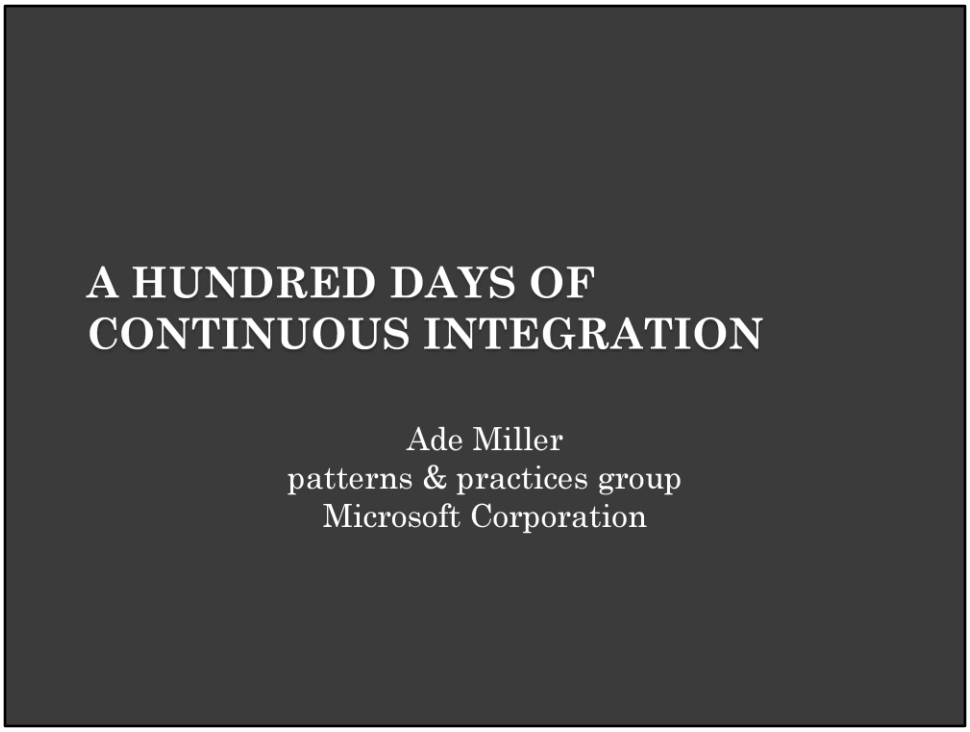


Notes

This deck contains some speaker notes. Most of the real content can be found in the paper that accompanied this deck. You can find a link to the paper on my talks page here:

http://www.ademiller.com/blogs/tech/talks/

OK… So here goes!

• Ade Miller is the Development Manager at p&p
• Small group within Microsoft Developer Division who ship guidance on application development to Enterprise customers
  Enterprise Library
  Software Factories
  Books
• Applies to p&p NOT all of Microsoft

WHAT ARE YOU GOING TO GET OUT OF THIS

• Data that shows CI has concrete benefit which may help you sell adoption
• Best practices around CI in a distributed organization
• Please try and keep questions to the end unless you're completely lost in which case wave your hand

