Open Source Software Development

Working on OSS Projects

Thái Minh Tuấn - Email: minhtuan@ctu.edu.vn

Learning Objectives

- Contributing to OSS Projects
- Respecting and Encouraging Diversity in OSS
- Constructing an Open Source Strategy
- TODO Group
- OpenChain Project, FOSSology, SPDX, CHAOSS
- Leadership vs. Control: Why Do Many OSS Projects Fail?

How to Contribute OSS Projects Properly (1/2)

- Before making contributions to any OSS project you should:
  - Investigate it, understand its workflow and style
  - Identify the scope and nature of your work
- Identify how the project communicates
  - Mailing list, existing archives
  - Joint Internet Relay Chat (IRC) network (if any)
- Understand how contributions are submitted
  - Mailing lists? Email?
  - A revision control system, such as git or subversion?
- Study previous history
  - Has your idea been considered before and rejected?
  - Is there already an individual or group working on the idea?
    - If so, them in many cases, not start over

How to Contribute OSS Projects Properly (2/2)

- Better to offer your services for testing, finding bugs, and similar work
  - Rather than beginning by submitting code
- Be competent at whatever programming or scripting language the project uses
  - Developers will be impatient in correcting your code
- Balance between asking for suggestions and review early in the process
- Be polite, respectful, avoid obscenities, flaming and trolling

Contributing to OSS Projects

- Things to consider when contributing to OSS projects:
  - Study and Understand the Project DNA
  - Figure Out What Itch You Want to Scratch
  - Identify Maintainers and Their Work Flows and Methods
  - Get Early Input and Work in the Open
  - Contribute Incremental Bits, Not Large Code Dumps
  - Leave Your Ego at the Door: Do Not Be Thin-Skinned
  - Be Patient, Develop Long-Term Relationships, Be Helpful

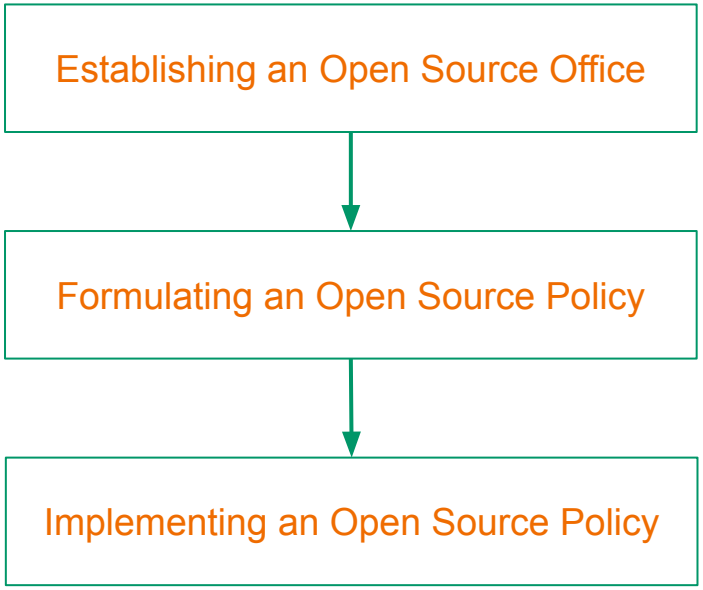
Respecting and Encouraging Diversity in OSS

- Diversity in an OSS project can mean many different things:
  - Race and national origin
  - Sex and gender identity
  - National and geographical/regional characteristics
    - Including both language and cultural differences
  - Religious beliefs and political views
  - Acceptance of different opinions and methods
    - How the project should take shape and develop

- *The right thing to do* to accept contributors and reviewers from divergent backgrounds,
- Also leads to a better project due to unleashing more sources of new ideas, approaches, and contributions

Constructing an Open Source Strategy

- Key considerations involved in developing a strategy for integrating open source:



