

Great Expectations (of a Build Engineer)

*Jeff A. Bowles
Piccolo Engineering*

Who is a build/release engineer?

- Ø You?
- Ø Me?
- Ø The guy down the hall who didn't want the position?
- Ø Joe, Jan, Sue, Marge, and Jack – each is a developer, doing a build and then letting the next developer do it the following time?

What is a handoff?

- Ø Strictly speaking, it's the act of making your work available to someone else.
- Ø There's two sides to any handoff.
 - *Imagine the four-year-old who **wants a tricycle** for Christmas, but **gets a box with parts** to build one, instead.*
 - *Or the test engineer who expects the code to be **ready on the 19th** but doesn't know that the developers think that means **finish the code at 5 PM on the 19th**.*

What is a handoff?

- Ø It's an Engineering "thing".
 - The *build engineer* can support Engineering standards and process in many ways.
 - The key is "just enough process."
 - Too much will irritate developers;
 - Too little undercuts downstream groups.
- "Process" and "make-work" are not interchangeable terms.*

Examples of build processes

Ø Automate build scripts.

- Build from a label, date, or submission/change number.

Examples of build processes

Ø Automate build scripts.

- Build from a label or change number.
- Start with an empty workspace. Empty the workspace before building, so that full builds are truly reproducible.

Examples of build processes

Ø Automate build scripts.

- Build from a label or change number.
- Start with an empty workspace.
- Guard against manual steps.
Consider asking developers for alternate ways to generate resource files and the like, to avoid running GUI tools for automated builds.

Examples of build processes

Ø Automate build scripts.

- Build from a label or change number.
- Start with an empty workspace.
- Guard against manual steps.
- Use the same [basic] scripts for overnight builds, release builds, and initial setup of a patch build. Share common code.

Examples of build processes

Ø Automate build scripts.

- Build from a label or change number.
- Start with an empty workspace.
- Guard against manual steps.
- Share common code.
- Ask about handoff contents. Who needs what items, in what form, and when?

Examples of build processes

Ø Automate build scripts.

- Build from a label or change number.
- Start with an empty workspace.
- Guard against manual steps.
- Share common code.
- Ask about handoff contents.
- Version the build/release scripts.
Include build/release scripts in your codeline / branches, so that the 1.0 script is saved with the 1.0 source and is separate from the 1.1 script.

Examples of build processes

Ø Automate build scripts.

- Build from a label or change number.
- Start with an empty workspace.
- Guard against manual steps.
- Share common code.
- Ask about handoff contents.
- Version the build/release scripts.

Examples of build processes

Ø Automate handoff report generation.

- Run the commands to generate bug lists, delta-between-labels, etc. from the build/staging harness.
- If you're ever asked for a report, incorporate its generation into the build/staging harness for next time.

Different groups need different information

Ø Testing groups

- need to know change information and results of smoke-test;

Ø Support groups

- need to know what's flakey and what's been reworked.

Ø Documentation groups

- need to see early versions for snapshots and to see behavior. (Real requirements docs are invaluable for writing documentation.)

Ask them what they need.

Give them more than you think they need.

Which is more useful?

Ø Changelists in this build:

- 10291
- 10293
- 10294
- 10295
- 10297

Ø Changelists in this build:

- 10291 – “fix GUI bug in menu.c”
- 10293 – “undo 10291 change, it had a typo.”
- 10294 – “fix GUI bug correctly, requires retesting all hot-key mechanisms in menu.”
- 10295 – “added new button to save/exit/quit dialog box”
- 10297 – “fixed typo in WinHelp file”

In general, you should run questions that save the Testing/QA group trips to the SCM database as possible. *Save them the trips to the well.* The build script can automate this since it's going there, anyhow.

How long to build a patch?

Ø A QA manager might want to regress every bit of a rebuilt product in the event of a patch. *“If it’s completely rebuilt, shouldn’t it be completely retested?”*

- Did you include build scripts/docs in branch?
- How much is recompiled for a patch? (Keep in mind that pulling from a new branch might entail a complete rebuild.)
- Is that reasonable?

Suggestion: build the final handoff as the first build from the ‘patch codeline’, to guarantee that compiles/tests at the moment you’re issuing the patch are manageable.

The unreasonable responses

- Ø “We can’t build this automatically.”
- Ø “It’s not the job of my group to do thus-and-such.”
- Ø “It’s in the bug database or Perforce database, they can look it up.”
- Ø “I can’t include that, since the document/format/etc isn’t finalized yet.”

The arsenal to carry...

- Ø Learn some scripting language well. (Unix shell, Python, Perl, Ruby. No compiled languages need apply.)
- Ø Script the build, script the creation of the release. Script everything.
- Ø Make your initial “build notes” in the form of a script that does the right things in the right order.
- Ø Get agreement on handoffs.
- Ø Notice others whose work you can encourage or directly use. Don't reinvent if you can avoid it.