The Cable Company Does Continuous WHAT?

COMCAST

cim COMCAST INTERACTIVE MEDIA

sh-3.2# whoami

Joe Campbell

- “Professional” Programmer for ~15 years
- Main job is to help developers “GO FAST”
- Find good ways to accomplish great things
- Pragmatism

Wally Eggert

- Programming for longer than I would care to admit

  (does the Timex Sinclair 1000 sound familiar?)
- Make software work…reliably and simply.
- The code I write today is often in production tomorrow.

sh-3.2# whoami

Photo by Peter Aaron/Esto

sh-3.2# whoami

http://www.resourcesforlife.com/docs/item1424

Disclaimer

What are we talking about

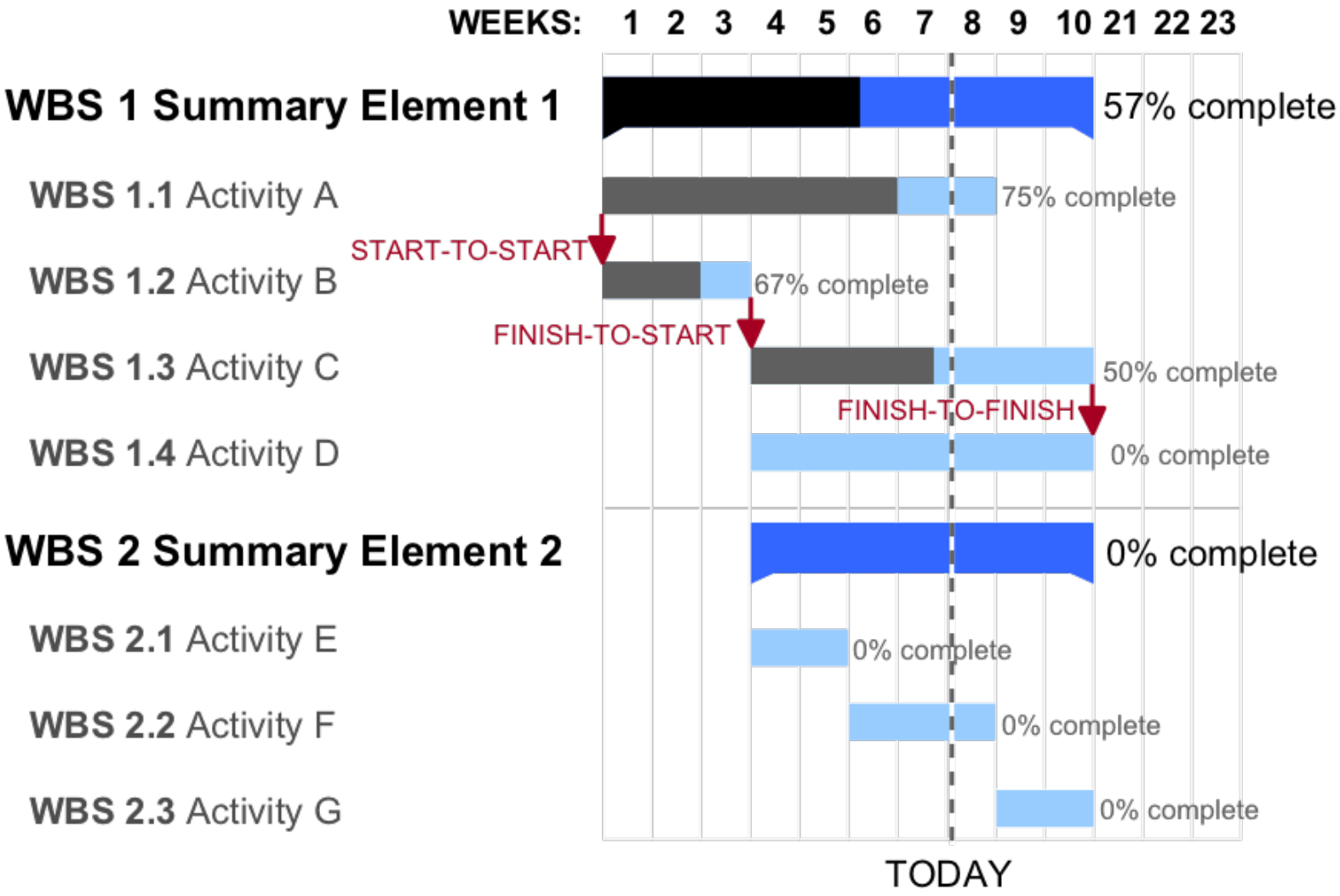
- Where did we start
  - What did we have to start from…
- How did we do it?
- What did we get out of it?
- Underpants Gnomes (Profit)

