

Continuous Integration

Chris.Read@ThoughtWorks.com

Why?

- Catch bugs
- Get rid of merge hell
- Make teams more efficient

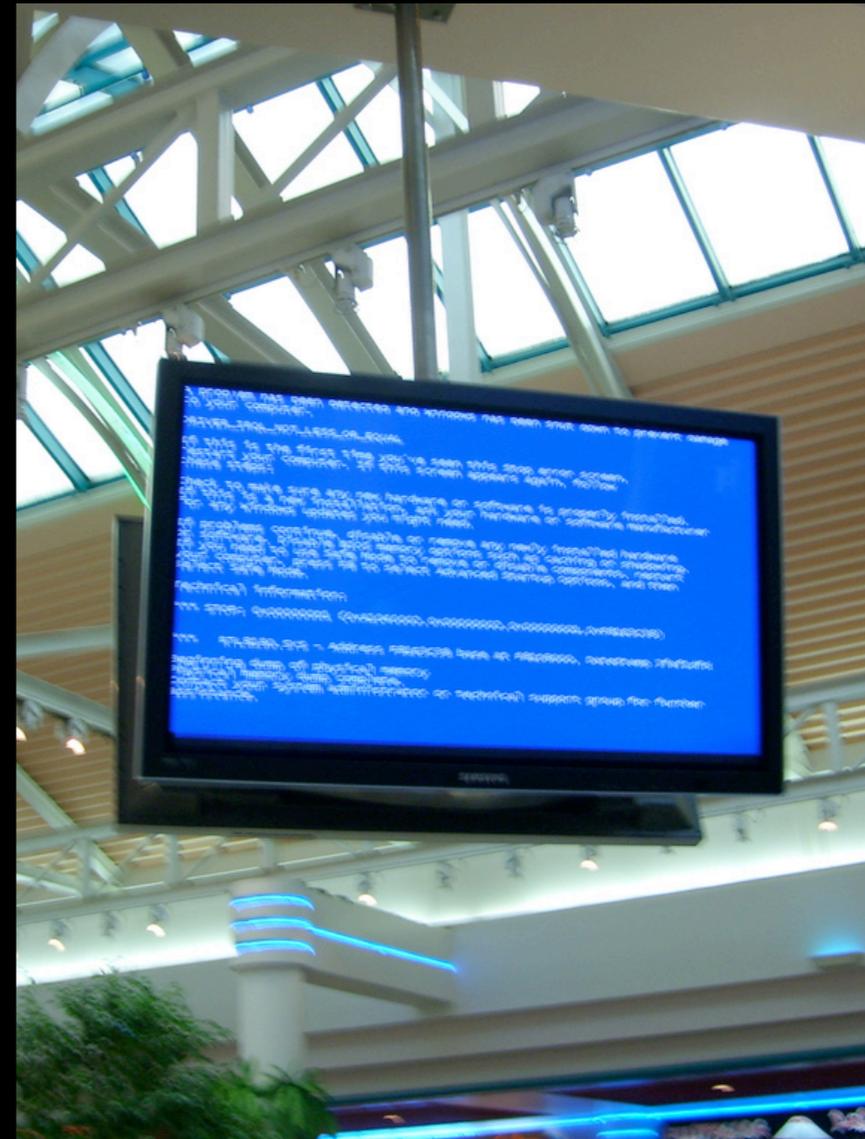


Photo by Osama ALASSIRY

How?

- Fast Feedback
- Repeatability
- Collective Ownership

Stage Activity in B2C-Website/code-compilation (What is this?)

B2C-Website/11/code-compilation - bu
anonymous

analysis	scheduled	[job details]
compile	scheduled	[job details]
setup-database	building(setup-da	details]

Elapsed: 00:00:08, ETA: 00:00:12

B2C-Website/10/code-compilation - pa
forced by anonymous

B2C-Website/9/code-compilation - pas
modified by cce

B2C-Website/8/code-compilation - pas
forced by anonymous

Core Practices

- Single Source Repository
- Automate Build
- Automate Testing
- Publish the Latest Distributable

