Welcome to the Jubula tutorial!

- If you haven’t already set up your installation, then come and get a USB stick…

- If you have, pat yourself on the back and relax until we start 😊
Jubula Tutorial

Alexandra Schladebeck
Markus Tiede

BREDEX GmbH
Agenda

- Quick introduction to Jubula
- Exercises: hands-on test specification and execution
- Exercises: automated testing during the project lifecycle
- Exercises: code coverage and reporting
- Further aspects & conclusions
- Show & Tell : Mylyn
Pre-flight checks

- All installed and setup?
- Anyone on Mac with Java 7?
Setup

- Start Jubula
- Open Project: EclipseCon2013_1.0.xml
  - Import via Test → Import
  - Open via Test → Open
- Connect to embedded AUT Agent
- Check for strange red square…
What is Jubula?

- Eclipse Project for automated functional testing (GUI testing)
- Released from core parts of GUIDancer commercial test tool (Eclipse Award Winner 2010)
- GUIDancer == Jubula. Both tools include Code Coverage, Reporting, Mylyn, and Teststyle.
What makes Jubula different?

- Drag and drop test creation
  No recording
  No program code
  According to best practices known from software development

### Programmed tests

- Testers write tests
- Structure
- Transparency for customer
- Independent from AUT

### Recorded tests

- Testers write tests
- Structure
- Transparency for customer
- Independent from AUT

---

Copyright © 2013 BREDEX GmbH. Made available under the Eclipse Public License v1.0.
What can Jubula do?

- User perspective
- Complete workflows
- All layers tested
- Tests through the GUI
Constant feedback about acceptance/regressions

- Why do we test?
- Quality info close to development
  - User perspective at forefront
- Safety net
  - Time to react
Quick demo: how are tests made?

- Step 1: Specify (drag and drop)
- Step 2: Add data and component names
- Step 3: Do Object Mapping
- Step 4: Execute
Version 1

Execute an existing test

Write a simple test

Write a less simple test

Finish off a test and run it
Exercise 0: Run existing test

- Project is imported and opened
- Connected to embedded AUT Agent
- Start AUT for your system in version one:
  - E.g. DVDTool (Win_V1): Win_V1@localhost
- Wait for AUT to start
- Start Test Suite FIRSTTEST
  - Yes to relevant
  - Yes to change perspective
  - Hands off the keyboard!
Exercise 1: Create simple test

- First test consists of actions on the Application and Menu components
  - which means no object mapping 😊
- Use Case:
  - Open waiting dialog from Edit menu
    - Unbound modules -> Actions -> Select -> Menu Bar
  - Wait for dialog
    - Unbound modules -> Actions -> Wait -> Application
  - Wait for dialog to close
    - Unbound modules -> Actions -> Wait -> Application
Exercise 2: Create slightly less simple test

- This time with object mapping as well
- Use Case:
  - Open Help → Info
  - Wait for Info dialog
    - Modules → Generic → Dialogs
  - Check version number
    - ubm → Actions → Check → Component with Text
  - Click OK
    - ubm → Actions → Click

Sounds like we need a module!
Exercise 3: Finish test case for name uniqueness

- Category names should be unique in the DVD Tool
- Open Test Case: Check uniqueness of category names
- Look at what we’ve got already
- Add final test actions:
  - Check enablement of OK button
    - Ubm -> Actions -> Check
  - Close dialog with cancel
    - Modules -> Generic -> Dialogs
- Do mapping and run FULLTEST
Break!

- 10 Minutes ;)
Version 2

Check that previous bug is gone

Make any necessary changes

Write a new test
Exercise 4: Re-run uniqueness test on new version

- Stop Version 1 AUT
- Start Version 2 AUT
- Re-run FULLTEST (to check that the bug is fixed)

- Do any changes need to be made?
Exercise 5: Comment-in deletion test and run

- In the FULLTEST Test Suite, comment in the Delete Single Category Test Case
- Run the test
- Is everything ok?
What do we do about errors?

- Error in startup module → EXIT
  - (e.g. when the menu path changed for load standard data)
- Error within a Use Case in a Test Suite
  - Can we fix it?
    - Yes → Retry Event Handler (later)
    - No / Don’t know → Return Event Handler
      - Use restart / reset to get AUT to correct state
About Event Handlers

- React to errors
  - Action error
  - Component not found
  - Check failed
  - Configuration error

- Continue test
  - Continue
  - Exit
  - Return
  - Retry
  - Break
  - Pause
Why do we need Event Handlers?

- What if an error occurs?
Why do we need Event Handlers?
Why do we need Event Handlers?

