

# Jubula Tutorial: Round 3

Alexandra Schladebeck

Achim Lörke

Zeb Ford-Reitz

BREDEX GmbH



# Agenda

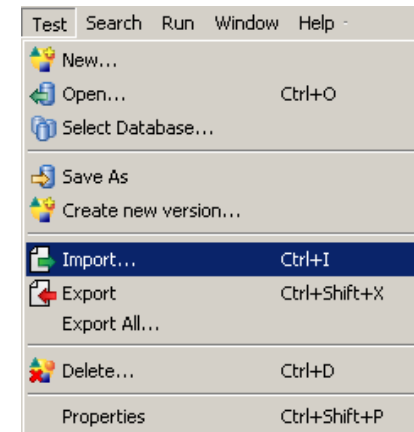
- Quick introduction to Jubula and short demo
- Exercises: hands-on test specification and execution
- Exercises: automated testing during the project lifecycle
- Further aspects
- Questions

# Pre-flight checks

- Firewalls off
- Write access to your home directory
- Default Java version is 1.6.x\_xx

# Setup

- Installation: *setup* file
- Start Jubula
- Import Project: *EclipseCon2012\_1.0.xml*
  - Via Test → Import
- Connect to embedded AUT Agent

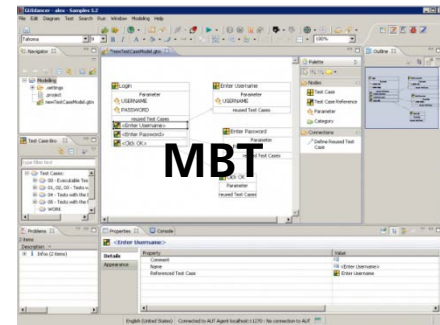
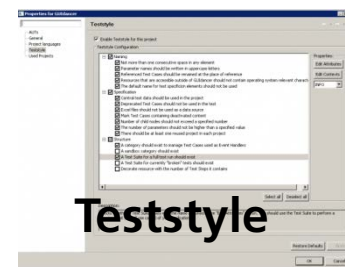
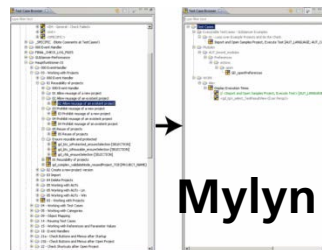


# What is Jubula?

- Eclipse Project for automated functional testing
- Released from core parts of *GUIDancer* commercial test tool (Eclipse Award Winner 2010)
- *GUIDancer* = Jubula +

