Smoke Tests to Signal Test Readiness

Aditya Dada (aditya.dada@sun.com)
Sun Microsystems

Presentation Agenda

- Typical Problems
- Release Models
- Smoke Test Types
- Smoke Test Features
- Design Challenges
- Smoke Test Creation Process
- Real-World Example
- Benefits
- Recap
- Q&A

Typical Problems

- Product consists of thousands of classes & interfaces
- Scores of developers working on 1 product
- Tests are run on
  - cross platform configurations
  - for 5 databases
  - multiple architectures
- Entire test base takes 30 hours to run
- ...all in all, the test execution is very expensive.

Test Matrix Sample

Platform O/S	Hardware	Build	Database	JDK	Test Run
Build 1					
Week 1					
Solaris 10 Sparc	Sparc	PE File	Oracle 10g	1.5 Bundled	Full tests
Windows 2003 Server	Intel	PE File	Derby	1.6 Beta	BAT
RH 4.0	X86 Intel	PE File	Oracle 10g	1.5 Bundled	BAT
Build 2					
Week 2					
Solaris 9 Sparc	Sparc	PE File	Oracle 10g	1.6 Beta	BAT
Windows XP	Intel	PE File	MS-SQL	1.5 Bundled	BAT
Windows 2000	Intel	PE File	Derby	1.4.2_10	Full tests
Solaris 10 x86	X86 Intel	PE File	Sybase	1.5.0_06	Full tests

Introduction: Smoke Test

- What is a smoke-test suite?
  - Designed to expose big problems with small number of tests
  - Don't tell you if the product is ready for shipping, rather if a product is ready for testing
    - Entry criteria for testing
- How is the smoke-test suite used?
  - Find defects at one or many stages of a software development life cycle.

Presentation Agenda

- Typical Problems ☑
- Release Models
- Smoke Test Types
- Smoke Test Features
- Design Challenges
- Smoke Test Creation Process
- Real-World Example
- Benefits
- Recap
- Q&A

Release Models

- Tinderbox
  - Continuous compile & build Machine
  - Immediate Feedback
  - Targeted for Developers
  - Reports red/green status & change on Web page
- Nightly
  - Compile & Build by Release Engineer
  - Catches Tinderbox slip-through mistakes
  - Great for validating bug-fixes, especially blocking defects.

Release Models (contd.)

