

Release Management Within Open Source Projects

*Justin R. Erenkrantz
Institute for Software Research
University of California, Irvine*

Importance

- *Release management focuses on delivery*
 - *“Getting software to user”*
 - *Directly user-focused task*
- *Lots of variations between projects*
- *Need for a taxonomy*
- *Process and tools are still evolving*

Decentralization & Distribution

- *Compensating for decentralization*
 - *Users may desire canonical releases*
 - *Creation of virtual organizations*
- *Compensating for distribution*
 - *Global reach*
 - *Often no contact for assistance*

Classification of Policies

- *Responsibility - who?*
- *Acceptance - when?*
- *Versioning - what?*
- *Distribution - where?*
- *Binaries - how?*

Examination of Three Projects

- *Linux Kernel*
- *Subversion*
- *Apache HTTP Server*
- *Chosen for variety*
- *Directly involved in two of these projects*

Linux Kernel

- *Highly centralized process*
- *Source tree separated into branches*
 - *Multiple active branches at one time*
- *Dedicated release manager per branch*
 - *Complete authority*

Linux Kernel

- *Extensive use of release candidates*
- *Stable branches, unstable branches*
- *Releases mirrored on kernel.org via FTP*
- *No official packaging contributions*
 - *Source only*
 - *Others provide binaries*

Subversion

- *Version control system*
 - *Principal funder is CollabNet*
- *CollabNet employees did releases*
- *Now have a volunteer release manager*
- *Milestones still determined by CollabNet*

Subversion

- *Must pass automated test suite*
- *Not yet at 1.0 release*
- *Centralized download location*
- *Users may contribute binaries*

Apache HTTP Server

- *Decentralized authority shared*
- *Committers volunteer to conduct release*
- *Releases do not have a formal schedule*
- *Three committers must approve release*

Apache HTTP Server

- *Automated test suite available*
- *Stable and unstable versions*
- *Distributed by mirrors*
 - *Custom download system*
- *Only committers can contribute binaries*

Conclusions

- *Core organizational structure dominates*
- *Room for improvements*
 - *Linux: Testing*
 - *Subversion: Scalability*
 - *Apache: Frequency*