Wild West

Anything worth doing once…

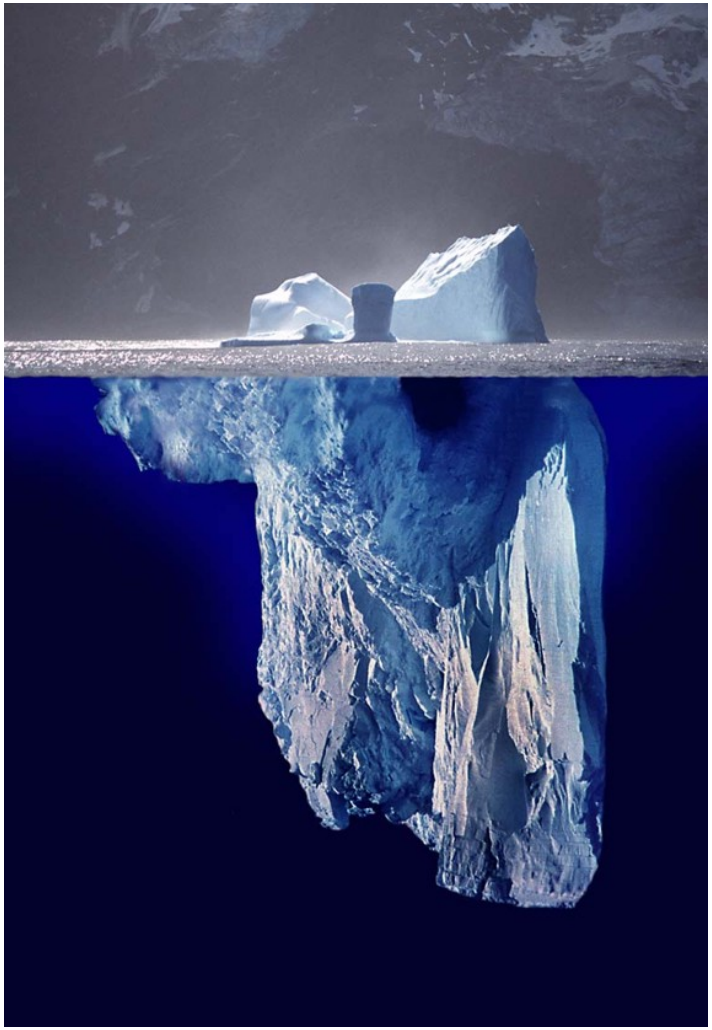
Ops folk – do it by hand

Deployment Schedules

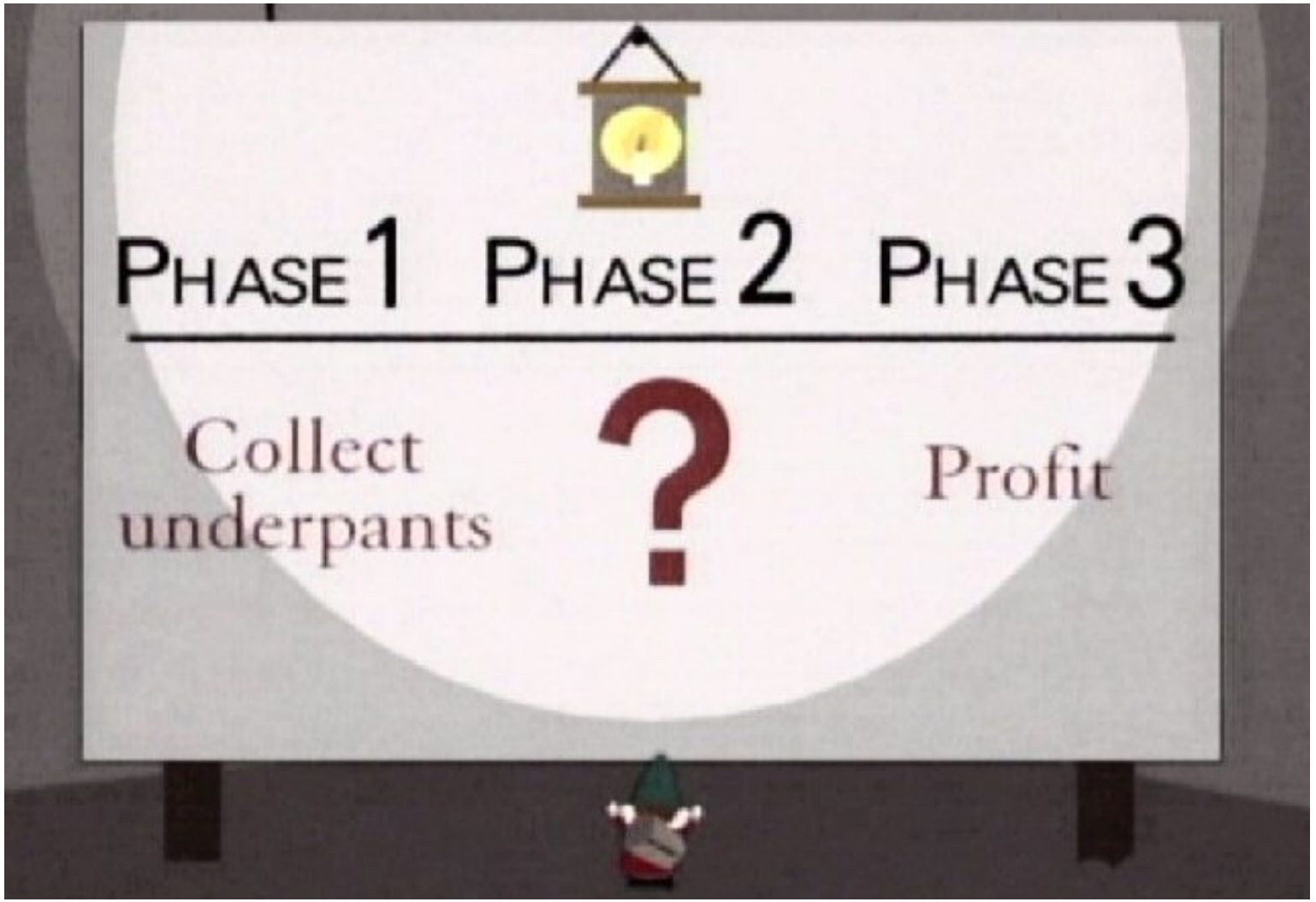


The tool box check list

Scratching the surface



Underpants Gnomes



http://www.alaskacommons.com/2012/10/14/paul-ryan-sean-parnell-and-the-curse-of-the-underpants-gnomes/

How do we start?

CC Image courtesy Marco Bellucci on Flickr
http://www.flickr.com/photos/marcobellucci/3534516458/

Let's pick a project!

By PeterPan23 [Public domain], via Wikimedia Commons
http://commons.wikimedia.org/wiki/File:Darts_in_a_dartboard.jpg

But which one?

CC Image courtesy of Mark Turnauckas on Flickr
http://www.flickr.com/photos/marktee/5590077419/

How we picked a project

- We picked a greenfield project.
- We put together a team that understood and believed in Continuous Delivery.
- We planned for extra up-front investment in the project.
- We let the team get their work done.