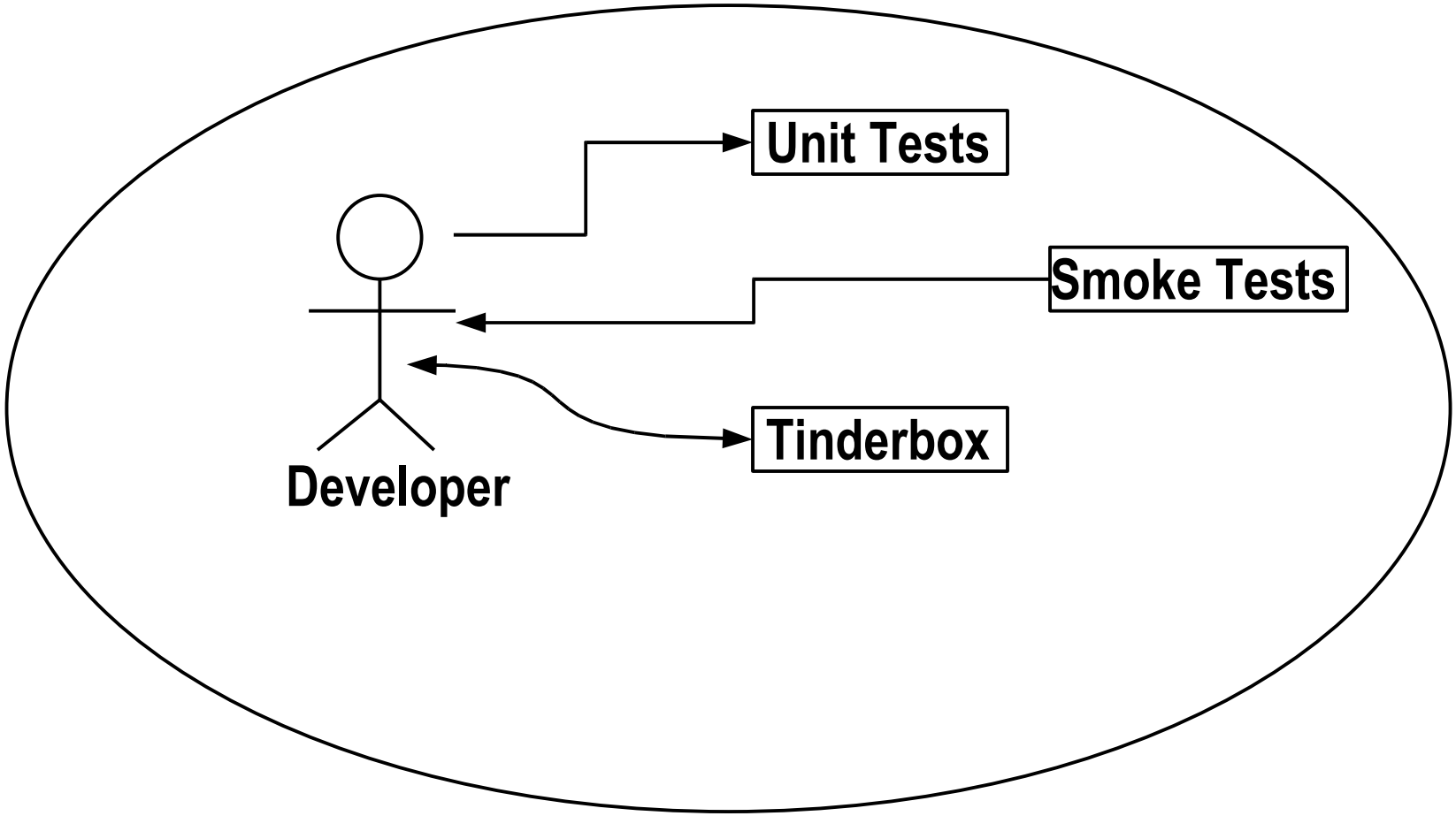
- Weekly
  - More comprehensive testing than Nightly
  - Build of higher certifiable quality
  - Used by Quality Team
  - Tracked by Quality and Project Teams
- Milestone
  - Before external release, and after code freeze
  - Delivered to external web sites, customers
  - Tracked by Upper Management

Presentation Agenda

- Typical Problems ☑
- Release Models ☑
- Smoke Test Types
- Smoke Test Features
- Design Challenges
- Smoke Test Creation Process
- Real-World Example
- Benefits
- Recap
- Q&A

