


IBM Software Group	
<h2>Agile SCM: Realising Continuous Integration</h2>	
Kevin Lee – Technical Consultant kevin.lee@uk.ibm.com	
	<small>© 2005 IBM Corporation</small>


IBM Software Group	
<h2>Agenda</h2>	
<p>§ What is Continuous Integration?</p> <p>§ Continuous Integration in Context</p> <ul style="list-style-type: none">– As part of a “normal” software build and release process <p>§ Realising Continuous Integration</p> <ul style="list-style-type: none">– Introduction to CruiseControl <p>§ Continuous Integration in Practice</p> <ul style="list-style-type: none">– Examples and suitability <p>§ References</p>	
2	<small>© 2005 IBM Corporation</small>

IBM Software Group 

Agenda


- § **What is Continuous Integration?**
- § **Continuous Integration in Context**
 - As part of a “normal” software build and release process
- § **Realising Continuous Integration**
 - Introduction to CruiseControl
- § **Continuous Integration in Practice**
 - Examples and suitability
- § **References**

3 | © 2005 IBM Corporation

IBM Software Group 

The Two Sides of Software Configuration Management

Governance versus Productivity



Governance
Ensuring Compliance and Control


Productivity
Delivering Developer Value

↓ ↓

Automation
Continuous Integration

- § *Process: CMM(I), ISO, SPICE*
- § *Control: Outsourcing, Offshoring*
- § *Regulatory: Basle II, Sarbanes Oxley*
- § *Reducing build times*
- § *Enabling parallel development*
- § *Increasing confidence*

4 | © 2005 IBM Corporation


IBM Software Group 

What is Continuous Integration?

... or rather what it is NOT!

- § **Continuous!**
 - It's really refers to an activity that is frequent or continual
- § **Dependent on the adoption of Agile Development**
 - although it enables such an approach
- § **A tool or set of tools**
 - although tools are required to implement it
- § **Difficult**
 - hey, I can do it!
- § **NEW!**
 - and I've been sort of doing it for years without knowing it!

5 | © 2005 IBM Corporation

IBM Software Group 

What is Continuous Integration?


... let's try some definitions

- § **"An important part of any software development process is getting reliable builds of the software. Despite it's importance, we are often surprised when this isn't done. We stress a fully automated and reproducible build, including testing, that runs many times a day. This allows each developer to integrate daily thus reducing integration problems."**

Martin Fowler and Matthew Foemmel, Continuous Integration
- § **"The macro process of object-oriented development is one of "continuous integration." ... At regular intervals, the process of "continuous integration" yields executable releases that grow in functionality at every release. ... It is through these milestones that management can measure progress and quality, and hence anticipate, identify, and then actively attach risks on an ongoing basis."**

Grady Booch, Object-Oriented Analysis and Design with Applications

6 | © 2005 IBM Corporation

IBM Software Group 

What is Continuous Integration?

... how about something more tangible

§ **A central repository for all members of a team, containing:**


- the latest code (at least)
- the latest executables

§ **An automated process for building AND testing all project assets**

- that can be run many times a day
- and that is self fulfilling

OK, but how does this fit into the rest of SCM?

