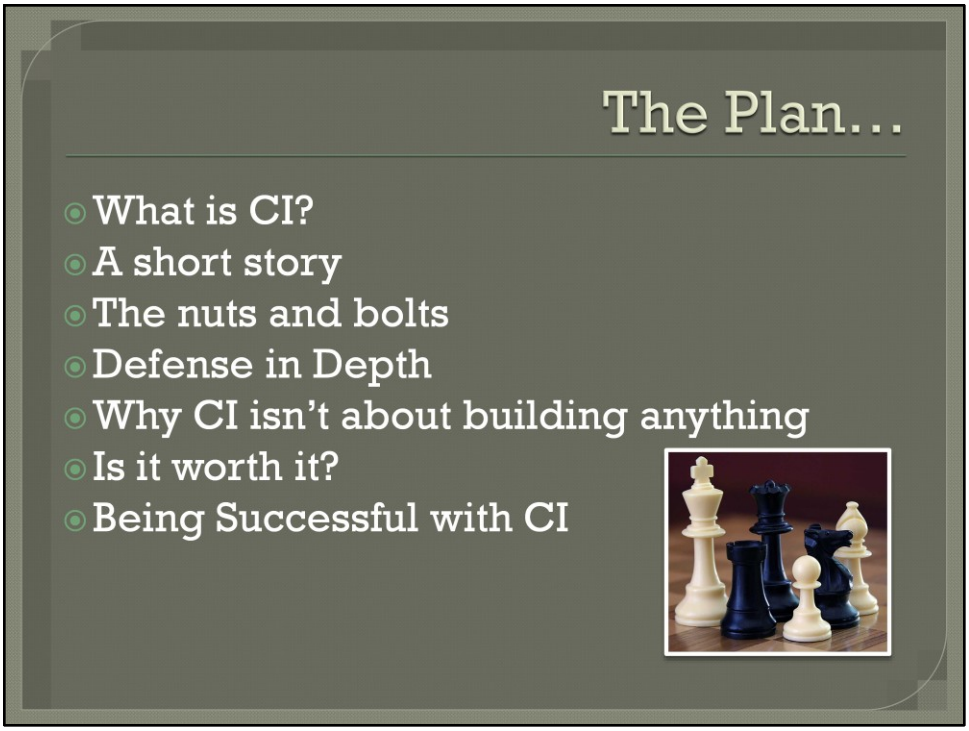
Continuous Integration and Defense In Depth

Experiences at Microsoft

Ade Miller
Development Manager
Microsoft patterns & practices

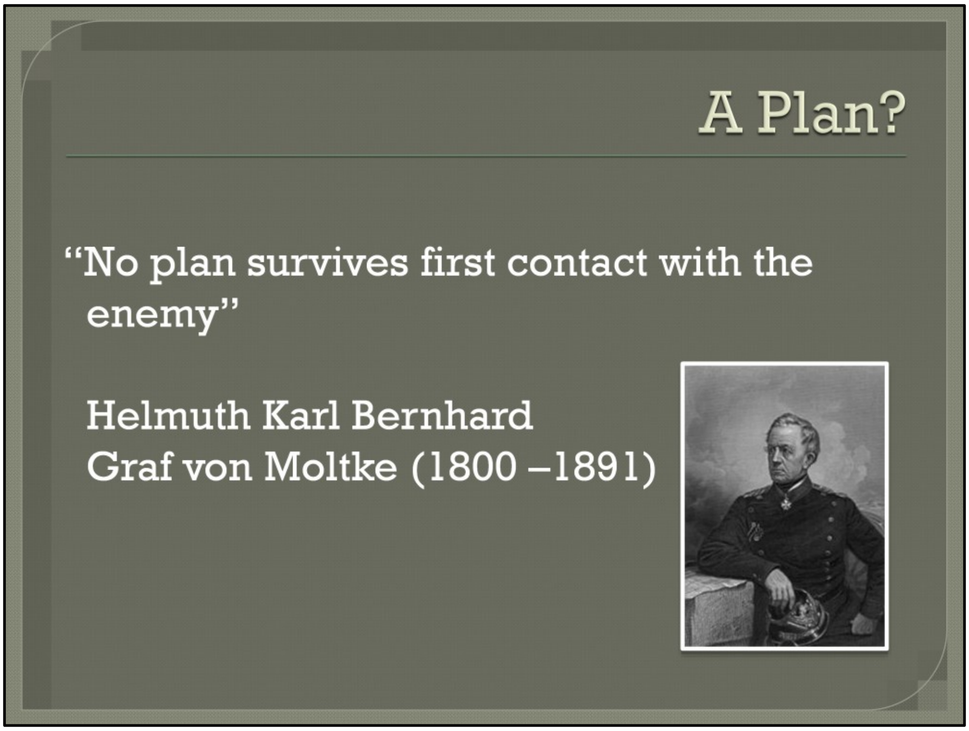
For more about patterns & practices: http://msdn.microsoft.com/practices/

My blog: http://ademiller.com/tech/



The Plan…

- What is CI?
- A short story
- The nuts and bolts
- Defense in Depth
- Why CI isn't about building anything
- Is it worth it?
- Being Successful with CI

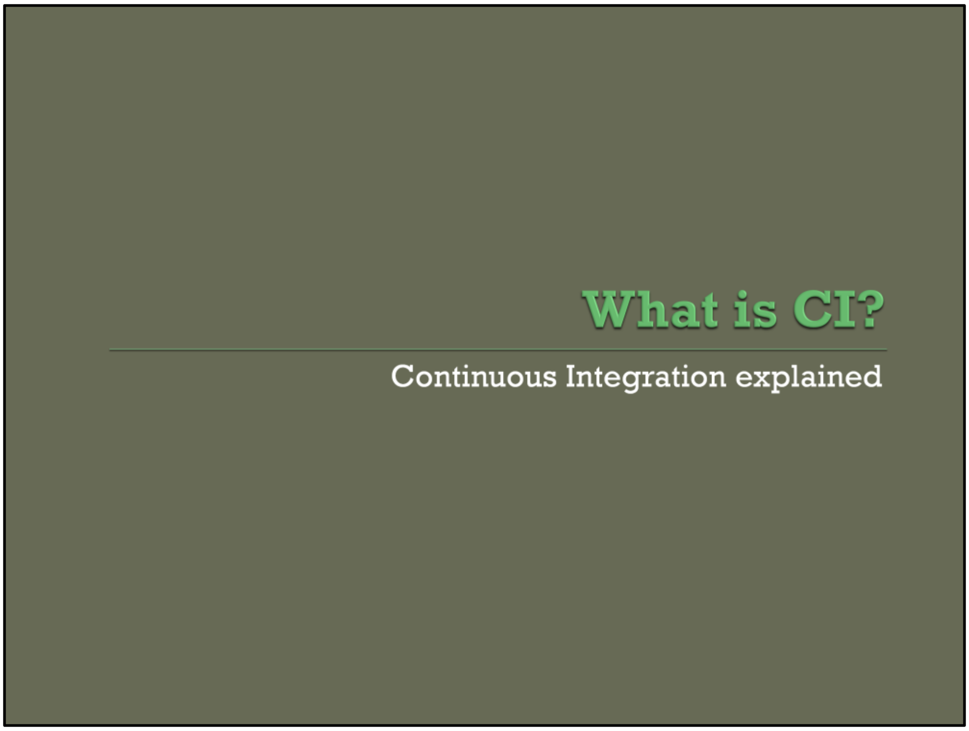


A Plan?

“No plan survives first contact with the enemy”

Helmuth Karl Bernhard
Graf von Moltke (1800 –1891)

Stop me. Ask questions. Tell me if you’ve heard it all before or you want to hear about something else!



What is CI?

Continuous Integration explained

What is CI?