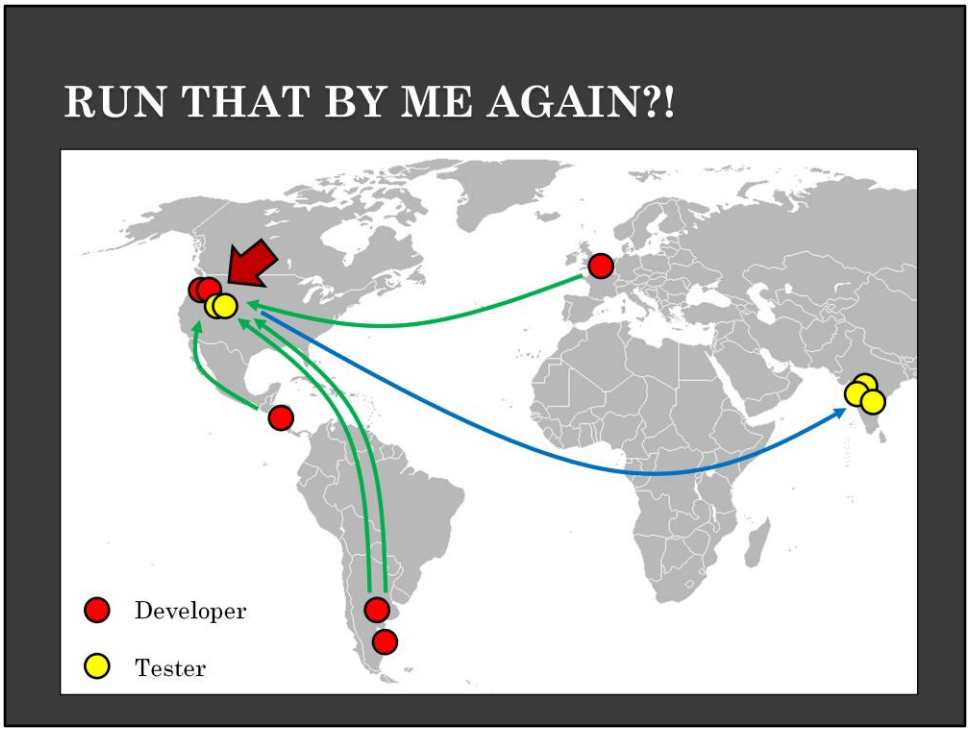
WHO? WHAT? WHERE?

The Service Factory: Modeling Edition team

- REDMOND
  - Two developers
  - Two testers
  - Architect
  - Product owner
  - Project manager
- ARGENTINA
  - Two developers
- COSTA RICA
  - One developer
- HOLLAND
  - One developer
- INDIA
  - Two or three testers

• AGILE TEAM! Scrum / XP
• Ade was the Dev Lead on this team (back in day when he wrote code)
• Building a set of Visual Studio packages that made up a software factory
• Mainly managed code but some C++
• You can download the binaries and source from
http://www.codeplex.com/servicefactory if your interested to see what we built

---
Stock photo:
http://www.sxc.hu/photo/858531



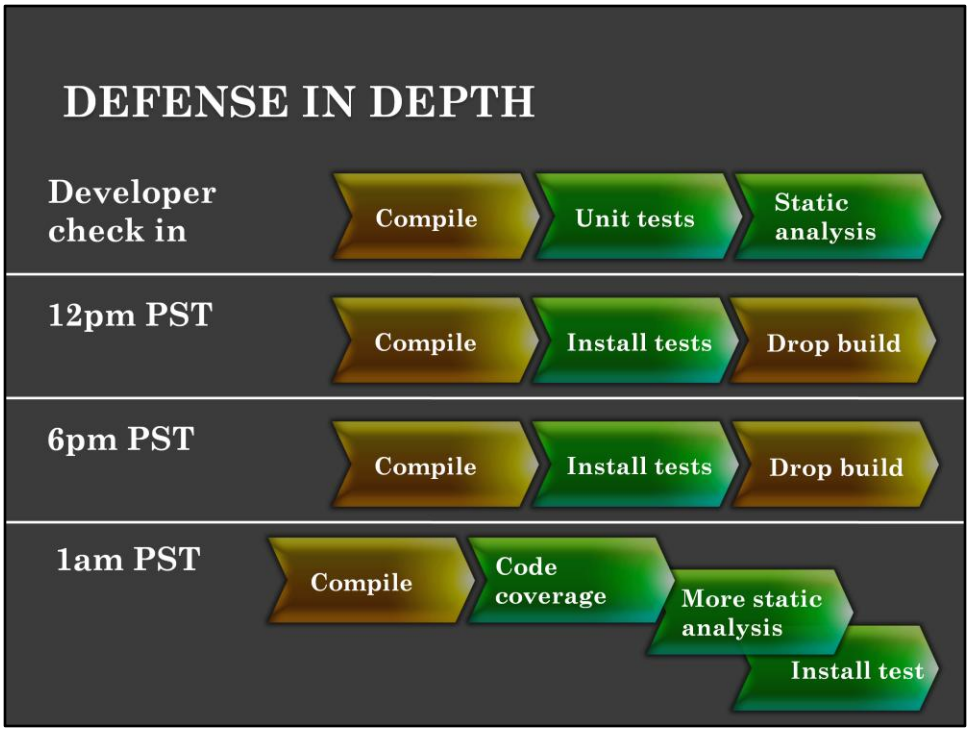
• CI server located in Redmond (West coast time)
• Devs in South America (+3-4) and Holland (+9)
• Some testers in India (-9)

