

Two years in the trenches

The evolution of a software project

Greg “groggy” Lehey
FreeBSD Project
grog@FreeBSD.org
grog@auug.org.au
Melbourne, 4 September 2002



The three ages of UNIX

UNIX® is now a third of a century old. It evolved in three phases of roughly 11 years each:

- 1969—1980: a research project, little known outside AT&T except at some universities.
- 1980—1991: UNIX developed into a commercial operating system (UNIX System V, XENIX, SunOS, Ultrix and friends).
- 1991—2002: Free UNIX gained momentum.



Free software, first cut

- Free software has been around since software was invented.
- Until the late 60s, software ran only on specific machines. Piracy was impractical.
- IBM/360 changed that: other companies built clones.
- IBM unbundled software.
- Vendors came to consider their software to be valuable intellectual property.



Free software, cut

- By 1980, most software cost money.
- Access to source code, even with a license, became increasingly difficult.
- Universities were an exception.
- One of the reasons for the foundation of the Free Software Foundation.



Free software, second cut

- In the late 70s, computers became affordable for individuals.
- Affordable computers were still inadequate for UNIX.
- Much free and “shareware” software developed for CP/M, Apple and Microsoft.
- UNIX became practicable with the Intel 80386 processor in the late 80s.
- At the same time, Internet access became widespread.
- Free software projects started independently of FSF.



Free UNIX: the beginnings

- In 1979, Electrolabs brought out “OS-2”, a UNIX-lookalike on Z-80 under CP/M.
- In the early 1980s, Mark Williams ported Coherent to the IBM PC.
- Andy Tanenbaum’s Minix ran on the IBM PC, amongst others.
- None were successful: too expensive, too slow, too unreliable (no memory protection).



The breakthrough

- The Intel 80386 became the standard PC processor in the early 90s.
- A student in Finland announced his toy OS project: “just a hobby, won’t be big and professional like gnu”.
- In Berkeley, Bill Jolitz worked to free BSD from the last traces of AT&T code.
- The Internet enabled others to join in the fun.



Motivations

- “Freedom” was not an important motivation.
- Main motivation was to hack.
- Free Software Foundation was a source of code, not initially an inspiration.



Semblances of organization

- Multiple players require some kind of organization.
- In BSD, those who hacked most became part of the “core team” or “core group”
- No attempt at fairness. “This is our game, this is the way we play it”.
- Teams still pretty small: in 1995, FreeBSD had had a total of 55 contributors.
- Some contributors never knew that they had contributed.



Growth and stability

- By mid-1990s, project aims had changed.
- BSDs were stable enough for commercial use on the Internet.
- Commercial BSD/386 suffered as a result.
- Linux took a little longer to become stable, since written from scratch.
- The people didn't change (much).
- The number of people involved did change.
- The general public started to become aware of free UNIX.



Free operating systems take over

- In the late 90s, free operating systems became even better known.
- Companies were formed to “market” free operating systems, especially Linux.
- (Groggy shows his scars)
- Projects became even larger.
- The people stayed the same.
- The FreeBSD project now has 320 committers, some of them inactive.



Social changes in the FreeBSD project

- By 2000, the core team was no longer the most active group of committers.
- Some core team members completely inactive.
- Project direction not as well defined.
- Position of chief architect vacant.
- No defined way to be retired from the core team.



Changes in project focus

- In the early days, functionality was the prime issue.
- Later stability became more important.
- After achieving stability, performance became important.
- Issues became more global.
- For example, SMP project touches the entire kernel.
- More global issues require better project coordination.



User friendliness

- Project lead by developers.
- End users not directly represented.
- End user focus provided by Walnut Creek CDROM, but not strongly.



View of core team

- Power vacuum.
- Some developers took advantage of the situation for their own purposes.
- One of them was a member of the core team.
- Accusations of favoritism were made.
- Core team had adopted a policy of silence.



Developer's view of the problem

In November 1999, Nate Williams wrote:

- What is the purpose of core? What determines if someone should become a core member?
- Is there any way to lose your core member status, in the same manner that you can lose the ability to be considered a maintainer?
- Do you have to quit in order to not become a core member?



Developer's view of the problem (2)

- My *biggest* fear is that we will lose active developers simply because we just plod along hoping that everything will work out.
- Once upon a time, core members were folks were *really* excited and highly motivated to work on this thing, and would spend nights/weekends and all sorts of time on this.
- Core is now older, and our real lives get in the way now.



