

How do you convert your office to build automation? [closed]

Asked 16 years, 3 months ago Modified 11 years, 10 months ago

Viewed 405 times



2



As it currently stands, this question is not a good fit for our Q&A format. We expect answers to be supported by facts, references, or expertise, but this question will likely solicit debate, arguments, polling, or extended discussion. If you feel that this question can be improved and possibly reopened, [visit the help center](#) for guidance.

Closed 11 years ago.

The title should say it all, then I can solidify 2 more ticks on the Joel test.

I've implemented build automation using a makefile and a python script already and I understand the basics and the options.

But how can I, the new guy who reads the blogs, convince my cohort of its inherent efficacy?

version-control

build-automation

visual-sourcesafe

Share

edited Feb 10, 2013 at 4:20

Improve this question

Follow

asked Sep 17, 2008 at 22:09



Peter Turner

11.4k ● 10 ● 70 ● 113

For the record, we use VSS 6.0 and Delphi 7. I modified a borland package group file to create my makefile, which I run nightly (using pycron) by running a python script that updates my source from VSS and copies my files out to a share we all update our binaries from. All windows, no dot net...

– Peter Turner Sep 17, 2008 at 22:12

9 Answers

Sorted by:

Highest score (default)



Ask for forgiveness, instead of permission.

6

Get it working in private (which it looks like you have) and then demonstrate its advantages.



One thing that always gets people is using CruiseControl's Tray utility - people love it when they can see, through their system tray, that the build succeeded. (this is assuming you're in a Windows environment, that CruiseControl will work with your existing systems, etc.)



NOTE: If asking for forgiveness instead of permission will result in instant termination, you might not want to do the above. You might also want to look for work somewhere else. Your mileage may vary.

Share Improve this answer

edited Sep 17, 2008 at 22:23

Follow

answered Sep 17, 2008 at 22:15



Tom Kidd

12.9k ● 20 ● 91 ● 134

There's also the Vista sidebar gadget (an unofficial Cruise Control.NET client), available from codeclimber.net.nz/archive/2007/07/15/... – alastairs Sep 17, 2008 at 22:19

I got it working because we're going into some vertical markets which require us to run the build with some different command line parameters. So that's how I justified building it. I like this answer, and I'm liking cruise control too, it's much better than my sorry python script. – Peter Turner Sep 22, 2008 at 0:23



5

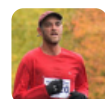


Implement [build lights](#) ... we did something similar with lava lamps and it was a huge hit. For added bonus marks give every developer a red light over their desk and have the right light come on when the build breaks.

Share Improve this answer

answered Sep 17, 2008 at 22:19

Follow



Rob Walker

47.4k ● 15 ● 100 ● 137

That's cool - I saw something similar done once with lava lamps (which had the added bonus that the developer who

broke the build had a chance to fix it before the red lava lamp got warmed up) – [Tom Kidd](#) Sep 17, 2008 at 22:31

True story: I once saw squad car lights, like what you see on a police cruiser, used as build lights. They were mounted from the ceiling and looked like the real deal. – [Parappa](#) Sep 17, 2008 at 22:56

I love the squad car lights idea ... coupled with a siren under the desk it would be a real incentive to not break the build! – [Rob Walker](#) Sep 18, 2008 at 0:12



3

Grab an old spare computer & put it in the corner of your office. Set it up to build your project. Write a small script that does:



1. Get latest version of all files.
2. If there was a file change, build
3. Notify you if there's a failure.



When you catch a break, compassionately get it fixed.

Consider adding a step to run unit tests, too.

If you can avoid scolding people for their mistakes, pretty soon people will be impressed with how reliable the build has been since you arrived. Build from there.

The trick is to spend very little of your time to generate a lot of value for your team, without pissing anyone off.

Follow



Jay Bazuzi

46.4k ● 16 ● 115 ● 170



1

Set up an autobuilder. Once you have it building and running the tests automatically, it won't matter if you convince other people to save their own time :)



If you're using git for version control, here's an autobuilder that automatically finds the exact checkin that started causing the tests to fail:



<http://github.com/apenwarr/gitbuilder/>

Share Improve this answer

answered Sep 17, 2008 at 22:25

Follow



apenwarr

11k ● 6 ● 50 ● 59



1

I would take a spare box, install a continuous integration server ([Hudson](#) or [CruiseControl](#) in the Java world) and set up a job that builds your application each time someone checks in some code.



You can either try to convince your coworker or just wait until someone breaks the build. In the latter case, just send the following email:



to: all developers

Guys,

I've just noticed that I can build our software using the latest version because of the following error:

...

I you want to be notified by our continuous build system (attached is the mail I received when it failed to build our application), just let me know.

Usually it doesn't take that long until everyone is on the list

Share Improve this answer

edited Sep 19, 2008 at 21:04

Follow

answered Sep 19, 2008 at 20:58



Vladimir

6,871 ● 2 ● 28 ● 25



0



I would set up the automated build as a nightly process such that every night it grabs the most recent code revision, builds it, and generates a report. Now you will know first thing every morning whether or not the build is broken, and if it is, you can notify the team. If broken builds are much of a problem on your project, people will probably start coming to you first to find out if it is safe to sync to the latest code, since you will be the person who tends to know on any given day whether or not the build is broken (by the way, an automated suite of unit tests helps a great deal with this as well). With any luck, people will start to realize that your nightly build is a useful thing to have, and you'll be able to just set up your daily build report as an email that goes out.

Share Improve this answer

answered Sep 17, 2008 at 22:20

Follow



Parappa

7,676 ● 3 ● 36 ● 39

Who would you send an email to? Everyone, your boss or the dude who you're build server determined broke the build?

– Peter Turner Sep 18, 2008 at 12:44



James Shora has two great links:

0

For hardware <http://jamesshore.com/Blog/Continuous-Integration-on-a-Dollar-a-Day.html>



For "Humanware" <http://jamesshore.com/Change-Diary/>



(The history of how he did it. The read is long but changing an organization is harder)



Share Improve this answer

answered Sep 17, 2008 at 22:28

Follow



borjab

11.6k ● 7 ● 74 ● 103



0

When the build is needed by the team on a regular basis, it's pretty easy. You appoint a team member (rotated periodically) to do the build. If the build process is complicated enough, the team will on its own come up with a way of at least partially automating the build. In the worst case, you'll have to automate the build yourself, but no-one will be against the automation.





Share Improve this answer

answered Sep 18, 2008 at 14:57

Follow



Alexander

9,380 ● 2 ● 28 ● 23



0

Demonstration is the best, and really the only way to change anyone's mind who is resistant to doing things differently.



Here we showed how useful automated builds are by having the ability for QA to grab a green light build straight from the build server and install it and test without any direction from the developers. They are able to continue working, they know that it at least passes its unit tests. It helped integrate testing and development reducing time bugs were in the system.

Share Improve this answer

answered Sep 19, 2008 at 21:06

Follow



Trevor Redfern

91 ● 1 ● 1 ● 5