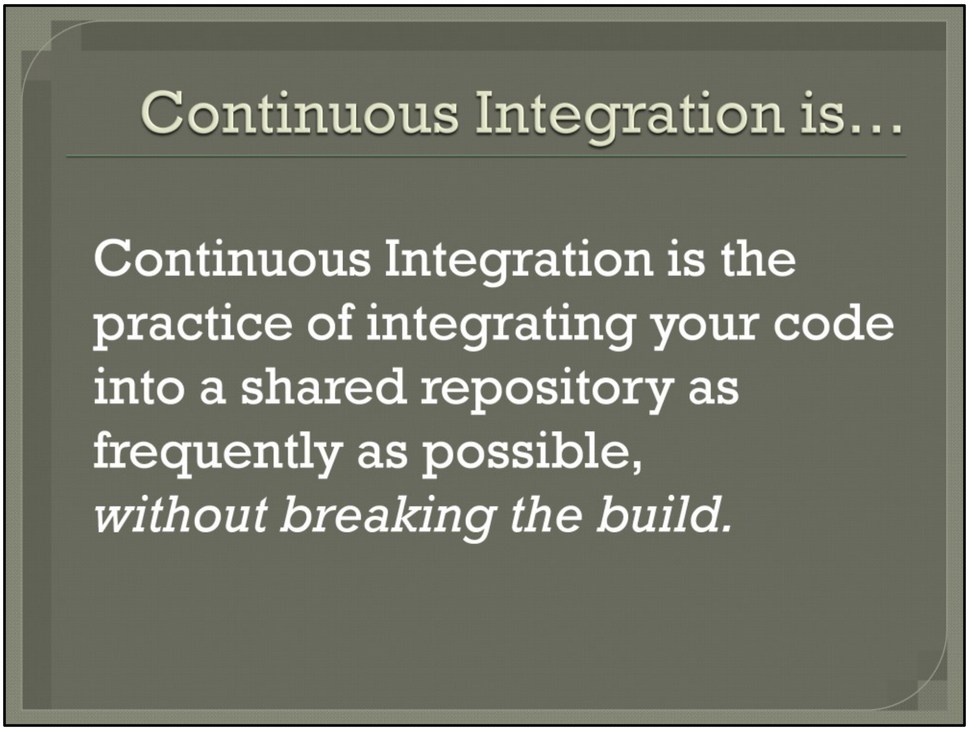
- Having your cake and eating it
- A free lunch
- Developer nirvana
- The agile gateway drug

There's a lot of good things about CI:

It's very low cost to adopt
It can spread within your organization virally
It's (fairly) easy to articulate the cost-benefit (to management)

If you are trying to get people to think about adopting better development practices (like agile) then in my experience CI is a good way to build credibility! It has a very high success rate.

Photo: http://www.sxc.hu/photo/982974

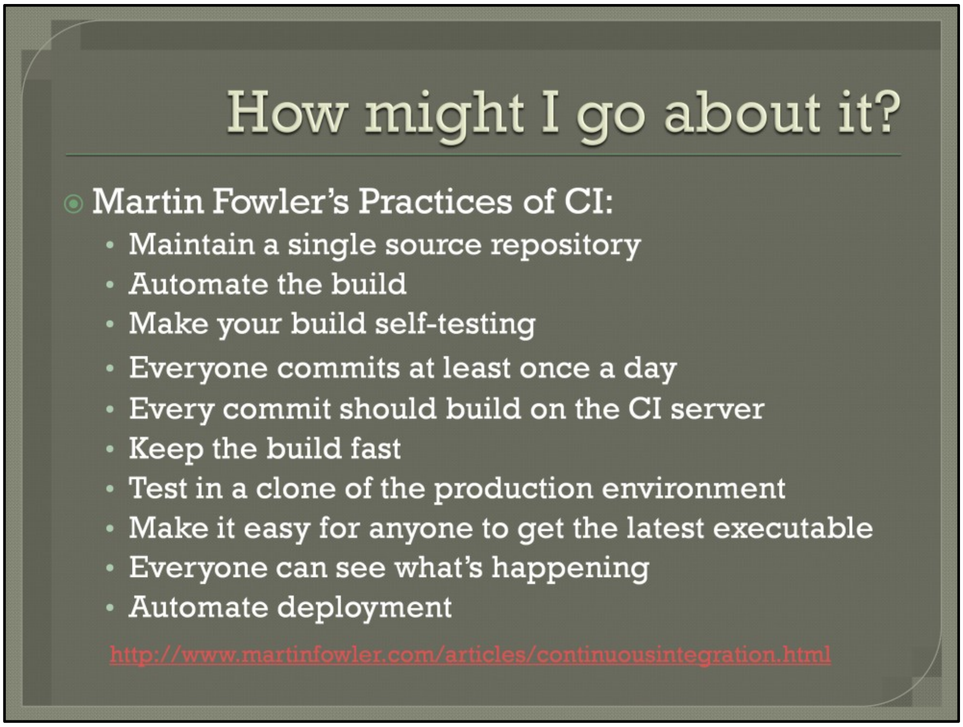


Continuous Integration is…

Continuous Integration is the practice of integrating your code into a shared repository as frequently as possible, *without breaking the build.*

The practice of CI is often used in conjunction with a CI Server, but it is possible to practice very simple CI without a CI Server.

For example every developer gets the source tree and merges changes rebuilds and runs tests before committing



How might I go about it?

- Martin Fowler's Practices of CI:
  - Maintain a single source repository
  - Automate the build
  - Make your build self-testing
  - Everyone commits at least once a day
  - Every commit should build on the CI server
  - Keep the build fast
  - Test in a clone of the production environment
  - Make it easy for anyone to get the latest executable
  - Everyone can see what's happening
  - Automate deployment

http://www.martinfowler.com/articles/continuousintegration.html

Who is Martin Fowler?

Chief Architect at ThoughtWorks

Author of numerous books, including Refactoring & Planning Extreme Programming

With Kent Beck, one of the first people to publish on the subject of CI

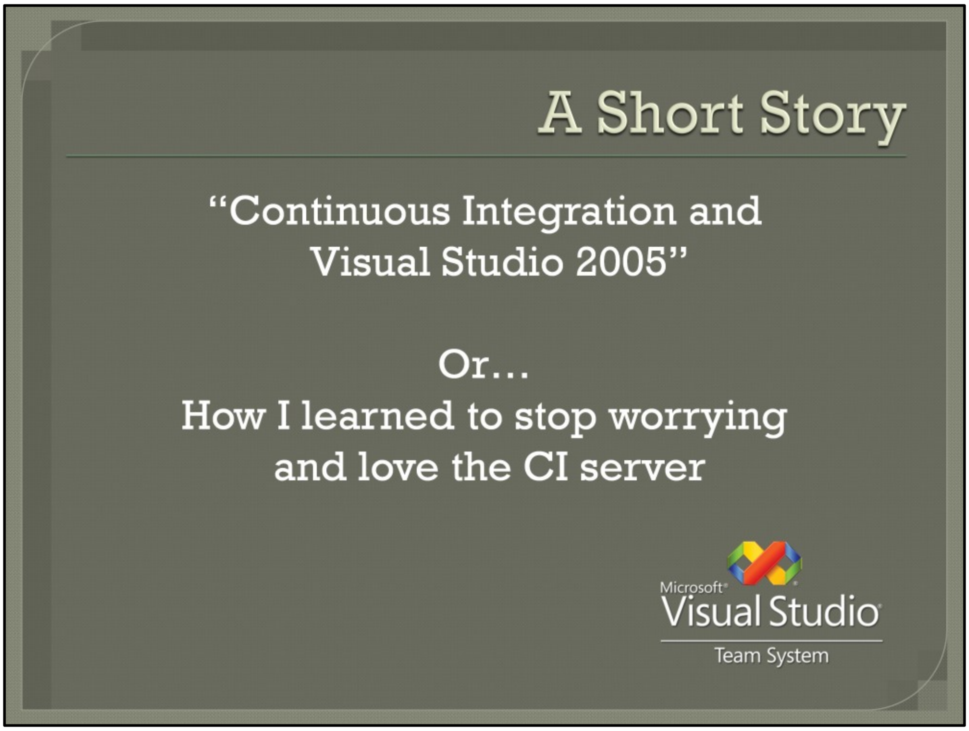
CI is also…

- One of the easiest agile practices to adopt

Copyright © 1990-2009 Ade Miller

All you need is a reasonable machine to host the server and one person willing to set it up and keep it running.

After 2-3 weeks people will start to see value in it.



A Short Story

"Continuous Integration and Visual Studio 2005"

Or…
How I learned to stop worrying and love the CI server

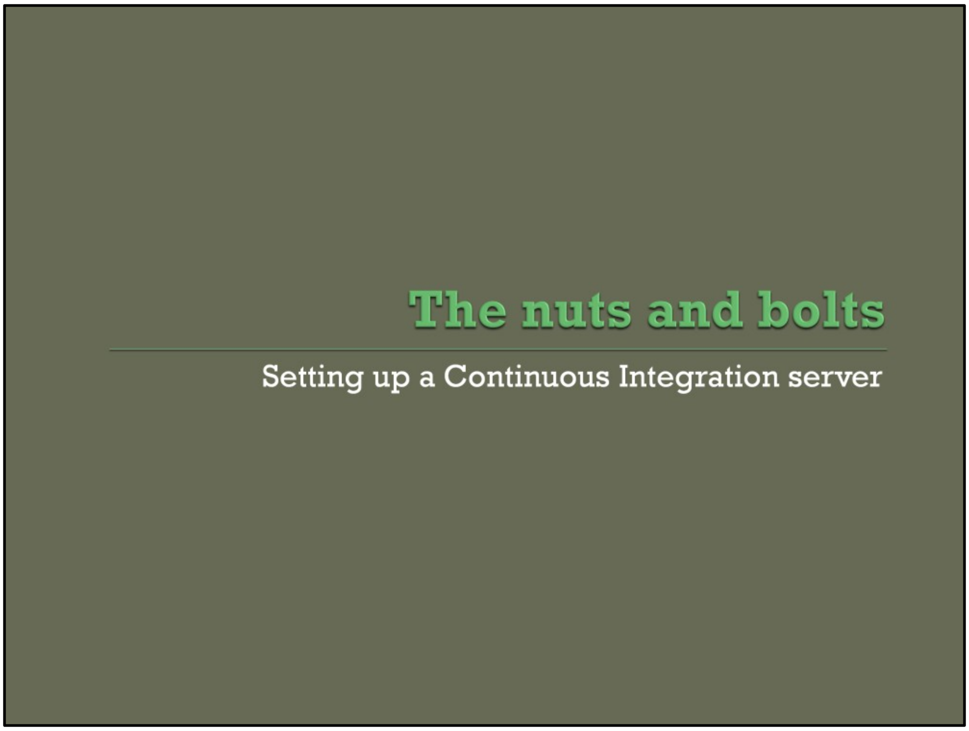
Microsoft Visual Studio Team System

The very short story…

We had a massive check in process which meant that it could take hours to submit even the smallest change.

