

Agile Development Infrastructure

MUUG – June 13, 2006

Steve Moffat

Agile Development

- Iterations of 3 – 4 weeks
- Emphasize real time communication
- Measured by working software
- Sometimes criticized as undisciplined

Agile Development

Agile methods are adaptive rather than predictive. Engineering methods tend to try to plan out a large part of the software process in great detail for a long span of time, this works well until things change. So their nature is to resist change. The agile methods, however, welcome change. They try to be processes that adapt and thrive on change, even to the point of changing themselves.

Agile methods are people-oriented rather than process-oriented. The goal of engineering methods is to define a process that will work well whoever happens to be using it. Agile methods assert that no process will ever make up the skill of the development team, so the role of a process is to support the development team in their work.

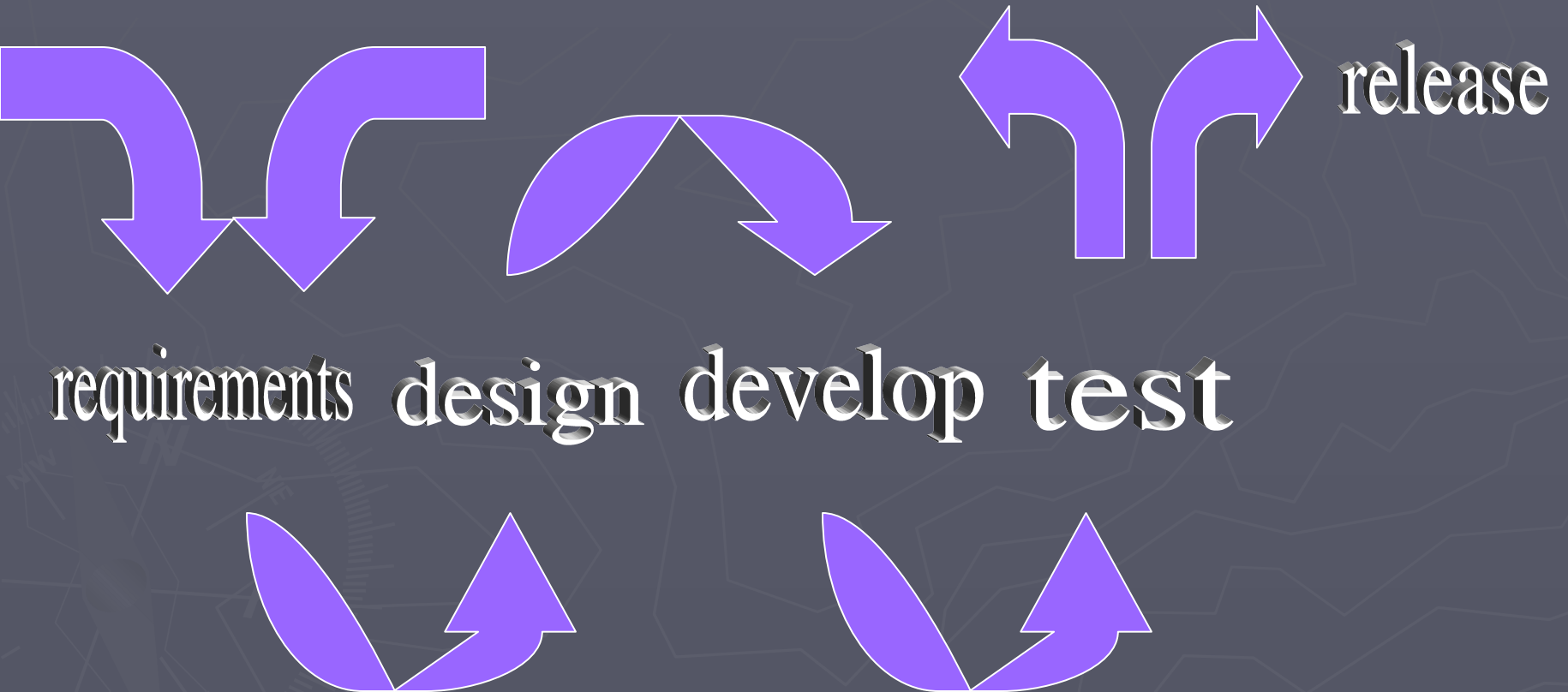
Martin Fowler

Agile Development

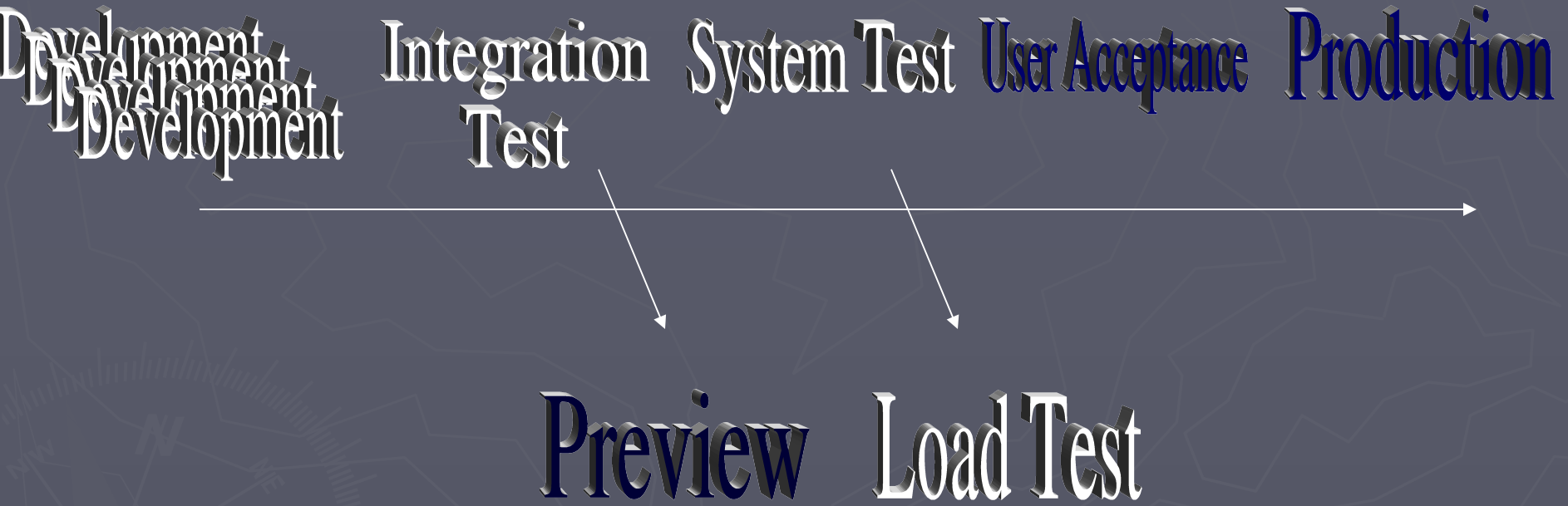
*Productivity of the
development team is key*

<http://agilemanifesto.org/>

The Cycle, or Iteration



Support Platforms



The pieces

CVS

CruiseControl

Apache
Tomcat

CVSWeb

Bugzilla

Cobertura

Apache
ANT

Eclipse

Apache
webServer

VQWiki

JUnit

Java

Development Workstations

- Eclipse
- Tomcat
- Windows
- Linux

Eclipse and Tomcat



JavaDoc...

```
/**  
 * Returns an Image object that can then be painted on the screen.  
 * The url argument must specify an absolute {@link URL}. The name  
 * argument is a specifier that is relative to the url argument.  
 * <p>  
 * This method always returns immediately, whether or not the  
 * image exists. When this applet attempts to draw the image on  
 * the screen, the data will be loaded. The graphics primitives  
 * that draw the image will incrementally paint on the screen.  
 *  
 * @param url an absolute URL giving the base location of the image  
 * @param name the location of the image, relative to the url argument  
 * @return the image at the specified URL  
 * @see Image  
 */  
public Image getImage(URL url, String name) {  
    try {  
        return getImage(new URL(url, name));  
    } catch (MalformedURLException e) {  
        return null;  
    }  
}
```

...turns into

getImage

```
public Image getImage(URL url,  
                      String name)
```

Returns an Image object that can then be painted on the screen. The url argument must specify an absolute URL. The name argument is a specifier that is relative to the url argument.

This method always returns immediately, whether or not the image exists. When this applet attempts to draw the image on the screen, the data will be loaded. The graphics primitives that draw the image will incrementally paint on the screen.

