

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 through 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



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:
 - Root name servers
 - Major web hosts, search engines
 - Routing infrastructure
 - Foundation for major commercial products
 - And much more...
- But the FreeBSD Project is not a commercial product

10 March 2007



What do you get with FreeBSD?

- Complete, integrated UNIX system
 - Multi-processing, multi-architecture
 - Intel/AMD 32/64-bit, Itanium
 - UNIX, POSIX, BSD protocols
 - Multi-protocol network stack
 - IPv4, IPv6, IPX/SPX, AppleTalk, 802.11, SCTP, ...
 - Standard and embedded
 - Extensive documentation
- Over 16,600 third party developers

10 March 2007



Locations of FreeBSD Committers (March 2007)



10 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

21 March 2007
Robert Watson
FreeBSD Project
Computer Laboratory
University of Cambridge

Introduction

- What is FreeBSD?
- What is the FreeBSD Project?
- How does the FreeBSD Project work?
- And does it all depend on who you ask?
 - Caveat: kernel developer!

17 May 2007



4

Who are the Committers?

Who are the Committers? (2006-2007)

- Locations
 - 34 countries
 - 6 continents
- Ages
 - Oldest (documented) committer born 1948
 - Youngest (documented) committer born 1989
 - Mean age 32.5, median age 31, stddev 7.3
- Professional programmers, hobbyists, consultants, university professors, students ...

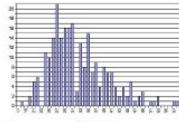
21 March 2007



Locations of FreeBSD Committers (March 2007)



FreeBSD Developer Age Distribution (March 2007)



- Try to make committers seem human
- Emp

17 May 2007

FreeBSD and Revision Control

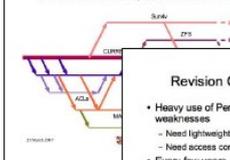
CVS

- Primary revision control system
 - Most project activity is in CVS
 - 10+ year revision history
 - One commit every
 - Technical limitations
 - Actually four repos
 - Homebrew - FreeBSD
 - Homebrew - FreeBSD
 - Homebrew - FreeBSD
 - Homebrew - FreeBSD

Perforce

- Secondary revision control system
 - Supports heavily branched development
 - FreeBSD developers
 - Guest accounts and project accounts
- Active project include
 - SMPing, TrustedBSD Audit, TrustedBSD MAC
 - TrustedBSD SEBSD, Alan Cox Superpages, uart
 - ARM, Summer of Code, trace, Xen, Sun4v
 - GEOM, Gjournal, ZFS, CAM locking, netperf, ...

Perforce Development Branches



Revision Control: the Future

- Heavy use of Perforce a symptom of CVS weaknesses
 - Need lightweight branching, history-aware merging
 - Need access control
- Every few years, consider options
 - Cost of migration very high - interrupt development, naitish developers, high risk
 - Currently evaluating several of revision control systems to see if any meet requirements

- FreeBSD Project lives and breaths CVS/P4
- Importance of heavily branched development

17 May 2007



17

AeroSource.aero.org

The screenshot shows a web browser window titled "Aerosource - Trac - Konqueror". The browser's address bar is empty, and the page content includes the following elements:

- Header:** "AEROSOURCE ENTERPRISE SOURCE SOFTWARE" with a search box and a "Search" button.
- Navigation:** A menu bar with links for "Login", "Settings", "Help/Guide", "About Trac", and "Contact Us".
- Project Management:** A row of buttons for "Project List", "Project Library", "Create Project", "Timeline", "Roadmap", "Browse Source", "View Tickets", "New Ticket", and "Search".
- Sorting:** A row of links for "Start Page", "Index by Title", "Index by Date", and "Last Change".
- Main Content:**
 - 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**
- Featured Projects (36 of 36 listed)**
 - [aero_idl](#)
 - [aeromatlib](#)
 - [Aerosource](#)
 - [Avant](#)
 - [Cell](#)
 - [cimus](#)
 - [cluster-portal](#)
 - [csdai](#)
 - [cuda-benchmarks](#)
 - [dance](#)
 - [deed-sandbox](#)
 - [DeviceDatabase](#)
 - [dtn](#)
 - [enterprise-person](#)
 - [eric-astro-tools](#)
 - [fellowship](#)
 - [FireWatch](#)
 - [Knowledge Management Apps](#)
 - [LabVIEW](#)
 - [m2html](#)
 - [mergesort](#)
 - [pad-redteam](#)
 - [phoenix-command](#)
 - [Quantum_Computing](#)
 - [RACE](#)
 - [reanalysis](#)
 - [reconfig-demod](#)