Test Execution History

- Unknown quality
- We haven't known about these test steps for a week
- Don't know if previous errors got fixed
- Looks like they didn't
Using Event Handlers
Adding Event Handlers

- Create Test Case
  - Can contain actions, just like any normal test case
    - E.g. restart
- Add the Test Case as an Event Handler to the Deletion Test Case
  - Error type: Action error
  - Reentry type: Return
- Use Case structure is important

..if an action error occurs in this Test Case, then restart and continue with the next Test Case in the hierarchy – it will be ready for you!
Version 3

Check that previous bug is gone

Check that there are no unwanted side effects
Exercise 6: Run test on new version

- Stop version 2
- Start version 3
- (Mac Java 7 users: switch context menu action for menu action)
- Re-run FULLTEST on latest version
Return to Event Handling

- What if the category to delete contains DVDs?
- Check for non-existence of confirmation dialog
  - If it is there, close it and check again

![Confirmation Dialog]

```
The category contains DVDs.
These are deleted likewise.
Would you like to delete the category nevertheless?
```

Retry EH
Exercise 7: Code coverage

- Stop the AUT
- Open the Project Properties
  - Select AUTs from the left
  - Select DVDTtool and select „Edit“
  - Find your system_v3_CC configuration and select „edit“
- Open the Expert properties
- In the Jacoco Configuration
  - Browse to the examples/AUTs/DVDTtool directory on your system for:
    - The installation directory
- Click all the OKs
- Start the AUT configuration for code coverage
Exercise 7: Code coverage

- Run the FULLTEST
- Open the Reporting Perspective
- See the code coverage values
- Open the report

- To discuss over a beer:
  - What should we use code coverage for?
  - http://testing.bredex.de/code-coverage.html

- http://www.eclemma.org/jacoco/
What do we do with the results? Reporting

- Daily test result analysis
  - Hudson: check that all jobs ran
  - Hudson: check that all Test Suites ran ("collectResults" script)
  - Jubula: Test Execution Overview
  - Analyse any tests that failed in ITE or dashboard
  - Add comments to test result summary view for failed tests
  - Print comments report for stand-up / status meeting

- Weekly analysis
  - History (with code coverage) report
  - Test execution histogram
  - Your own reports (use cases, performance, …)
## Daily test result analysis: Hudson

<table>
<thead>
<tr>
<th>Job</th>
<th>Last Success</th>
</tr>
</thead>
<tbody>
<tr>
<td>gdTestCentOS5</td>
<td>15 hr (#1357)</td>
</tr>
<tr>
<td>gdTestiOS</td>
<td>16 hr (#267)</td>
</tr>
<tr>
<td>gdTestITE</td>
<td>15 hr (#609)</td>
</tr>
<tr>
<td>gdTestMac</td>
<td>15 hr (#1322)</td>
</tr>
<tr>
<td>gdTestVista</td>
<td>15 hr (#1351)</td>
</tr>
<tr>
<td>gdTestWin7</td>
<td>15 hr (#1307)</td>
</tr>
<tr>
<td>gdTestWin8</td>
<td>15 hr (#235)</td>
</tr>
</tbody>
</table>

**echo.noReport:**

[echo] ################################################################################
[echo] The following 0 out of 184 test suites did not write a report:
[echo] ################################################################################
[echo] ################################################################################
Daily test result analysis: Test Execution Overview
Daily test result analysis: Failed tests analysis

<table>
<thead>
<tr>
<th>Test Suite</th>
<th>AUT OS</th>
<th>Duration</th>
<th>%</th>
<th>COMMENT TITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FULLTEST-Win</td>
<td>Windows Vista</td>
<td>02:18:15</td>
<td>99.94%</td>
<td>Swing - Button - Select Context Menu Entry by Textpath (Specify Position); Action Error - Not found, though the context menu is visible on screenshot</td>
</tr>
<tr>
<td>FULLTEST-Win</td>
<td>Windows Vista</td>
<td>02:50:37</td>
<td>0.02%</td>
<td>aa_complex_waitToStartup, activate AUT, test hung up for 1:50:32h and ran into timeout</td>
</tr>
<tr>
<td>FULLTEST-86centos5</td>
<td>Linux</td>
<td>02:03:09</td>
<td>38.04%</td>
<td>Check that Connect to AUT Agent is enabled, check for toolbaritem &quot;localhost 2001&quot; failed - after EH (editing preferences) failed again</td>
</tr>
</tbody>
</table>
Weekly / long-term analysis
Exercise 8 : Reporting

- Add a comment to a failed test in the reporting perspective
- Open the comments report
  - Yesterday – today
- See your comment 😊

BIRT reports:
- Only look at “relevant” tests
- Comments report ignores BROKEN

- http://www.eclipse.org/birt/phoenix/
Further information: From the command line

