Contributing to Open Source Software Projects

THE LINUX FOUNDATION

How to Contribute Properly to Open Source Software?

- Investigate the project, understand its workflow and style, identify the scope and nature of your work
- Identify how the project communicates:
  - Review mailing lists
  - Study archives
  - Join an Internet Relay Chat (IRC) Channel, if one exists
- Understand how contributions are submitted:
  - Mailing lists
  - Email
  - Revision control system (e.g. git or subversion)
- Study previous history

How to Contribute Properly to Open Source Software? (Cont.)

- Check if the project offers veteran contributors as mentors
- Offer your services for testing, finding bugs, etc., before you begin submitting code
- Make sure you are competent at whatever programming or scripting language the project uses
- Find the right balance between asking for review and suggestions early in the process, and waiting too long and dumping a lot of work on people at once
- Be polite and respectful, and avoid obscenities, flaming and trolling

Study and Understand the Project DNA

- Remember that unless you are starting a new project, there will already be a community established ethos and formal or informal leadership structure
- Ask questions:
  - Why does this project exist and why was it started?
  - Has it diverged far from its original purpose?
  - Is the contributing community large or small? Continuously active or only sporadically so?
  - Does it have a collective or singular governance structure?
  - What kind of licenses does it adopt? Is there a Contributor License Agreement (CLA) you have to agree to?
- Keep in mind that the majority of OSS project never really take off

Figure Out What Itch You Want to Scratch

- Most contributors get involved in open source projects because:
  - There is a bug/problem that interferes with their use of the project
  - They want to add additional functionality to the project
  - They want to learn something new
- Some projects have a “janitorial” list of things to do; these relatively straightforward tasks can help you get your feet wet
- Beginning contributors who go on to become important to a project rarely start with patches that do not affect functionality