AeroSource Goals

- Promote Enterprise Source Software
 - Encourage code reuse
- Provide developers with good development tools
- Modernize development processes
 - 1990s vs 1970s

AeroSource.aero.org

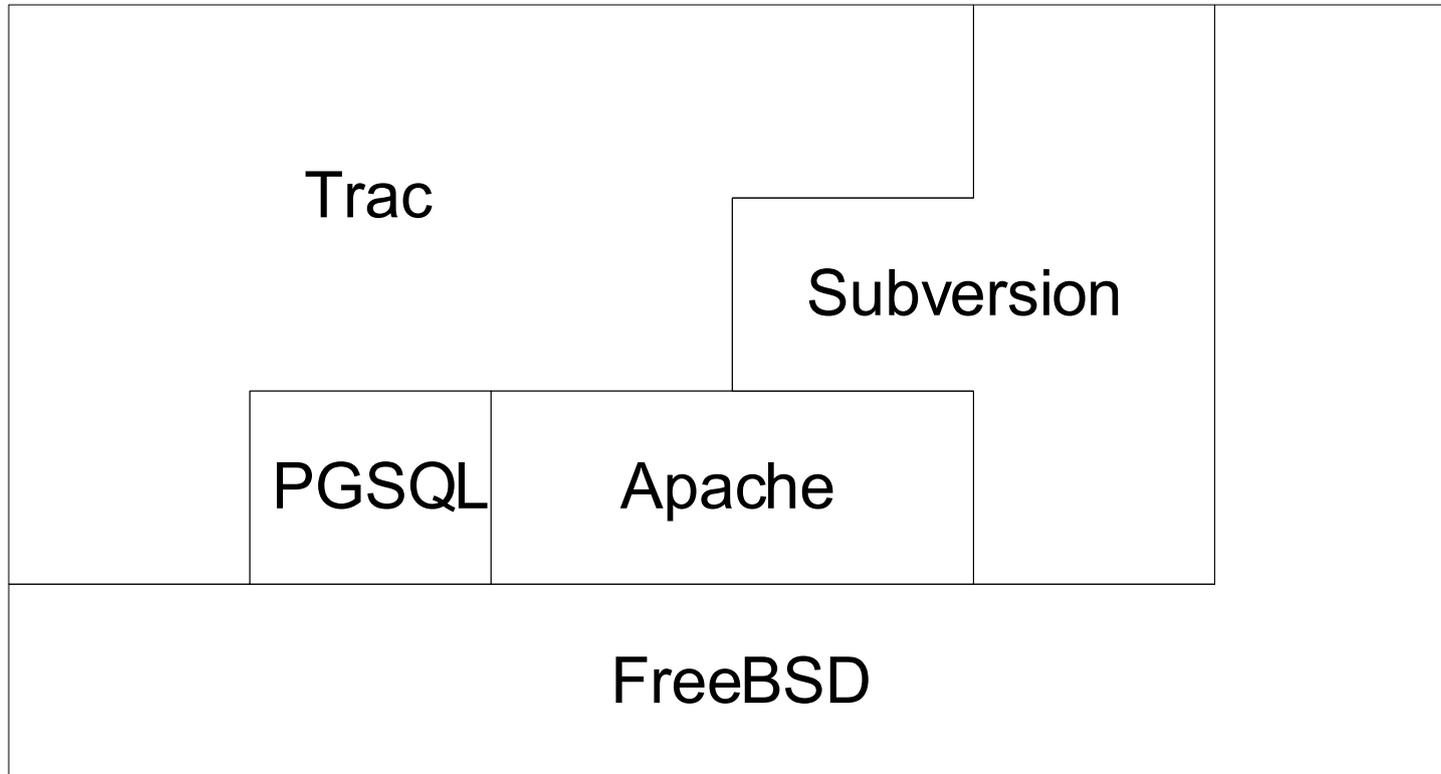
Implementation Requirements

- 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
- ...

AeroSource.aero.org Implementation



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 it
- 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
 - Bootstrapping 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 `portsnap` and `svn`
 - `portsnap` updates ports and constructs `INDEX*` with “`-l descfile`” option
- `portsnap` like interface
 - `init-basic`, `init-aeronet` `fetch` and `extract`
 - Add some `/etc/make.conf` entries
 - `fetch` wraps “`portsnap fetch`”
 - `update` similar to “`portsnap 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