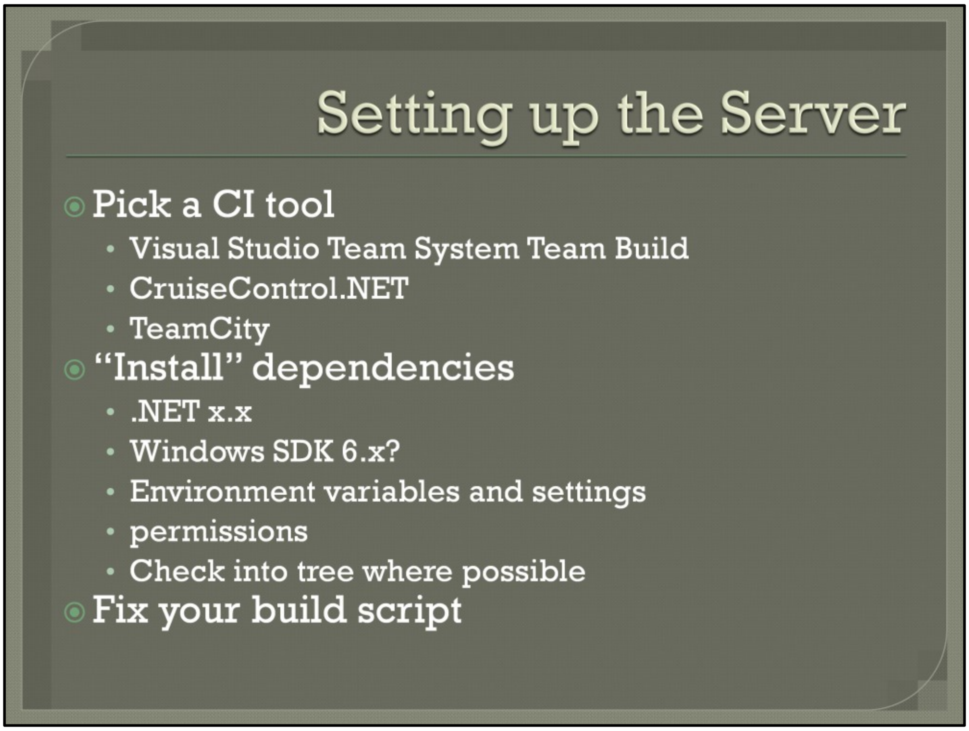
We adopted CI and this cut our debt by a huge margin, ~4 hours to under an hour.

Other sub teams within my group started to use the same process and piggy back on our server!



The nuts and bolts

Setting up a Continuous Integration server



Setting up the Server

- Pick a CI tool
  - Visual Studio Team System Team Build
  - CruiseControl.NET
  - TeamCity
- "Install" dependencies
  - .NET x.x
  - Windows SDK 6.x?
  - Environment variables and settings
  - permissions
  - Check into tree where possible
- Fix your build script

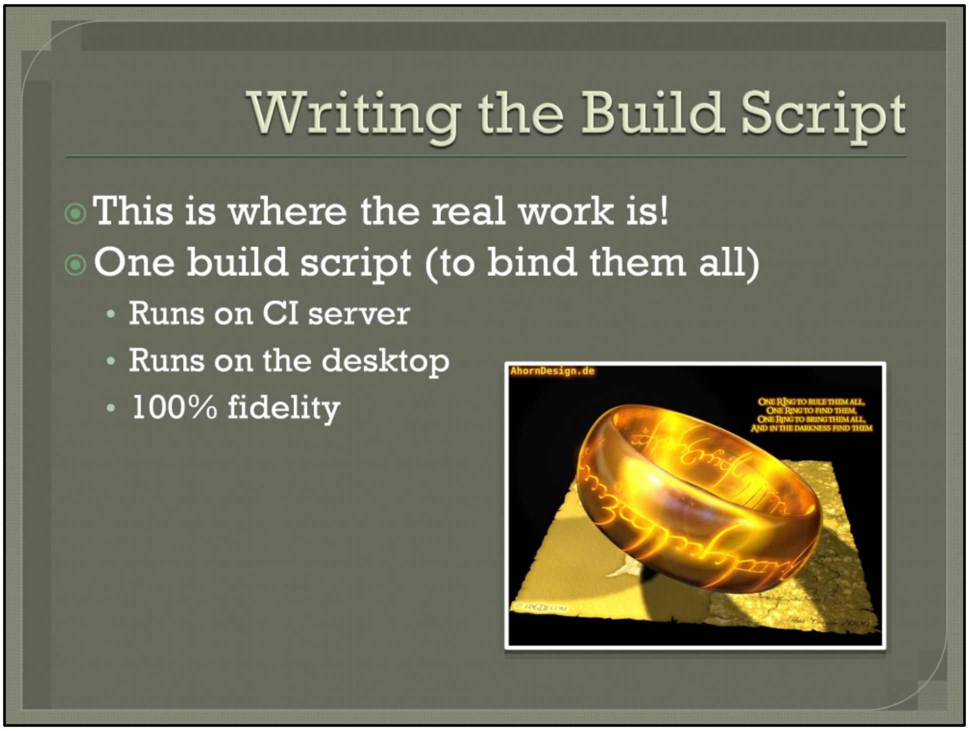
CC.NET and TeamCity are very lightweight. Ideal for small team CI.

TeamBuild has many more features for larger projects and teams:

support drop management and scheduling
multiple build machine support (in VS2010)
visual designer for Windows Workflow (in VS2010)
gated check-in (in VS2010)

In the process of doing this you'll gain a better understanding of your application dependencies
Fix a load of issues with your build environment

I'm not going to show you how to setup a CI server or really dig into which one to use (see appendix).
Download/install on and read the manual!



Writing the Build Script

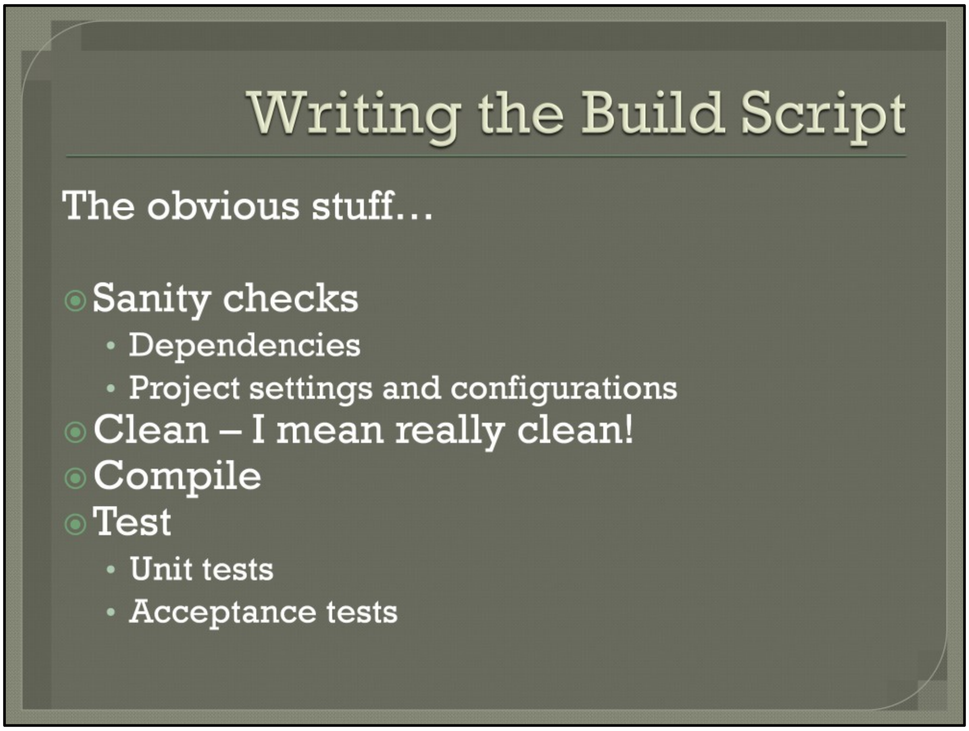
- This is where the real work is!
- One build script (to bind them all)
  - Runs on CI server
  - Runs on the desktop
  - 100% fidelity

AhornDesign.de
One Ring to rule them all,
One Ring to find them,
One Ring to bring them all,
And in the darkness find them

100% fidelity is *really* important!

