

# Continuous Integration

ThoughtWorks<sup>®</sup>  
The art of heavy lifting™

## Continuous Integration

Nat Pryce & Manish Shah  
{npryce, mshah}@thoughtworks.com

Copyright © ThoughtWorks, 2004

---

---

---

---

---

---

---

---

ThoughtWorks<sup>®</sup>  
The art of heavy lifting™

Traditionally, different modules are developed by different programmers

- Divide work to allow parallel development
- Integrate modules at late state of project
- Integration problems can easily knock project off schedule... or cause it to fail altogether
- *The earlier you can detect problems, the easier it is to resolve them*

2

---

---

---

---

---

---

---

---

ThoughtWorks<sup>®</sup>  
The art of heavy lifting™

In agile processes, any programmer can change any part of the code

- Module clients can change interfaces to suit their needs
- SCM tools detect conflicting commits
- Parallel commits can cause *semantic* incompatibilities
- *The earlier you can detect problems, the easier it is to resolve them*

3

---

---

---

---

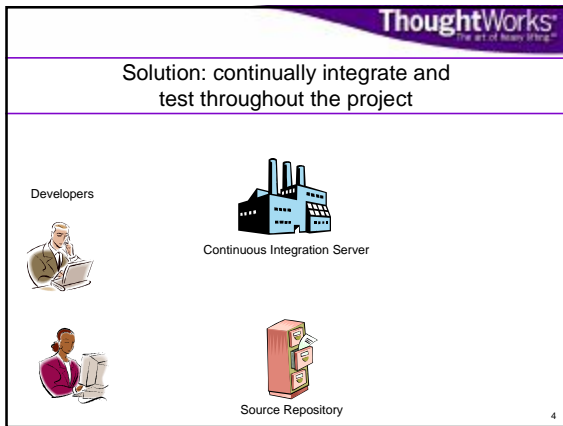
---

---

---

---

# Continuous Integration



---

---

---

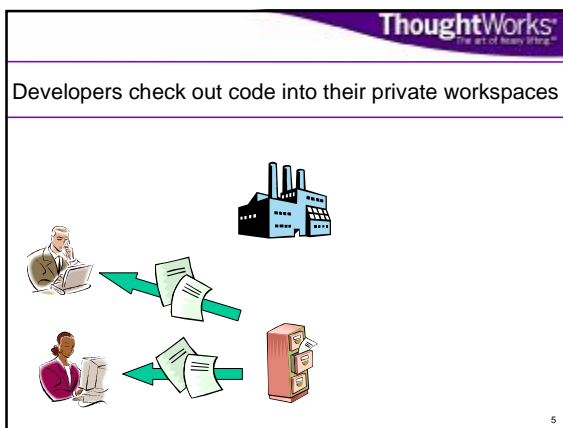
---

---

---

---

---



---

---

---

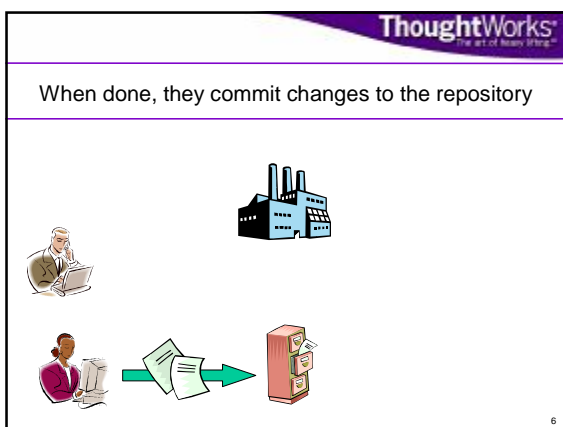
---

---

---

---

---



---

---

---

---

---

---

---

---

# Continuous Integration

ThoughtWorks<sup>®</sup>  
The art of heavy lifting<sup>™</sup>

The CI server monitors the repository and checks out changes when they occur

The diagram illustrates the first step of Continuous Integration. On the left, two people are shown working at computers. In the center, a blue factory icon represents the CI server. A green arrow points from a document icon (representing a code change) to the CI server. Another green arrow points from the CI server to a server rack icon (representing the repository).

7

---

---

---

---

---

---

---

---

ThoughtWorks<sup>®</sup>  
The art of heavy lifting<sup>™</sup>

The CI server builds the system and runs unit and integration tests

The diagram illustrates the second step of Continuous Integration. On the left, two people are shown working at computers. In the center, a blue factory icon represents the CI server with purple smoke coming out of its chimneys. A green arrow points from the CI server to a server rack icon (representing the repository).

8

---

---

---

---

---

---

---

---

ThoughtWorks<sup>®</sup>  
The art of heavy lifting<sup>™</sup>

The CI server releases deployable artefacts for QA testing

The diagram illustrates the third step of Continuous Integration. On the left, two people are shown working at computers. In the center, a blue factory icon represents the CI server with purple smoke coming out of its chimneys. A green arrow points from the CI server to a cardboard box icon (representing a deployable artefact). Another green arrow points from the cardboard box to a server rack icon (representing the repository). A person is shown pushing a shopping cart next to the server rack.