7 | © 2005 IBM Corporation

IBM Software Group 

Agenda

§ **What is Continuous Integration?**

§ **Continuous Integration in Context**

- As part of a “normal” software build and release process

§ **Realising Continuous Integration**

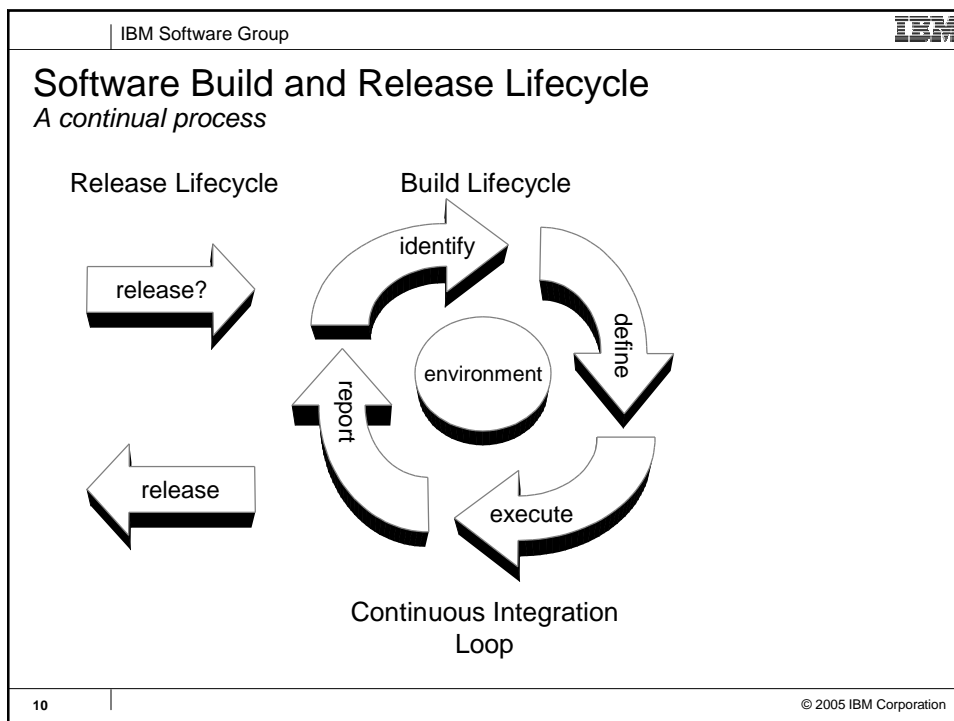
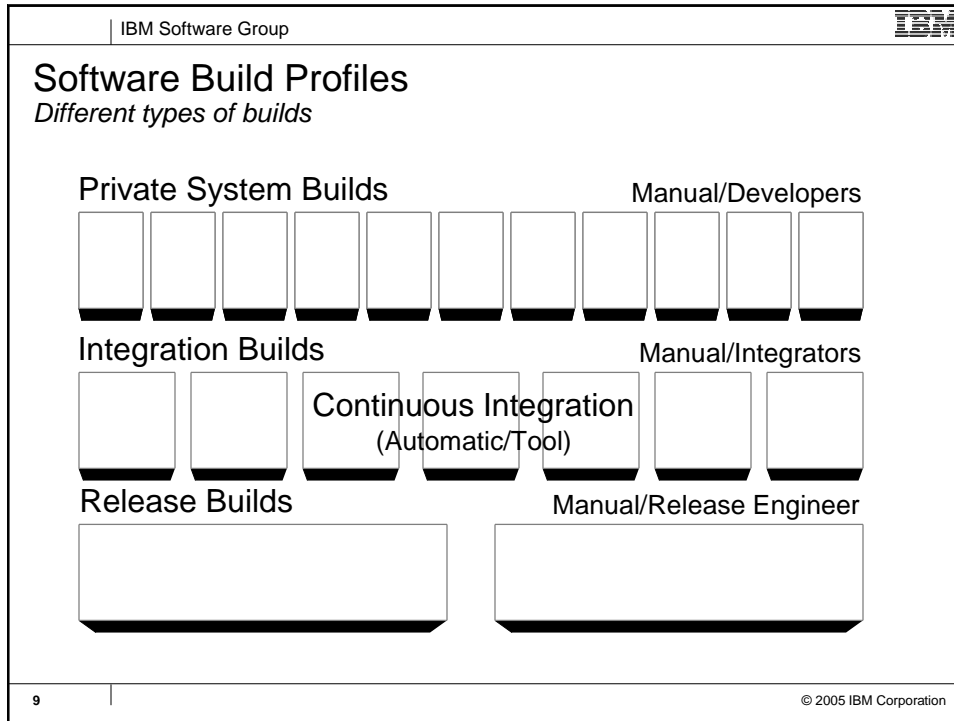
- Introduction to CruiseControl

§ **Continuous Integration in Practice**

- Examples and suitability

§ **References**

8 | © 2005 IBM Corporation



IBM Software Group

Software Build and Release Lifecycle Tools

In the Java domain

The diagram illustrates a continuous cycle of software build and release. At the center is 'CVS environment'. Four arrows form a circle around it, labeled 'identify', 'define', 'execute', and 'report'. Surrounding this cycle are logos for tools: ClearQuest, BugZilla, Apache Maven Project, THE APACHE ANT PROJECT, JUnit.org, and CruiseControl.