You will waste a lot of time chasing down build breaks that only happen on the server otherwise.

If you make it easy to run a subset of the CI build locally developers will do this before checking in. But you have to make it easy!



Writing the Build Script

The obvious stuff…

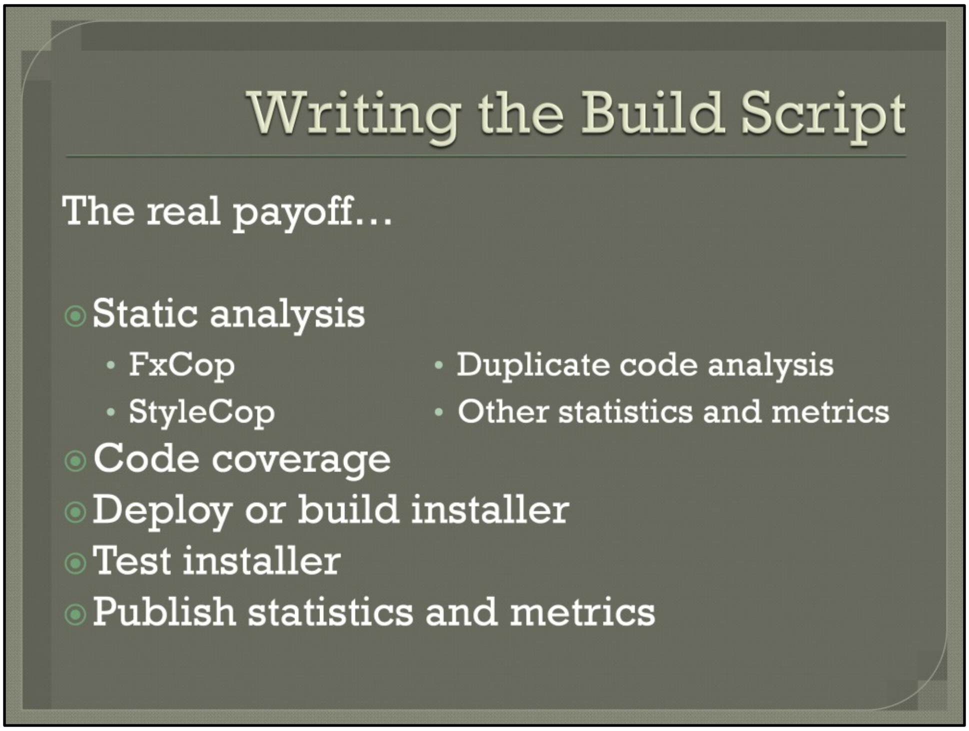
- Sanity checks
  - Dependencies
  - Project settings and configurations
- Clean – I mean really clean!
- Compile
- Test
  - Unit tests
  - Acceptance tests

I don't like the default VS clean behavior.
It leaves a lot behind. I implement a custom build target that tries to remove everything in bin, obj etc.

Sanity checks like…

All project files are configured correctly?
Rogue solution files
Correct dependencies are installed?

All of this you might reasonably expect the developer to be running locally *before* they check in.



Writing the Build Script

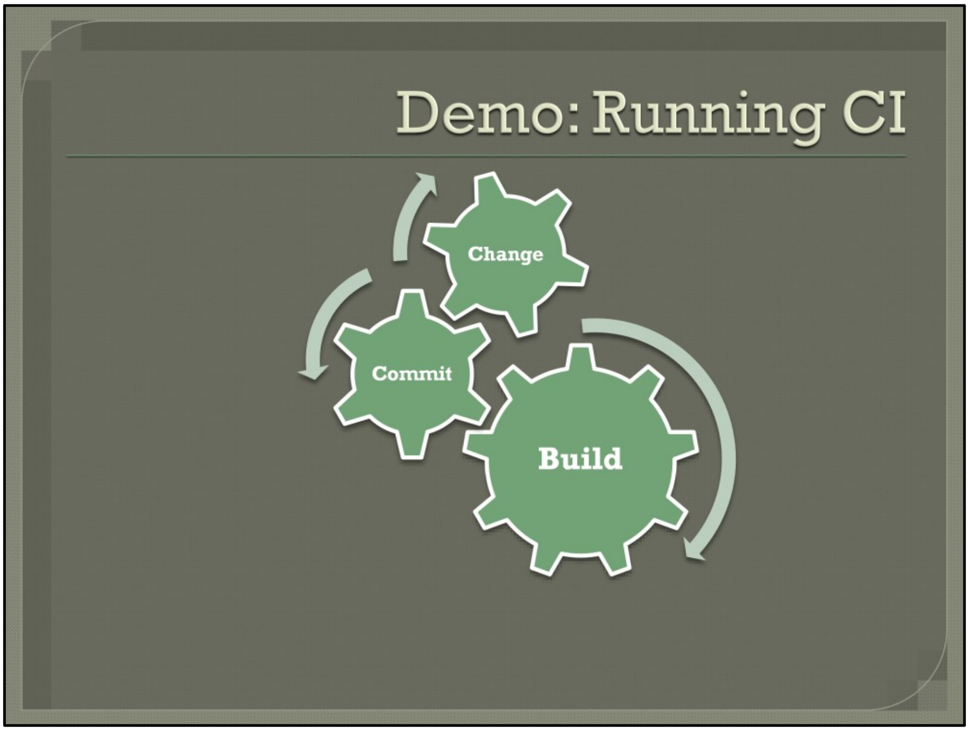
The real payoff…

- Static analysis
  - FxCop
  - StyleCop
  - Duplicate code analysis
  - Other statistics and metrics
- Code coverage
- Deploy or build installer
- Test installer
- Publish statistics and metrics

Other statistics and metrics like…

FxCop & StyleCop warnings and errors
LOC for product vs. LOC for tests
“TODO”s

The goal is to drive these metrics to 100% or 0% over time and then “lock” them by causing the build to fail.



Demo: Running CI

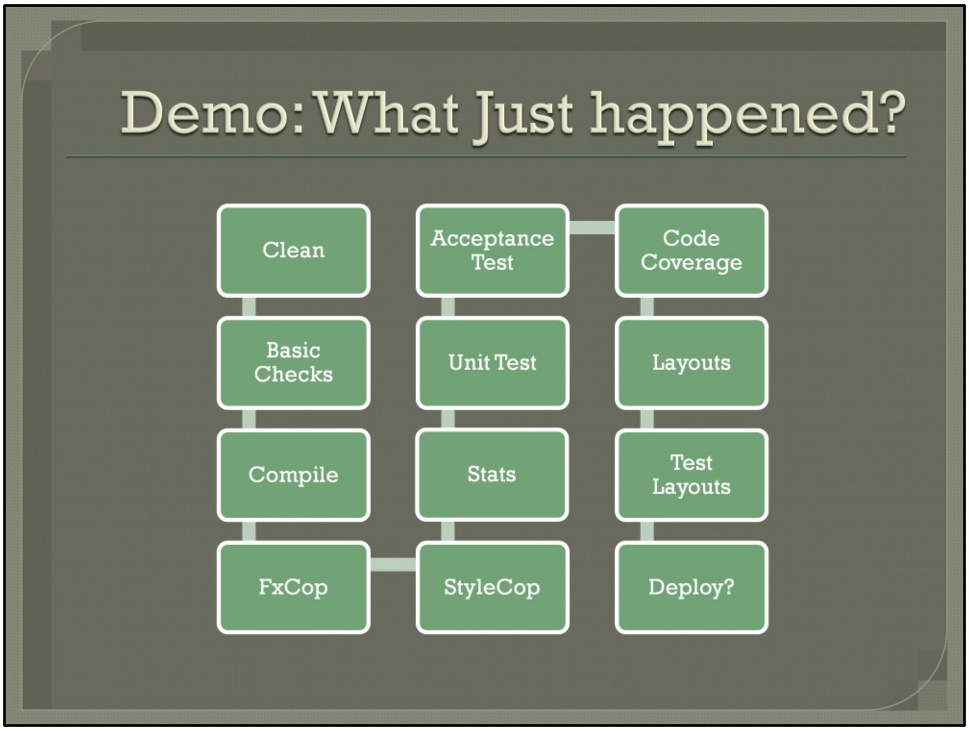
Change

Commit

Build

Here I demo:

- Open project
- Make a change
- Run sub-build locally (build and some unit tests) from the IDE or command line
- Check in
- Do something else… while the build finishes
- Review the results in the UI



Demo: What Just happened?