• EIGHTEEN HOUR TIMEZONE DIFFERENCE!

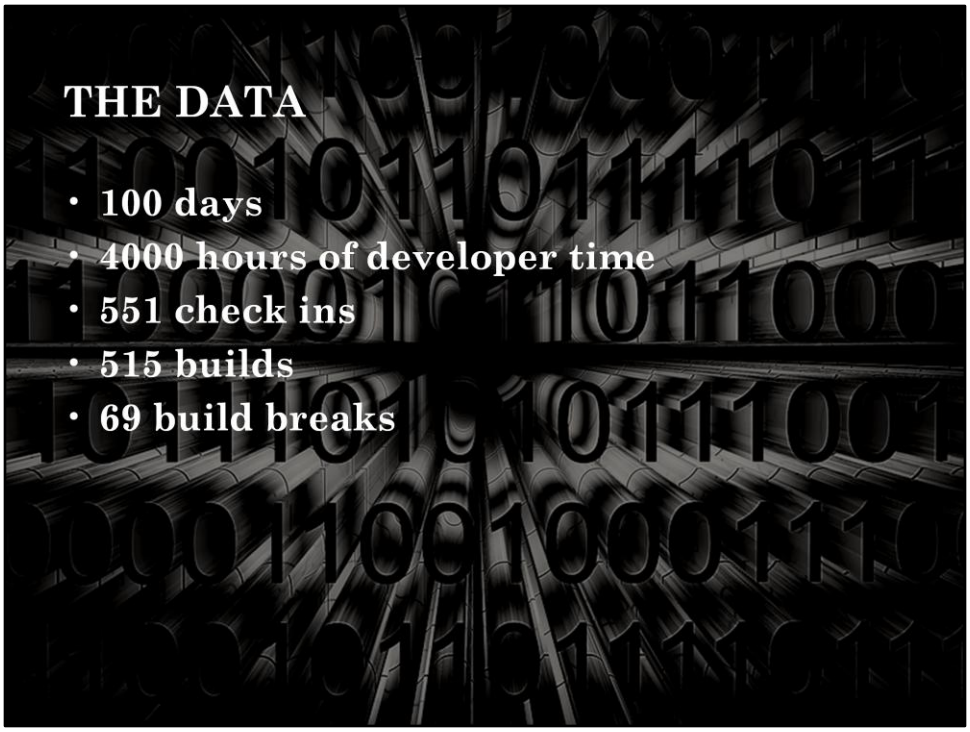
Stock photo:
http://en.wikipedia.org/wiki/Image:BlankMap-World.png

- Lynchpin of the development process
- Make the CI server do the work!
- Defense in depth
- Staged build strategy