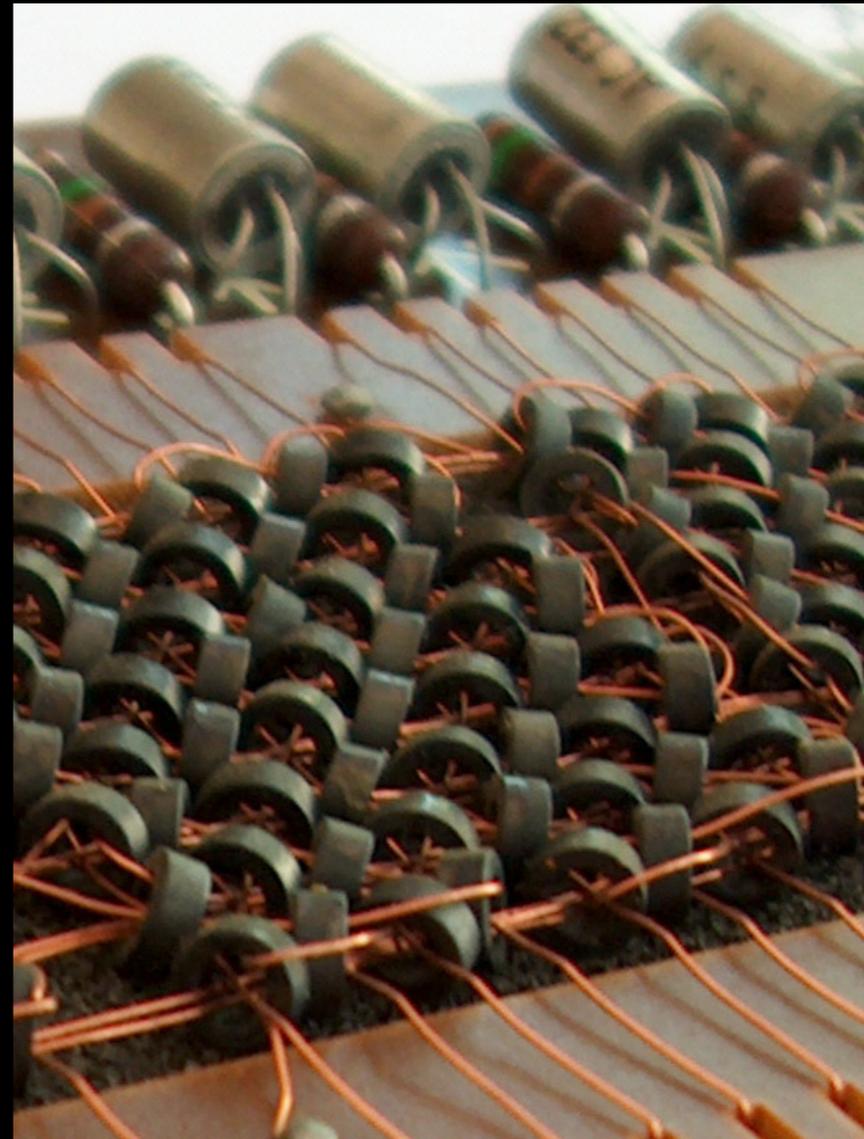


Photo by teclasorg

Single Source Repository

- Single point of truth
- Everyone's code in the same place
- NOT a branch per developer
- Shared ownership

