available as Open Source software.*

  - All the Apache Taverna dependencies are available as Open Source software.

- **LC40:** *Committers are bound by an Individual Contributor Agreement (the "Apache iCLA") that defines which code they are allowed to commit and how they need to identify code that is not their own.*

  - All Apache Taverna committers are bound by an ICLA. (Committers are listed on the Apache Taverna home page.)

- **LC50:** *The copyright ownership of everything that the project produces is clearly defined and documented. [5]*

  - All Apache Taverna source code contains the appropriate Apache copyright text.

Releases

- **RE10:** *Releases consist of source code, distributed using standard and open archive formats that are expected to stay readable in the long term. [6]*

  - Apache Taverna is based on open source, community maintained languages (list needed - Java, Android).

- **RE20:** *Releases are approved by the project's PMC (see CS10), in order to make them an act of the Foundation.*

  - Apache Taverna releases are voted for by the project PMC following the appropriate Apache release procedures and the Taverna release preparation and release voting guidelines.

- **RE30:** *Releases are signed and/or distributed along with digests that can be reliably used to validate the downloaded archives.*

  - Apache Taverna releases are distributed along with SHA1 and MD5 digests to allow validation of downloads.

- **RE40:** *Convenience binaries can be distributed alongside source code but they are not Apache Releases -- they are just a convenience provided with no guarantee.*

  - Apache Taverna ...???

Quality

- **QU10:** *The project is open and honest about the quality of its code. Various levels of quality and maturity for various modules are natural and acceptable as long as they are clearly communicated.*

  - Apache Taverna uses the Apache JIRA bug tracker system to list current issues.

- **QU20:** *The project puts a very high priority on producing secure software. [7]*

  - Apache Taverna ...

- **QU30:** *The project provides a well-documented channel to report security issues, along with a documented way of responding to them. [8]*

    - Apache Taverna uses mailing lists as well as IRC channels to communicate, all of which are documented on the website.

- **QU40:** *The project puts a high priority on backwards compatibility and aims to document any incompatible changes and provide tools and documentation to help users transition to new features.*
    - Apache Taverna ...backwards compatibility of what?

- **QU50:** *The project strives to respond to documented bug reports in a timely manner.*
    - Apache Taverna strives to respond to bug reports as quick as is practicable. Most are responded to within 48 hours.

Community

- **CO10:** *The project has a well-known homepage that points to all the information required to operate according to this maturity model.*
    - Apache Taverna ...

- **CO20:** *The community welcomes contributions from anyone who acts in good faith and in a respectful manner and adds value to the project.*
    - Apache Taverna ...

- **CO30:** *Contributions include not only source code, but also documentation, constructive bug reports, constructive discussions, marketing and generally anything that adds value to the project.*
    - Apache Taverna ...

- **CO40:** *The community is meritocratic and over time aims to give more rights and responsibilities to contributors who add value to the project.*
    - Apache Taverna ...

- **CO50:** *The way in which contributors can be granted more rights such as commit access or decision power is clearly documented and is the same for all contributors.*
    - Apache Taverna ...

- **CO60:** *The community operates based on consensus of its members (see CS10) who have decision power. Dictators, benevolent or not, are not welcome n Apache projects.*
    - Apache Taverna ...

- **CO70:** *The project strives to answer user questions in a timely manner.*
    - Apache Taverna ...

Consensus Building

- **CS10:** *The project maintains a public list of its contributors who have decision power -- the project's PMC (Project Management Committee) consists of those contributors.*
    - Apache Taverna ...

- **CS20:** *Decisions are made by consensus among PMC members [9] and are documented on the project's main communications channel. Community opinions are taken into account but the PMC has the final word if needed.*
    - Apache Taverna ...

- **CS30:** *Documented voting rules are used to build consensus when discussion is not sufficient. [10]*
    - Apache Taverna ...

- **CS40:** *In Apache projects, vetoes are only valid for code commits and are justified by a technical explanation, as per the Apache voting rules defined in CS30.*
    - Apache Taverna ...

- **CS50:** *All "important" discussions happen asynchronously in written form on the project's main communications channel. Offline, face-to-face or private discussions [11] that affect the project are also documented on that channel.*
    - Apache Taverna ...

Independence

- **IN10:** *The project is independent from any corporate or organizational influence. [12]*
    - Apache Taverna ...

- **IN20:** *Contributors act as themselves as opposed to representatives of a corporation or organization.*
    - Apache Taverna ...

Footnotes from Apache Project Maturity Model

[1] "For distribution to the public at no charge" is straight from the from the ASF Bylaws at http://apache.org/foundation/bylaws.html.
[2] See also LC40.
[3] It's ok for platforms (like a runtime used to execute our code) to have different licenses as long as they don't impose reciprocal licensing on what we are distributing.
[4] http://apache.org/legal/resolved.html has information about acceptable licenses for third-party dependencies
[5] In Apache projects, the ASF owns the copyright for the collective work, i.e. the project's releases. Contributors retain copyright on their contributions but grant the ASF a perpetual copyright license for them.
[6] See http://www.apache.org/dev/release.html for more info on Apache releases
[7] The required level of security depends on the software's intended uses, of course. Expectations should be clearly documented.
[8] Apache projects can just point to http://www.apache.org/security/ or use their own security contacts page, which should also point to that.
[9] In Apache projects, "consensus" means widespread agreement among people who have decision power. It does not necessarily mean "unanimity".
[10] For Apache projects, http://www.apache.org/foundation/voting.html defines the voting rules.
[11] Apache projects have a private mailing list that their PMC is expected to use only when really needed. The private list is typically used for discussions about people, for example to discuss and to vote on PMC candidates privately.
[12] Independence can be understood as basing the project's decisions on the open discussions that happen on the project's main communications channel, with no hidden agendas.