Open Source Program Office

- Serves as a center for functions:
  - Choosing OSS code to be used
  - Facilitating adoption of the code and popularizing its availability and usage
  - Keeping track of usage
  - Auditing compliance considerations
  - Making sure proper information and training is available to all employees, as needed
- Determine the placement of such an office within the organization
  - Should be relatively high in the organizational chart
    - Research and development command structure
    - Legal department
  - Whomever is responsible must be sensitive to both the developer and legal needs

Designing an Open Source Policy

- How much of the product is desired to be OSS?
  - Replacing current closed source components open source ones
- Which license(s) are a good fit to your needs
  - Which licenses can you use in various OSS components you integrate, such as libraries, APIs, etc?
- Enough understanding (and legal help) to set up procedures for using both proprietary and open source code?
- How will you review the code to make sure things are being done properly?
  - Automated license compliance tools?
  - Linux Foundation's Open Compliance Program

Implementing an Open Source Policy

- A policy is worthless without:
  - A structure for ensuring implementation, and
  - Having a dedicated staff tasked with carrying it out
- Training will need to be developed (or outsourced):
  - Developers, Procurement, Legal, Security
- Community participation and outreach can also be critical
  - Who are designated spokespeople?
  - Who monitors developer participation and contributions?
- Clear assignment of responsibility is required in all areas

TODO Group

- Founded in 2014, fostering the adoption and spread of open source software in the corporate business community
- Founding members: Box, Dropbox, Facebook, GitHub, Google, HPE, Khan Academy, Microsoft, Square, Stripe, Twitter, Walmart and Yahoo

TODO Group

- Focuses on:
  - Sharing best practices for running large open source programs.
  - Codifying goals for successfully managed projects.
  - Spreading knowledge about helpful tooling and instrumentation.
  - Listing and describing member projects that are successful examples.
- Every organization has its own unique set of needs, challenges and practices. Sharing allows for faster and better establishment of enabling open source strategies.

TODO Guides

- Guides on building a successful open source program:
  - How to create an open source program
  - Measuring your open source program's success
  - Tools for managing open source programs
- Guides for open source program management:
  - Using open source code
  - Participating in open source communities
  - Recruiting open source developers
  - Starting an open source project
  - Open source reading list
  - Improve your open source development impact
  - Shutting down an open source project
  - Building leadership in an open source community
  - Setting an open source strategy

TODO Case Studies

- Case studies by a diverse group of companies, that describe how they adopted and manage open source
  - Autodesk
  - Capital One
  - Comcast
  - Dropbox
  - Facebook
  - Microsoft
  - Oath
  - Red Hat
  - Salesforce
  - SAP

OpenChain Project

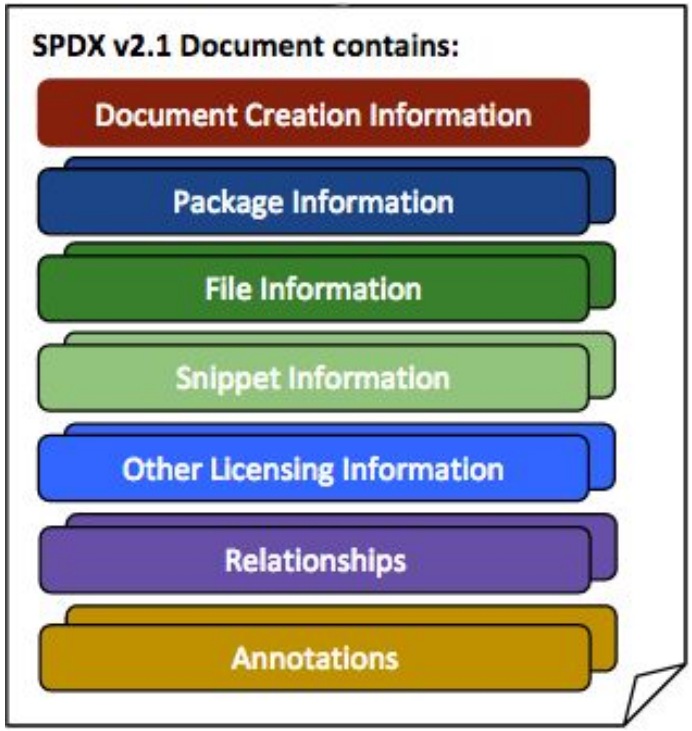
- Established an open set of requirements for effective management of FOSS
- An organization meets this specification of requirements
  - Certified as OpenChain Conforming
  - Receive the right to display the logo indicating that it is fully compliant

