Continuous Integration

a brief introduction to creating quality software continuously

presented by

Davisbase
Developing People Who Develop Software

ASPE
SDLC TRAINING
Introduction and Agenda

- Andy Painter, Davisbase Consulting
  - 15+ years in software development.
  - 5+ years working with software development teams, training, leading, and coaching Agile teams.
  - Trained and coached teams ranging from start-ups to Fortune 50 corporations.

- Agenda
  - What is Continuous Integration?
  - Evolving to Nirvana
  - Success Patterns & Challenges
  - Next steps
Continuous Integration Attitude

‣ Contrary to popular belief, continuous integration is an attitude, not a tool. It's a shared agreement by the team that:
  • When we get the latest code from the repository, it will always build successfully and pass all tests.
  • We will check in our code every two to four hours.

--- James Shore, “The Art of Agile”
Continuous Integration Practices

- Maintain a code repository
- Automate the build
- Make the build self-testing
- Everyone commits every day
- Every commit (to mainline) should be built
- Keep the build fast
- Test in a clone of the production environment
- Make it easy to get the latest deliverables
- Everyone can see the results of the latest build
- Automate deployment

Continuous Integration Defined

A process that supports teams in making small, working incremental changes to software. Continuous Integration is an automated process that provides feedback and supports:

- Transparency
- Consistency
- Continuous Builds
- Continuous Inspections
- Continuous Testing
- Continuous Deployment
Typical Development

- Large grained commits or Version control as a backup
- Independent development
- IDE driven manual builds
Scripted Build
Integrated Build
Continuous Builds

- Published Artifacts & Test Results
- Detect Changes
- Integration Build Machine
- Scripted Build
- Transparency & Continuous Builds
- Server

Check Out Project

Scripted Build
Continuous Inspections
(Static Analysis)
Automated Unit Testing

Published Artifacts & Test Results → Detect Changes

Unit Tests (i.e., xUnit) → Static Analysis

Scripted Build
Integration Build Machine
Transparency & Continuous Builds
Continuous Inspections
Automated Unit Testing
Automated Acceptance Testing

- Published Artifact & Test Results
- Detect Changes
- Unit Tests (i.e., xUnit)
- Static Analysis
- Scripted Build

CI
Integration Build Machine
Server

- Scripted Build
- Integration Build Machine
- Transparency & Continuous Builds
- Continuous Inspections
- Automated Unit Testing
- Automated Acceptance Testing
Automated Deployment

- Scripted Build
- Integration Build Machine
- Transparency & Continuous Builds
- Continuous Inspections
- Automated Unit Testing
- Automated Acceptance Testing
- Automated Deployment

CI
Integration Build Machine
Server

- Deployment
- Detect Changes
- Check Out Project
- Published Artifact & Test Results
- Acceptance Tests
- Unit Tests (i.e., JUnit)
- Static Analysis
Functional Testing
Through the Eye of Testing

- Manual & Exploratory Testing
- GUI Tests
- Other Tests
- Unit Tests

Diagram showing the integration of build and CI/CD processes.
### Sample Tools for CI

<table>
<thead>
<tr>
<th>Tools</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>QTP/Selenium</td>
<td>Functional Tests</td>
</tr>
<tr>
<td>FIT, Fitnesse, BDD frameworks</td>
<td>Acceptance Tests</td>
</tr>
<tr>
<td>Cobertura, NCover</td>
<td>Code Coverage Metrics</td>
</tr>
<tr>
<td>Junit, Nunit, xUnit</td>
<td>Unit Testing</td>
</tr>
<tr>
<td>FxCop, CPD, PMD</td>
<td>Static Analysis</td>
</tr>
<tr>
<td>Java, MSBuild</td>
<td>Execution/Compilation</td>
</tr>
<tr>
<td>Ant, Nant</td>
<td>Build Scripting</td>
</tr>
<tr>
<td>Hudson, CC, Ant Hill Pro</td>
<td>Continuous Integration</td>
</tr>
</tbody>
</table>
Value Proposition

› Reduce risk
› Automate frequent manual processes
› Improve transparency & visibility
› Generate deployable software
Symbiotic Agile Engineering Practices

- Automated Unit & Integration Tests
- Coding Standards
- Refactoring
- Collective Code Ownership
- Automated Acceptance Tests
Building Successful Habits

- Commit Often
- Don't commit broken code
- Fix broken builds immediately
- Write Automated Developer Tests
- Fail the Build if any test or inspection fails
- Run Local Builds before commits
- Don't check out code from a broken build
Fighting Resistance

- Another Process & System?
- More change?
- Cost?
- Our developers should already be doing this, right?
Next Steps

- Find experts that can point you in the right direction.
- Recognize that automating is an essential foundation that supports agile software teams.
- It takes time to get good at anything, Continuous Integration is no exception, but the rewards are well worth it.
- Getting started is easier than you might think.
- Try something, see what works
About Us

- A leading, national provider of Agile training and coaching services.
  - Since 2007, over 300 classes taught to over 5000 people nationwide.
  - Training, coaching, and consulting work for over 50 Fortune 500 Clients.
- Experienced; “in the trenches” doing what we teach
  - Extensive experience with all aspects of software development and project lifecycle across multiple industries and verticals.
  - Every title and role from Developer, QA, Project Manager through Sr. Leadership.
  - Average of 14+ years of applied IT Leadership experience.
Final Thoughts...

- “Simplicity does not precede complexity, it follows it.”
  - Alan Perlis

- “Whether your next project is a success or a failure is not a matter of chance, it is a matter of choice.”
  - A wise Agile coach and trainer
Questions and Answers

› Your questions, my answers.