

---

# Ship It!

CSE 403, Winter 2003  
Software Engineering

<http://www.cs.washington.edu/education/courses/403/03wi/>

---

# Readings and References

- References
  - » Release Engineering Information, FreeBSD
    - <http://www.freebsd.org/releng/index.html>
    - <http://www.freebsd.org/releases/4.8R/schedule.html>
- Acknowledgment
  - » Most of the slides in this lecture are derived from a similar lecture by G. Kimura in an earlier instance of CSE 403

---

# Rules about breaking the build

- Rule 1: Never ever break the build
- Rule 2: If you do break the build, fix it immediately and beg for forgiveness
- Rule 3: Don't change other people's files without their permission
- Rule 4: The closer one gets to the finish line the more important rule one becomes
- Rule 5: Don't make superfluous last minute changes
- Rule 6: Whenever in doubt, see rule 1

---

# Consequences of breaking the build

- Increases development cost
- Adds risk to the project
- Undermines group synergy
- Personally you may
  - » Have your name published widely within the company
  - » Receive 2:00 AM phone calls
  - » Wear the goat horns
  - » Pay \$ into the build lab's refreshment / antacid fund

## Some ways to avoid breaking the build

- Do a clean build and run all the tests
- Buddy builds
- Have your changes reviewed
- Develop using separate development trees
- Do what's right by you and your teammates

5-March-2003

cse403-18-Shiplt © 2003 University of Washington

5

## Why is it so important not to break?

- Lots of other people are making decisions and schedule commitments
  - » The software is growing up and moving away from the development team
- Many of the dates were almost random when initially picked
  - » But changing them now is very expensive, very inconvenient, and very embarrassing

5-March-2003

cse403-18-Shiplt © 2003 University of Washington

6

Schedule				FreeBSD 4.8 Release Process			
Action	Expected	Actual	Description				
Reminder announcement	16 Jan 2003	16 Jan 2003	Release Engineers send announcement email to developers@FreeBSD.org with a rough schedule for the FreeBSD 4.8 release.				
Another Reminder announcement	7 Feb 2003	10 Feb 2003	1 week reminder.				
Another Reminder announcement	14 Feb 2003	14 Feb 2003	Final reminder, with exact time that freeze begin.				
4.8-PRERELEASE Testing Guide published	15 Feb 2003	2 Mar 2003	A <a href="#">testing guide</a> should be published with information about recent changes and areas of the system that should be thoroughly tested during the pre-release/RC period.				
4.8-PRERELEASE	15 Feb 2003	15 Feb 2003	newvers.sh, and release.ent updated.				
FTP site updated	15 Feb 2003	15 Feb 2003	unmessaary pkg_add -r works for 4.8-PRERELEASE machines.				
Code freeze begin	15 Feb 2003	15 Feb 2003	After this date, all commits to the PRELNC_4 branch must be approved by <a href="mailto:rs@FreeBSD.org">rs@FreeBSD.org</a> . Certain highly active documentation committers are exempt from this rule for routine man page / release note updates. Head-up email should be sent to the developers@, stable@, and qa@ lists.				
4.8-RC	1 Mar 2003	2 Mar 2003	newvers.sh and release.ent updated.				
Announce the Ports Freeze	1 Mar 2003	1 Mar 2003	Someone from portmgr@ should email freebsd-ports@ and BCC: developers@ to set a date for the week long ports freeze and tagging of the ports tree.				
First release candidate	1 Mar 2003	--	The first release candidate for the x86 and Alpha architecture is released. ISO images should be uploaded to ftp-master.freebsd.org and release.freebsd.org. A network install directory should be uploaded to ftp-master.freebsd.org. The packages directory should be a relative symlink, as described in the releng article.				
Head up to -stable	2 Feb 2003	--	A message should be sent to qa@FreeBSD.org and stable@FreeBSD.org after the snapshot is uploaded.				
Second release candidate	--	--	Note: the release date of this candidate depends on the user experience with RC1.				
Third release candidate	--	--					
Ports tree frozen	5 Mar 2003	--	Only approved commits will be permitted to the ports/ tree during the freeze.				
Announce doc/ tree shuh	6 Mar 2003	--	Notification of the impending doc/ tree shuh should be sent to doc@.				
doc/ tree shuh	10 Mar 2003	--	Non-essential commits to the en_US.ISO8859-1/ subtree should be delayed from this point until after the doc/ tree tagging, to give translation teams time to synchronize their work.				
Ports tree tagged	13 Mar 2003	--	RELEASE_4_8_0 tag for ports/.				
Ports tree unfrozen	13 Mar 2003	--	After the ports tree is tagged, the ports/ tree will be re-opened for commits, but commits made after tagging will not go in 4.8-RELEASE.				
Final package build starts	14 Mar 2003	--	The ports cluster and <a href="#">bento</a> build final packages.				
doc/ tree tagged.	14 Mar 2003	--	Version number bumps for doc/ subtree. RELEASE_4_8_0 tag for doc/ doc/shuh ends at this time.				
RELEASE_4_8 branch	13 Mar 2003	--	The release branch is created.				
Note to freebsd-stable@	13 Mar 2003	--	A note should be sent to the freebsd-stable@ list to let over-ambitious users know that the tags have been created but the release still isn't ready. Tags may be sld before the announcement goes out. Point users to freebsd-qa@ for details.				
Version numbers bumped.	13 Mar 2003	--	The files listed <a href="#">here</a> are updated to reflect the fact that this is FreeBSD 4.8.				
Update man.cgi on the website	15 Mar 2003	--	Make sure the 4.8 manual pages are being displayed by default for the man- <a href="#">web</a> gateway. Also make sure these man pages are pointed to by docs.sgml.				