Stock photo:
http://www.sxc.hu/photo/801819

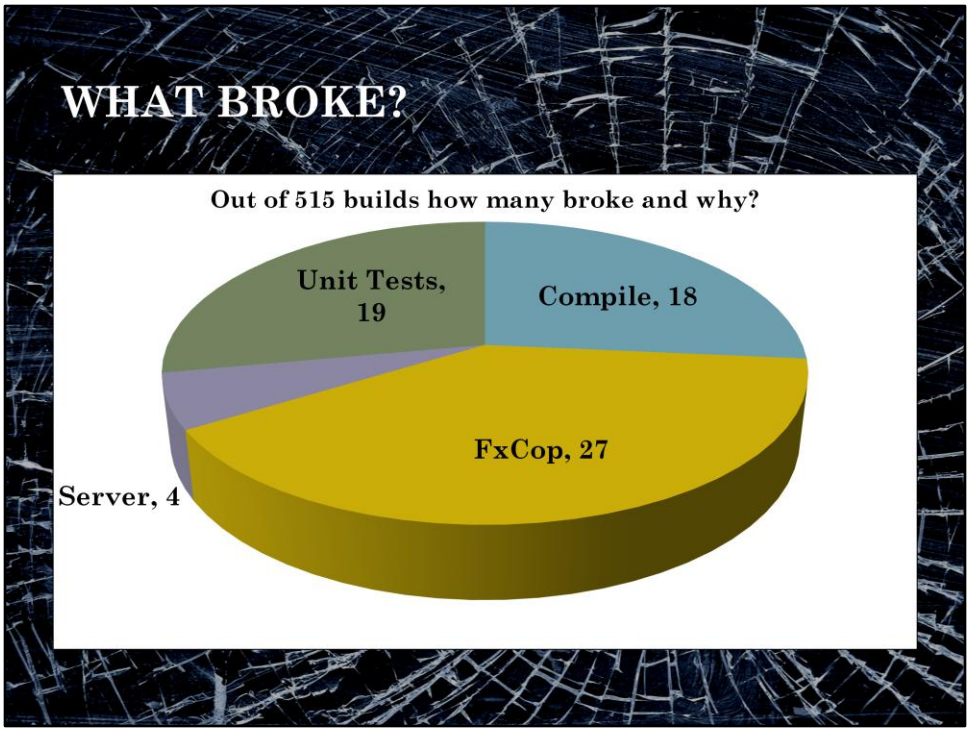


- Use your CI server to drive quality into your product
- Add additional builds for time zones