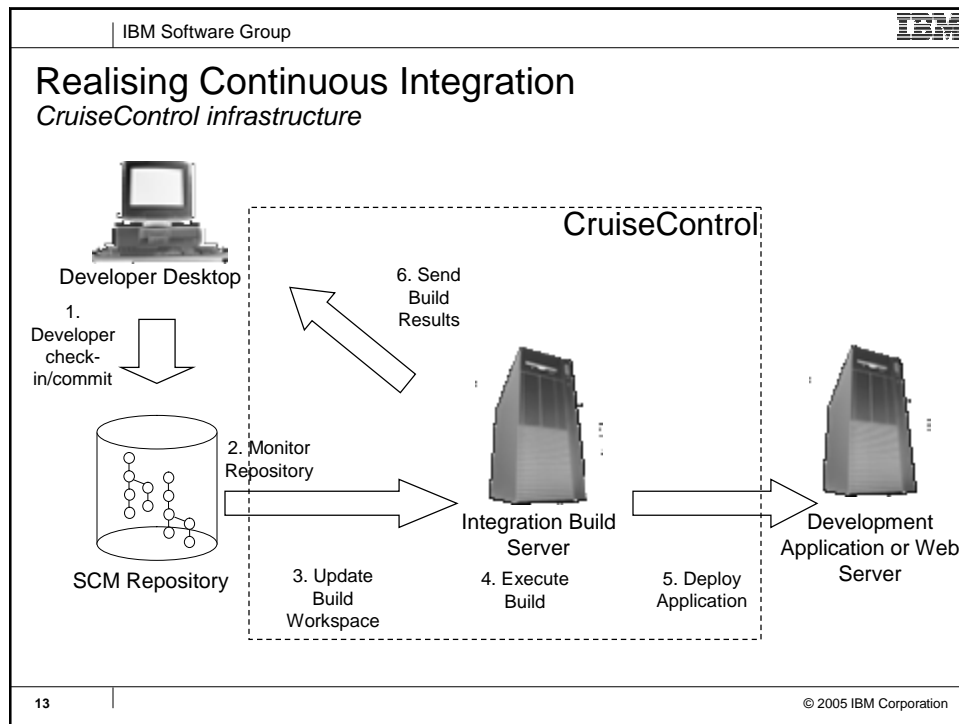
11 | © 2005 IBM Corporation

IBM Software Group


Agenda

- § **What is Continuous Integration?**
- § **Continuous Integration in Context**
 - As part of a “normal” software build and release process
- § **Realising Continuous Integration**
 - Introduction to CruiseControl
- § **Continuous Integration in Practice**
 - Examples and suitability
- § **References**

12 | © 2005 IBM Corporation



- IBM Software Group
- ## Realising Continuous Integration
- CruiseControl process*
- § **Monitors an SCM repository for changes**
 - CVS = check for commits in the repository since last build time
 - ClearCase = check for check-ins on integration branch
 - § **If changes are found, then on a schedule start the build**
 - § **Build your application**
 - through your existing Ant or Maven scripts
 - § **Run your JUnit Test suite**
 - § **Report on the build results**
 - send formatted email notifications
 - publish results to a website
 - § **(Optionally) publish the application**
 - § **Configuration is through a central XML file**
- 14 | © 2005 IBM Corporation


IBM Software Group 

Realising Continuous Integration

The CruiseControl schedule

- § **Schedule is the most important definition in Continuous Integration**
 - how long can you wait before being informed of integration errors?
- § **Continuous Integration means building many times a day, every 20 minutes, every hour...**
 - but if there is nothing checked in/committed, there will be nothing to build
 - you need to find your own “project rhythm”
- § **However, even if you only want to do a nightly build you can still get benefits from the CruiseControl toolset.**

15 | © 2005 IBM Corporation


IBM Software Group 


Realising Continuous Integration

What is a successful build?


- § **When is your build successful?**
 - When it compiles?
 - When all the unit-tests have run?
 - When it has been deployed?
- § **In fact, every failure is a success**
 - You have exposed a potential problem – early!