Smoke Test Types

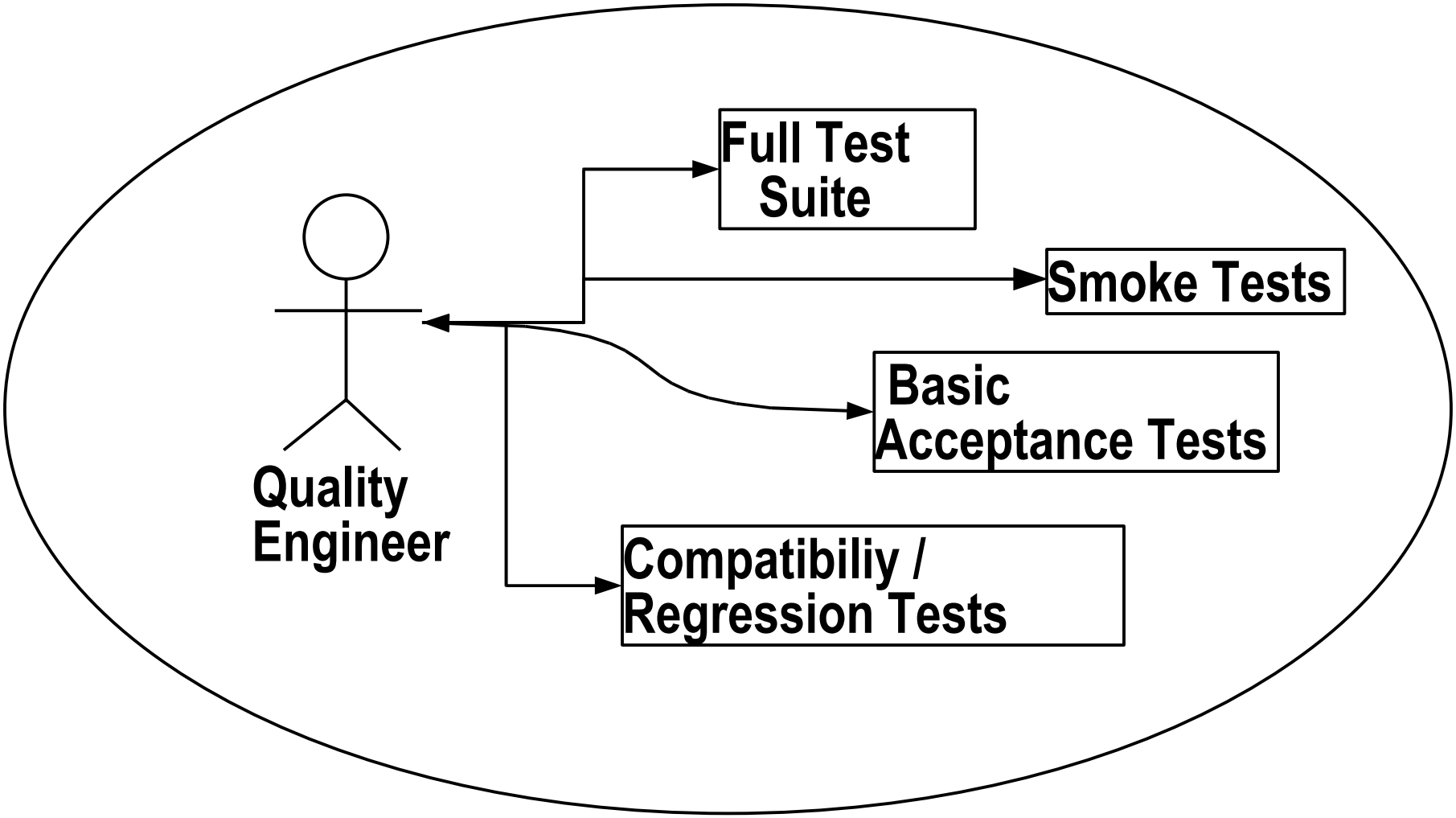
- Catching Defects with Every Integration
  - Smoke Tests run with every change.
  - Tinderbox catches the rest.
  - Development Team Machine used
- Catching defects every night
  - Automation Scripts report status first thing every morning.
  - Quality team machines used



Development Engineering Domain

Smoke Test Types (contd.)

- Catching defects every week
  - Release Engineering certify builds using Smoke tests, every week.
  - Smoke tests maybe used for quick turn-around. Full test suite used otherwise.
  - Quality Engineering lab machines used with coverage matrix
- Catching defects with every milestone
  - Smoke tests certify build of reasonable quality.
  - Full test base execution along with optional tests.



Quality Engineering Domain

Presentation Agenda

- Typical Problems ☑
- Release Models ☑
- Smoke Test Types ☑
- Smoke Test Features
- Design Challenges
- Smoke Test Creation Process
- Real-World Example
- Benefits
- Recap
- Q&A

Features for Smoke tests

- Coverage
  - Purpose:
    - Identify Dead-On-Arrival (D.O.A.) Builds.
    - Identify D.O.A. Components
  - Minimal Depth
    - Full Test Suite covers entire depth
  - Maximum Breadth
    - Certify all functional components' quality

Features for Smoke tests (contd.)

- Speed
  - Purpose: efficiency
  - One click execution
    - Developers run smoke test with every change.
  - Execution time under 30 minutes
    - Quick turn around, and feedback mechanism
  - Customers:
    - Development team,
    - Release Team,
    - Quality Team

Features for Smoke tests (contd.)

- Reporting
  - Multiple reporting formats
    - HTML for management
    - Text for development
    - XML for flexibility
- Management
  - Buy-in from management
    - 30 minutes for every major change. May affect schedules.
  - Gatekeepers needed
    - Large number of developers making enhancements cause instability

Features for Smoke tests (contd.)