- Clean
- Basic Checks
- Compile
- FxCop
- StyleCop
- Stats
- Unit Test
- Acceptance Test
- Code Coverage
- Layouts
- Test Layouts
- Deploy?

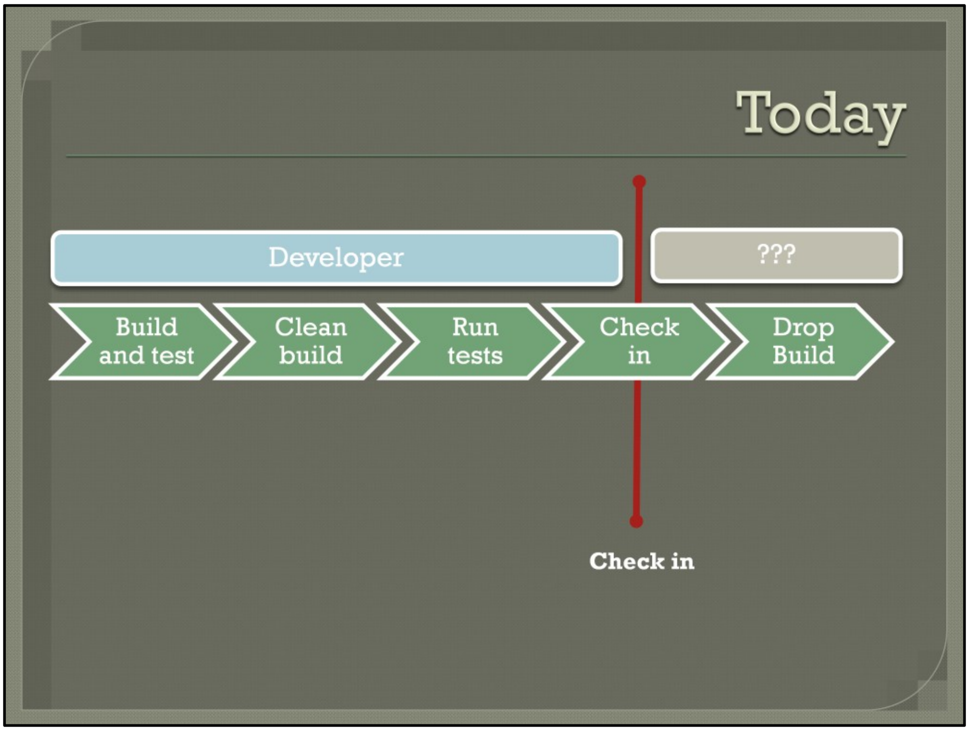
All this stuff happened when I pressed the submit button and I did NOTHING!

Let’s review the build script…

Defense in Depth

Improving quality with CI

Here’s another way to think about this…



Today

Developer

???

Build and test

Clean build

Run tests

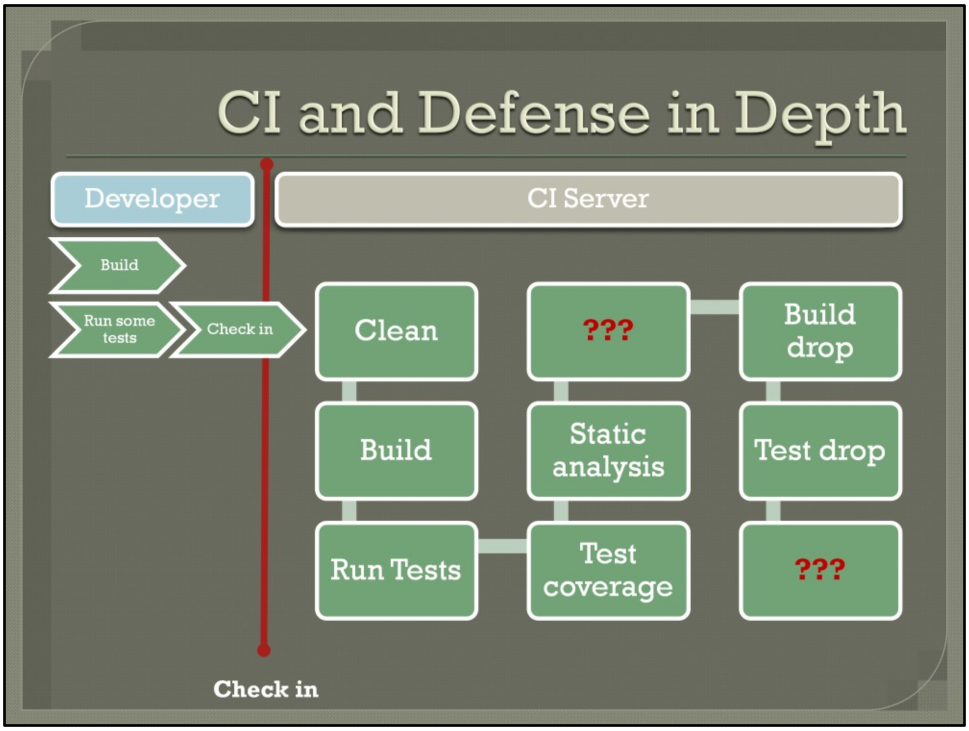
Check in

Drop Build

Check in

Without CI we have to add further developer TAX as we add more tests.

Remember the Visual Studio story? This tax was *hours per check in!*



CI and Defense in Depth

With CI I have to modify the build and setup the CI server but after that the cost is very low… ALMOST FREE!

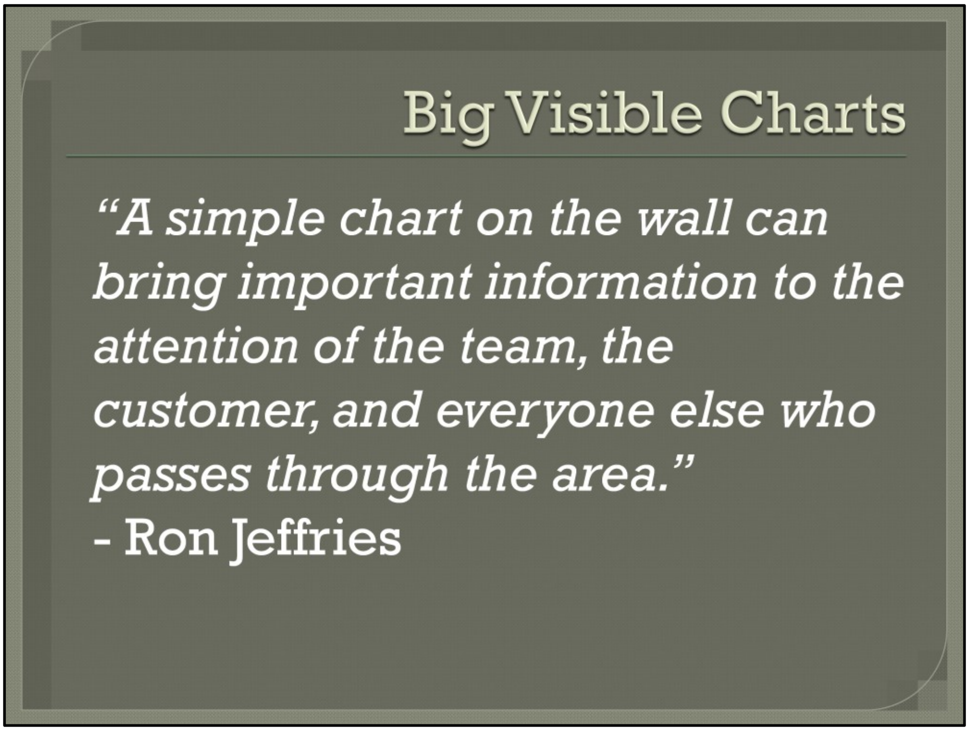
Welcome to the free lunch!

Why CI isn't About Building Anything

Big Visible Charts

So now we understand what CI is and how you can use it to improve quality.