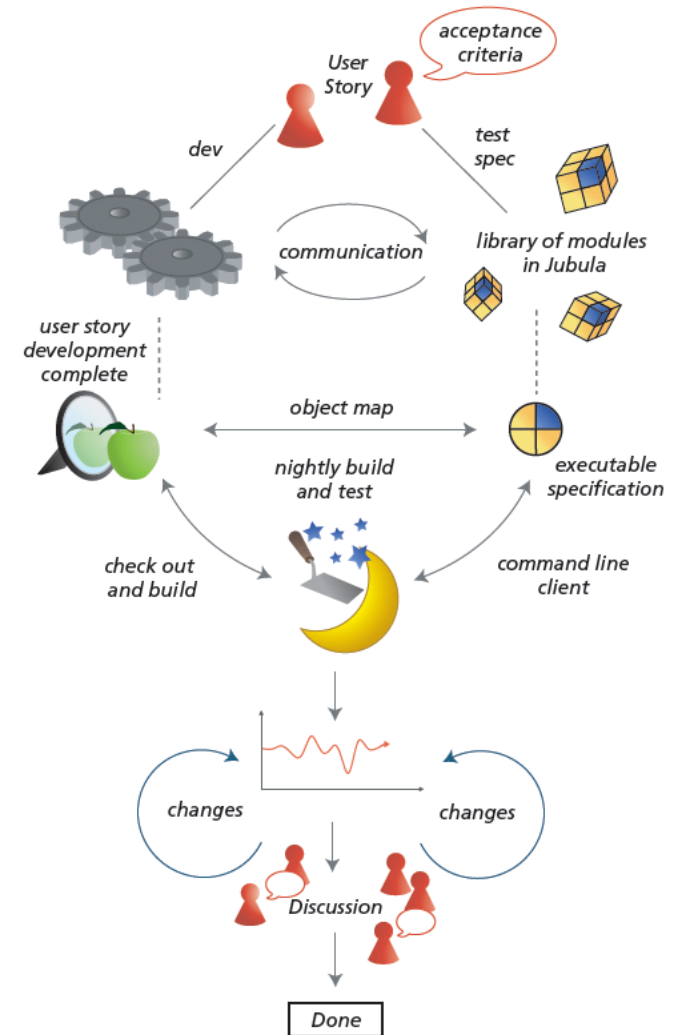


Element	Missed Instructions %	Missed Branches %	Cov	Missed	Methods	Missed	Classes
org.eclipse.jubula.examples.and.dvttool.console	21%	22%	156	202	29	40	
org.eclipse.jubula.examples.and.dvttool.gui	6%	43%	63	94	3	12	
org.eclipse.jubula.examples.and.dvttool.model	52%	16%	27	73	3	6	
org.eclipse.jubula.examples.and.dvttool.persistence	0%	0%	0	0	4	4	
org.eclipse.jubula.examples.and.dvttool.resources	4%	1%	0%	0	0	1	
org.eclipse.jubula.examples.and.dvttool	0%	n/a	2	2	1	1	
<b>Total</b>	<b>0%</b>	<b>0%</b>	<b>234</b>	<b>310</b>	<b>306</b>	<b>41</b>	<b>65</b>



# What makes Jubula different?

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



## What can Jubula do?

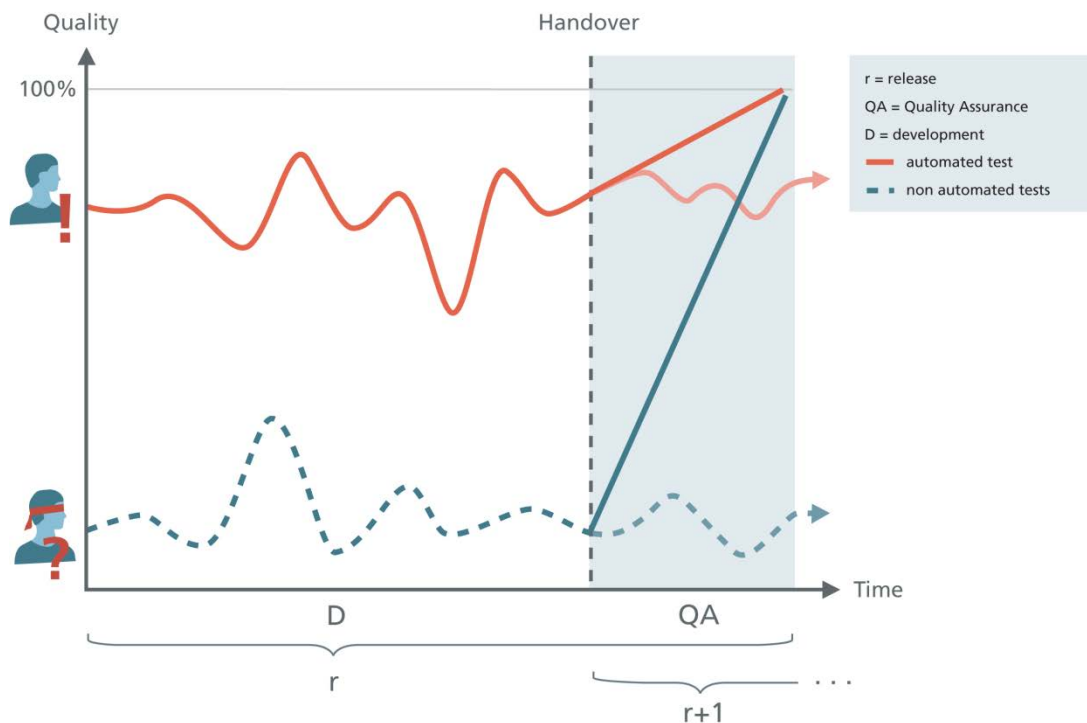
- User perspective
- Complete workflows
- All layers tested