- Tools / Harness Services
  - Purpose: simplification
  - Tools for test failure analysis
    - Server/back-end logging
    - Console/client logging
    - Reporting with test location
    - Re-run list for failures
  - Services for ease of test development and execution
    - Common targets
    - Common properties
    - Common configuration & setup
    - Common Database access

Presentation Agenda

- Typical Problems ☑
- Release Models ☑
- Smoke Test Types ☑
- Smoke Test Features ☑
- **Design Challenges**
- **Smoke Test Creation Process**
- **Real-World Example**
- **Benefits**
- **Recap**
- **Q&A**

Smoke Tests Design Challenges

- Complex Software Stack
  - Integration testing
  - Tests for multiple modules needed
  - Integration delays raise costs
- Multiple Developers & Users
  - Managing the developers & users is time consuming
  - Personal preference conflicts
    - Reporting
    - Platforms
    - Execution time
  - Troubleshooting & support for all is difficult

Smoke Tests Design Challenges

- Tests for Tinderbox & Developers:
  - Requirements come from developers
  - Execution time about 15 minutes
  - Single command execution.
    - Only one-time setup needed.
  - Text reporting a must.
  - Automation provided by Quality Team
  - Tests maintained & enhanced by Development Team
  - May include unit tests

Smoke Tests Design Challenges

- Tests for Release Engineering
  - Requirements gathered from Release Engineering
    - Types of machines used.
    - Text & HTML reporting preferred.
    - Time limit allocated for execution
  - Automation provided by Quality team.
  - Tests maintained & enhanced by Quality Team

Smoke Tests Design Challenges

- Tests for QE Verification
  - Requirement gathering from Quality team
    - Number & Types of machines used:
      - Major platforms, most supported configurations
    - HTML & XML Reporting mechanism preferred
    - Time limit allocated for execution < 10 hours
  - Automation provided by Quality team.
  - Tests maintained & enhanced by Quality Team

Smoke Tests Design Challenges

- Test enhancement strategy
  - Simple tests in the beginning
  - Replace with enhanced tests as product development progresses
  - Additional tests may be added depending upon
    - execution time requirements
    - Integration of new features

Smoke Tests Design Challenges

- Baselining
  - Easy Handoff mechanism
  - Create baseline on most supported configuration
  - Use baseline for handoffs
  - Baseline replication avoids test configuration issues

Smoke Tests Design Challenges

- Portability across products and teams
  - Ensure portability across multiple databases
    - Create separate DDLs for each database
    - Never hardcode. Read properties from the common file
    - Minimize dependencies on stored procedures.
    - Resolve differences in multiple JDBC drivers' vendor implementation
    - Use easily portable datatypes
    - XA and non-XA transaction
  - Make no assumptions about encoding schemes for data exchange
    - LDAP (UTF-8),Oracle (UTF-16),JVM default client encoding

Smoke Tests Design Challenges

- Sources and binaries to stay in Sync:
  - Sources required to eliminate test case bugs,version inconsistency issues.
  - Promotes ease of enhancement by all
  - Reasonable payoff against execution time

Presentation Agenda

- Typical Problems ☑
- Release Models ☑
- Smoke Test Types ☑
- Smoke Test Features ☑
- Design Challenges ☑
- **Smoke Test Creation Process**
- **Real-World Example**
- **Benefits**
- **Recap**
- **Q&A**

Smoke Test Creation Process

- Setup a process
  - Automation is the key to following through with the process!
    - Without automation, process implementation become unwieldy
- Pre-requisites
  - Requirement gathering
  - Development architect & manager buy-in
  - Quality architect & manager buy-in
  - Gatekeeper required
  - Tests Identification
    - Simple tests in beginning
    - Enhance/Add tests parallel to development effort

Smoke Test Creation Process

- Requirement Gathering
  - What is the purpose of the test suite?
  - Which tests need to be executed?
  - Who will execute the tests?
  - When and how often will the tests be executed?
  - Where will the tests be run?
  - Who will monitor the results?
  - Who will monitor issues like hangs, crashes etc?
  - What are the guidelines for filing, tracking and fixing defects?