The crisis

- In early 2000, the problem reached crisis proportions.
- A prominent developer threatened to leave the project because the rogue core team member was trampling over his work.
- Developer 1 asked core to make a statement.
- Nothing happened.
- Jordan Hubbard summarized the current state with historic background.



jkh's view of the problem

- First there was a simple mail alias `freebsd-core` to allow “project insiders” to communicate with one another.
- Over time, core morphed into a combined working group and “mark of recognition” committee.
- Once public perception and general committer desire took over and basically turned it into a management and steering committee, despite the fact that core had never showed itself to be very effective at doing either job.



jkh's view of the problem (2)

- I think that core's time actually “passed” some months back, but we don't want to admit it.
- Core should be broken up like AT&T, so to speak, and become smaller operating groups who actually have clear and limited mandates.
- Much discussion ensued. Suggestions included complete anarchy and an elected core team.
- jkh suggested a number of modi for reforming core, called for a vote.



Reforming core

- The idea of core is fine, its membership simply needs a shake-up and some mechanism added for voting in new blood. (58 votes)
- The idea of core is fine, but some of members need to leave. (12 votes), most of which identified one specific member.
- Core needs to be broken up into an oversight/human resources group, leaving architectural decisions to developers. (9 votes)
- Don't change anything, core is fine the way it is. (7 votes)
- Disband core entirely and let committers create a new structure in its place. (7 votes)



Electing the new core team

More discussions:

- What should the new core team look like?
- Who is eligible to be a member of core?
- Who is eligible to vote?
- How should we vote?
- Jonathan Lemon, Warner Losh and Wes Peters formed a team to decide on answers.



Bylaws

- Active committers have made a commit to the tree in the last 12 months.
- Core consists of 9 elected active committers.
- Core elections are held every 2 years, first time September 2000.
- Core members and committers may be ejected by a 2/3 vote of core.
- If the size of core falls below 7, an early election is held.
- A petition of 1/3 of active committers can trigger an early election.



Holding elections

Elections will be run as follows:

- Core appoints and announces someone to run the election.
- 1 week to tally active committers wishing to run for core.
- 4 weeks for the actual vote
- 1 week to tally and post the results.
- Each active committer may vote once in support of up to nine nominees.



Holding elections

- New core team becomes effective 1 week after the results are posted.
- Voting ties decided by unambiguously elected new core members.
- These rules can be changed by a 2/3 majority of committers if at least 50% of active committers cast their vote.
- These “bylaws” passed by 117 yes votes to 5 no votes, thus also disproving the concern that committers wouldn’t be interested enough to vote for the core team.



The results

The election completed in time for the second BSDCon in Monterey. The new core consisted of:

- Satoshi Asami, member of the old core team. Guardian of the Ports Collection. Japanese.
- David Greenman, one of the founders of the FreeBSD project, and member of the old core team. Kernel hacker and former principal architect of the FreeBSD project. American.
- Jordan Hubbard, one of the founders of the FreeBSD project, and member of the old core team. Release engineer and former president of the FreeBSD project. American.



The results (2)

- Greg Lehey, newly elected. Kernel hacker, author of the Vinum Volume manager. Australian (Adelaide).
- Warner Losh, newly elected. Network hacker. American.
- Doug Rabson, member of the old core team. Kernel hacker, responsible for the port of FreeBSD to the Alpha platform. British.



The results (3)

- Mike Smith, newly elected. Low-level kernel hacker. Australian (Adelaide).
- Robert Watson, newly elected. Network hacker, FreeBSD security officer. British.
- Peter Wemm, member of the old core team. Universal Kernel Hacker. Australian (Perth).



First meeting

- First and only meeting ever of the entire core team in Monterey on 14 October 2000.
- Attempted to decide the charter of the core team after the event.
- Few decisions.
- The FreeBSD core team does not decide the architectural direction of the project.
- There will be no officers on the core team.
All members are equal.



What core does

- The FreeBSD project is a volunteer organization, so the core team does not have a mandate to tell anybody to do anything.
- That's conceding a lot. So what was left?
- The core team awards “commit bits”.
- In case of extreme misbehaviour, the core team can expel a committer from the project.
- In case of dispute between two committers, the core team mediates.
- (later) The core team produces a monthly report.



Unanswered questions

- Still no architectural direction.
- Intention was to form consensus on the mailing lists.
- If no consensus could be formed, core would mediate.
- Attractiveness to end users. The majority of the members of the core team, being developers themselves, were not very interested in this aspect.
- Rogue developers. No agreement.
- The core team did not have a “big stick”.
- About the only thing that it could do would be to expel a member from the project.



Unanswered questions (2)

- Project morale, including behaviour of developers towards each other.
- Again, no good solution for this problem
- Theoretically expulsion from the project would have been a solution.



Acceptance of core.2

Parts of it were excellent.

- Core members *still* unresponsive.
- No individual responsibilities: who should do the work?
- Core reports very slow.
- Core appointed a secretary (Wilko Bulte) to handle reports, gradually things got better.