16 | © 2005 IBM Corporation

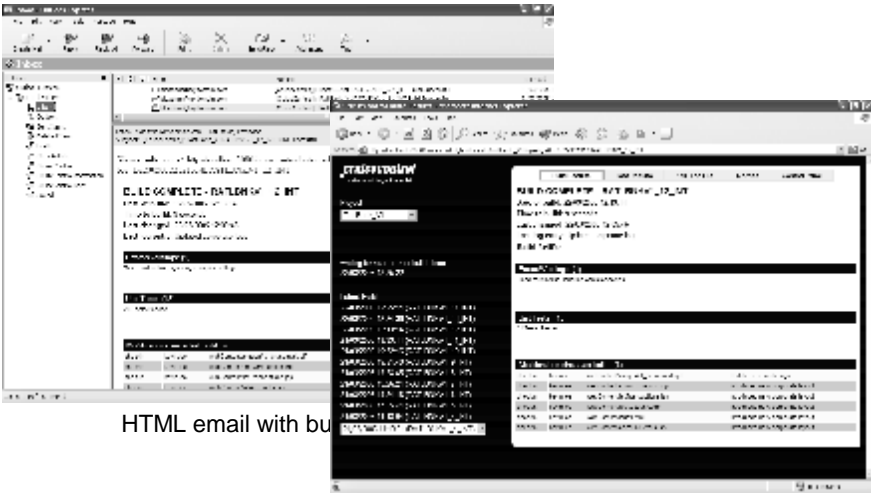
IBM Software Group 



17 © 2005 IBM Corporation

IBM Software Group 


CruiseControl Screenshots



HTML email with build results

Build Results web


18 © 2005 IBM Corporation

IBM Software Group 

Agenda

- § **What is Continuous Integration?**
- § **Continuous Integration in Context**
 - As part of a “normal” software build and release process
- § **Realising Continuous Integration**
 - Introduction to CruiseControl
- § **Continuous Integration in Practice**
 - Examples and suitability
- § **References**

19 | © 2005 IBM Corporation

IBM Software Group 

Continuous Integration in Practice

Will it work on my project?

- § **Continuous Integration is only one aspect of an overall process, for it to work best, you need to:**
- § **Plan iteratively**
 - schedule regular releases with evolving levels of functionality (CRs)
 - be wary of inflexible Change Control Boards!
- § **Implement incrementally**
 - identify and implement small work tasks
 - refactor if necessary!
- § **Report proactively**
 - identify exactly the contents (CIs) of any build, in both file and content
 - automate reports!

20 | © 2005 IBM Corporation

IBM Software Group

Continuous Integration in Practice

Will it work on my project?

- § **Proven for small-medium sized projects**
 - Short integration cycles require a comprehensive unit test suite for validation
 - Large projects can benefit from “programme” based approach with multiple continuous integration streams, one for each system or component
- § **Works best with single mainline development**
 - Latest stream, maintenance stream, patch stream at most
 - Developing multiple releases in parallel can be problematic, streams tend to live longer and changes can be potentially integrated to multiple releases
- § **You need a development workspace**
 - i.e. ClearCase stream/view, CVS checked out workspace, so that ground does not change under your feet


21 | © 2005 IBM Corporation

IBM Software Group

Agenda

- § **What is Continuous Integration?**
- § **Continuous Integration in Context**
 - As part of a “normal” software build and release process
- § **Realising Continuous Integration**
 - Introduction to CruiseControl
- § **Continuous Integration in Practice**
 - Examples and suitability
- § **References**

22 | © 2005 IBM Corporation


IBM Software Group 

References

Supporting and alternate tools

- § **CruiseControl**
<http://cruisecontrol.sourceforge.net>
- § **Apache Ant**
<http://ant.apache.org>
- § **Apache Maven**
<http://maven.apache.org>
- § **JUnit**
<http://www.junit.org>
- § **ClearCase**
<http://www-306.ibm.com/software/awdtools/clearcase>
- § **CVS**
<http://www.cvshome.org>

23 | © 2005 IBM Corporation


IBM Software Group 

References

Supporting information

- § **Continuous Integration (according to Martin Fowler)**
<http://www.martinfowler.com/articles/continuousIntegration.html>
- § **The Continuous Integration Wiki**
<http://c2.com/cgi/wiki?ContinuousIntegration>
- § **CM Crossroads**
<http://www.cmcrossroads.com>
- § **The Buildmeister**
<http://www.buildmeister.com>

24 | © 2005 IBM Corporation

IBM Software Group	
Summary	
<i>What have we learned?</i>	
<ul style="list-style-type: none">§ Continuous Integration is a variation on an old theme§ It works best with an incremental approach to software development§ However, the tools can be used for “traditional development”<ul style="list-style-type: none">– Even if you just want to implement an automated nightly build process§ Using them can potentially yield increased productivity<ul style="list-style-type: none">– Immediate feedback via email if the build breaks§ There are no limits to the level of automation that can be achieved<ul style="list-style-type: none">– Once you have an automated build and testing, you can automate baselining, reporting, deployment ...§ It makes software development fun!	
25	© 2005 IBM Corporation