- Continuous integration and testing
  - Check out sources
  - Build software
  - Prepare test environment
  - Install software (AUT) on test systems
  - Run tests – Test Executor (testexec)

- Check results
  - HTML / XML
  - In ITE
  - In Dashboard

```xml
<configuration>
  <project>YourProjectName</project>
  <version>1.0</version>
  <autconfig>YourAUTConfiguration</autconfig>
  <dbscheme>YourDatabaseSchemeName</dbscheme>
  <dbuser>YourDatabaseUsername</dbuser>
  <dbpw>YourDatabasePassword</dbpw>
  <server>YourAUTAgentHostname</server>
  <port>PortNumberForAUTAgent</port>
  <language>en_US</language>
  <testsuite>TestSuite</testsuite>
  <datadir>DirectoryWhereExternalDataIsStored</datadir>
  <resultdir>DirectoryToStoreTestResultFiles</resultdir>
</configuration>
```
Summary

- Complete workflows tested based on acceptance criteria
  - Even before application available
  - Automated regression testing – quick feedback
  - Whole team can participate, discussion basis for stakeholders
- Manual intelligence incorporated
  - Synchronization
  - Reacting to clues in the application
  - Error handling
- Reuse instead of copy
  - Tests made up of similar units
  - Reusing ensures structure & maintainability
  - Central place for any changes
- Testing is important
  - So do it right!
Jubula project status

- Current release:
  - BREDEX release: 7.2
  - Next release: 8.0, Kepler SR2
- Keep updated:

http://testing.bredex.de

Module examples, articles, presentations, ...

@EclipseJubula

www.bredex.de
Further information

Community

- [http://www.eclipse.org/jubula](http://www.eclipse.org/jubula)
- Forum
- Webinar
- Mailing lists

Professional services

- [http://testing.bredex.de](http://testing.bredex.de)
- Proof of Concept
- Appetizer packages
- Training
- Workshops
- Support
- Test reviews
- Best practices
- Customized extensions
- Sponsored development
Show and Tell: ALM with Mylyn

- View and edit tasks from repository in ITE
- Create tasks directly from test result reports
- Automatically comment on tasks in external repositories after a test run

- [www.eclipse.org/mylyn](http://www.eclipse.org/mylyn)
View and edit tasks from ITE
Create tasks directly from test result reports
Automatically comment on task after test run

Disable OK button when category name already exists

Details
Type: Story  Priority: P3  Labels: None  Story Points: 1
Status: In Progress (View Workflow)  Resolution: Unresolved

Description
Categories can be created in the DVD Tool. To make it easier to keep an overview of the structure, it should not be possible to create two categories with the same name.
In the "Add New Category" dialog, if a category name is entered that already exists, the OK button should be disabled.

Activity
- [ ] Alexandra Schladebeck added a comment - 4 minutes ago
  Test results from Thu Oct 24 08:42:02 CEST 2013 for Test Case "UC: Test that no two categories can have the same name": status failed http://localhost:11270/dashboard7/summary?tid=6293&resultcode=2

- [ ] Alexandra Schladebeck added a comment - 3 minutes ago
  The test shows that the button remains enabled, even if the name already exists.

- [ ] Alexandra Schladebeck added a comment - 2 minutes ago
  Test results from Thu Oct 24 08:44:18 CEST 2013 for Test Case "UC: Test that no two categories can have the same name": status passed http://localhost:11270/dashboard7/summary?tid=6314&resultcode=2
Join us at other talks!

- **Usability and testability: two unloved birds with one stone**
  - Wednesday, 15:15 - 15:50 Bürgersaal 2

- **Testing mobile applications with Jubula**
  - Wednesday, 17:00 - 17:35 Theater Stage
  - [http://www.eclipsecon.org/europe2013/testing-mobile-applications-jubula](http://www.eclipsecon.org/europe2013/testing-mobile-applications-jubula)

- **To infinity and beyond – making Jubula surpass its limits**
  - Thursday, 10.30-11.05 Wilhelm-Krämer-Zimmer

- **Hackathon**

- **And don‘t forget to evaluate the talk!**
Evaluate!

- Excellent
- Very good
- Good
- Average
- Poor

☑ Excellent
Exercise 9 : Add the Return Event Handler

- Create a Test Case called „Restart Application“
- Add the Test Case as an Event Handler to the Deletion Test Case
  - Error type: Action error
  - Reentry type: Return
- Run the FULLTEST on version 2
Exercise 10: Continue writing tests

- If we have time:
  - Startup
  - Import DVD library
  - Select category (Science Fiction)
  - Select DVD from table
  - Select technics tab
  - Select region
  - Select languages