Automate Build

- Using the IDE is not automating!
- Use a build tool
- Compile, package, test

Automate Testing

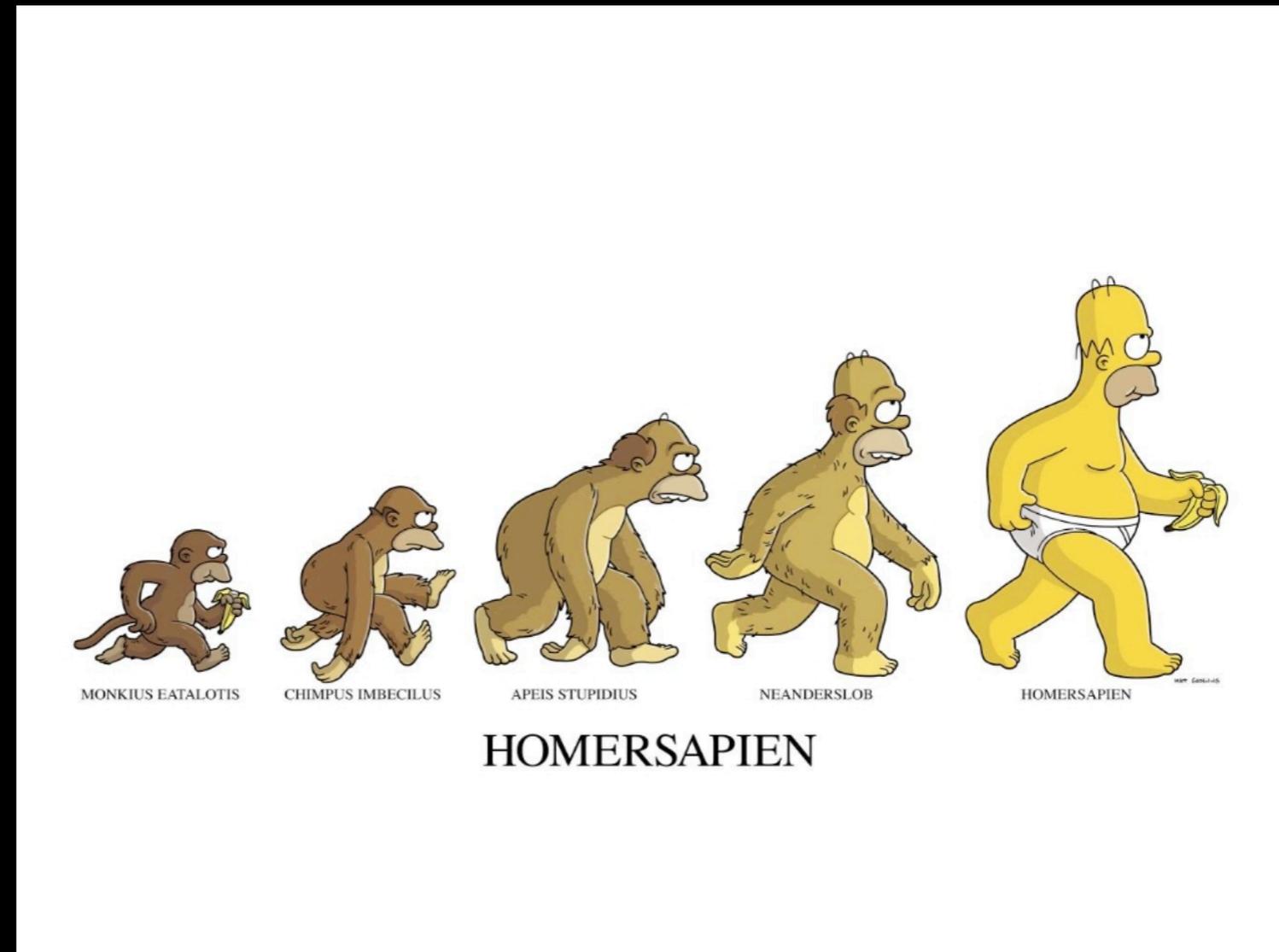
- Not just Unit Tests
- Failing tests fail the build
- Fix it if it's broken or you're wasting your time

Publish Latest Distributable

- Make it easy to get the final product
- Should only be built once
- Configuration is separate

