Ship It!

CSE 403, Spring 2003
Software Engineering

http://www.cs.washington.edu/education/courses/403/03sp/

Readings and References

- ## References

  - » Release Engineering Information, FreeBSD

    - http://www.freebsd.org/releng/index.html

    - http://www.freebsd.org/releases/5.1R/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

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

Schedule

Action	Expected	Actual	Description
-CURRENT code freeze	5 May 2003	5 May 2003	The src/ code freeze for 5.1. Commits to HEAD require re@FreeBSD.org approval.
5.1-BETA	5 May 2003	--	5.1-BETA release of x86, alpha, sparc64, and ia64.
RELENG_5_1 branched	19 May 2003	--	Branch of src/ from HEAD for the release. Note: no branch for RELENG_5 will happen at this time.
Turn off debugging for RELENG_5_1	19 May 2003	--	Turn off WITNESS, INVARIANTS, and malloc debugging options similar to what was done for 5.0.
First release candidate	19 May 2003	--	x86, alpha, sparc64, and ia64 images released and uploaded to ftp-master.FreeBSD.org.
src/ unfrozen	19 May 2003	--	Unfreeze HEAD src. Continue to coordinate significant check-ins with re@FreeBSD.org in order to work towards 5-STABLE.
Second release candidate	27 May 2003	--	--
Ports tree tagged	27 May 2003	--	Tentative date of RELEASE_5_1_0 tag for ports.
Version numbers bumped	30 May 2003	--	The files listed here are updated to reflect FreeBSD 5.1.
src/ tree tagged	30 May 2003	--	Tag the RELENG_5_1 branch with RELENG_5_1_0_RELEASE.
doc/ tree tagged	30 May 2003	--	Tag the doc/ branch with RELEASE_5_1_0.
Final builds	30 May 2003	--	Start x86, alpha, sparc64, ia64, and pc98 builds.
Warn hubs@FreeBSD.org	30 May 2003	--	Heads up email to hubs@FreeBSD.org to give admins time to prepare for the load spike to come. The site administrators have frequently requested advance notice for new ISOs.
Upload to ftp-master	1 Jun 2003	--	Release and packages uploaded to ftp-master.FreeBSD.org.
FreeBSD 5.1 Released	2 Jun 2003	--	FreeBSD 5.1 is announced to the mailing lists.
FreeBSD 5.1 Press Release	2 Jun 2003	--	A formal press release statement is in the works and should be released at this time to the www.FreeBSD.org website and various tech publications.

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?

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 macabre term, but also rather accurate

- 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