You can also use it to improve your team using Big Visible Charts

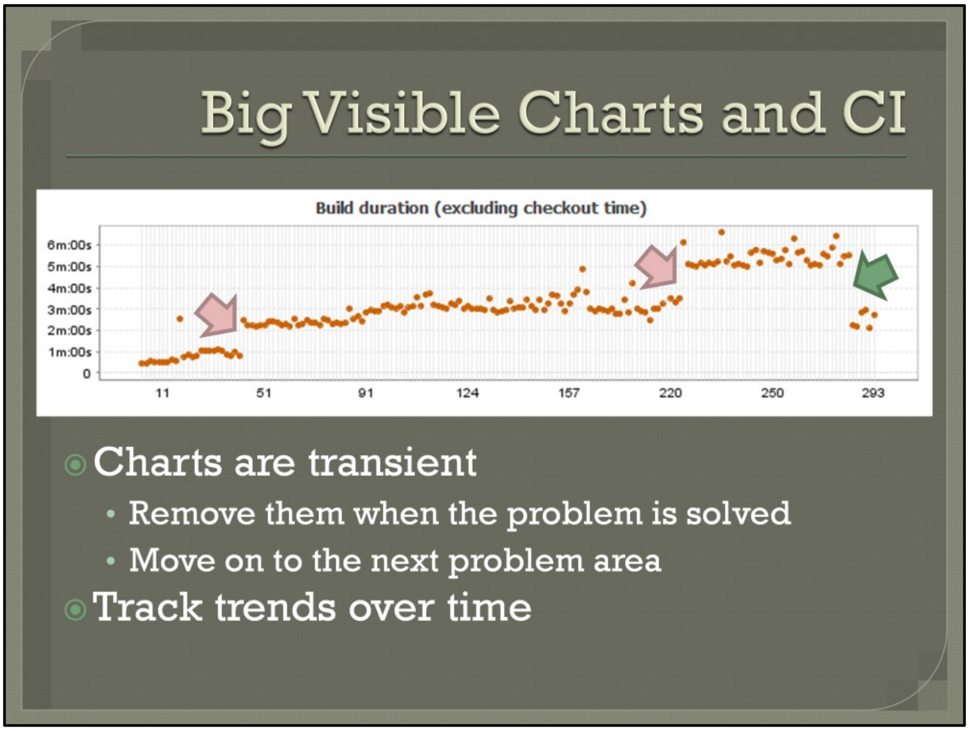


Big Visible Charts

"A simple chart on the wall can bring important information to the attention of the team, the customer, and everyone else who passes through the area."
- Ron Jeffries

Big Visible charts should really be on a wall in your team space but the CI server comes close, especially if you have it setup with a system tray tool to notify developers when the build breaks.

You can always print off a chart and stick it on the wall.



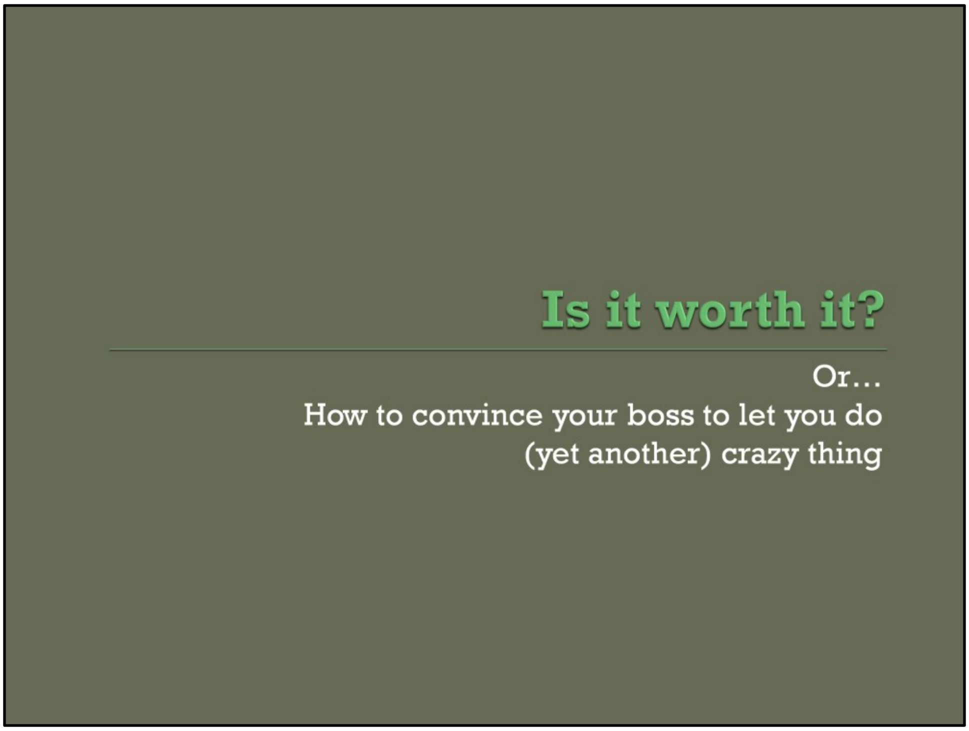
Big Visible Charts and CI

- Charts are transient
  - Remove them when the problem is solved
  - Move on to the next problem area
- Track trends over time

What happened in this chart.

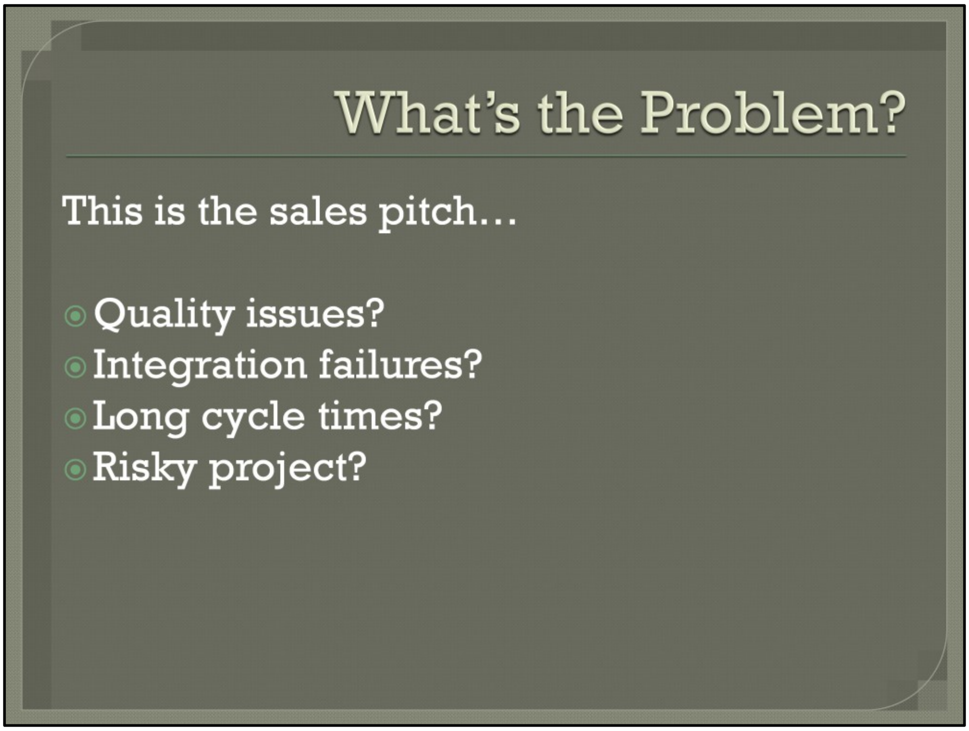
I added more and more tests (good)
The build took longer and longer (bad)
Eventually the chart embarrassed me so much I made some of my tests a lot faster.

Note: All my unit tests run in << 1s but I had a couple of acceptance tests that took a lot longer especially when running to gather code coverage data. This was the problem.



Is it worth it?

Or…
How to convince your boss to let you do
(yet another) crazy thing



What’s the Problem?

This is the sales pitch…

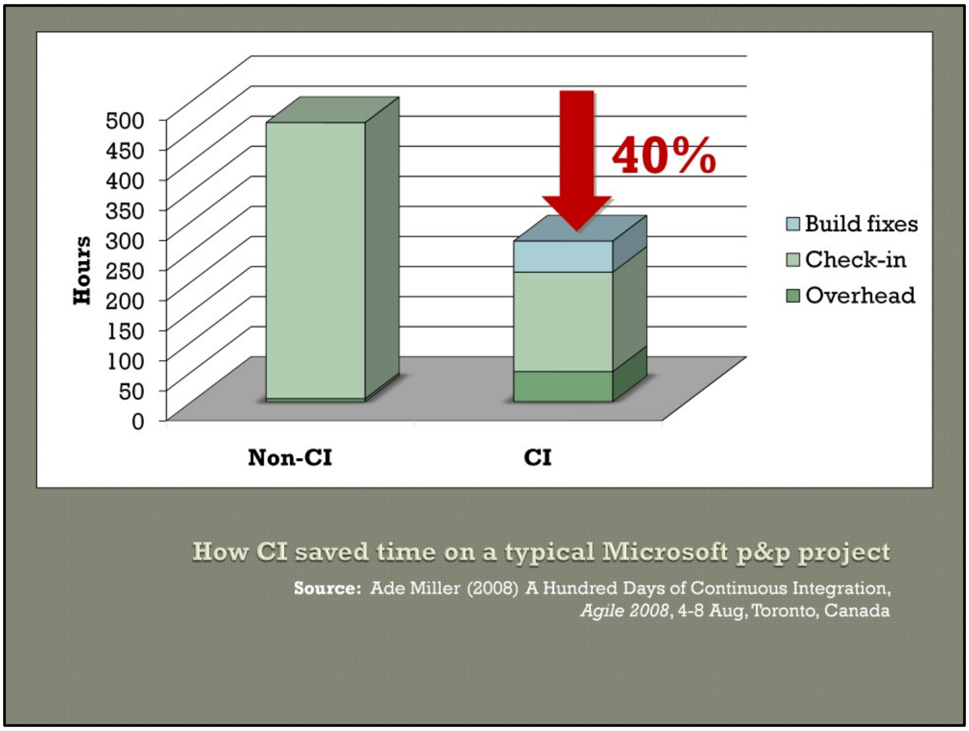
- Quality issues?
- Integration failures?
- Long cycle times?
- Risky project?