Parameters:

url - an absolute URL giving the base location of the image
name - the location of the image, relative to the url argument

Returns:

the image at the specified URL

See Also:

[Image](#)

JUnit

- **JUnit – is a program to test a program**
- **Easily integrated into new or existing java classes**
- **JUnit tests allow you to write code faster while increasing quality.**
- **JUnit tests check their own results and provide immediate feedback.**

JUnit

```
public class HelloWorld extends TestCase
{
    public void testMultiplication()
    { // Testing if 2*2=4:
        assertEquals ("Multiplication", 4, 2*2);
    }
}
```

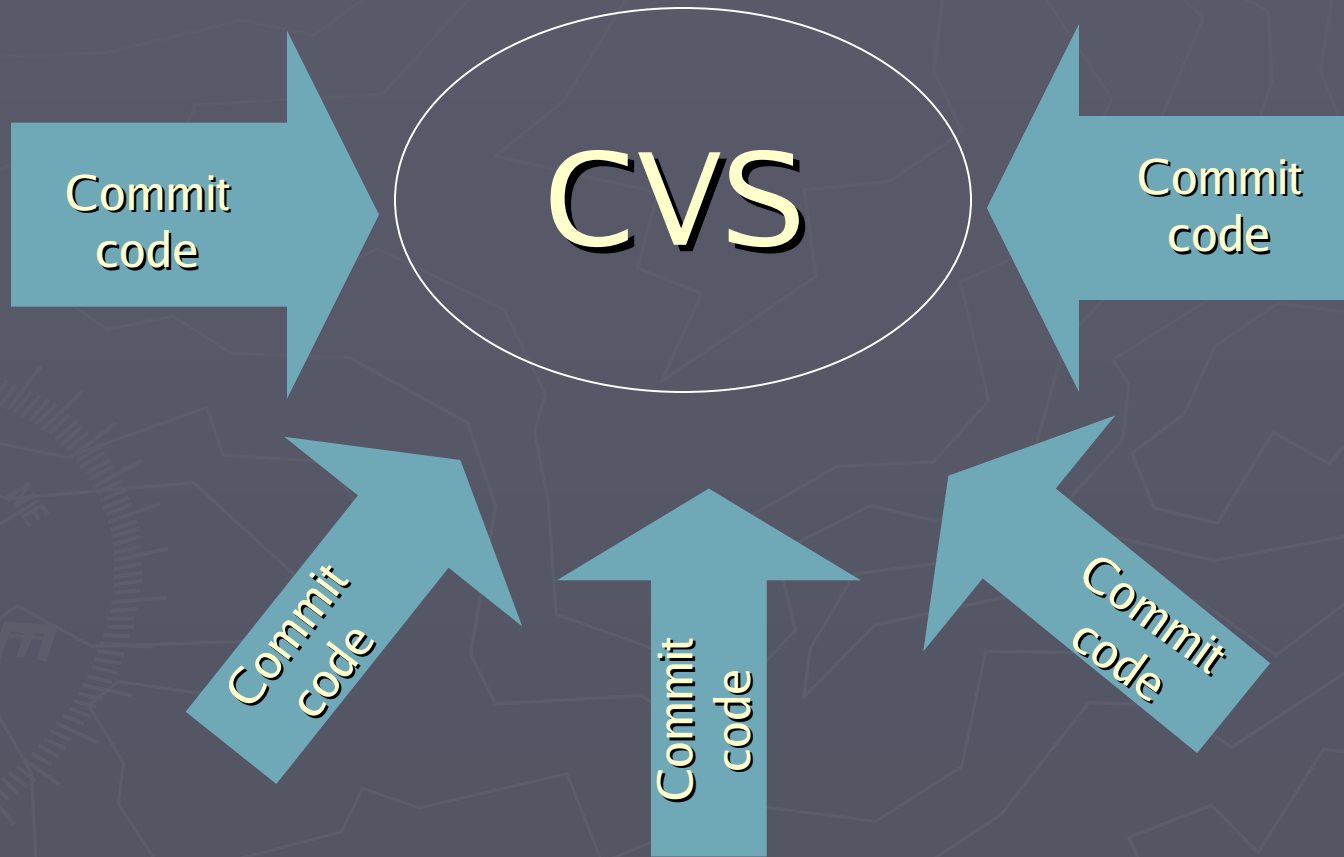
A simple example

JUnit

```
@Test(expected= IndexOutOfBoundsException.class) public void empty() {  
    new ArrayList<Object>().get(0);  
}
```

Another simple example

Many developers archive



CruiseControl

- Build Automation
- Continuous Integration
- Web Interface for status and control
- Supports multiple projects
- Status Email

Cobertura

- Calculates the percentage of code accessed by JUnit tests
- Based on jcoverage
- Works by inserting instrumentation instructions into compiled code
- Records which lines of code have been tested
- Calculates complexity value
- Produces output in html or xml

