Does Open Source need Standards Bodies?

Doug Davis, STSM, IBM
Start a new OSS Project

New project is started:

• Pick a spot: e.g. github.com, sourceforge.net
  • No barrier to entry

• Could be seeded with existing code
• Could be starting from scratch

• Could be an individual
• Could be one or more companies collaborating

• Goal: develop a common/shared code base
• Pool development resources across the community
**Open Source**

**Code and API specification developed in tandem:**

- Typically, driven by customer/end-user requirements
- Testing is done simultaneously as the code and API specification are developed
  - **Code correctness testing**
- Continuous cycle of testing and development
Open Source

At some point a "release" is created:

• Binaries, code and API specifications are deemed "ready"

• Feedback from community

• The cycle continues for the next release
Open Source

Code can now be used in production:

• Usually some statement about API, and functional, stability and versioning is made

• Whether they adhere to that or not....
Key points:

- No/low barrier to entry
- Code and API specifications are jointly developed
- Usually community based developed. But sometimes driven by one company
- Testing is focused on this codebase

Question: Is the "API specification" a "standard"?
What is a "Standard"?

• Websters.com:
  
  **Noun**
  • something considered by an authority or by general consent as a basis of comparison; an approved model.

  **Adjective**
  • of recognized excellence or established authority
  • authorized or approved

• In Open Source who is the "authority"?
  • Perhaps the marketplace?
Challenges with OSS

• The language-of-the-day influences the API
  • Language specific artifacts are exposed in the API
  • E.g. go-lang text templating

• At some point the language you're using today will become COBOL!
  • The "single" implementation becomes, almost, a single point of failure
  • When (if) a new version is created will they fork the spec?
    • What is the impetus to remain backwards compatible?

• Often lacks "enterprise" requirements
  • E.g. Internationalization, multi-arch support are often afterthoughts

• Differentiation is limited to extensions
Standards Development Organizations

Initial Draft of Specification

**Starts with an idea:**

- Typically one or more *companies* collaborate on an input specification
- Might be based on existing code
- But not a requirement
Standards Development Organizations

Propose the idea to an SDO:

• Find an appropriate SDO
  • Convince them the idea is worthy

• Each SDO has their own "process"
  • Some more open than others
  • Some require $ (pay to play)
  • Some are very political

• Some SDOs do spec and code at the same time - e.g. W3C/HTML 5.0
  • But, typically not grass roots
  • Much more overhead, bureaucracy than github
Standards Development Organizations

**Develop the Specification:**

- Specification is developed
- Input from working group members
- Sometimes the public
- Working group members may represent customers
- No code is necessary yet...
Testing:

- Most SDOs will require some level of testing prior to publication

- **Spec correctness**
  - PoC code, NOT necessarily product code

- **Interop testing**
  - Many SDOs require multiple implementations of the specification
  - **Goal is to avoid vendor and implementation lock-in**

- Cycle between development and testing
Standards Development Organizations

**Standard is published:**

- After SDO/WG requirements are met
- Statement of compliance and versioning are usually required
- The cycle continues for the next release
Implementation and productizing:

- Now the "production" grade code is developed
- Many companies will not touch a spec until it's released
- Real-world issues may not be found before v1.0
- Feedback is provided for next release
- Plugfests to test interop on product level code
International Standards Bodies:

- When ready, a version is taken to an International Standards Body
  - Second level of standardization - at a global level

- Promote specification/interop at global level
  - National Bodies get to weigh-in

- Some companies and countries require a specification to be approved by an ISB as part of their contract
Standards Development Organizations

Key points:

• Focus is on spec first, code second

• **Multiple implementations** of the specification is critical
  • Can differentiate in implementation choices

• Processes followed are well-established/trusted
• It's about providing **interoperability** and **stability** to the industry
  • To some these are what define a "Standard"

• Lacks real-world verification until standard is published
Challenges with developing a spec in an SDO?

• Sometimes too bureaucratic, political, not grounded in reality

• Feedback loop is too long
  • Waiting until after a spec is published for feedback is too late
  • SDO release cycles are not normally as short as OSS projects'

• Perception: APIs are rarely static and SDOs prevent innovation
  • Most projects view taking their specification to an SDO as something that will negatively impact their ability to continue to evolve

• Reality: This is no different than creating a release of an OSS API
  • Both are (or should be) set in stone from that point on
  • All changes are for subsequent versions of the API
What Value does an SDO offer?

• Access to different types of customers
  • Open the aperture to customers who are not part of existing OSS projects
    • Perception: OSS is just for coders (wild west), right?
    • OSS might exclude some companies, governments, influential customers - more later...

• Encourages multiple implementations to void "lock-in" and language atrophy

• Seal of approval as a "real standard"
  • Due to its well defined, trusted, processes and governance models

• Avoids risk of a country trying to fork the community due to no official "standard"
  • Causes pain and confusion for everyone, especially customers
Potential Procurement Roadblocks

• Many countries, e.g. in the EU, have stringent procurement regulations:
  • Can reference formal standards (e.g. ISBs)
  • Can reference consortia standards (e.g. w3c) if standard identified by EU Commission
  • If not, can only reference the technical features desired, but not a spec/OSS directly

• In many EU countries hospitals, universities, etc. fall under public procurement regulation
Both have a "consortium" working together towards a common goal

**Open Source**
- Goal: shared code/spec
- Self derived authority
- Code proves spec prior to release
  - Immediate real-world feedback
- Possible **implementation** lock-in
- Testing: coding bugs & pluggability
- Differentiation? extensions

**Standards Body**
- Goal: shared spec/API
- Recognized authority
- Well established/trusted processes
  - Inclusive and open
- Discourages **vendor** lock-in
- Testing: interop
- Differentiation? impl & extensions
Is there a path forward where they can co-exist?
A Proposal...

• **OSS projects continue as they do today**
  • Develop the code and API specification in tandem
  • After agreed upon releases, submit API specification to ISB for "ratification"
  • OSS project incorporates feedback into next version of the specification

• **Increases the aperture to a new set of customers/requirements**

• **Attain the SDO "seal of approval" for a "standard"**
  • Meet certain customer requirements
  • Reduces chances of a national body "forking" the community
  • Would encourage additional implementations of the APIs

• Exploring a trial run of this idea with a Cloud Native specification
Thank You!
Some humor...

https://xkcd.com/927/

http://www.gocomics.com/foxtrot