

Automation in XP - Why and How?

Sudhindra Rao
Anand Joglekar
ThoughtWorks

Scope

- Software and Automation
- Automation in Action
- Tools for automation
- Automating for Distributed Teams

Software development

- Software industry makes money by optimizing and automating business processes.
- Typical targets to automate are large volume, low complexity processes.
- High variability / complexity and low volume processes are less viable for computerization.

Why automate?

- Cost of change – stop the exponential increase
- Better value add
- Quicker feedback
- documentation
- Visibility

When to Automate?

- Automate when
 - Performing mundane tasks repetitively
 - Building an application
 - Re-running Unit Tests
 - Need to scale up – that's when drudgery takes toll.
 - Need to reduce risk

What to Automate?

- Automate anything that the machine can do for you
- Build Automation
- Test automation
- Refactoring
- Tools
 - IntelliJ(Eclipse)
 - ant
 - junit
 - CruiseControl

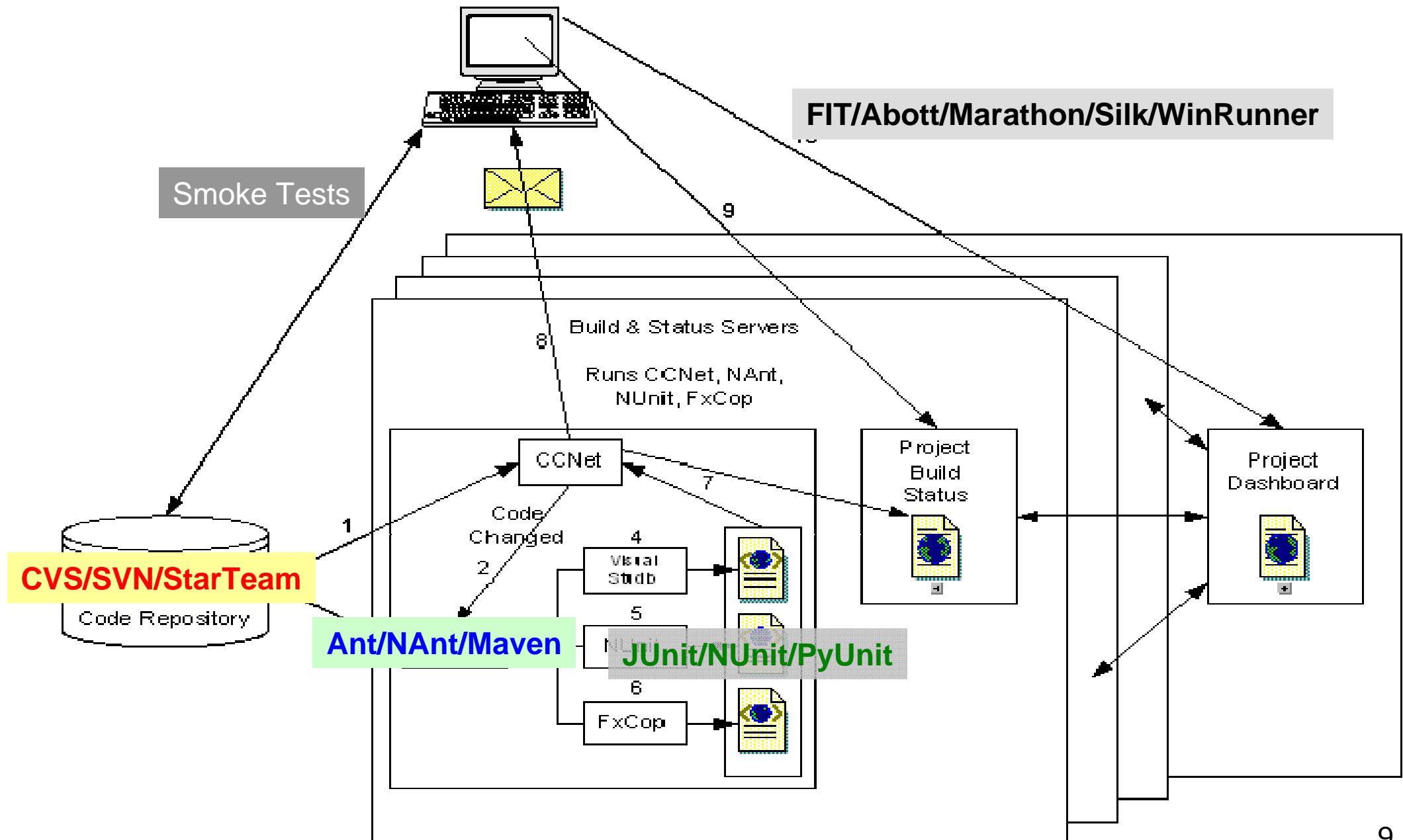
XP, Agile and Automation

- Automation makes XP viable.
- Focussed automation
- Improving feedback
- Many XP assumptions – linear growth in cost of change, ‘yagni’ depend implicitly on automation.

