Equipping Yourself for the Agile Wave

Alan Kan, IBM Rational Technical Specialist
alankan@nz.ibm.com
Discussion Topics

- **Agile has gone Mainstream**
  
- The Road to Agility
  
- Best Practices for Agile Testing
  
- Latest and Upcoming Rational Technologies
  
- Summary
What is Agile?

- An iterative and incremental (evolutionary) approach performed in a highly collaborative manner with **just the right amount of ceremony** to produce high quality software in a cost effective and timely manner which meets the changing needs of its stakeholders.

- **Core principles**
  - Frequent delivery of working software
  - Continuous testing and validation
  - Consistent team collaboration
  - Rapid response to change
  - Ongoing customer involvement
  - “Fits just right” process
Agile has gone Mainstream  
*From the analyst community*

"Thirty-five percent of … respondents have projects or pilots underway, and *only 12 percent do not see a fit* for Agile processes in their organizations.

The fact that 88 percent of these organizations (one-third of which have over 10,000 employees) are using or evaluating Agile processes proves that Agile processes have truly hit the mainstream."

- Excerpt from “And the Agile Survey Says…”  
  Agile Journal, March 6, 2006

**Third-party research suggests even wider adoption**

*Have you adopted any Agile techniques?*

- **“No”** 35%
- **“Yes”** 65%

Source: Ambler ‘Agile Adoption Rate Survey’ of over 4200 Dr. Dobb’s subscribers, March 2006
Discussion Topics

- Agile has gone Mainstream

- The Road to Agility

- Best Practices for Agile Testing

- Latest and Upcoming Rational Technologies

- Summary
Challenges in Achieving Agility

- Following the process
- Change management
- Collecting team status
- The larger role of a tester
- Heavier testing workload
- Traditional tools do not fit well with Agile testing
- Cost of face-to-face meetings
- Why is this change in the build?
Challenges with Agile in Larger Organisations

- **Compliance requirement**
  - Low risk
  - Critical, Audited

- **Geographical distribution**
  - Co-located
  - Global

- **Application complexity**
  - Simple, single platform
  - Complex, multi-platform

- **Organization distribution**
  - In-house
  - Third party

- **Entrenched process, people, and policy**
  - Minimal
  - Significant

- **Team size**
  - Under 10 developers
  - 100’s of developers

- **Degree of Governance**
  - Informal
  - Formal

**Agile Methodology**
Discussion Topics

- Agile has gone Mainstream
- The Road to Agility
- **Best Practices for Agile Testing**
- Latest and Upcoming Rational Technologies
- Summary
Testing Best Practices in Agile

- User Acceptance Testing
- Manual Exploratory Testing
- Security Testing
- Unit Testing
- Test Management and Team Collaboration
Best Practices – User Acceptance Testing

- Get educated on requirements definition/gathering
- Automate UAT
- Centralise generic test assets for reuse
- Use mainstream scripting languages
- Automate Policy Testing

- Used to uncover hidden requirements
- Pair up with a domain expert or a business analyst to do exploratory testing
- Keep a catalog of reusable test assets
- Automate part of your exploratory tests
Best Practice - Security Testing

- We are not security experts – get help
- Automate security tests
- Security information updates are important
- Recommendations to developers would be ideal
Best Practices – Unit Testing

- Pick your favourite tools
- IDE
- xUnit family
- Reuse
Best Practices – Test Management and Team Collaboration

- Clearly define and publish the process and keep it up-to-date
- Everyone keeps a task list
- Test planning for one iteration
- Generate documentation instead of writing them
- Use team collaboration technologies to replace some face-to-face meetings
- End-to-end automation of regression testing
Complexity Changes the Approach for Tools & Process

Organizational Drivers
- Team Size
- Geographical Distribution
- Organization Distribution
- Entrenched process, people, policy

Technical and Regulatory Drivers
- Application complexity
- Compliance
- Governance

End-to-end tool focus
- Auditable, reproducible process

Focus on tools integration
- Process support for distributed development

Best of breed tools
- Documented lifecycle, shared best practices

Open source tool sweet spot
- Principles and individual practices

Focus on tools integration
- Process support for distributed development

Best of breed tools
- Documented lifecycle, shared best practices

Open source tool sweet spot
- Principles and individual practices
Discussion Topics