Photo Courtesy PDPhoto

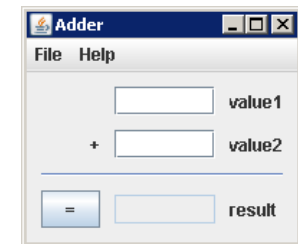
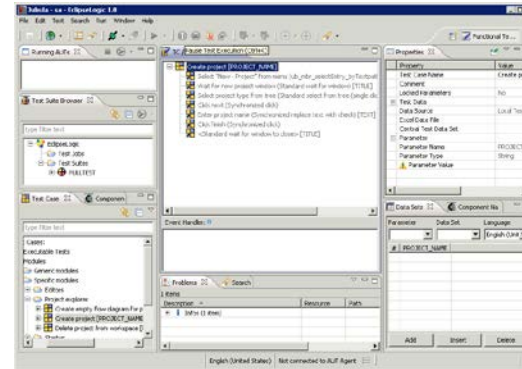
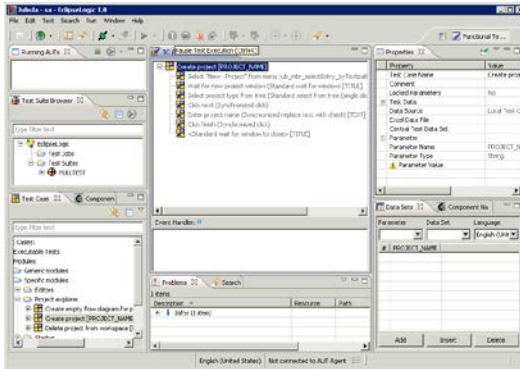
# Constant feedback about acceptance

- Quality info close to development
  - User perspective at forefront
- Safety net