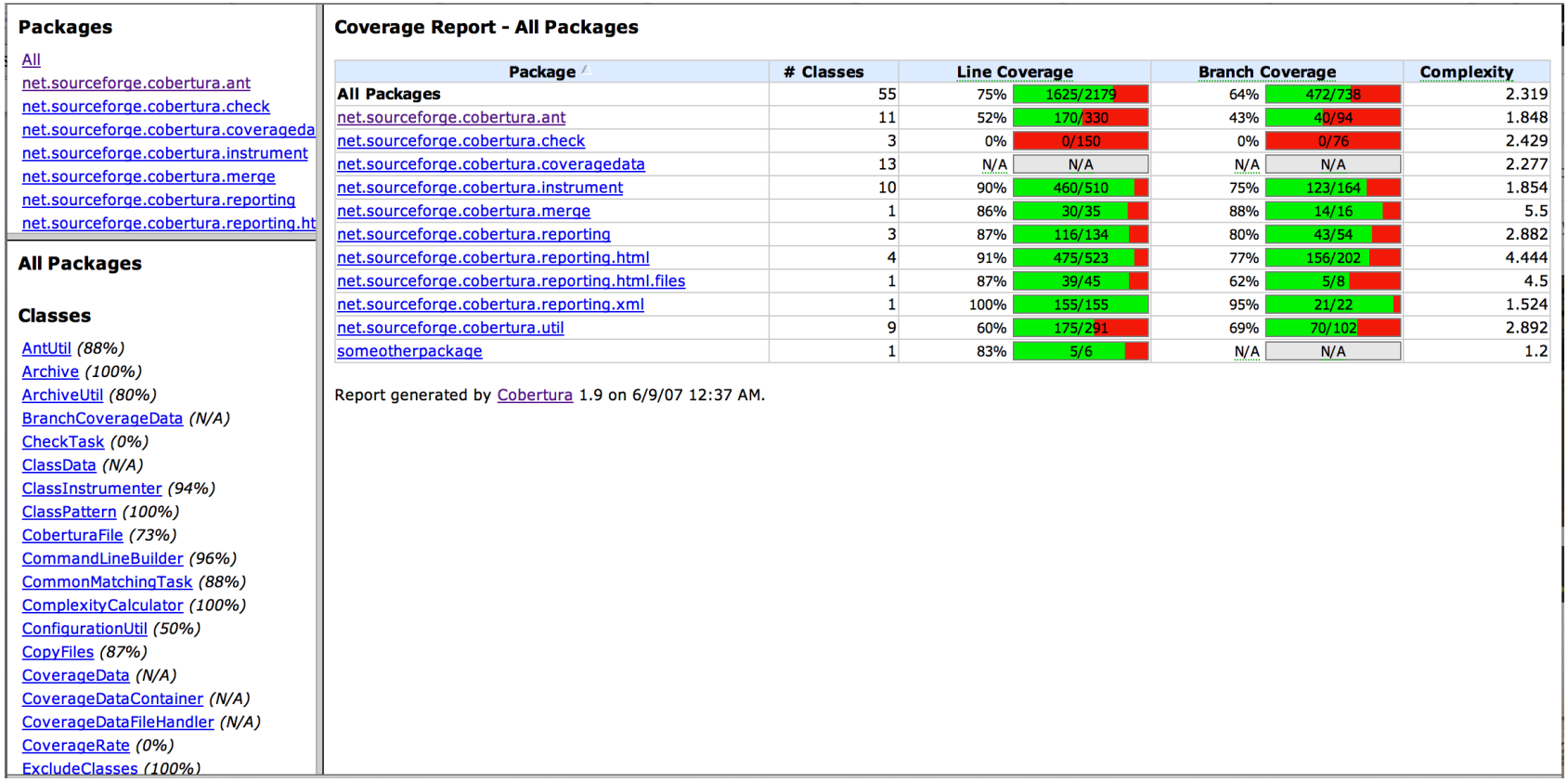
We automated CI and deployment

CC Image courtesy exfordy on Flickr
http://www.flickr.com/photos/exfordy/2867696831/

We defined the deployment environment

By Dream out loud (formerly Crashintome4196) (English Wikipedia) [GFDL (http://www.gnu.org/copyleft/fdl.html), CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0/) or CC-BY-SA-2.5-2.0-1.0 (http://creativecommons.org/licenses/by-sa/2.5-2.0-1.0)], via Wikimedia Commons

We measured unit test coverage



Packages

All
net.sourceforge.cobertura.ant
net.sourceforge.cobertura.check
net.sourceforge.cobertura.coverageda
net.sourceforge.cobertura.instrument
net.sourceforge.cobertura.merge
net.sourceforge.cobertura.reporting
net.sourceforge.cobertura.reporting.ht

All Packages

Classes

AntUtil *(88%)*
Archive *(100%)*
ArchiveUtil *(80%)*
BranchCoverageData *(N/A)*
CheckTask *(0%)*
ClassData *(N/A)*
ClassInstrumenter *(94%)*
ClassPattern *(100%)*
CoberturaFile *(73%)*
CommandLineBuilder *(96%)*
CommonMatchingTask *(88%)*
ComplexityCalculator *(100%)*
ConfigurationUtil *(50%)*
CopyFiles *(87%)*
CoverageData *(N/A)*
CoverageDataContainer *(N/A)*
CoverageDataFileHandler *(N/A)*
CoverageRate *(0%)*
ExcludeClasses *(100%)*