Smoke Test Creation Process

- Requirement Gathering (contd.)
  - Will other groups use the test base?
  - Will other groups contribute to the test base?
    - Do we need check-in procedures for all teams?
  - Will support be added in future for more configurations?

Smoke Test Creation Process (contd)

- Automation
  - Invest in proper build framework
    - Multiple configurations support suggests platform-independent build framework. e.g. ANT, Maven etc.
  - Identify common functionality, properties, targets, libraries
  - Establish common configuration directory with common files
  - Establish common reporting guidelines
    - One page summary for easy reference
  - Use build framework to create single click execution
    - Improve to reduce setup and running times

Presentation Agenda

- Typical Problems ☑
- Release Models ☑
- Smoke Test Types ☑
- Smoke Test Features ☑
- Design Challenges ☑
- Smoke Test Creation Process ☑
- **Real-World Example**
- **Benefits**
- **Recap**
- **Q&A**

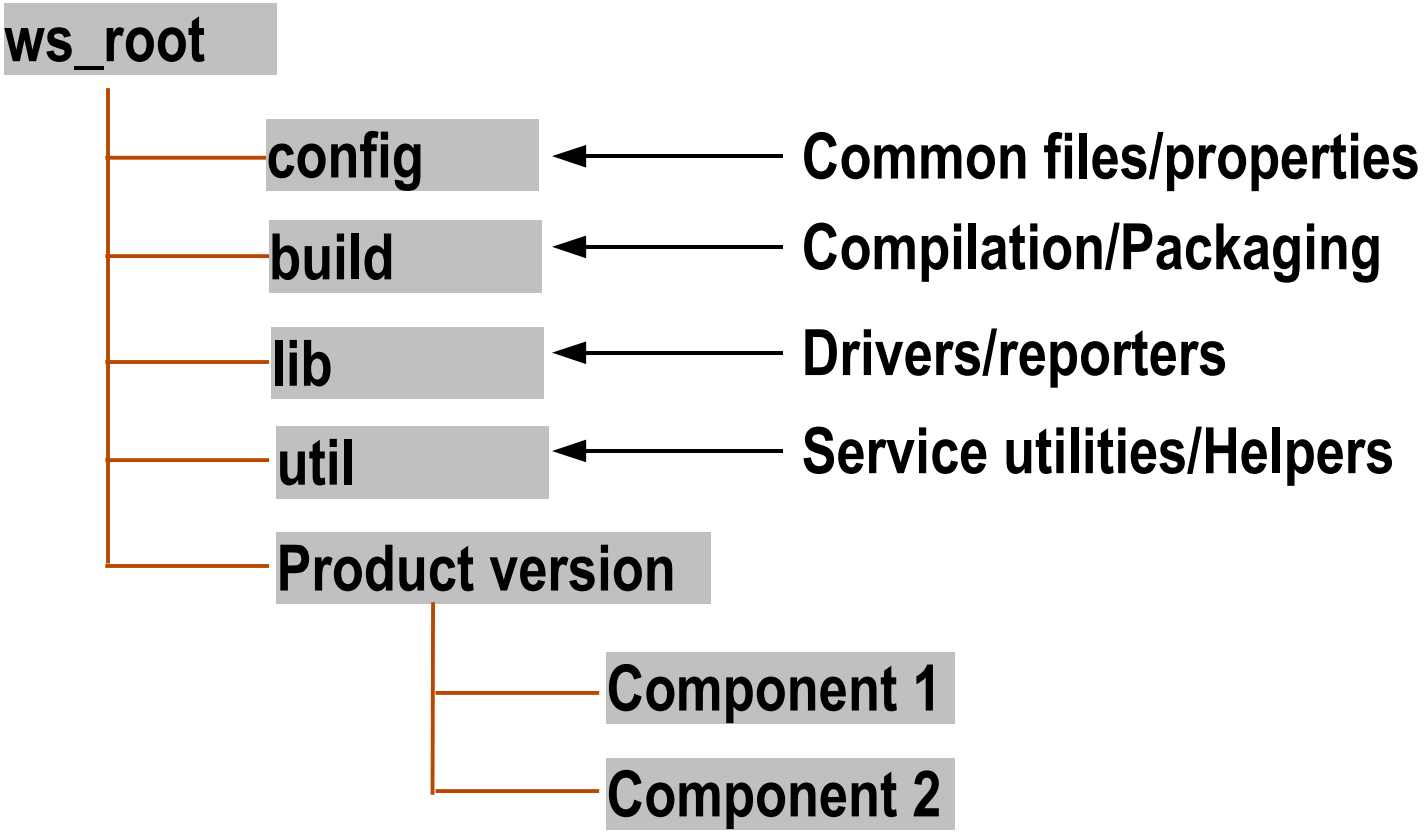
Real-World Example

- Sun Java System Application Server, Glassfish
  - Support on
    - 6 databases
    - multiple versions of Linux, Solaris, Mac & Windows
    - Intel, SPARC & AMD architectures
    - multiple browsers, loadbalancers, webtier modules.
  - In all, they add up to over 1500 configurations!

Real-World Example (contd.)

- Application Server consists of ...
  - EJB Container
  - Web Tier
  - Java Message Service
  - Java Transaction Service
  - Java Connector Architecture
  - Java Server pages....and many more.
  - Test applications cover all above!

Real-World Example (contd.)



Real-World Example (contd.)

```
<target name="deploy-common" depends="init-common">
 <property name="as.props"
        value="--user ${admin.user}
        --password ${admin.password}
        --host ${admin.host} --port ${admin.port}"/>
  <exec executable="${ASADMIN}" failonerror="false">
    <arg line="deploy"/>
    <arg line="${as.props}"/>