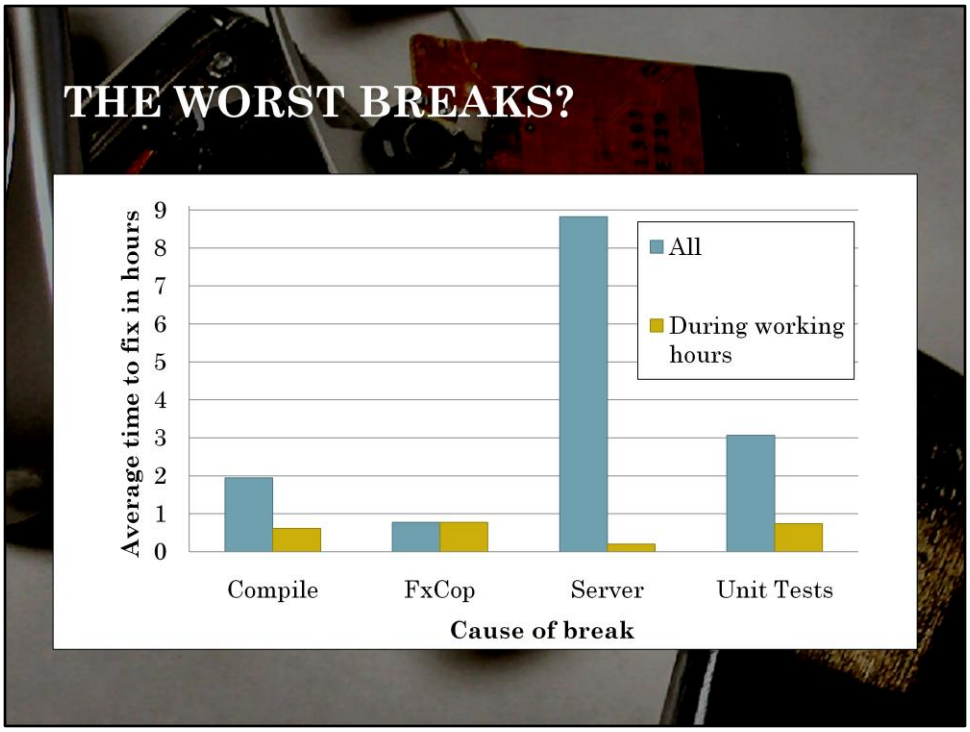


- Checkins (515 builds) and 69 build breaks (13%)
- 100 days (approx 4000 hours) of development time mid project
- On average people checked in once per day (not often enough)

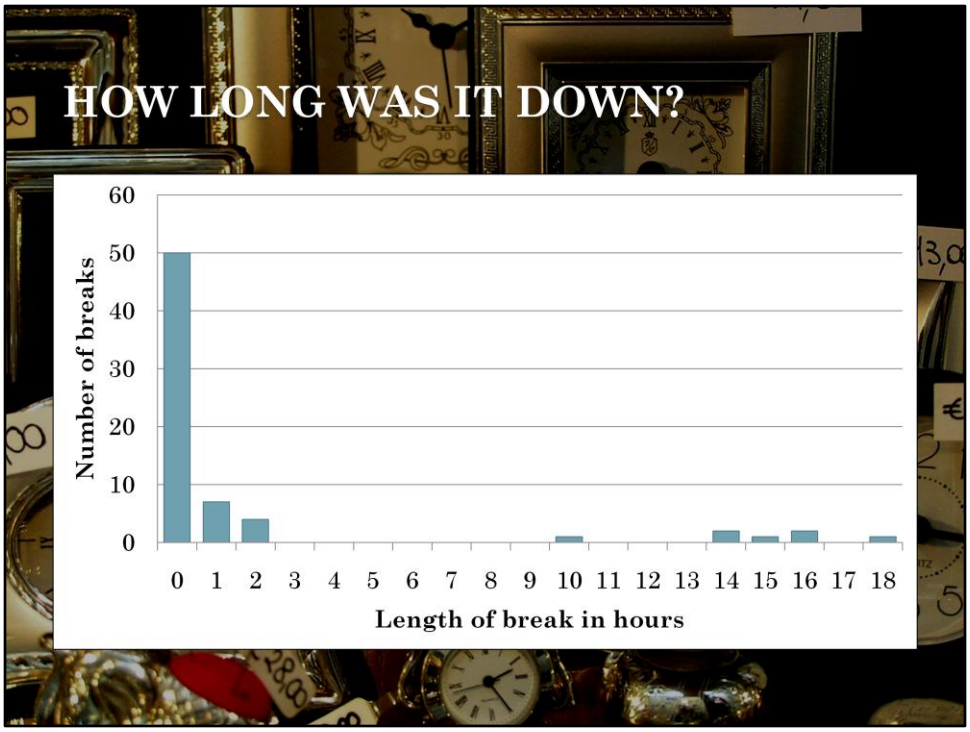
Stock photo:
http://www.sxc.hu/photo/1005800