The three step guide to win over key decision makers:

1) Identify the problem
2) Sell a solution to a problem not a practice
3) Back up your statements with evidence

Try and pitch it as solving a problem *not* “I want to do CI because it’s GOOD”.

It is good but middle managers with pointy hair (like me) need something they can grasp.

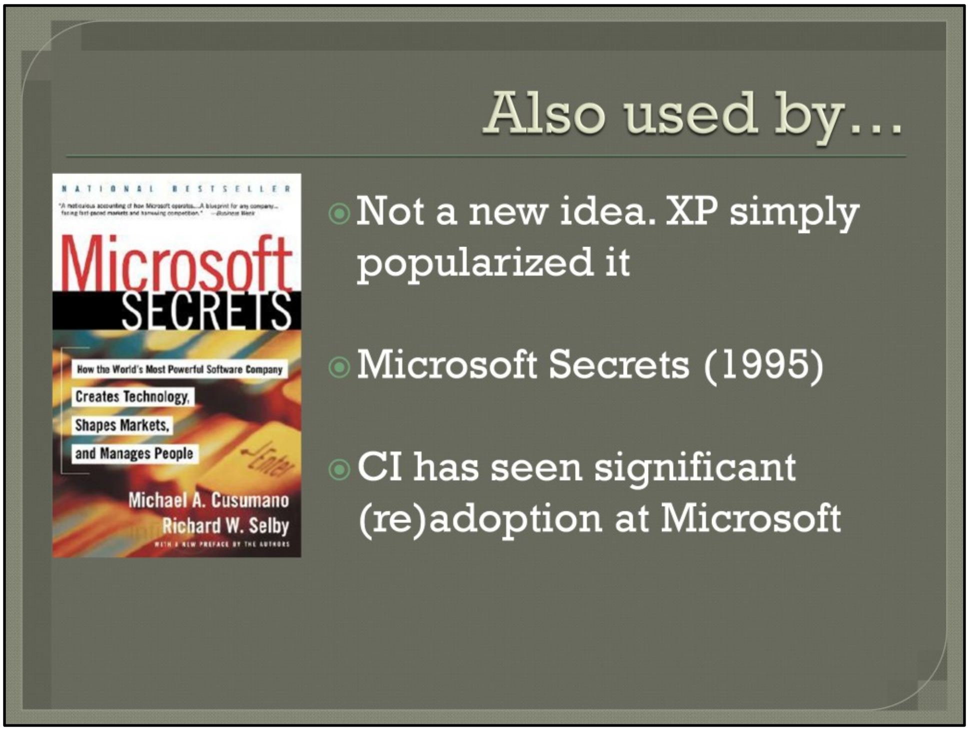


How CI saved time on a typical Microsoft p&p project

Source: Ade Miller (2008) A Hundred Days of Continuous Integration, *Agile 2008*, 4-8 Aug, Toronto, Canada

You can download the paper from my web site.

http://www.ademiller.com/blogs/tech/about-2/



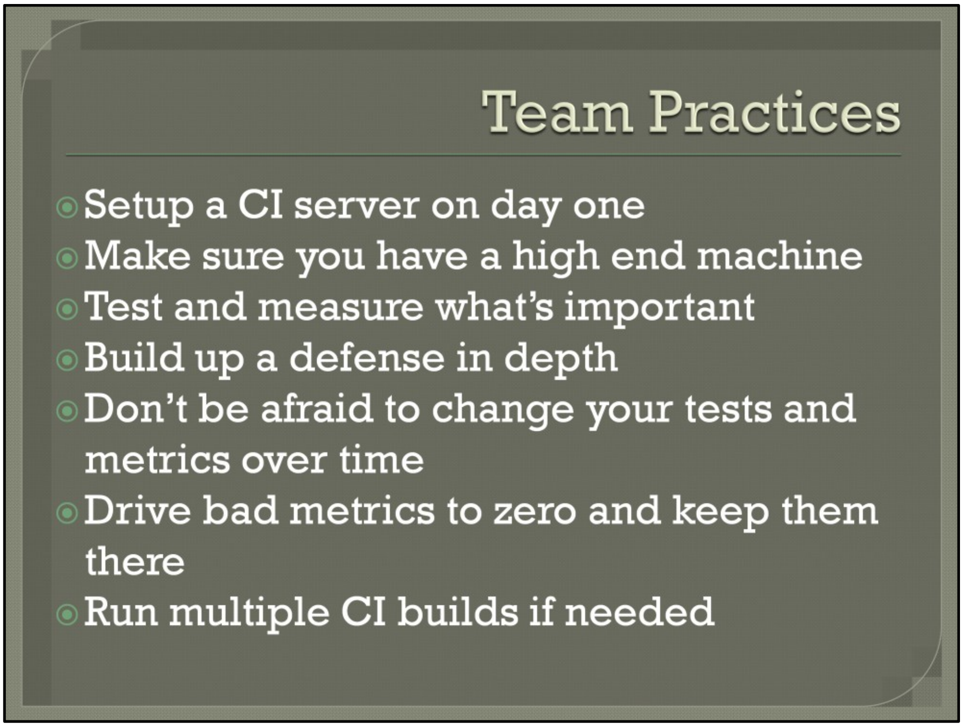
Also used by…

- Not a new idea. XP simply popularized it
- Microsoft Secrets (1995)
- CI has seen significant (re)adoption at Microsoft

I’m not saying Microsoft invented CI but they have always had a “build every day” culture which has now seen significant adoption of CI.

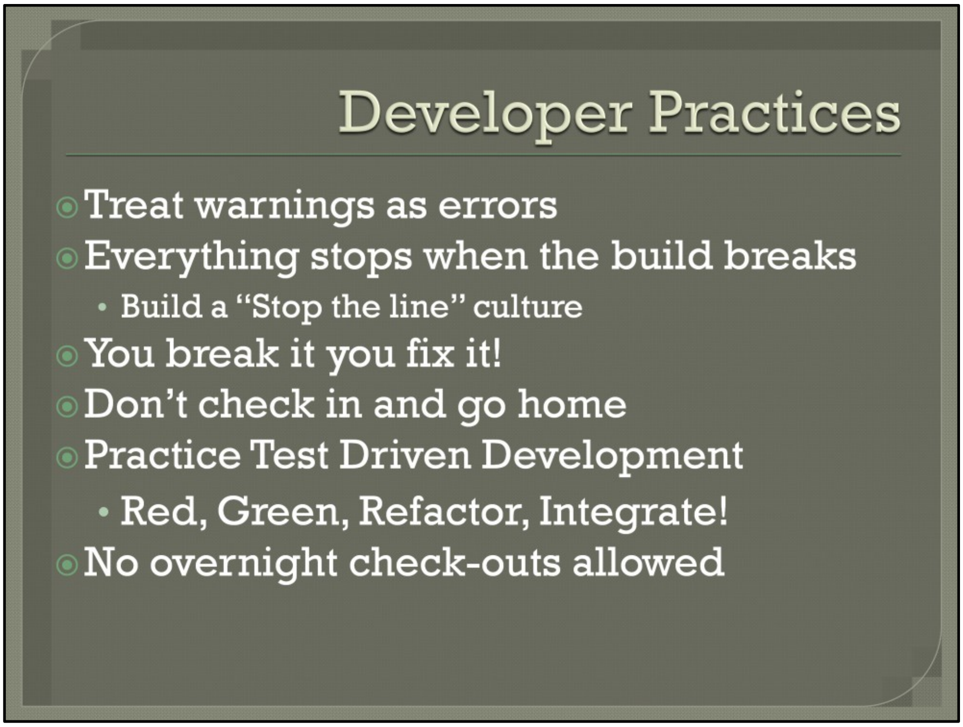
Being Successful with CI

Some proven practices



Team Practices

- Setup a CI server on day one
- Make sure you have a high end machine
- Test and measure what's important
- Build up a defense in depth
- Don't be afraid to change your tests and metrics over time
- Drive bad metrics to zero and keep them there
- Run multiple CI builds if needed