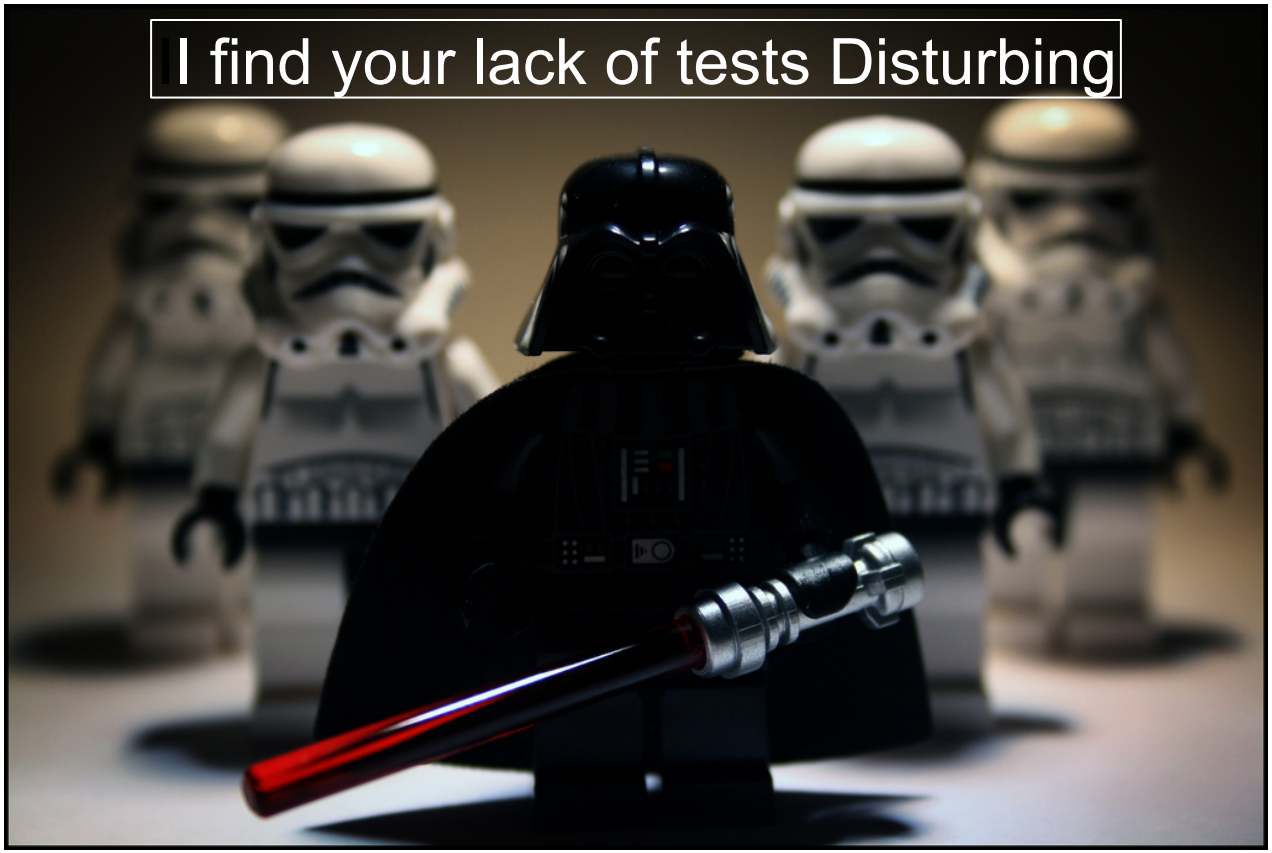
Coverage Report - All Packages

Package	# Classes	Line Coverage	Branch Coverage	Complexity
All Packages	55	75% 1625/2179	64% 472/738	2.319
net.sourceforge.cobertura.ant	11	52% 170/330	43% 40/94	1.848
net.sourceforge.cobertura.check	3	0% 0/150	0% 0/76	2.429
net.sourceforge.cobertura.coveragedata	13	N/A N/A	N/A N/A	2.277
net.sourceforge.cobertura.instrument	10	90% 460/510	75% 123/164	1.854
net.sourceforge.cobertura.merge	1	86% 30/35	88% 14/16	5.5
net.sourceforge.cobertura.reporting	3	87% 116/134	80% 43/54	2.882
net.sourceforge.cobertura.reporting.html	4	91% 475/523	77% 156/202	4.444
net.sourceforge.cobertura.reporting.html.files	1	87% 39/45	62% 5/8	4.5
net.sourceforge.cobertura.reporting.xml	1	100% 155/155	95% 21/22	1.524
net.sourceforge.cobertura.util	9	60% 175/291	69% 70/102	2.892
someotherpackage	1	83% 5/6	N/A N/A	1.2

Report generated by Cobertura 1.9 on 6/9/07 12:37 AM.