FOSSology

- A system/toolkit for measuring and documenting open source license compliance
  - Scan individual files/entire source archives, and identify licenses, copyrights etc.,
  - Document the results, and help identify where information is lacking or is poorly documented
- FOSSology Scanning Features (searches for licenses using)
  - Nomos: regular expression matching + context identification
  - Monk: text-based searching, report back on license variations
  - Copyrights: searching for “copyright” and “(C)” in the text
  - Export Control Codes (ECC): Regular expression searching

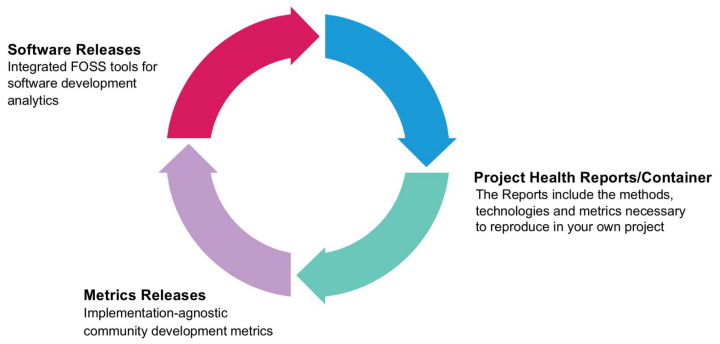
Software Package Data Exchange (SPDX)

- Open standard provides a common format for sharing data about software copyrights, licenses, security information, etc.



Community Health Analytics Open Source Software (CHAOSS)

- Developing tools and metrics for evaluation of the health of open source projects and ecosystems
  - Increasing sustainability over time
  - Making good, informed decisions about how to be involved
- Goals:
  - Establish excellent metrics for measuring community activity, contributions, and health
  - Produce software tooling or doing the analysis
  - Build project health reports



FOSSology

- A system/toolkit for measuring and documenting open source license compliance
  - Scan individual files/entire source archives, and identify licenses, copyrights etc.,
  - Document the results, and help identify where information is lacking or is poorly documented
- FOSSology Scanning Features (searches for licenses using)
  - Nomos: regular expression matching + context identification
  - Monk: text-based searching, report back on license variations
  - Copyrights: searching for “copyright” and “(C)” in the text
  - Export Control Codes (ECC): Regular expression searching

Leadership vs. Control

- OSS projects can be organized in many ways,
  - From very tight control, to an almost anarchistic ecosystem
  - Wide choice of operating methods
- Advantages of leading a project, as opposed to just controlling it
  - Loosen the Chains
    - Good ideas can originate from many different contributors
    - Make the final decision, but listen and give proper consideration to debates in the community
  - Mentoring
    - One level of leadership, or more, including subsystem maintainers
    - Proper mentoring: Empowering people, more efficient workflow, long term vitality
  - Building Trust
    - Without trust, an open source project cannot function
    - Contributors must trust that their submissions will be treated with respect and given reasonably prompt consideration
    - Project leaders must trust that subsystem maintainers are qualified and capable of doing their job

Why Do Many OSS Projects Fail?

- Vast majority of OSS projects do not succeed
- Reasons are varied, but include:
  - Insufficient interest
  - Competition from more established projects that duplicate the intended functionality
  - Poor leadership.
  - Not enough developers.
  - Insufficient funding
  - Insufficient or uninformed attention to licensing issues