Rogue developers

- In February 2002, a developer announced his intention to commit some significant changes to the SMP code.
- At the time, the most active SMP developer, John Baldwin, was offline.
- Others involved pointed out that these changes were in conflict with changes that John was currently testing and asked the developer to hold off.
- The developer committed the changes anyway.



Handling the crisis

- This issue became a test of core's authority.
- For the first time, core.2 decided to revoke the developer's commit privileges if he did not back out the commits.
- He did so in the nick of time and asked core to resolve the issue.
- Resolution was hard, looked more like tactics rather than strategy.
- After a month of discussion, core appointed John Baldwin to the position of technical lead for the SMP project, with the power to approve or reject changes.



Behaviour

Core used the experience to formulate rules on developer behaviour:

1. Committing during code freezes results in a suspension of commit bits for two days.
2. Committing to the security branch without approval results in a suspension of commit privileges for 2 days.
3. Commit wars will result in both parties having their commit bits suspended for 5 days.
4. Impolite or inappropriate behaviour results in suspension of commit bits for 5 days.



Behaviour (2)

5. Any single member of core or appropriate other teams can implement the suspension without the need for a formal vote.
6. Core reserves the right to impose harsher penalties for repeat offenders, including longer suspension terms and the permanent removal of commit privileges. These penalties are subject to a 2/3 majority vote in core.
7. In each case, the suspension will be published on the developers mailing list.
8. No provision for milder penalties.



The big stick

- In June 2002, core received another formal complaint about the same committer who had caused so much grief in February.
- He had committed code in an area on which another developer was working, without discussing the matter.
- The other developer was annoyed to the point that he relinquished the maintainership of this part of the tree.
- Core decided that this conflicted with rule 4 (inappropriate behaviour).



The big stick (2)

- Extenuating circumstances, but the rules were rigid.
- Developer was suspended for five days.
- Public reaction was unfavourable.
- Claims of political motivation: core elections were under way, developer was a candidate.
- Reprieve after two days.



The big stick (3)

- A few weeks later, two highly respected developers engaged in a commit war.
- Clear violation of rule 3.
- As usual, extenuating circumstances.
- Core decided on a 24 hour suspension.
- Less protest this time.



The collapse of core.2

- Round May 2001, Satoshi Asami became sick and disappeared from the scene for some time.
- After his return, he did not participate in core discussions,
- After several months, we finally decided that he was *de facto* no longer a member of the core team.
- According to the “bylaws”, we carried on with only eight members.



The collapse of core.2 (2)

- After the SMP commit war in February and March 2002, core members were feeling tired.
- On 29 April 2002, Jordan Hubbard dropped a bombshell: he resigned from core.
- Reasons: being on core is now painful.
- No longer “fun”.
- Slashdot picked on it with glee: “Death of FreeBSD project, film at 11”.



The collapse of core.2 (3)

- On 4 May 2002, Mike Smith resigned.
- In his message, he wrote: “FreeBSD used to be fun.”
- Found it too bureaucratic.
- After that, we only had six members.
- The “bylaws” required an early election.



Electing core.3

- Should we change the bylaws?
- Decided to stick with the old bylaws.
- Record number of candidates, including four members of core.4.
- More politicking, including nomination of running mates.
- Results announced immediately after polling closed.



core.3

The following candidates were elected:

- John Baldwin, newly elected. FreeBSD SMP technical lead. American.
- Jun Kuriyama, newly elected. Japanese.
- Greg Lehey, member of core.2. Kernel hacker, author of the Vinum Volume manager. Australian (Adelaide).
- Warner Losh, member of core.2. Network hacker. American.



core.3

- Mark Murray, newly elected. Security hacker. Zimbabwean.
- Wes Peters, newly elected. Network hacker. American.
- Murray Stokely, newly elected. FreeBSD Release Engineer. American.
- Robert Watson, member of core.2. Network hacker. British.
- Peter Wemm, only member of the original core team left. Universal Kernel Hacker. Australian (Perth).



What has changed?

- Much better composition of core.
- SMP technical lead.
- Release engineer.
- Still only developers, no end users.



Other bodies

- Security officer was around for a long time.
- Now a team.
- In core.1, Satoshi Asami was “Mr. Ports”.
- Since 2001, replaced by *portsmgr* team.
- In the time of core.1, FreeBSD machines administered by two or three people who happened to have physical access.
- Now administered by admin team.
- New bug fixing team (*bugmeister*).
- Planning an architectural review board.
- Expect more to come in the future.



The future

- As the project grows, expect more organization.
- Many project members do FreeBSD work for their day job.
- FreeBSD has grown up.
- No, it's not fun any more.

