Getting Involved in an Open Source Project

- Why listen to me?
- Reasons to get involved in a project
- What fuels a project?
- How to choose a project
- How to be a good contributor
  - Challenges
- How to join a project
- How to be a good guru

Bob Igo, StormLogic, LLC

Why listen to me?

- Projects I've worked on or contributed to:
  - KnoppMyth (major: core contributor; automation guru)
  - Ångström (minor: bug reporing, bug fixes, mailing list help, documentation)
  - Linux kernel (minor: fixed buggy ATI Remote Wonder driver)
  - kmymoney (minor: bug reports, documentation improvement)
  - XML::Twig perl module (trivial: filed a single bug report that led to a fix)

Reasons to Get Involved in a Project

- Learn or improve skills
- Apply skills
- Technical skills
  - QA, Technical Writing, Coding, Design, Telecommuting
  - Training, Revision Control, Hosting
- Social skills
  - influence at a peer level
  - communicate efficiently, effectively, and unambiguously
  - project management

Learning or improving skills

- Personal satisfaction
  - ”Everybody stand back! I know regular expressions!” --xkcd
- Career opportunities
  - Learn skills that your job won't let you learn... then get a better job.
- Experiment
  - Ever wanted to try out some new software on 30,000 people?
  - Want to refine your new tech writing style before using it with clients?

Applying skills

- ”We are the ones we've been waiting for.”
  -- The Elders Oraibi, Arizona Hopi Nation
- Is some piece of Open Source software not doing what you want it to do?
- Does project XYZ release too infrequently?
- Do poorly translated documentations lead to the injury of thinking organ?
- Ever wanted random internetizens to thank you?

Project Fuel

- What keeps an Open Source contributor (and, by extension, a project) productive?
- My hypothesis:
  - Productivity = Free Time * Enthusiasm * Skill
- One of your unwritten duties is to try to maximize those.
- If you've ever experienced a grumpy project guru, check to see if it related to one of the fuels above.
- When you think about how to maximize project fuel, lots of things become clear. (more later)

How to Maximize Time

- Prevent or catch things that would take time away from productive work.
  - Solve problems that cost time regularly (e.g. bad documentation, lack of Revision Control, lack of bug tracking, etc.)
- Answer questions to which you know the answer, even if you're new to the project.
  - Much like seeding a torrent, this frees up your peers to contribute in ways that benefit you.

How to Maximize Enthusiasm

- Give people credit for their ideas.
- Thank everyone for their contributions, no matter how small.
- If it's clear that someone is burning out, reach out to them.
- When conflict arises, try to be the voice of reason.
- Find new and cool ways to use the project, and let the community know.
- Remember that you don't employ the community; remind others, too.

How to Maximize Skill

- Lift people up to your level when you can.
- Don't discourage people for imperfect work; point out what isn't right, but praise them for their progress.
- Good training resources can help get the untrained up to speed.

How to Choose a Project

- Determine how members communicate
  - Forum, mailing list, IRC?
  - Are contributors separate from users?
- What's the organizational structure?
  - Is the structure compatible with you?
- Who are the major personalities involved?
  - From their writings, do they seem to be people you'd enjoy working with?
- Is the project healthy?
  - Time since last release?
  - Number of unanswered questions

Being a Good Contributor

- You can be a valuable contributor without being a guru.
  - Just remember: you may be enthusiastic, but one of your goals is to give contributors more free time.
- Try to read up on the project-centric theme of your contribution, whether it's a process, UI design, coding, QA, etc.
  - this can avoid rehashing something the others have already seen
  - it also lets you say that you did search before making your contribution

Being a Good Contributor

- Remember, an Open Source project is done at a peer level. Any authority is granted by the good will of the project members.
  - Even the project leads are potentially a fork away from not being leads anymore.
    - c.f. XFree86/xorg
  - You need to influence others based on the merits of your contributions. Ultimately, nobody can tell anybody what to do.
    - Although if you implement your idea and get much negative feedback, you may have no choice but to redo it.

Challenges: Communication

- All of these interactions tend to be different:
  - Meeting new people face-to-face
  - Meeting old friends face-to-face
  - IMing or emailing new people
  - IMing or emailing old friends
  - IMing or emailing people you've never met
- Which of those best describes social news sites, newsgroups, or mailing lists, where flame wars rule?
- Which of those apply when you start to work with an Open Source project?

Challenges: Communication

- Unlike most office jobs, contributors may be from all over the world. Factors here:
  - Timezones
  - Language barriers
  - Different standards
    - mm/dd/yyyy vs dd/mm/yyyy
- For all these reasons, it's very important to learn to communicate clearly, concisely, and unambiguously.

How to Join a Project: Programmer

- An example of how to introduce yourself and begin contributing:
  - "Hi! I've noticed that there are some instances in the Foo GUI where the default selections don't make sense for the hardware I have. I don't see any bug reports that cover this problem. I'm a programmer, and I would like to help fix this. Who should I talk to about this?"
- It's polite, concise, and it helps the gurus know how to reply without asking more questions. You save them free time while offering your enthusiasm.

How to Join a Project: Informal QA

- Another example of how to introduce yourself and begin contributing:
  - "Hi! I've noticed that there are some instances in the Foo GUI where the default selections don't make sense for the hardware I have. I don't see any bug reports that cover this problem. **I'm not a programmer, but I'm willing to provide information and testing to help fix the problem.** Who should I talk to about this?"
- Assuming you back up your words with actions, this is a great start.

How to Join a Project: Formal QA

- A final example of how to introduce yourself and begin contributing:
- File a bug report:
  - Summary: The Foo GUI crashes when users backtrack too much.
  - Description: Proceed as normal to the Foo GUI's final screen, then hit ”BACK” until you get to screen 1. Repeat this process once more, and the Foo GUI crashes.
- Developers love good bug reports because they save time.

How to Join a Project

- An example of how NOT to introduce your contribution (based on a real example):
  - "I have been involved in the software industry for almost 30 years, and in that time I have yet to see a GUI so thrown-together and haphazard as the Foo GUI. It violates 9 of the 10 guidelines of good design, and I have no idea how this got past everyone. Here are all the ways in which it violates the guidelines..."
  - It's arrogant, self-promoting, and insulting, and shows ignorance of peer-level collaboration.

Being a Good Guru

- Try to cultivate and maintain enthusiasm.
  - Encourage people
  - Thank them
  - Give them credit
- Common sense isn't common; it's experiential.
  - When someone does something wrong, teach them what is right. Consider improving documentation to prevent the problem from recurring.
- When someone makes a valid contribution in a horrible way, try not to let others stigmatize the contribution.

That's It!

- Questions?