Screen Capture by Walter Eggert

We wrote lots and lots of tests



CC Image courtesy CJ Isherwood on Flickr
http://www.flickr.com/photos/isherwoodchris/7102871211/

Testing on many levels

CC Image University of Aberdeen, new Library by Alan Findlay
Copyright Alan Findlay and licensed for reuse
http://www.geograph.org.uk/photo/2628795

We deployed something to all environments

By Dbxsoul (Own work) [CC-BY-3.0 (http://creativecommons.org/licenses/by/3.0)], via Wikimedia Commons

We iterated and matured

CC Image by psd on Flickr
http://www.flickr.com/photos/psd/100757720/

CC Image by Erik Daniel Drost on Flickr
http://www.flickr.com/photos/edrost88/6158540686/

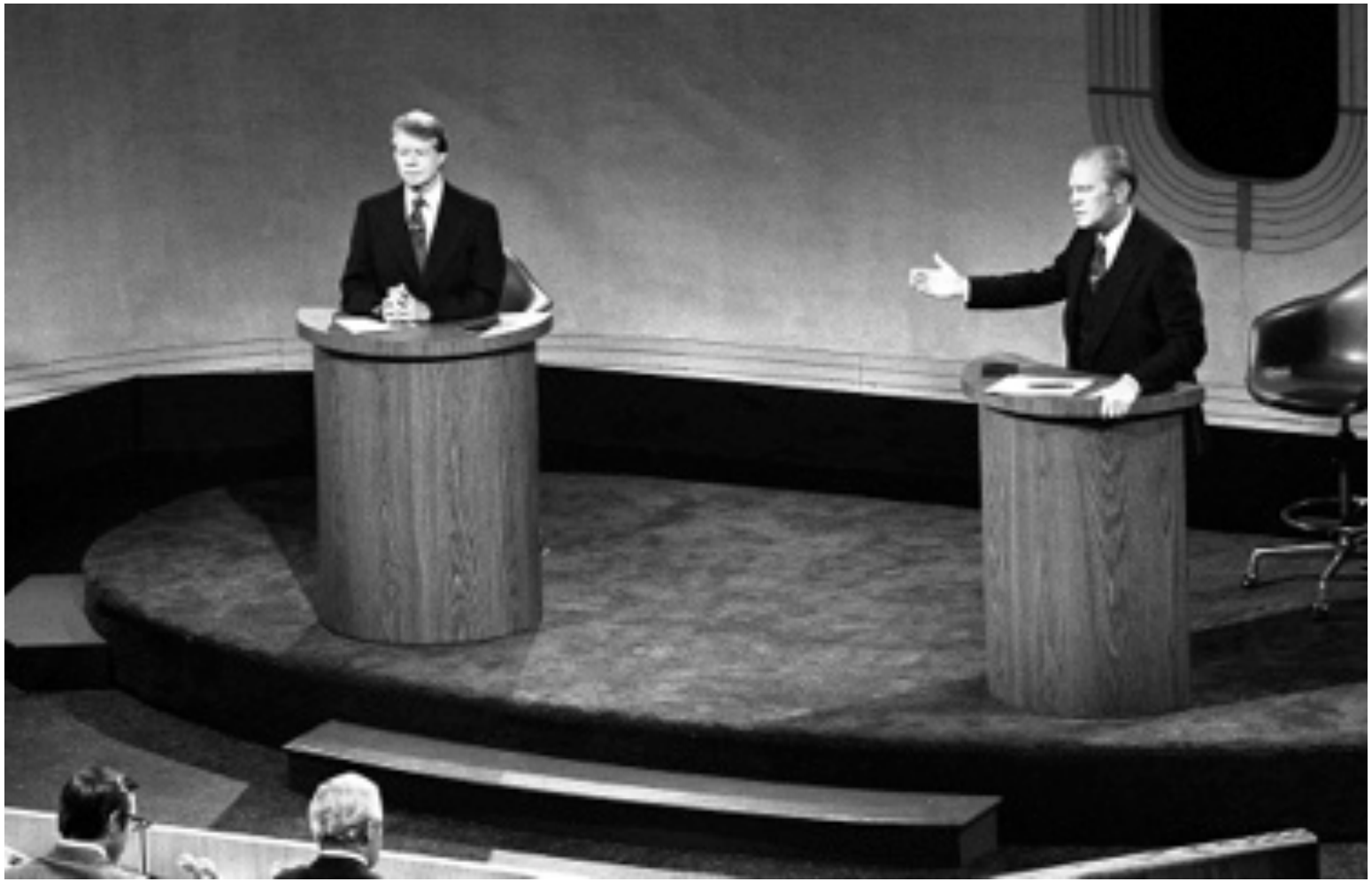
CC Image by Joaquin Martinez Rosado on Flickr
http://www.flickr.com/photos/25876167@N08/3694927599/

We had a pretty good pipeline going

By U.S. Geological Survey employee [Public domain], via Wikimedia Commons
http://commons.wikimedia.org/wiki/File%3A800px-Trans_Alaska_Pipeline_Denali_fault_shift.JPG

But it was time to take out the garbage

Pre or Post Commit Reviews?

