Using FreeBSD to Promote Open Source Development Methods

Brooks Davis
Computer Systems Research Department
The Aerospace Corporation
brooks@aero.org
What is this talk about anyway?

• AeroSource
  – Set of tools to aid software development
  – Open Source software and methods evangelism
  – Demonstrating good software practice
  – Vehicle for cultural change

• FreeBSD's role
  – Hosting platform
  – Shining example of what Open Source Methods can achieve
What is The Aerospace Corporation?

• From www.aero.org:
  – Since 1960 The Aerospace Corporation has operated a federally funded research and development center in support of national-security, civil and commercial space programs. We're applying the leading technologies and the brightest minds in the industry to meet the challenges of space.
What is The Aerospace Corporation?

- Approximately 2400 engineers in nearly every discipline working most aspects of national security space
- Over 20 locations
Today's Software Culture(s)

- Two main camps
- Classic Software Engineers
  - Big, important software
    - Flight control, navigation, other mission software
  - Big expensive process
    - Lots of verification and validation
    - Complex, often painful change control process
Today's Software Culture(s)

- Engineering support software
  - Written to solve today's problem
    - ...but often lives for years or decades
  - Often extremely minimal process
    - Advanced revision control is file server and white board to manage locks
Problems With Today's Engineering Software Culture

• Code everywhere, mostly inaccessible to others
  – Duplicate code of highly varying quality

• Archaic code
  – Deprecated or dead language features
  – Obsolete practices
Problems With Today's Engineering Software Culture

- Often no revision control
  - Can't figure out which change broke things
  - Features get “lost” between revisions
  - Releases aren't repeatable
  - Locks on files or manual conflict resolution waste developer time
Open Source Software (OSS) and Methods to the Rescue

• An alternative to traditional, heavyweight software development methods
  – Can still write software to achieve a goal, not just write software

• OSS methods can produce high quality software
  – See Apache, FreeBSD, PostgreSQL, etc

• OSS methods are necessarily low friction
  – OSS developers usually have day jobs, lives, etc
Open Source Software (OSS) and Methods to the Rescue

- Making code accessible reduces duplication
  - May improve quality
- OSS tools reduce cost objections
Enterprise Source Software (ESS)

- Open Source, but restricted to the enterprise
  - Read the code within the enterprise
  - Build and run the code within the enterprise
  - Make changes to the code
  - Redistribute modified versions within the enterprise
Enterprise Source Software (ESS)

- Not merely using open source software within the enterprise
- ESS can become OSS
- OSS with modifications can be ESS
The FreeBSD Project as an Example

• Shows what can be achieved though open source methods

• Provides examples of working practices
  – Communications
  – Repository management
The FreeBSD Project as an Example

How the FreeBSD Project Works

10 March 2007

Robert Watson

FreeBSD Project

Computer Laboratory
University of Cambridge
The FreeBSD Project as an Example

The FreeBSD Project

- One of the most successful open source projects in the world
  - Can’t throw a stone without hitting a FreeBSD server
  - Root name servers
  - Major web hosts, search engines, and search traffic
  - Routing infrastructure
  - Foundation for major commercial systems
  - And much more...
- But the FreeBSD Project is much more than just a system:
  - Complete, integrated UNIX system
    - Multi-processing, multi-threading
    - Intel/AMD 32/64-bit, Itanium, PowerPC
    - UNIX, POSIX, BSD and System V interfaces
    - Multi-protocol network protocols
      - IPv4, IPv6, IPX/SPX, AppleTalk, 802.11, SCTP, ...
    - Standard and embedded
    - Extensive documentation
  - Over 16,600 third party packages

What do you get with FreeBSD?

Locations of FreeBSD Committers (March 2007)
The FreeBSD Project as an Example

How the FreeBSD Project Works: Self-Description as Advocacy

Robert N. M. Watson
Computer Laboratory
University of Cambridge

Introduction

- How the FreeBSD Project Works
- Introduction
- FreeBSD and Revision Control

Who are the Committers?

- Copyright (documented) committer born 1946
- Youngest committer born 1980
- Mean age 32.5, median age 31, standard 7.3
- Professional programmers, hobbyists, consultants, university professors, students...