- Agile has gone Mainstream
- The Road to Agility
- Best Practices for Agile Testing
- Latest and Upcoming Rational Technologies
- Summary
Rational Performance Tester Extension for SAP Solutions

- The process of testing SAP applications
  - Follows the same steps as testing web based applications
    - Record
    - Edit
    - Execute & measure response times
  - Requires new technology to enable the process
    - SAP Recorder
    - SAP Protocol Browser for test editing
    - SAP Performance Reports
**Rational Performance Tester Extension for SAP Solutions**

**SAP Protocol Browser**

- Protocol Browser allows users to insert actions into the script by acting upon captured screenshots
  - No re-recording or complex test editing required

- Verification points can be inserted to validate SAP data
Rational *Functional* Tester Extension for Siebel Test Automation

**Unique Features**

- Lowers the cost of functional testing
  - Recognition of Siebel custom High Interactivity (HI) and Standard Interactivity (SI) Objects

- Offers extensive customization options
  - Exposing of Siebel Script-Only methods
  - Siebel Automation Framework
How Rational AppScan works

1. Scan
2. Analyze
3. Report

Detailed, Actionable Information

Security, Privacy, Quality, Standards, Compliance
Rational AppScan Screenshots
**Blind SQL Injection**

**Severity:** High  
**Type:** Application-level test  
**WASC Threat Classification:** Command Execution: SQL Injection  
**CVE Reference(s):** N/A  
**Security Risk:** It is possible to view, modify or delete database entries and tables.

**Possible Causes**  
Sanitization of hazardous characters was not performed correctly on user input.

**Technical Description**  
Web applications often use databases at the backend to interact with the enterprise data warehouse. The de-facto standard language for querying databases is SQL (each major database vendor has its own dialect). Web applications often take user input (taken out of the HTTP request) and incorporate it in an SQL query, which is then sent to the backend database. The query results are then processed by the application and sometimes displayed to the user.

This mode of operation can be exploited by an attacker if the application is not careful enough with its treatment of user (attacker) input. If this is the case, an attacker can inject malicious data, which when incorporated into an SQL query, changes the original syntax of the query into something completely different. For example, if an application uses users’ input (such as username and password) to query a database table of users’ accounts in order to authenticate the user, and the attacker has the ability to inject malicious data into the username part of the query (or the password part, or both), the query can be changed into a different data-yielding query; a query that modifies the database, or a query that runs shell commands on the database server.

Typically, the attacker achieves this goal in steps. Helise will first learn the structure of the SQL query, and then use this knowledge to thwart the query (by injecting data that changes the query to perform functions different from those intended. Sometimes the query in injection is...
AppScan Screenshots

Blind SQL Injection

Fix Recommendation

General

App.Net

J2EE

PHP

There are three possible ways to protect your application against SQL injection, i.e. malicious tampering of SQL parameters. Instead of directly inserting user input into SQL statements, one could enforce the following:

1. Prepared Statements, which is precompiled and stored in a pool of PreparedStatement objects. PreparedStatement defines setters for supported JDBC SQL data types. For example, setString should be used for input parameters of type VARCHAR or LONDDVARCHAR. Using input parameters prevents an attacker from manipulating the SQL statement through injection of bad characters, such as apostrophe.

Example of how to use a PreparedStatement in J2EE:

```java
// J2EE PreparedStatement Example
// Get a connection to the database
Connection myConnection;
if (isDataSourceEnabled()) {
    // using the DataSource to get a managed connection
    Context ctx = new InitialContext();
    myConnection = (DataSource)ctx.lookup(datasourceName).getConnection(dbuserName, dbPassword);
} else {
    try {
        // using the DriverManager to get a JDBC connection
        Class.forName(jdbcDriverClassName);
        myConnection = DriverManager.getConnection(dbURL, dbUserName, dbPassword);
    } ...
```
What is Jazz?

Jazz is IBM's **next-generation technology platform** for collaborative software delivery. Uniquely attuned to global and distributed teams, the Jazz platform is designed to **transform how people work together to build software**—making software delivery more **collaborative, productive and transparent**.

**Innovation**

A major investment by IBM to create a scalable, extensible team collaboration platform

**Community**

Jazz.net – an online venue for open commercial development of the Jazz platform and Jazz-based products.

**Evolution**

Many Rational products will evolve to support the Jazz technology platform over time, bringing a host of next-generation capabilities to the Rational Software Delivery Platform.