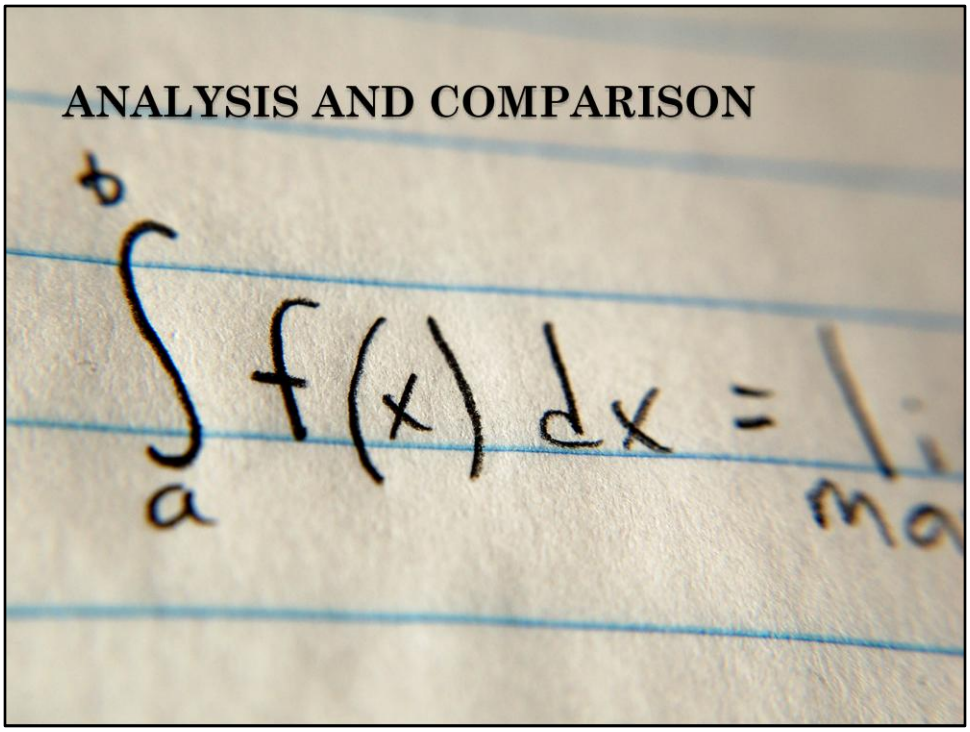
Stock photo:
http://www.sxc.hu/photo/890258



Stock photo:
http://www.sxc.hu/photo/257954



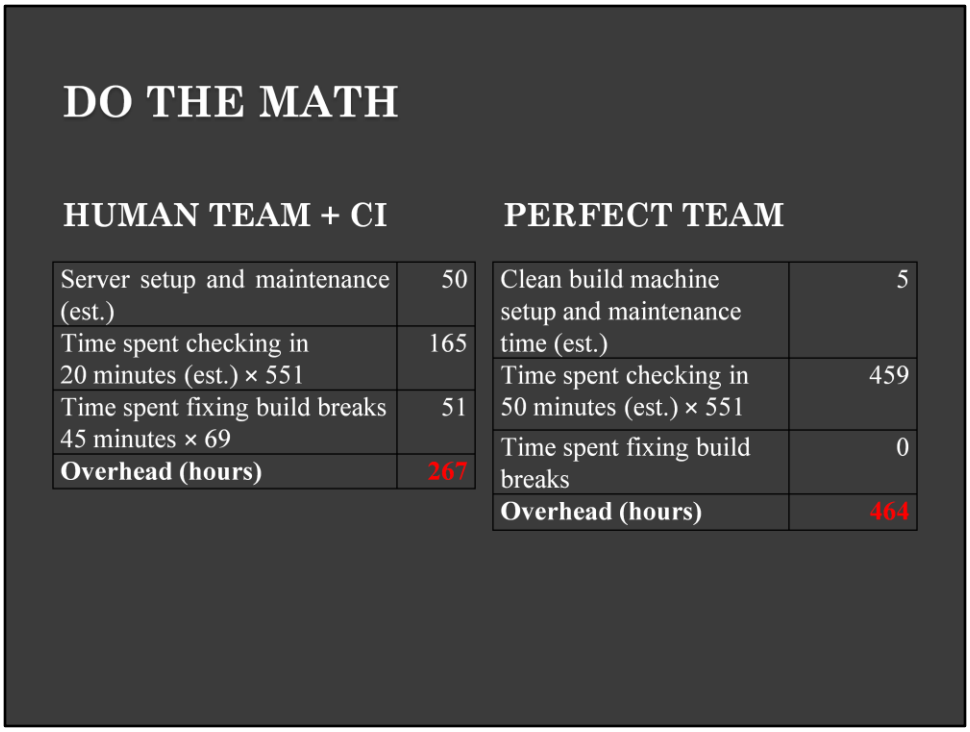
Stock photo:
http://www.sxc.hu/photo/418441