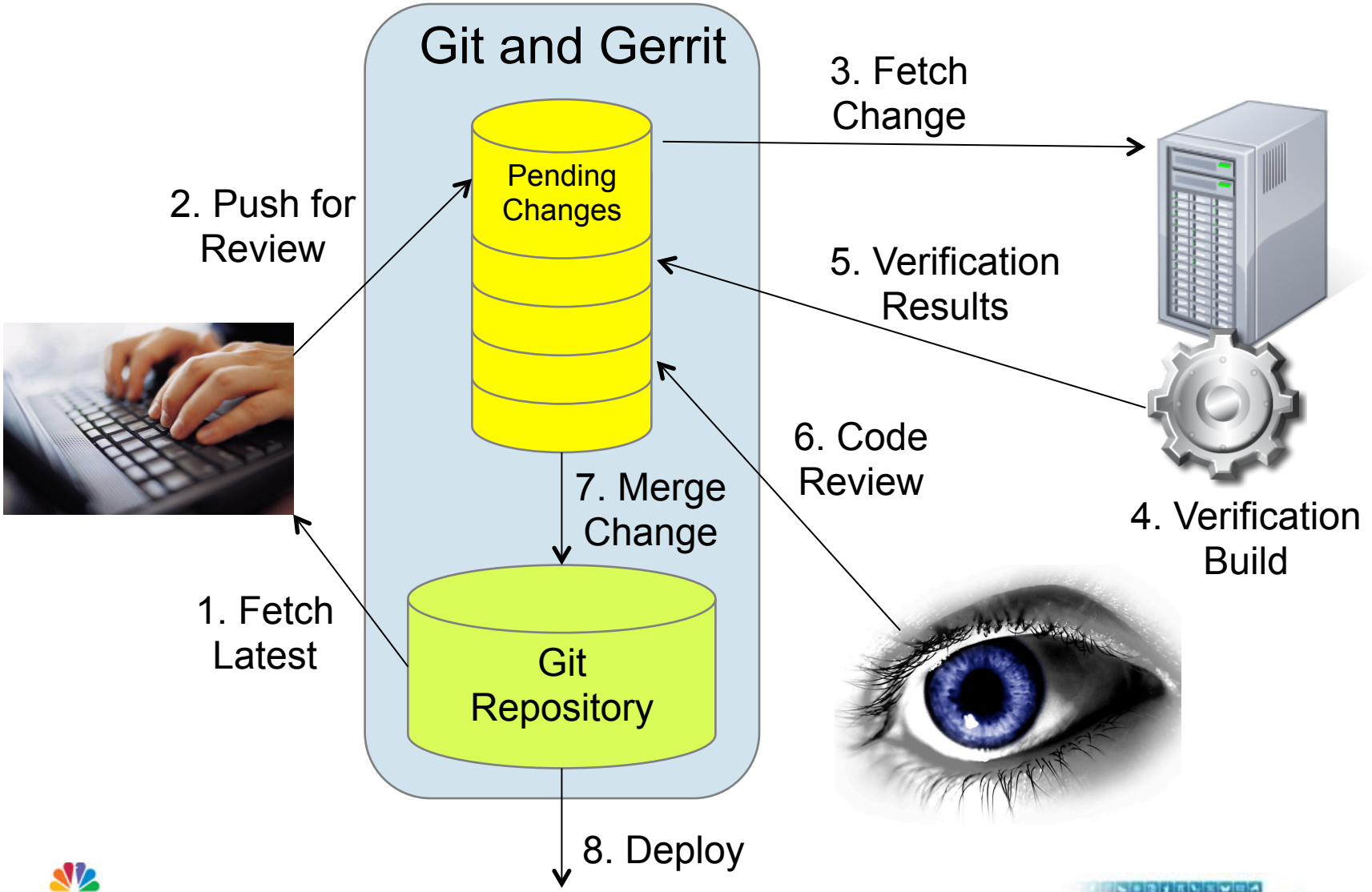


David Hume Kennerly [Public domain], via Wikimedia Commons
http://commons.wikimedia.org/wiki/File%3ACarter_and_Ford_in_a_debate%2C_September_23%2C_1976.jpg

Pre or Post Commit Reviews?