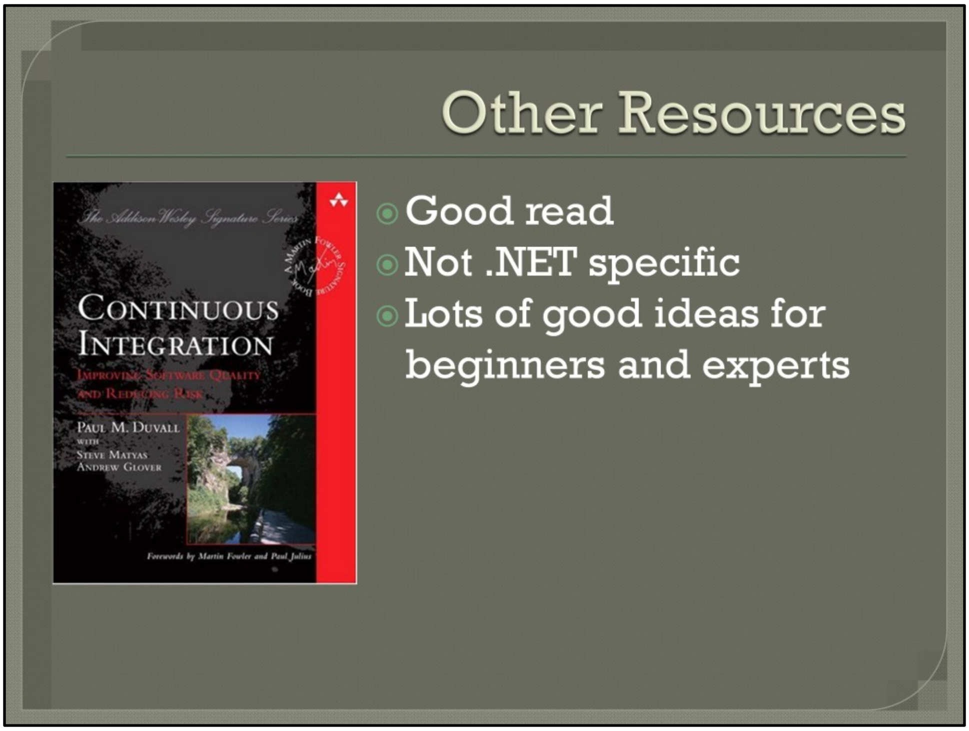
Developer Practices

- Treat warnings as errors
- Everything stops when the build breaks
  - Build a "Stop the line" culture
- You break it you fix it!
- Don't check in and go home
- Practice Test Driven Development
  - Red, Green, Refactor, Integrate!
- No overnight check-outs allowed

The build server is the only one that counts!

"It works on my machine" More on what this means here:

http://www.ademiller.com/blogs/tech/2008/06/it-works-on-my-machine-award/



Other Resources

- Good read
- Not .NET specific
- Lots of good ideas for beginners and experts

The Addison-Wesley Signature Series
A Martin Fowler Signature Book
Continuous Integration
Improving Software Quality and Reducing Risk
Paul M. Duvall
with
Steve Matyas
Andrew Glover
Forewords by Martin Fowler and Paul Julius

Team City is available from JetBRAINS:

http://www.jetbrains.com/teamcity/

They also offer a free hosting service for OpenSource projects on CodeBetter.com. For details see this blog post:

http://codebetter.com/blogs/james.kovacs/archive/2009/02/24/announcing-teamcity-codebetter-com.aspx

Slides and Other Cool Stuff

http://ademiller.com/tech

- This and other decks
- Experience reports

http://microsoft.com/agile

- White papers
- Videos of the p&p team rooms

ade.miller@microsoft.com

#2782
Home About Publications Talks Recommended Reading
Building a Green Windows Home Server: Fan Configuration
Welcome...
patterns & practices Agile Development Showcase
amazon.com Music
Agile

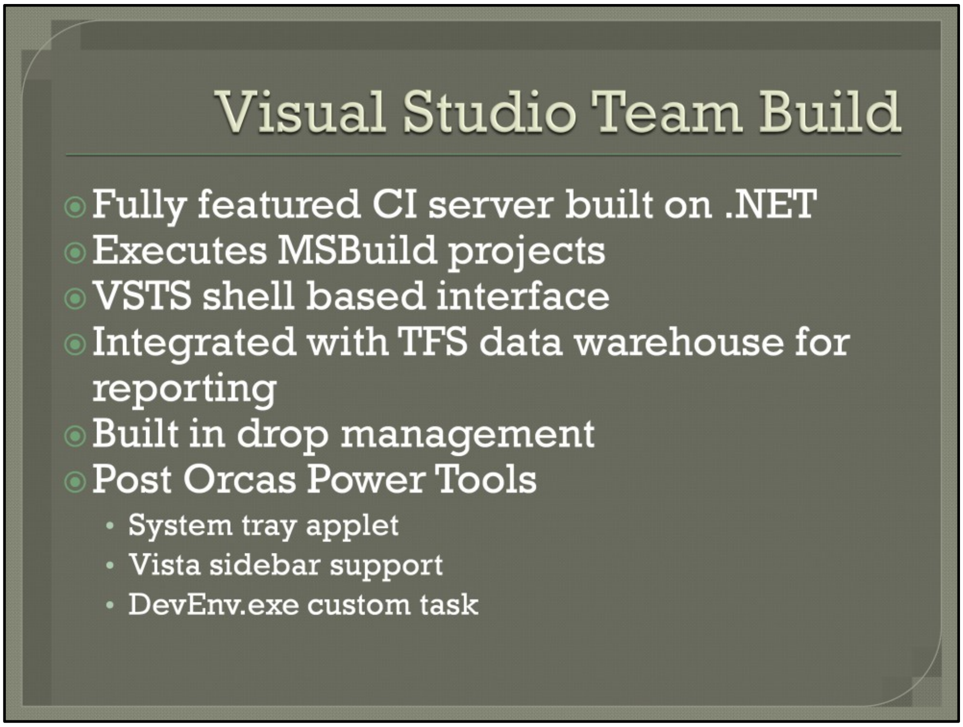
Questions? Feedback?

Photo credit: http://www.sxc.hu/

I’m done!

Stock photo:
http://www.sxc.hu/photo/728931
http://www.sxc.hu/photo/418215

Appendix

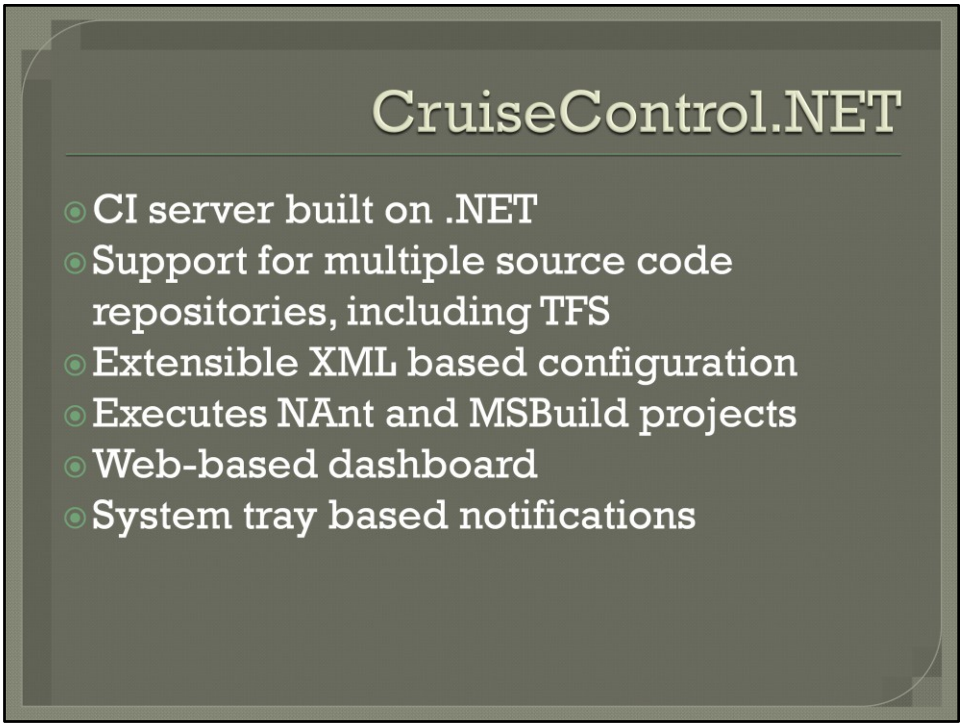


Visual Studio Team Build

- Fully featured CI server built on .NET
- Executes MSBuild projects
- VSTS shell based interface
- Integrated with TFS data warehouse for reporting
- Built in drop management
- Post Orcas Power Tools
  - System tray applet
  - Vista sidebar support
  - DevEnv.exe custom task

More details of Team Build can be found on MSDN:

http://msdn.microsoft.com/en-us/library/ms181710(VS.80).aspx



CruiseControl.NET

- CI server built on .NET
- Support for multiple source code repositories, including TFS
- Extensible XML based configuration
- Executes NAnt and MSBuild projects
- Web-based dashboard
- System tray based notifications

CruiseControl can be downloaded from:

http://cruisecontrol.sourceforge.net/