Supporting Distributed and Decentralized Projects: Drawing Lessons from the Open Source Community

Justin R. Erenkrantz
Institute for Software Research
University of California, Irvine
http://www.erenkrantz.com/oopsla/

What can we learn?

- *Lots of successful open-source projects*
- *Identify areas of impact*
  - *Decentralization and distribution*
- *Examine techniques and tools*
  - *Variations in tools and processes*
- *Helpful for starting new projects*

Decentralized and Distributed

- *Independent collections form together*
  - *Should have self-interest at heart*
  - *Should believe can be beneficial*
- *People will not be face-to-face*
  - *Global reach and openness*
  - *Time delays must be accounted for*

Decision-Making

- *How do we decide what to do?*
- *Techniques: Project leader, meritocracy*
- *Exemplar: Linux, Apache*
- *Enhancements: Recognizing tradeoffs*

Accountability

- *Who will stand behind the product?*
- *Techniques: for-profit, non-profit*
- *Exemplar: PostgreSQL, FreeBSD*
- *Enhancements: Introducing clarity*

Communication

- *Where do we exchange ideas?*
- *Techniques: Mailing lists, asynchronous*
- *Exemplar: All*
- *Enhancements: Balancing granularity*

Awareness

- *How do I know what others are doing?*
- *Techniques: Status updates, Discussion*
- *Exemplar: Apache HTTP Server*
- *Enhancements: Better tools*

Historical Rationale

- *Why was this activity performed?*
- *Techniques: Archives, design documents*
- *Exemplar: Perl*
- *Enhancements: Creating standards*

Design Rationale

- *Why does the code look like this?*
- *Techniques: Developer docs, examples*
- *Exemplar: AbiWord*
- *Enhancements: Synchronization*

Participation

- *How can we entice others to join?*
- *Techniques: Clear tutorials, guidelines*
- *Exemplar: KDE*
- *Enhancements: Creating standards*

Controlling Participation

- *How do we manage people?*
- *Techniques: Annotating contributions*
- *Exemplar: Python*
- *Enhancements: Integration*

Source Code

- *How do people know what we're doing?*
- *Techniques: Public, optimistic resolution*
- *Exemplar: All*
- *Enhancements: Decentralized repositories*

Issue Tracking

- *How do users report problems?*
- *Techniques: Soliciting developer help*
- *Exemplar: Mozilla*
- *Enhancements: Easy-to-use tools*

Documentation

- *How do users learn about the system?*
- *Techniques: Distinct team, annotations*
- *Exemplar: PHP*
- *Enhancements: Separate code and docs*

Testing

- *How do we know if what we have is good?*
- *Techniques: Reviews, automated tests*
- *Exemplar: Subversion*
- *Enhancements: Optimizing test runs*

Release Management

- *How do users receive the project?*
- *Techniques: Mirroring, versioning*
- *Exemplar: Debian*
- *Enhancements: Managing distributions*

Discussion

- *Variety of projects and domains examined*
- *Variety of techniques and tools used*
- *Only a few areas have consensus*
- *Beginning of a roadmap for adoption*