What happens if you crunch the numbers?
Ade is a Physics PhD so this is what he does given a bit of spare time and a lot of data

Exact details are in the paper and I'm not going to go into great depth here
This is an experience report not a research paper, you should judge the rigor accordingly

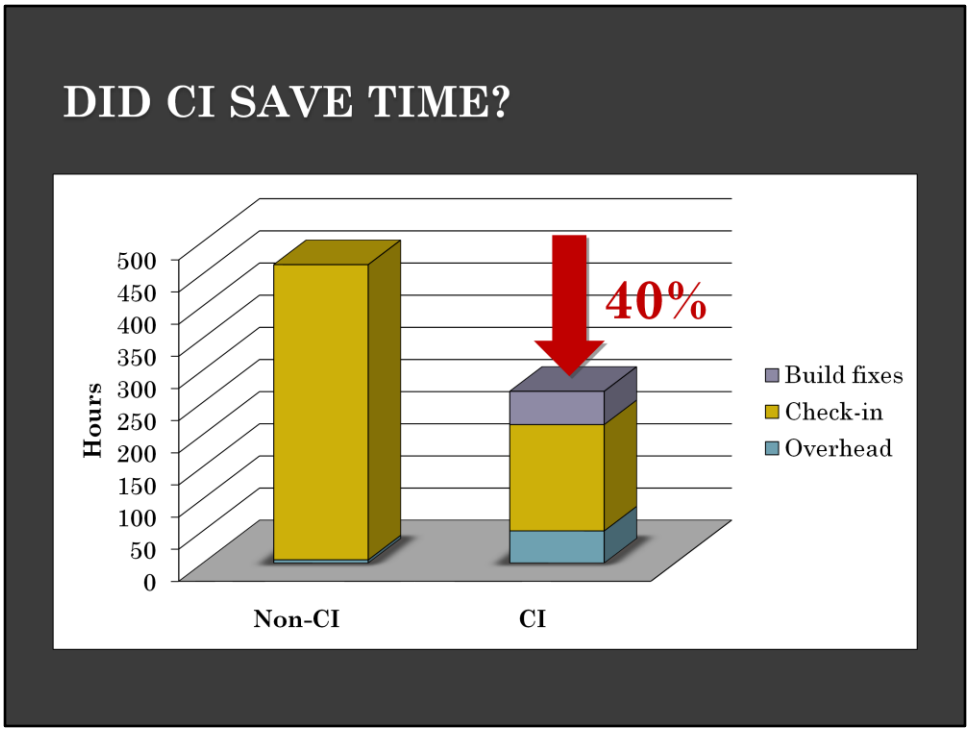
---
Stock photo:
http://www.sxc.hu/photo/943188



This is covered in more depth in the paper
Essentially:

- Take data from the TFS and CruiseControl.NET logs and figure out how long we spent fixing things
- Make estimates for typical check in time (short)
- Amount of time spent working on the CI server itself
- Compared this to a theoretical model based on the data for a perfect team who never broke the build but took much longer to check in because they did a lot of the stuff the CI server did manually
- Deliberately skewed the estimates in favor of the perfect team
- CI STILL COMES OUT AHEAD!

The perfect team can do other things to reduce their check in time, like batch up work and only check in every few days but this effects quality.