- Try to make committers seem human
- Empathy

FreeBSD and Revision Control

- CVS
  - Primary revision control system
  - Many project activities in CVS
  - One control center
  - Technical control
  - Actually four rep
    - FreeBSD
    - Development
    - FreeBSD Ports
    - FreeBSD

- Perforce
  - Secondary revision control system
  - Supports heavily branched development
  - Branches development
  - Many clients
  - Good support
  - Multi-user access
  - Multi-user access
  - Perforce Development Branches
  - Revision Control the Future

- FreeBSD Project lives and breaths CVS/P4
- Importance of heavily branched development
Aerosource: Enterprise Source Software at Aerospace

Aerosource is a project management environment for Aerospace that allows the free sharing of ideas and source code within the Aerospace Corporation. Aerosource is built using the Trac project management software and utilizes the Subversion version control system for managing source code.

Getting Started

Create a new project

- Everything you need to know about Subversion
- Everything there is to know about Trac
- AerosourceLinks contains links to tutorials and other helpful documentation.
- ProjectList contains all current Aerosource projects.
- VideoTutorials
- Try it out in our Sandbox

Admin Links
AeroSource Goals

- Promote Enterprise Source Software
  - Encourage code reuse
- Provide developers with good development tools
- Modernize development processes
  - 1990s vs 1970s
AeroSource.aero.org
Implementation Requirements

• Required features
  – Version control including web access
  – Bug, issue, task tracking
  – Website and documentation management
  – All well integrated
  – Internal access only

• Basically SourceForge.net for internal use

• Also needed to be easy to use, operate, maintain, and modify
AeroSource.aero.org
Implementation Options

- SourceForge
- Gforge
- Trac
- Retrospectiva
- CVS/SVN + Bugzilla + Wiki
- ...

THE AEROSPACE CORPORATION
AeroSource.aero.org Implementation

- Trac
- Subversion
- PGSQL
- Apache
- FreeBSD
AeroSource experiences

- Existing CVS/SVN users eager to move
  - Less work for them
- New projects generally interested in many organizations
  - People are realizing that version control is good
- Some instances used for system management tasks
- Strong resistance to Enterprise Source Software ideas in some camps
Some Objections to ESS

- It's my code
  - Actually, the company owns is
- It's embarrassingly bad code
  - and others can help you improve it
- Rewriting basic algorithms is a right of passage
  - ...or a pointless waste of time?
Some More Objections to ESS

- Only I can maintain this code
  - Possibly true, but your co-workers aren't stupid
- Only I can use this code correctly
  - Hard to believe if it's documented
- People might submit changes and I'd have to review them
  - Isn't this a good situation to be in?
AeroSource Projects

- AeroSource
- AeroPorts
- Avant
- fellowship
- FireWatch
- SOAP
- many others
AeroSource Maintenance

• Eating our own dog food
  – Configuration, custom modules, scripts stored in an AeroSource project
  – Frontpage is the project wiki
  – Use the ticket system where possible

• Basic FreeBSD
  – Standard ports
  – plus local special use ports

• Separate backups with off site storage
AeroPorts

- AeroPorts
  - Aerospace specific ports
    - Bootstraping meta-port
    - Ports of internal software
    - Incubator for OSS ports
  - Adds ports/aero/<category>/<port> directories
  - Based on a posting to ports by Scot Hetzel
    - Generally works well
    - Minor problems with portupgrade
APT (AeroSpace Ports Tool)

- Integrates ports and AeroPorts
- Wrapper around portsnapsnap and svn
  - portsnapsnap updates ports and constructs INDEX* with “-l descfile” option
- portsnapsnap like interface
  - init-basic, init-aeronet fetch and extract
    - Add some /etc/make.conf entries
    - fetch wraps “portsnapsnap fetch”
    - update similar to “portsnapsnap update”
Conclusions

- AeroSource is attracting new customers regularly
- OSS methods are attracting developers
  - Efficient and effective methods
  - Free tools
- ESS is gaining traction
  - Work to do in some areas
The End
Future Work

- Improve automation
- Increase tutorial content and startup information
- Open source more of our tools
- Consider Trac follow on if deficiencies aren't corrected
- Introduce distributed version control
Acknowledgments

• Thanks to Michael Gorlick for pushing the initial idea of AeroSource

• Thanks to Mark Thomas for performing the majority of initial Trac setup and customization and being the front line advocate of AeroSource and ESS

• Slides from How The FreeBSD Project Works are used with the permission of Robert Watson

• All trademarks, service marks, and trade names are the property of their respective owners
Promoting AeroSource and ESS

- Internal and External presentations
  - Talks on OSS
  - Tutorial on OSS methods and ESS at GSAW2007
- Trac and Subversion demonstrations
- Management pressure