See page for author [Public domain], via Wikimedia Commons
http://commons.wikimedia.org/wiki/File%3AREH_Vinson_Beachcombers'_Argument.PNG

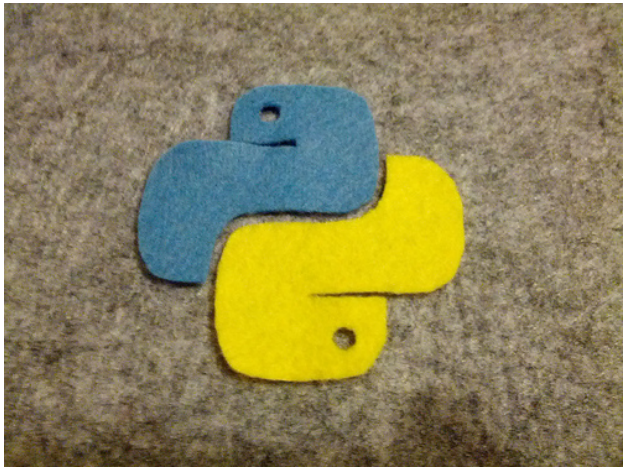
Git and Gerrit



We switched to an industrial-strength build-and-deploy framework

By Bitjungle (Own work) [CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons
http://upload.wikimedia.org/wikipedia/commons/5/5c/INdustrial_piping.jpg

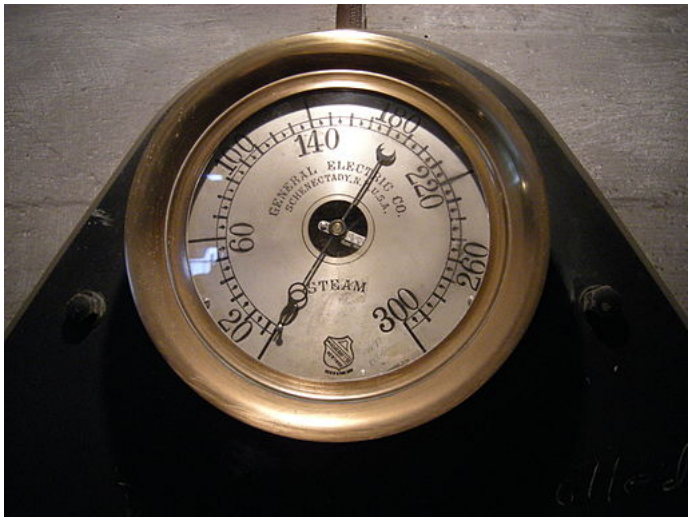
We matured our functional test framework



CC Image courtesy osakanyan on Flickr
http://www.flickr.com/photos/osakanyan/8103810833/

+ Function Testing = PyFuncTest

We modernized our monitoring infrastructure



Joe Mabel [GFDL (http://www.gnu.org/copyleft/fdl.html) or CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0/)], via Wikimedia Commons http://commons.wikimedia.org/wiki/File%3AGeorgetown_PowerPlant_Museum_gauges_04.jpg

CC Photo courtesy dawnhops on Flickr http://www.flickr.com/photos/seenoevil/343753843/

COMCAST

cim COMCAST INTERACTIVE MEDIA

We found that we had inconsistent deployment environments

CC Image courtesy certified su on Flickr
http://www.flickr.com/photos/certified_su/229016531/

We matured our automated deployment framework (libdeploy)

http://pixabay.com/en/burlap-texture-background-fabric-19189/

Evangelism

Brown Bags

Shop it around

Get Generative

What did we get…

Deploys went from:

- ~30Min + → ~5 minutes (in some cases)

Of Builds Executed

- ~20 per week → ~20 per day (project dep.)

Almost no humans

- You still have to push a button

Thousands of test cases run

- They run very very fast

Profit

What did we learn

You have to start SOMEWHERE…

- Don’t let the size of the thing paralyze you

TEST TEST TEST

- Highly automated
- Everyone can run

People are involved

- Personalities
- Agendas

What did we learn

Combine similar things

- Having two of everything is a problem
- Caution though: There is likely an upper limit

Having the numbers matters

- Being able to tell a success story

Joe Campbell - @joercampbell

Walter Eggert - @wallyeggert