Test Data Management Challenges and Solutions

Software testing naturally involves some "housekeeping chores". One of these chores is managing test data. Perhaps you keep your test data in a relational database. Maybe it's a set of text files or spreadsheets. Some of your tests might require automatic generation of input values when the test is executed.

A variety of data sources are used to set up for, or provide input to, test cases. The actual and expected results from test execution also require varying levels of management. With many available options it can be a challenge to determine the most effective or efficient alternative for handling diverse data sets. This presentation reviews some of the strategies and techniques for obtaining, storing, tracking, and deleting various test data.

Definitions for this presentation

- Test data is information used while executing test cases, or resulting from the execution of test cases
  - Test case management is not our focus
  - Data management implies creating or obtaining, storing, tracking, or otherwise guiding or handling test data throughout its life cycle.
Agenda

- Our Motivation
- Roles Played by Data and The Data Lifecycle
- Data Packaging
- Scenarios and Challenges
- Distilled Requirements
- Some Solutions
- Future Directions

Why Manage Test Data?

- Test case data is an asset
- For data-driven or keyword-driven testing, the data specifies the test cases
- To reproduce any reported bugs, the data used must be available
- Finding relevant data for a test case must be easy to allow efficient execution of the test case
- Quality of the test data is directly related to the quality of the end product
- Redundant data may be cheap to store, but expensive to use
Roles for Test Data

- **Input**
  - Data elements
  - System setup data
  - Application configuration
  - Transactions

- **Output**
  - Calculated Results
  - State of the application
  - State of the system

Test Data Life Cycle

- Analysis
- Creation
- Use
- Maintenance
- Disposition
How might data be packaged?

- Delimited text file (e.g. CSV, logs)
- Flat markup text files (e.g. HTML, XML)
- Workbooks (e.g. Microsoft Excel or OpenOffice Calc)

How might data be packaged?

- Binary Data Files (e.g. Executable, image, or archive files)
How might data be packaged?

- Streaming data through the computer's ports (e.g. Manual input through any human interface, game controllers, other devices)

- Relational or object-based databases with string, numeric, date-time, or BLOB data types. Might be extracts from customer databases, or subsets of testing data.
Common Scenario – Text files

- Comma Separated Values (CSV) text file with data where each row is a test case

- Changes that might be tracked include adding new rows, removing old rows, editing existing rows

<table>
<thead>
<tr>
<th>User</th>
<th>Password</th>
<th>ExpectedLogo</th>
</tr>
</thead>
<tbody>
<tr>
<td>rtillerson</td>
<td>bigprofits</td>
<td>exxon.jpg</td>
</tr>
<tr>
<td>kdlewis</td>
<td>bigdebt</td>
<td>bankAmerica.jpg</td>
</tr>
<tr>
<td>cbartz</td>
<td>nomicrosoft</td>
<td>yahoo.jpg</td>
</tr>
<tr>
<td>llelsenhans</td>
<td>cash30pershare</td>
<td>sunoco.jpg</td>
</tr>
<tr>
<td>jmimmelt</td>
<td>wwdomain</td>
<td>ge.jpg</td>
</tr>
</tbody>
</table>

Common Scenario – Shared Data

- System configuration setup required by multiple test cases

- Input data used as part of the same script/instructions by multiple test cases

- Changes required by one test script may adversely affect another test script
Common Scenario – Database Upgrades

- Schema changes from one release to another
- New default data required for new data elements (may or may not be visible to the user)
- Consumption of test data

Common Scenario – Image files

- New version of the file required for a new release of the product
- May contain embedded metadata relevant to a test that must be changed for different test cases, e.g. date/time information
Common Scenario – Streaming Data

- Testing Devices or a Service
- Could effectively be generated in advance or in real time
- Sometimes sample data stored for later replay
- Stored data editable to match requirements for test cases

Common Scenario – Localized Data

- Need to run the same script using different languages = different text and numeric input
- Data packaging and storage must handle different character encodings
### Common Scenario – Exploratory Testing

- Manual or online record of actions and input values
- Previous setup and configuration data
- System output (results, transactions, logs, etc)

### Common Scenario – Business Rule Changes

- Steps for some actions might change
- Additional (or less) test data required for existing test cases
- Additional data required as expected results
- New data required for new test cases
Challenges