Automation supports XP Practices

- Small Releases
- Simple Design
- Testing
- Refactoring
- Collective Ownership
- Continuous Integration
- 40-hour week
- Coding Standards

Continuous Integration



Automation Tools

- CruiseControl, DamageControl, AntHill, ...
- Few more tools listed on <http://damagecontrol.codehaus.org>
- Selenium, SharpRobo, Sahi, ...
- Your own ingenious automation

CruiseControl

- Builds every quantum of checked in code
- Runs tests and breaks code with errors for an easy fix
- Notifications about health of code
- Fix the build before working on new code

CruiseControl build page

cruisecontrol
continuous integration toolkit

Current Build Started At:
03/30/2004 15:54:54

Latest Build
03/30/2004 15:46:41 (build.2)
03/30/2004 15:42:17
03/30/2004 15:39:47 (build.1)

Build Results | Test Results | XML Log File | Control Panel

BUILD FAILED

Ant Error Message: file:/Users/mike/builds/checkout/dms/build.xml:123: Tests failed! Check test reports.
Date of build: 03/30/2004 15:42:17
Time to build: 11 seconds
Last changed: 03/30/2004 15:42:02
Last log entry: I'm too busy to test

[Build Artifacts](#)

Unit Tests: (4)

failure	testTitleSearch	com.pragprog.dms.SearchTest
---------	-----------------	-----------------------------

Unit Test Error Details: (1)

Test: testTitleSearch
Class: com.pragprog.dms.SearchTest
Type: junit.framework.AssertionFailedError
Message: expected:<2> but was:<0>

```
junit.framework.AssertionFailedError: expected:<2> but was:<0>  
  at com.pragprog.dms.SearchTest.testTitleSearch(Unknown Source)  
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)  
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
```

Modifications since last build: (1)

modified	mike	src/com/ragprog/dms/Search.java	I'm too busy to test
----------	------	---------------------------------	----------------------

Automation at the database layer

- Can databases be created /destroyed / initialized with the build?
- `<sql></sql>` task in ant
- `<exec/>` task in ant
- Database migration scripts with every build
`ant upgradeFrom=111 upgradeTo=222`

Database Migration

```
<script language="javascript">
<![CDATA[
importClass(java.io.File)
importClass(java.lang.Integer)
function UpgradeException(msgTxt){
    this.messageText = msgTxt;
}
try{
    intFrom = new Integer(project.getProperty("upgrade.From"));
    intTo = new Integer(project.getProperty("upgrade.To"));
    dbUser = new String(project.getProperty("db.user"));
    dbPassword = new String(project.getProperty("db.password"));
    dbTnsname = new String(project.getProperty("db.tnsname"));
    oldFileName = new String("UpgradeDBTo_release-one.");
}catch(ex){
    echo = createTask("echo");
    echo.setMessage("*****CANNOT UPGRADE!*****\n"+
        "NEED NUMERIC VALUES FOR upgrade.From AND upgrade.To\n"+
        "e.g. ant dbupgrade -Dupgrade.From=222 -Dupgrade.To=233");
    echo.perform();
    userEx = new UpgradeException("Invalid value for upgrade.From or upgrade.To");
    throw userEx;
}

    echo = createTask("echo");
    echo.setMessage("setlocal\n");
    echo.perform();
    echo.setMessage("set USER="+dbUser+"\n");
    echo.perform();
    echo.setMessage("set PASSWORD="+dbPassword+"\n");
    echo.perform();
    echo.setMessage("set DB="+dbTnsname+"\n");
    echo.perform();

```

Automating deployment

- Generate build artifacts (war, ear, jar files) – every build
- Automate pulling the artifacts and deploying on a testable environment
- Run sanity checks by running smoke tests
- Integrate smoke tests with the build - **build can fail on UAT !!!**

Distributed Projects

- Teams working in different timezones
- Geographically separated
- Build on the same Continuous Integration system
- Web based interface to control/automate building
- Send alerts on a broken build
- It is like teams **pairing** on the same code

Automation in Distributed Projects

- Version Control systems – code sharing among teams
- Visibility of code health to distributed teams – Continuous Integration

Learning

- You improve chance of success with automation
- Automation is cheap – not automating is costly.
- You can win as a customer if you make sure that the development dollar is really spent on solving the business problem – not drudgery.

Thank you

sudhindra.rao@thoughtworks.com

ajoglekar@thoughtworks.com