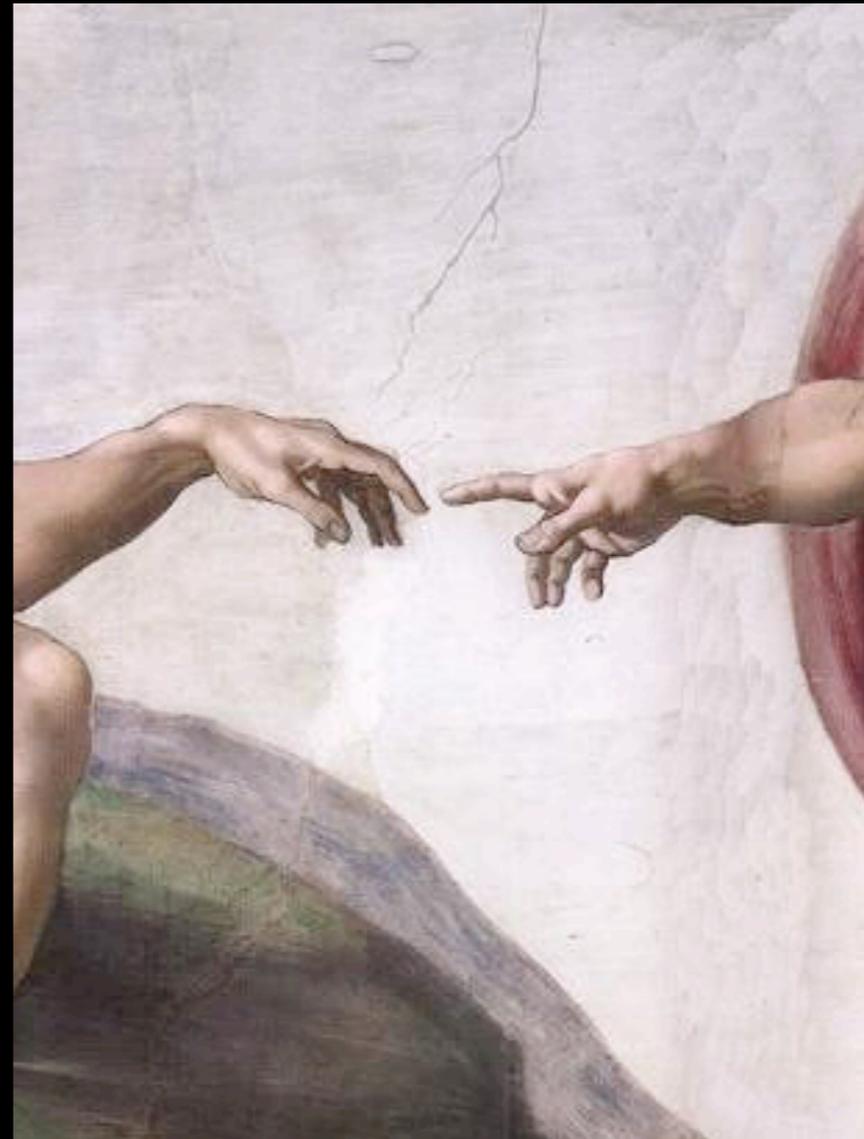
Evolution

- Practice
- Language
- Tools



Practice Evolution

- Everyone commits more often
- Every commit should build
- Test in production clone
- Keep the build fast
- Everyone sees what's happening
- Automate deployment



Commit More Often

- “At least once a day” - aim for at least once an hour
- Needs a small unit of work
- To commit cleanly you need to update first

Every Commit Builds

- It's all about fast feedback
- Small changes
- Less to merge and/or fix

Test in Production Clone

- Detect multi-threaded or cluster issues
- Tests system architecture
- Includes databases!

Keep Build Fast

- It really is all about feedback
- If things break you find out about it while it's still fresh in your mind
- Keep up with frequent check ins

Everyone Sees What's Happening

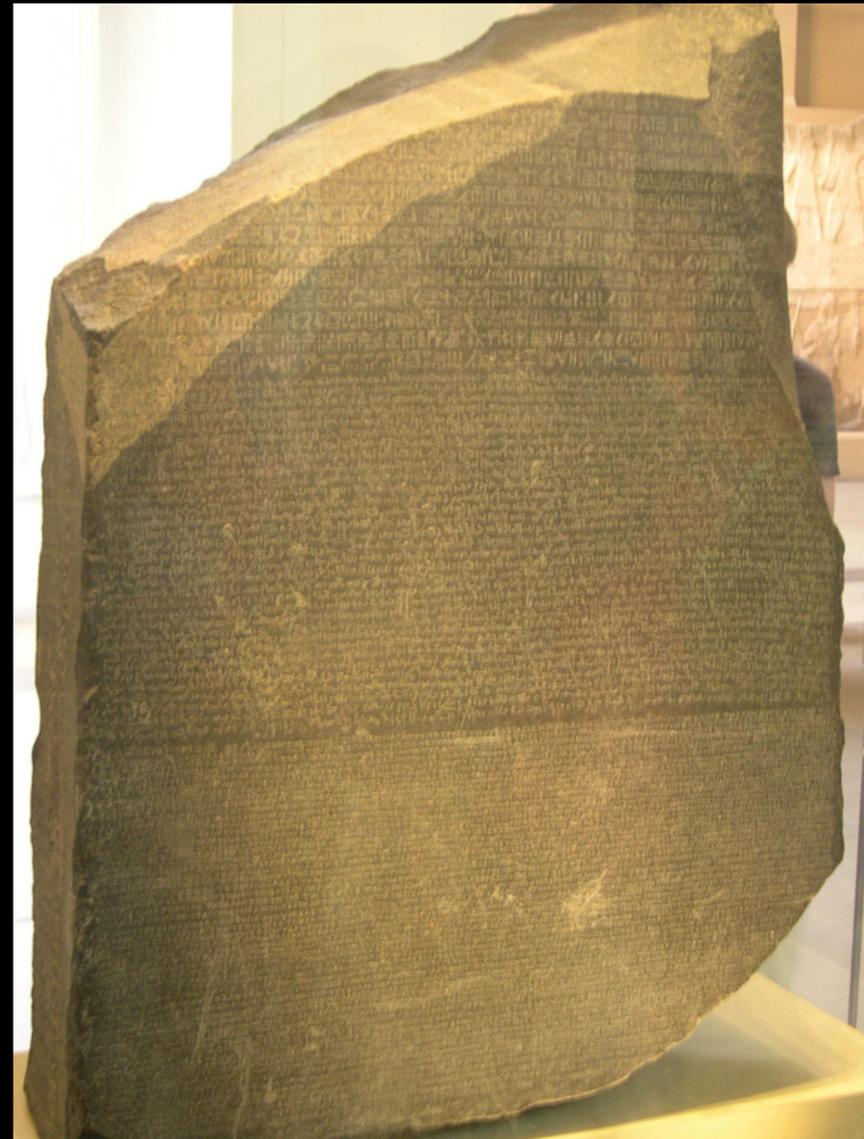
- Reduce time to fix
- No excuse to check in on broken build
- It's not about blame, it's about feedback