**Vision**

Our vision for the future of software delivery -- supporting diverse types of teams and a workforce that is increasingly organizationally and globally distributed, fluid and dynamic.
The value of the Jazz technology?

- The Jazz platform enables teams to:
  - **Collaborate in context** of the work they are doing. Jazz maintains the relationships between artifacts so you don’t have to.
  - **Right-size governance** by capturing, sharing and automating best practices at varying degrees of rigor. Fine-tune your governance process over time—avoiding the need to define everything up front.
  - **Choose your own path.** An open and extensible architecture is designed to give you the flexibility to assemble your own software delivery platform, relying on your pref

---

**The Jazz Platform**

*An Open, Services Oriented Architecture*

- **Eclipse Client**
  - Jazz Client Extensions
  - Eclipse Platform

- **Lotus Sametime or open source Jabber**

- **Jazz Team Server**
  - Web UI
  - Jazz Server Extensions
  - Jazz Kernel
  - Eclipse Equinox
  - J2EE Web Services

- **DB2 or open source Derby**

- **Web 2.0 or Web Services clients**
What is IBM Rational Team Concert?

A New Family of Products based on Jazz Technology
- Optimized for agile development teams in midsized & large businesses
- Integrates the entire team around an integration server
- Includes Enhanced Eclipse Development environment

Primary Features
- In-place collaboration among team members
- Support and enforcement for development processes
- Transparency of status and trends through automated data-gathering and reporting

Motivation
- Enable flexible, agile application lifecycle management
- Low administrative footprint optimized for agile teams in SMB & large businesses
- Adds value for existing ClearCase and ClearQuest customers
- Step one in the rollout of “lifecycle service integration” middleware in the Rational Software Delivery Platform
Evolving the Rational Software Delivery Platform
An open ecosystem based on IBM middleware

Built for development efficiency: Allows developers to innovate rather than duplicating efforts, figuring out who to hand off to, or tracking and reporting status
Work items at jazz.net
Iteration plan at jazz.net
Dashboards at jazz.net

Web UI Team Dashboard

Team Description

Web UI Foundation
Provides frameworks and APIs for web browser-based user interfaces. Also develops the Admin Web UI.

Team Members

- Bill Higgins - component lead, contributor
- Richard Backhouse - contributor
- Kristen Behoff - contributor
- Reb Reddix - part-time, contributor
- Matthew Jarvis - contributor
- Matt Lavin - part-time, contributor

Planning

PMC MS Iteration Plan
Relicensing
Team Concourse 1.0 Release Plan
Web UI MS Iteration Plan
Web UI Wiki Page

Recent Team Events

- Improve Web UI handling of situation when a user does not belong to the groups required to access… (40667)
- Improve Web UI handling of situation when a user does not belong to the groups required to access… (40667)
- Improve user experience when there are problems with login (38433)
- Improve Web UI handling of situation when a user exists in the repository, but not in the user… (40662)

In Progress Work Items

- 36128 Adopt Diga 1.0.2
- 39996 Display error icon on the project application page for Admins
- 38971 Adopt Diga 1.0 (Repository)
- 33697 Need a way to dynamically show/hide 'show partial/maximize' controls
- 38416 Set 'Cache-Control: public' header for web UI resources

Team Work Item Queues

Open (Current Milestone) (29)
Recently Resolved (1)
Unresolved (40)
Backlog (67)
Discussion Topics

- Agile has gone Mainstream
- The Road to Agility
- Best Practices for Agile Testing
- Latest and Upcoming Rational Technologies

- Summary
Summary: Succeeding with Agile

- **Agile Development is transforming how development is done**
  - Placing greater demand on people, processes and tools

- **Following best practices, Agile can scale to accommodate technical and organizational complexity**
  - Automation is a must
  - Equip your testers

- **There are a lot of cool Rational technologies**
  - Rational AppScan
  - Rational testing tools for SAP, Siebel
  - Jazz.net
  - Rational Team Concert – read Agile!
Resources

- Agile Community http://www.agilealliance.org/
- Manifesto for Agile Development http://www.agilemanifesto.org/
- Scott Ambler http://www.ambysoft.com/
- Jazz or Rational Team Concert http://www.jazz.net
- Alan Kan, alankan@nz.ibm.com