## Is it ready to ship?

- Who decides when it is ready to ship?
  - » The developer?
  - » The customer?
  - » The managers?
  - » The executive?
- What factors are involved in the decision?
  - » Is the software ready?
  - » Do we need the revenue?
  - » Is there an external factor that is setting the completion date?

5-March-2003

cse403-18-Shiplt © 2003 University of Washington

8

## It's not just the software ...

---

- Development and testing plays a big role in getting the product ready, but there is also...
- Documentation
- Manufacturing
- Sales and marketing
- Support and more

## Does it meet the requirements?

---

- Is the quality sufficient for the customer
  - » PC mentality
  - » Mission critical systems
  - » Financial, medical, imbedded systems in planes, factories, safety critical equipment

## Is the software ready and tested?

---

- What are the remaining bugs like?
  - » Are there showstopper bugs (i.e., bugs that you cannot ship knowing they are in the product)
  - » What is the cost of keeping a known bug in the product
    - Is it an obscure bug
    - Are there easy workarounds for the bug
    - Is the product “bug free”

## Is the software ready and tested?

---

- In house testing
  - » Is the daily pass rate acceptable? The definition of acceptable open to discussion.
  - » Has it successfully run through its validation suite?
  - » Is everyone happy with the product
- Off-site testing through release candidates
  - » Are the beta customers satisfied
- Is the product stable enough for its intended market
- What do we do when we stop finding bugs

## Documentation done and accurate?

---

- Is manufacturing ready for the release
  - » There is physical manufacturing and shipping of the release media and documentation
  - » Filling the distribution channel
  - » Even with internet distribution there needs to be well planned capacity
- Is product support ready
  - » Whether the support is “on-site”, “9 to 5”, “24 / 7”, “phone”, or “mail only”, the support staff needs to be trained and have adequate communication capacity.
- Is sales and marketing ready

## Customer ready?

---

- A lot of this depends on the type of customer
  - » The shrink warp market
  - » ISV – More technically savvy than the regular shrink wrapped market, however also less tolerant of stupid errors
  - » IHV – They have a different set of issues from ISV including automated setup and installation, and customization support
  - » Contract work – Various levels from delivering an entire turnkey system to only a small component that the buyer will integrate.
  - » Internal customer group

## Ship mode

---

- In the Windows NT group, this was called “showstopper mode” or the “death march” a rather insensitive macabre term
- All source changes must be reviewed and approved. It must be for a specific showstopper bug. Bugs are recorded in a bug database called “raid”. Sometimes kept an open showstopper bug active just to check-in more “fixes”
- Daily builds and stress runs (yes, seven days a week including most holidays)

## A typical day near the end

---

- 5 AM results are starting to be gathered for the previous night stress run
- 7 AM release of the previous nights stress results. Developers then have until noon or so to debug all the crashed machines. Sometimes you need to keep the machine a lot longer.
- 8 AM meeting of the development team managers looking at the nightly stress results and new bugs review (they decide which bugs need to be fixed, and when to ship the product). Near the very end this becomes a twice a day meeting
- 10 AM to 3 PM the build lab is willing to accept any bug fixes for approved showstopper bugs
- 5 PM dinner is served
- 6 PM the next build is released and everyone installs the new system and starts up stress, and those with showstopper bugs continue to work.

## Finally

---

- When it is finally decided to ship the product then the bits go into escrow as the golden media is produced and manufacturing starts ramping up. Testing continues and if necessary the bits can be recalled from escrow and the release done over again.
- Work continues on the subsequent release for the various server editions and international language versions.

## Ancillary issues

---

- Media hype
  - » Setting expectations
  - » Beta previews
  - » Getting beta customer testimonials might be important
- Competitive pressure
  - » Market share before quality
  - » First one defines the market and grabs market share even with junk
  - » The followers often play catch-up with mixed success (unless you are a monopoly)

## More issues

---

- Timing the release
- When do we get paid and are we ready for the IPO?
- Major release vs. minor release
  - » Big delta or small delta
  - » Customer perception based on version number
  - » Some IHV contracts are based on version number
- Where to have the ship party

## Intellectual property rights

---

- Copyrights and patents are ways to protect your ideas using the legal system
  - » Copyright term (in the US) was life+50 or 75 years
    - now life+70 or 95 years (extensions at Mickey Mouse-2)
  - » Design patent term (in the US) is 14 years
- The patent application itself is somewhat time consuming but can be worth quite a bit for its owner
  - » This is growing to be a significant issue for developers
  - » The patent office will apparently patent almost anything brought before it