Common Themes
From Scenarios

- Multiple sources with data in different formats required for test cases
- Version control of content
- Large Volumes
- Small Volumes
- Generated Data Consistency
- Dynamically created data at the time of test execution (e.g. Transactions, calculated results, etc.)
Challenges

- Ever growing libraries of DB extracts
- Data Subsetting
  - What's the required data volume?
  - What data needs to be added?
  - How to mask for security reasons?
- Historic relationship between test results and test data makes it hard to delete old data

Challenges

- Different testing goals with different releases means different sets of data required, but there will be overlap
- Different base states required for different test cases
- Aging the data where required
- Test Data consumption
- Multiple languages and character encodings
- Advance specification of setup data can be hard
Requirements

For Different Solutions

Data Requirements

- Configuration management support
  - Version control
  - History

- Provenance:
  - Where does it come from?
  - How has it been modified?
  - Where was it used?

- Package content, or metadata, is accessible
Data Requirements

- Reliable and consistent generation
- Adaptable to growth in volume over time
- Discovery - Searchable
- Maintenance needs – fixes and updates
- Expect reuse of data across test cases and application versions
- Archivable

Data Requirements

- Dynamically harvest new data (results, transactions, etc.)
- Access by different people/Security
- Data should be defined in context – what is it's role, and how does it depend on other data and their roles.
Options for Solutions

To Challenges

Examine Roles

- Partition Data by Role
  - Environment
  - Setup
- Execution
  - Result
- Use UML Test Profile to clarify requirements
Processes and Conventions

- Sequenced execution of test cases for shared, aged, or consumable data
- Conventions for referencing, naming, or storing data in tools to address data sharing and metadata
- Well defined baselines (especially for databases)
- Test case definition include all relevant data roles
- Introduce periodic reviews of data files and content associated with test cases to remove unused data

Generated Data

- Static = pre-generated
  - Derive from simple and well known rules
  - Fully documented output of generation tools
- Most useful working with aging or consumable data
- Dynamic = at time of test execution
  - Process immediately to avoid storing
Version Control

- Versions of data for specific app versions
- Use where it makes sense
- Generally only available to flat files
- Some databases support it
- Data generation utilities should be under version control too
- Baseline the data to be able to return to the same state

Self-Verifying Data

- Test data elements encode the expected result
- Eliminates maintenance of result data
- May also provide metadata in the data
- Most useful for automated testing
Change the Packaging

Tools

- Data Management or Digital Asset Management
- Something with API that allows integration with test automation and test case management tool.
- Home-grown lists or folder structures
- May need multiple tools with diversity of data packaging or requirements.
Future

- Grid or Cloud Computing
- GrayWulf – clustered architecture for data intensive computing
- Your Own Data Warehouse

Summary Recommendations

- There is no one "right way" to meet the diversity of data management needs
- Careful analysis of needs – do you really need to manage it?
- Review options
- Adopt a composite, or hybrid approach, blending techniques that fit your organization's needs
Recommendations

- Consider maintenance consequences when defining your data
- Design the structure and content of your data to be transparent
- Data should improve understanding of testing
- Buy or build practical tools that meet your needs. No single tool will do the full job.

References

- IT Toolbox "A New Day at the Office"
  

- SQA Forums topic "Test Data Management War Stories"
  
  http://www.sqaforums.com/favlinker.php?Cat=0&Entry=18297&F_Board=UBB15&Thread=559466

- UML Testing Profile
  
  http://www.omg.org/technology/documents/formal/test_profile.htm
References


- Noel Nyman, "Self Verifying Data - Testing without an Oracle," StickyMinds.com
  http://www.stickyminds.com/sitewide.asp?Function=edetail&ObjectType=ART&ObjectId=2918

References: tools

- Solix  http://www.solix.com
- Grid-Tools  http://www.grid-tools.com
- IBM Optim  http://www.optimsolution.com
- Mosaic DSTAR™  http://www.mosaicinc.com/mosaicinc/pdf_files/DSTAR_Mosaic_Data_Profile_Manager.PDF
References: tools

- DSpace  http://www.dspace.org
- DBPrism  http://www.dbprism.com.ar
- Alfresco  http://www.alfresco.com/
- FITnesse  http://fitnesse.org

Thank You!

- Questions or Comments to derekk62 at Yahoo dot com
Welcome to SQGNE’s 15th season!