    <arg line="--retrieve ${assemble.dir}"/>
    <arg line="${assemble.dir}/${appname}App.ear"/>
  </exec>
</target>
```

This a snippet of a simple target to deploy an EAR file to the application server. In our e.g., this is common functionality used by most components. This is kept in a 'common.xml' file under the 'config' directory that contains all the common and shared code.

Real-World Example (contd.)

Results Summary Report

(Automatically generated on Wed May 04 16:51:39 PDT 2005)

Configuration

OS	J2SE	Machine
Mac OS X 10.4	1.4.2_07	.sfbay.sun.com

Summary

S.No.	Test Area	Passed	Failed	DidNotRun	Total	% Pass
1.	Tomcat-Servlet2_4	57	0	0	57	100%
2.	Tomcat-Servlet2_3	52	0	0	52	100%
3.	J2EE-Deployment	7	0	0	7	100%
4.	GTest-Transaction	0	0	0	0	0%
5.	J2EE-Transaction	0	0	0	0	0%
6.	J2EE-Connector	24	0	0	24	100%
7.	GTest-Security	27	0	0	27	100%
8.	J2EE-EJB	169	0	0	169	100%
9.	J2EE-JSR109	14	0	0	14	100%
10.	J2EE-Classloader	0	0	0	0	0%

Presentation Agenda

- Typical Problems ☑
- Release Models ☑
- Smoke Test Types ☑
- Smoke Test Features ☑
- Design Challenges ☑
- Smoke Test Creation Process ☑
- Real-World Example ☑
- **Benefits**
- **Recap**
- **Q&A**

Benefits

- Integration Model
  - Checks a Complex Software Stack
  - Re-used by Many Teams, multiple developers
- Timely Feedback
  - Catches defects as close to introduction as possible.
  - Saves expensive execution costs for DOA builds.
  - Simplifies test execution by splitting load effectively
- Effective use of resources
  - Only certified builds are tested.
  - No time wasted.

Recap

- Typical Problems
- Release Models:
  - Tinderbox, Nightly, Weekly, Milestones
- Smoke Test
  - Types
  - Features
  - Design Challenges
  - Creation Process
- Real-World Example
- Benefits:
  - Timely Feedback, Effective use of Resources

Presentation Agenda

- Typical Problems ☑
- Release Models ☑
- Smoke Test Types ☑
- Smoke Test Features ☑
- Design Challenges ☑
- Smoke Test Creation Process ☑
- Real-World Example ☑
- Benefits ☑
- Recap ☑
- **Q&A**