9

---

---

---

---

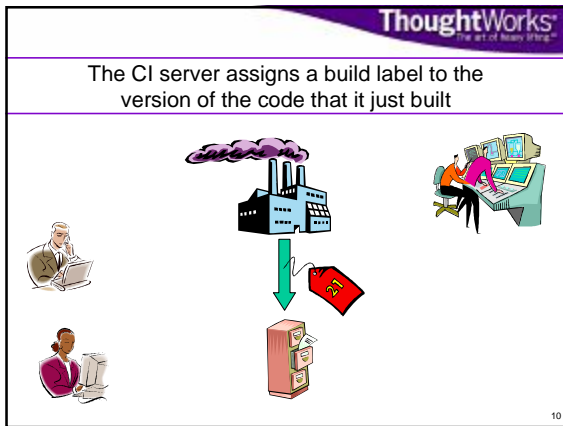
---

---

---

---

# Continuous Integration



---

---

---

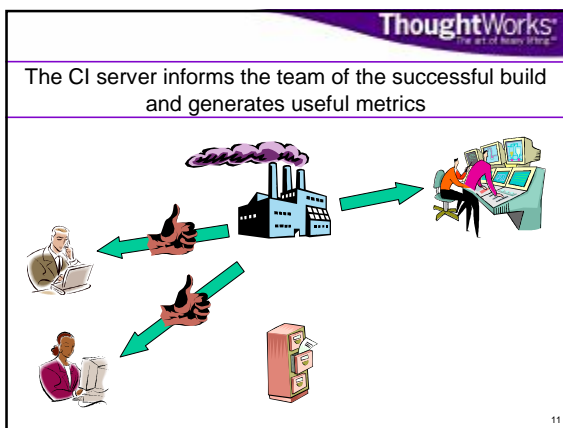
---

---

---

---

---



---

---

---

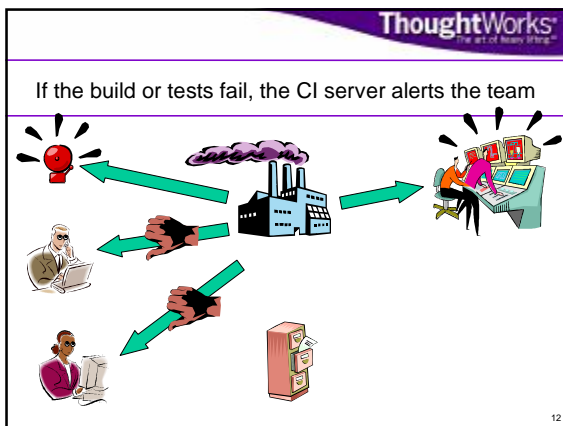
---

---

---

---

---



---

---

---

---

---

---

---

---

# Continuous Integration

ThoughtWorks<sup>®</sup>  
The art of heavy lifting<sup>™</sup>

The team fix the issue at the earliest opportunity

13

---

---

---

---

---

---

---

---

ThoughtWorks<sup>®</sup>  
The art of heavy lifting<sup>™</sup>

Continually integrate and test throughout the project

14

---

---

---

---

---

---

---

---

ThoughtWorks<sup>®</sup>  
The art of heavy lifting<sup>™</sup>

Demonstration

---

---

---

---

---

---

---

---

# Continuous Integration

ThoughtWorks<sup>®</sup>  
The art of heavy lifting™

Metrics generated by the CI server can be a useful indication of project "health"

16

---

---

---

---

---

---

---

---

ThoughtWorks<sup>®</sup>  
The art of heavy lifting™

Continuous Integration supports team communication, it does not replace it

- Supported by policies and practices
  - Check in frequently
  - Don't check in broken code
  - Don't check in untested code
  - Don't check in when the build is broken
  - Don't go home after checking in until the system builds
  - ...
- Peer pressure within team ensures policies are followed
  - No need to enforce policies from on high

17

---

---

---

---

---

---

---

---

ThoughtWorks<sup>®</sup>  
The art of heavy lifting™

"Oops, I did it again...."

• Developers undergo a forfeit for breaking the build!

18

---

---

---

---

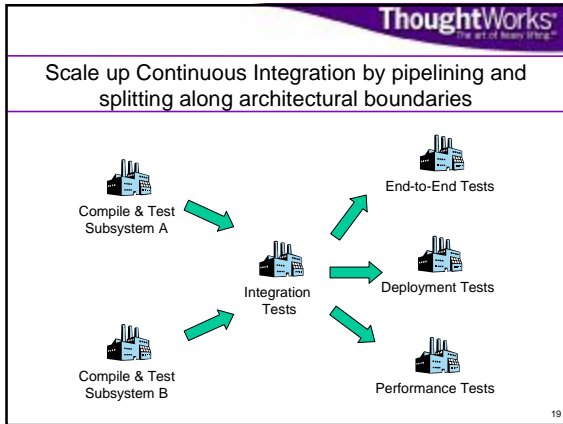
---

---

---

---

# Continuous Integration



---

---

---

---

---

---

---

---

- 
- ### Conclusion
- Continually integrate and test to detect problems early
  - Always have a deployable build
  - Generate metrics to guide project management
  - Continuous Integration is:
    - A good practice in any software development method
    - Vital for agile development
- 20

---

---

---

---

---

---

---

---

- 
- ### Continuous Integration Tools
- Cruise Control / Cruise Control .NET
    - [cruisecontrol.sourceforge.net](http://cruisecontrol.sourceforge.net)
    - [ccnet.thoughtworks.com](http://ccnet.thoughtworks.com)
  - Damage Control
    - [damagecontrol.codehaus.org](http://damagecontrol.codehaus.org)
  - Anthill OS / Anthill Pro
    - [www.urbancode.com/projects/anthill](http://www.urbancode.com/projects/anthill)
  - Tinderbox
    - [www.mozilla.org/tinderbox.html](http://www.mozilla.org/tinderbox.html)
  - Feature Comparison:
    - [damagecontrol.codehaus.org/Continuous+Integration+Server+Feature+Matrix](http://damagecontrol.codehaus.org/Continuous+Integration+Server+Feature+Matrix)
- 21

---

---

---

---

---

---

---

---

## Continuous Integration



---

---

---

---

---

---

---