Automate Deployment

- Reduce Human Error
- Verify you can get it running somewhere other than “my machine”
- Test not only the code but your deployment mechanism too

Language Evolution

- Repository
- Build
- Integrate
- Test



Build

- Compile
- Link
- Run Tests
- Analyse Code
- Package
- Deploy

Integrate

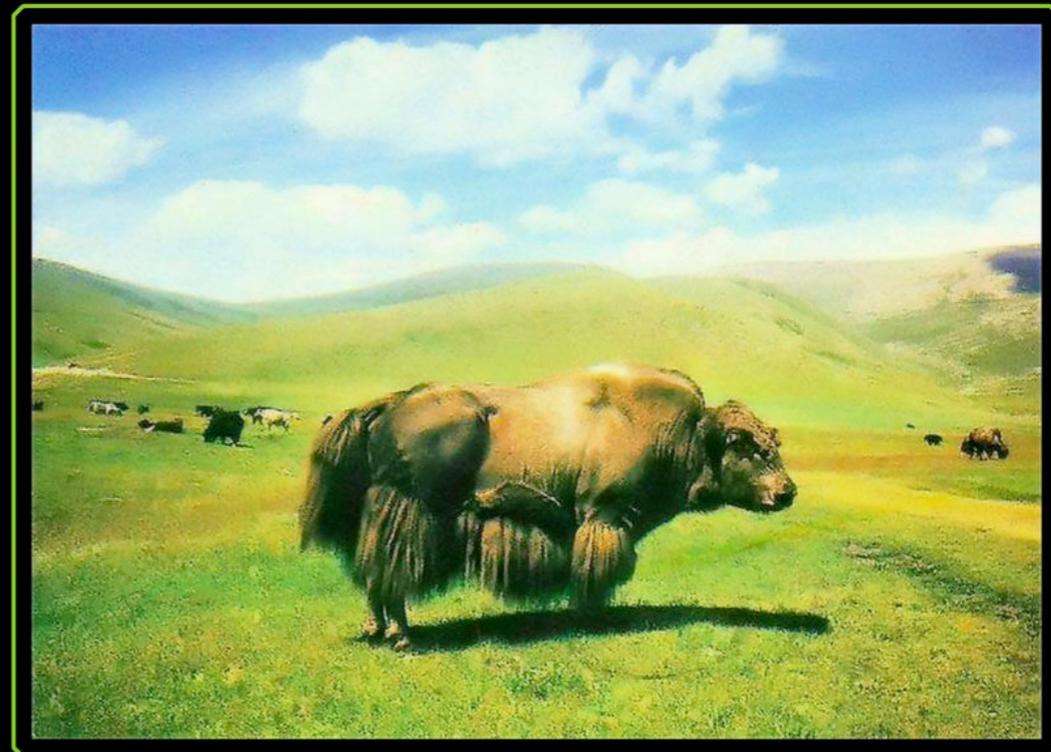
- Team members on a project getting their code to work together
- Getting projects working together
- Other systems
- Other libraries

Test

- Just test my code in isolation
- It compiles, I'm done
- Manual testing
- Unit Tests
- Integration Tests
- Acceptance Tests

Making CI Work

- Can't be done in isolation
- Pick the right tools for the job
- It's not a silver bullet



DO NOT SHAVE

A HAIRY YAK IS A HAPPY YAK

Image by LiminalMike

Tools Evolution & Trends

- Source Control
- Build Analytics
- Metrics
- Testing Tools
- CI Servers



Photo by docman

Source Control

- Inter-SCM Integration
- See which revision broke the build
- Cope with the load
- Artefact versioning

Build Analytics

- Test Code Coverage
- Bugs
- Style
- Complexity
- Visualisation

Metrics

- Interesting code stats
- Track quality trends
 - Performance
 - Code coverage
 - Build times

Testing Tools

- Test doubles - test interaction
 - Stubs
 - Mocking
- BDD
- FIT
- Selenium

CI Servers

- Agents and Build Grids
- Personal Builds
- Pipelines
- Reporting
- Ease of use

Q&A

Thanks for participating...