- An all-volunteer group with no membership dues!
- Supported entirely by our sponsors...
- Over 700+ members
- Monthly meetings - Sept to July on 2nd Wed of month
- E-mail list - contact John Pustaver pustaver@ieee.org
- SQGNE Web site: www.swqual.com/sqgne/main.html

Volunteers / Hosts / Mission

Volunteers
- John Pustaver - Founder and Director
- Steve Rakitin - Programs and web site
- Gene Freyberger - Annual Survey
- Dawn Wu - our new greeter!!

Our gracious Hosts
- Paul Ratty - room, copies, cookies
- Tom Anakel - room, copies, cookies
- Margaret Shinkle - room, copies, cookies
- Jack Guilderson - A/V equipment

SQGNE Mission
- To promote use of engineering and management techniques that lead to delivery of high-quality software
- To disseminate concepts and techniques related to software quality engineering and software engineering process
- To provide a forum for discussion of concepts and techniques related to software quality engineering and the software engineering process
- To provide networking opportunities for software quality professionals

ASQ Software Division

- Software Quality Live - for ASQ SW Div members...
- Software Quality Professional Journal www.asq.org/pub/sqp/
- CSQE Certification info at www.asq.org/software/getcertified
- SW Div info at www.asq.org/software
- ICSQ Nov 9-11 2009 Northbrook, IL www.asq-icsq.org/

SQGNE 2008-09 Schedule

<table>
<thead>
<tr>
<th>Speaker</th>
<th>Affiliation</th>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lou Cohen</td>
<td>None</td>
<td>9/10/08</td>
<td>Introduction to using Quality Function Deployment on Software Projects</td>
</tr>
<tr>
<td>John Pustaver</td>
<td>None</td>
<td>10/6/08</td>
<td>A Survey of Test Automation Approaches</td>
</tr>
<tr>
<td>Steve Rakitin</td>
<td>None</td>
<td>11/12/08</td>
<td>Estimating using Work Breakdown Method - An interactive exercise</td>
</tr>
<tr>
<td>J. Douglas</td>
<td>None</td>
<td>12/9/09</td>
<td>Integrating Agile into the Development Process</td>
</tr>
<tr>
<td>A. Vinogradov</td>
<td>None</td>
<td>1/13/09</td>
<td>Route Optimization Etc.</td>
</tr>
<tr>
<td>A. Vinogradov</td>
<td>None</td>
<td>2/10/09</td>
<td>Route Optimization Etc.</td>
</tr>
<tr>
<td>Brian LeSueur</td>
<td>None</td>
<td>3/11/09</td>
<td>Introduction to using Quality Function Deployment on Software Projects</td>
</tr>
<tr>
<td>Brian LeSueur</td>
<td>None</td>
<td>4/8/09</td>
<td>Coding with a Purpose</td>
</tr>
<tr>
<td>Brian LeSueur</td>
<td>None</td>
<td>5/13/09</td>
<td>Automated Functional Test Design</td>
</tr>
<tr>
<td>Brian LeSueur</td>
<td>None</td>
<td>6/10/09</td>
<td>Test Tool - Make or Buy?</td>
</tr>
<tr>
<td>Brian LeSueur</td>
<td>None</td>
<td>7/9/09</td>
<td>Test Tool - Make or Buy?</td>
</tr>
</tbody>
</table>

Test Data Management Challenges and Solutions

Derek Kozikowski

Software testing naturally involves some “housekeeping chores”. One of these chores is managing test data. Perhaps you keep your test data in a relational database. Maybe it’s a set of test files or spreadsheets. Some of your tests might require automatic generation of input values when the test is executed.

A variety of data sources are used to set up for, or provide input to, test cases. The actual and expected results from test execution also require varying levels of management. With many available options it can be a challenge to determine the most effective or efficient alternative for handling diverse data sets. Derek will review some of the strategies and techniques for obtaining, storing, tracking, and deleting various test data.

Bio:

Derek is a Principal Software Quality Engineer with SAP, in Cambridge, MA, and holds an ASQ CSQE certification. He has 22 years experience in the software quality field working with real time systems, operating systems, image processing systems, and most recently enterprise web applications.

Working in large, medium, and small companies, Derek has introduced software development process improvements, developed automated and manual testing strategies and frameworks, written and executed too many test cases to count, and implemented various tools that help make software testing just a little bit easier.