

Automating Development Processes with Open Source Tools

Ian Robinson, Engage Software irobinson@engagesoftware.com



What is Continuous Integration?

- > Team members integrating work **frequently**
- Automate build and build related tasks
 - Important but tedious tasks



Why Continuous Integration?

- > Reduce integration problems
- > Find errors early
- Enforce best practices
- Increase team communication/awareness
- > Develop better software faster



Local Environment

- > Your machine
 - Application development environment
 - This is where the magic happens.
 - Source control client
 - Backup code
 - Collaborate with your team
 - Branch your project



Source Control Server

- Assuming (client/server model)
- > The definitive source for your app



Build Server

- > Can be the same as source control server
- > Clean and consistent environment
 - Not exposed to frequent changes like a developer's environment



Build Server

- > Fundamental features
 - Builds application
 - Report success or failure
 - Report detailed built output



Build Server

- Additional (optional) tasks
 - Email notifications
 - Code metrics / static analysis
 - Run automated tests
 - Deploy application



Without Continuous Integration...

How do you push changes to the test environment?

- How do you notify QA?
- A team member checks in bad code right before they leave. Ah crap...



So how do we do it?

- > We have lots of options
- > I just happened to pick:
 - Subversion for source control
 - Cruise Control .NET for CI



Source Control with Subversion

- > Open Source Development
- Large and Active Community
- Client Server Model
- Well established
- Wide community support and compatibility

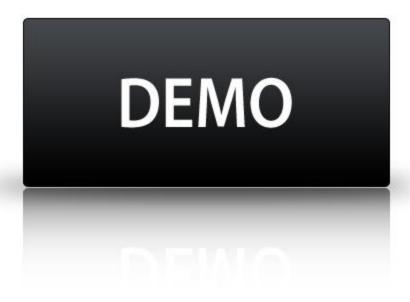


Setting up Subversion

- Svn1clicksetup
 - Installs SVN client and server
 - Installs TortoiseSVN
- > TortoiseSVN
 - Windows Shell Extensions
- > AnkhSVN
 - Visual Studio Integration



Subversion + Visual Studio Project





What is Cruise Control .NET?

- > Automated integration server
- Monitoring and reporting
 - Monitors project repository
 - Integration builds launched on commit/check-in
 - Developer notification
 - Integration with other tools to perform additional tasks



Setting up the basics

- Monitor source control repository
- > Trigger a build event



Rudimentary CruiseControl.NET Setup



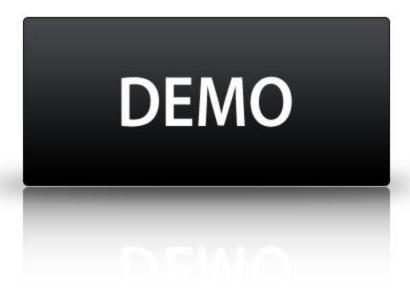


Custom project build scripts

- > Several options
 - MSBuild or NAnt?
- Common tasks
 - Packaging
 - Deployment



Custom Build Script





CC.NET Additional Integration Options

- Source Control
 - Subversion
 - o CVS
 - Visual Source Safe
 - SourceGear Vault
 - StarTeam
 - + many more

- > Build Tools
 - NAnt
 - Visual Studio
 - MSBuild
 - FinalBuilder



CC.NET Additional Integration Options

- Test + Coverage
 - NUnit
 - MSTest
 - NCover
 - MBUnit
 - FitNesse

- > Metrics
 - FxCop
 - o Vil
 - Simian
 - NDepend
 - Gendarme



SVN Additional Integration Options

- > Software collaboration sites
 - Codeplex
 - Host and collaborate on open source projects
 - Unfuddle
 - Track bugs and feature requests
 - Subversion repository hosting



Resources

- Continuous Integration
 - Continuous Integration (Martin Fowler)
 - Why use Continuous Integration? Beginner's Overview (Scott C Reynolds)
- Subversion
 - Subversion Home Page
 - svn1clicksetup Home Page
 - TortoiseSVN Home Page
 - AnkhSVN Home Page
 - Setting up Subversion (Rick Strahl)
 - Subversion Quick start for .NET Developers (Polymorphic Podcast)
- Cruise Control .NET
 - CruiseControl.NET Home Page (Great Documentation!)
 - CruiseControl with MSBuild, MSTest, Deveny (Gaurav R)
- > NAnt
 - NAnt Home Page



WE WOULD LIKE TO THANK OUR SPONSORS