Build Server



Java Doc
Package
CVS Tag
JUnit test
Instrument
Compile
CVS Export

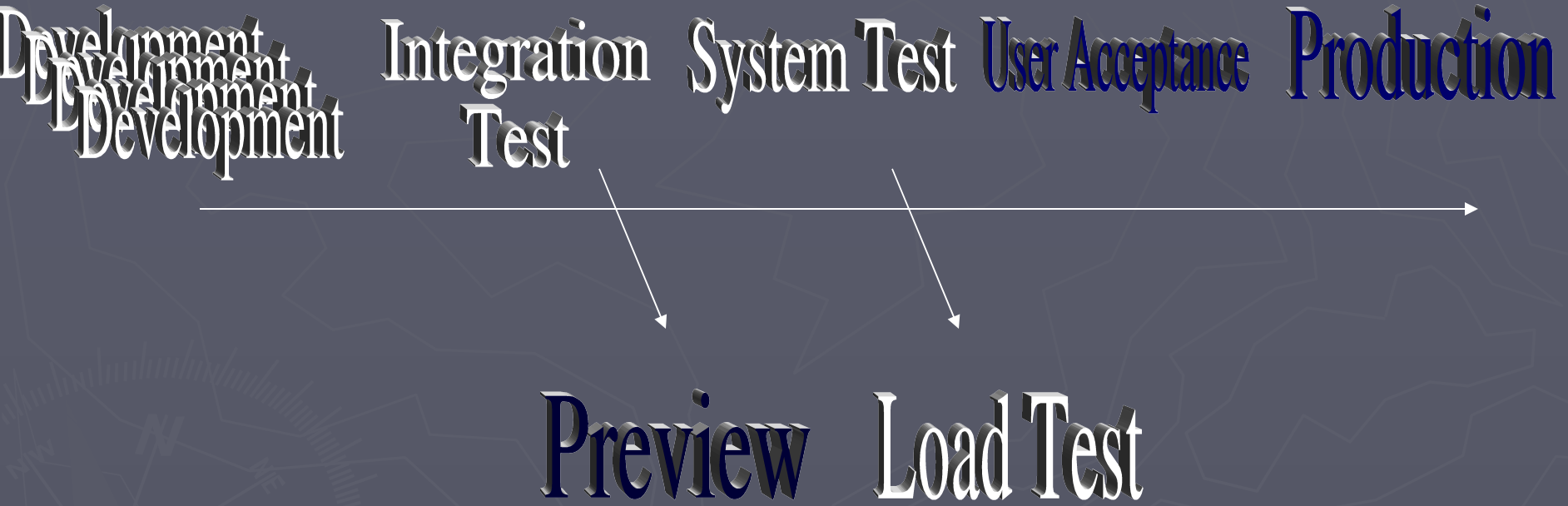
Development Server

- CVS / CVSWeb
- Ant
- Cobertura
- VQWiki
- Bugzilla
- HP-UX
- Oracle Java Application Server

Runtime

- HP-UX
- Linux
- Oracle Application Server
- Oracle Database Server

Support Platforms



The pieces

- Apache Ant: <http://ant.apache.org>
- Apache Tomcat: <http://tomcat.apache.org>
- Apache WebServer: <http://httpd.apache.org>
- Bugzilla: <http://www.bugzilla.org>
- Cobertura: <http://junit.sourceforge.net>
- CruiseControl: <http://cruisecontrol.sourceforge.net/>
- CVS: http://ximbiot.com/cvs/wiki/index.php?title=Main_Page
- CVSWeb: <http://www.freebsd.org/projects/cvsweb.html>
- Eclipse: <http://www.eclipse.org>
- Java: <http://java.sun.com>
- JUnit: <http://junit.sourceforge.net>
- VQWiki: <http://www.vqwiki.org>
- Linux: redhat, fedora, ubuntu

More pieces.. Not so free

- HP-UX: <http://www.hp.com> or
<http://h20338.www2.hp.com/hpux11i/cache/324545-0-0-0-121.html>
- Oracle Application Server: <http://www.oracle.com> (middleware) or
<http://www.oracle.com/appserver/index.html>
- Oracle Database Server: <http://www.oracle.com> (database) or
<http://www.oracle.com/database/index.html>
- Rational Tools: RequisitePro, TestManager, Robot:
<http://www-306.ibm.com/software/awdtools/suite/index.html>
- Windows: desktop, server

Even more pieces...

- Password Manager:
http://www.geocities.com/ramix_info/passwordmanager.ht%6dl
- Cronolog: <http://cronolog.org>
- CGI Scripts for deploy automation

The Future...

- Bugzilla to Jira \$\$
- CVS to Subversion
- Ant to Maven
- XPlanner
- Upgrades upgrades upgrades...