Estimated 40% saving over a perfect team not using CI

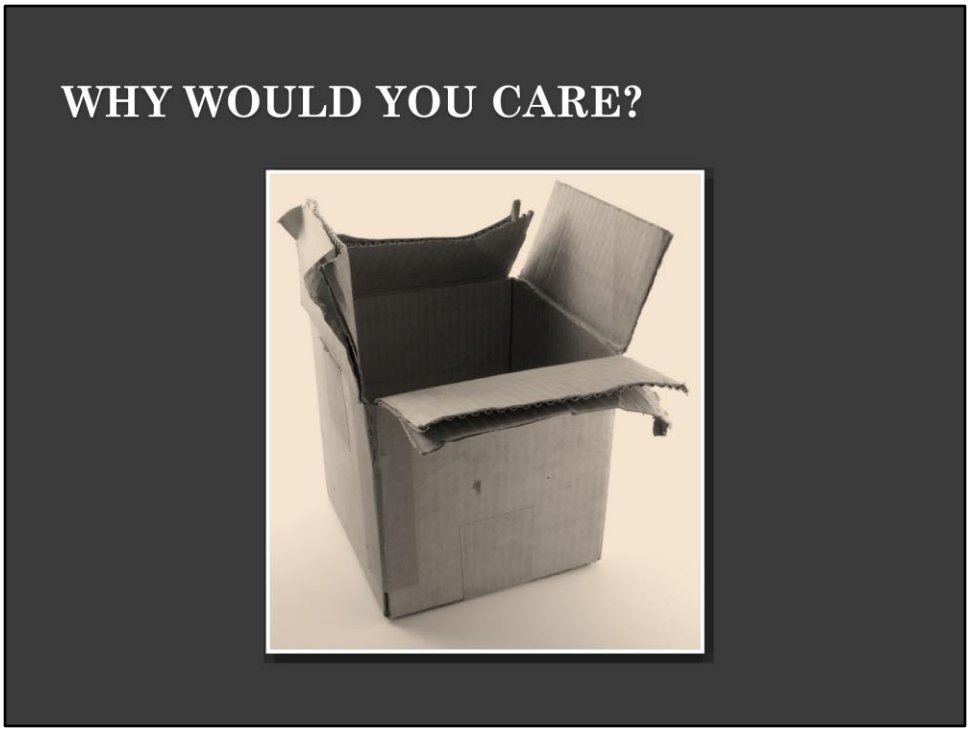
Label axes

Stock photos:
http://www.sxc.hu/photo/851553
http://www.sxc.hu/photo/535933

WHAT DID WE LEARN?

- CI saves you time
- Over spec your CI server
- Add additional builds for time zones
- Capture the data
- Add real time analysis
- Warnings are always errors!
- Don't check in and go home
- You break it you fix it

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



I see a lot of talks where we seem to have lost track of why we care about various practices

Because it's about continuously delivering value to customers in the face of changing (business) requirements.
CI lets you do this faster by reducing waste

It helps the (development) team say "Yes" to the customer!

Stock photo:
http://www.sxc.hu/photo/918252

I'm done!

Stock photo:
http://www.sxc.hu/photo/728931
http://www.sxc.hu/photo/418215
http://www.sxc.hu/photo/858531

RESOURCES

Microsoft patterns & practices
http://msdn.microsoft.com/practices

Service Factory: ME
http://msdn.microsoft.com/servicefactory

Ade Miller's blog
http://www.ademiller.com/tech/

Stock photos in this presentation
http://www.sxc.hu/

Ade blogs a lot about many of the things discussed here!

Stock photo:
http://www.sxc.hu/photo/971068

BONUS SLIDES!

DISTRIBUTED TEAMS

- Co-locate if at all possible
- Align team locations by feature (not discipline)
- Time zones add further tax
- Onshore representative for offshore team
- Pay tax associated with distribution – don't ignore it
- Improve communication where possible
- Focus on team consistency and minimize churn
- Everyone periodically visits the main site

See also…
http://www.ademiller.com/blogs/tech/2008/08/agile-2008-distributed-agile/

Some general things we've learnt about distributed agile projects
---
Stock photo:
http://www.sxc.hu/photo/728931