# Quick demo: Jubula tests Jubula



## Quick demo: how are tests made?

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

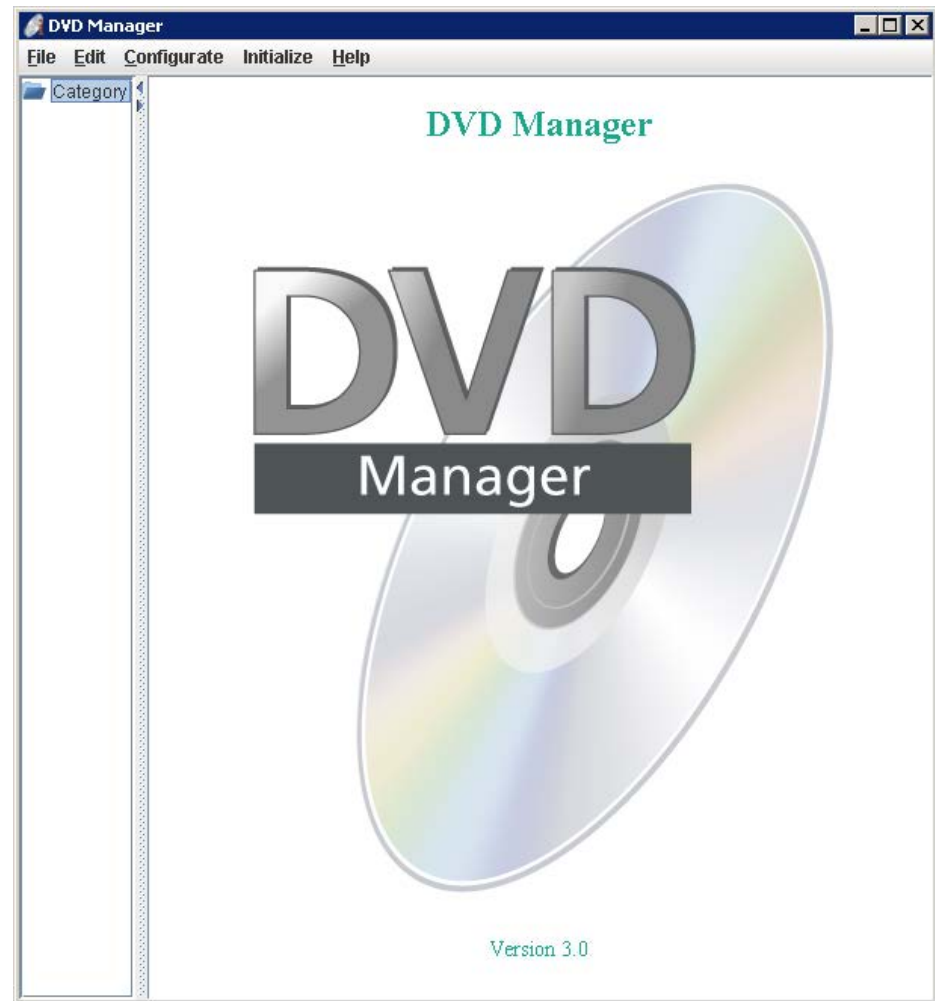
## Version 1

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* component
  - → which means no object mapping ☺
- Use Case:
  - Open waiting dialog from Edit menu
  - Wait for dialog
  - Wait for dialog to close

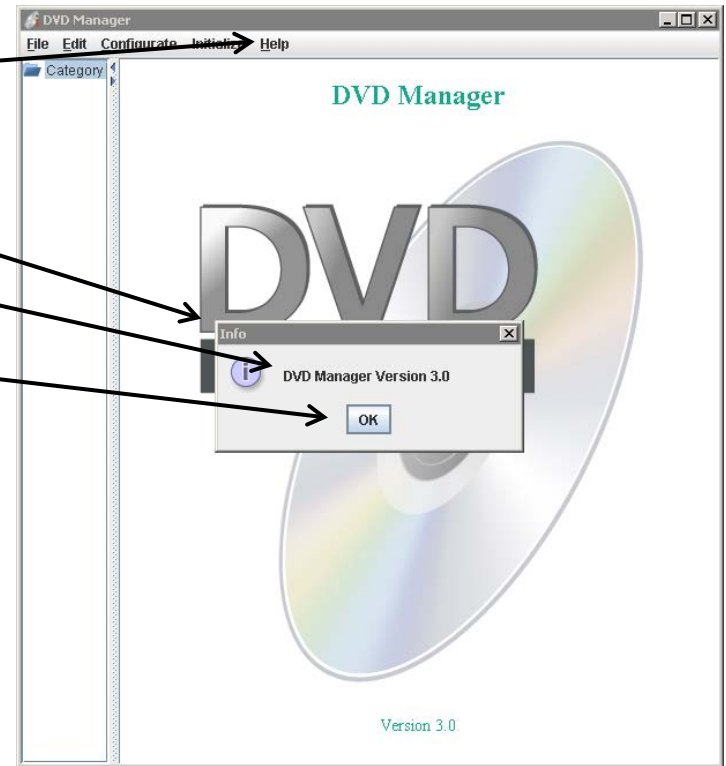


## Exercise 2: Create slightly less simple test

- This time with object mapping as well
- Use Case:

Sounds like we need a module!

- Open Help → Info
- Wait for Info dialog
- Check version number
- Click OK



## Exercise 3: Finish test case for name uniqueness

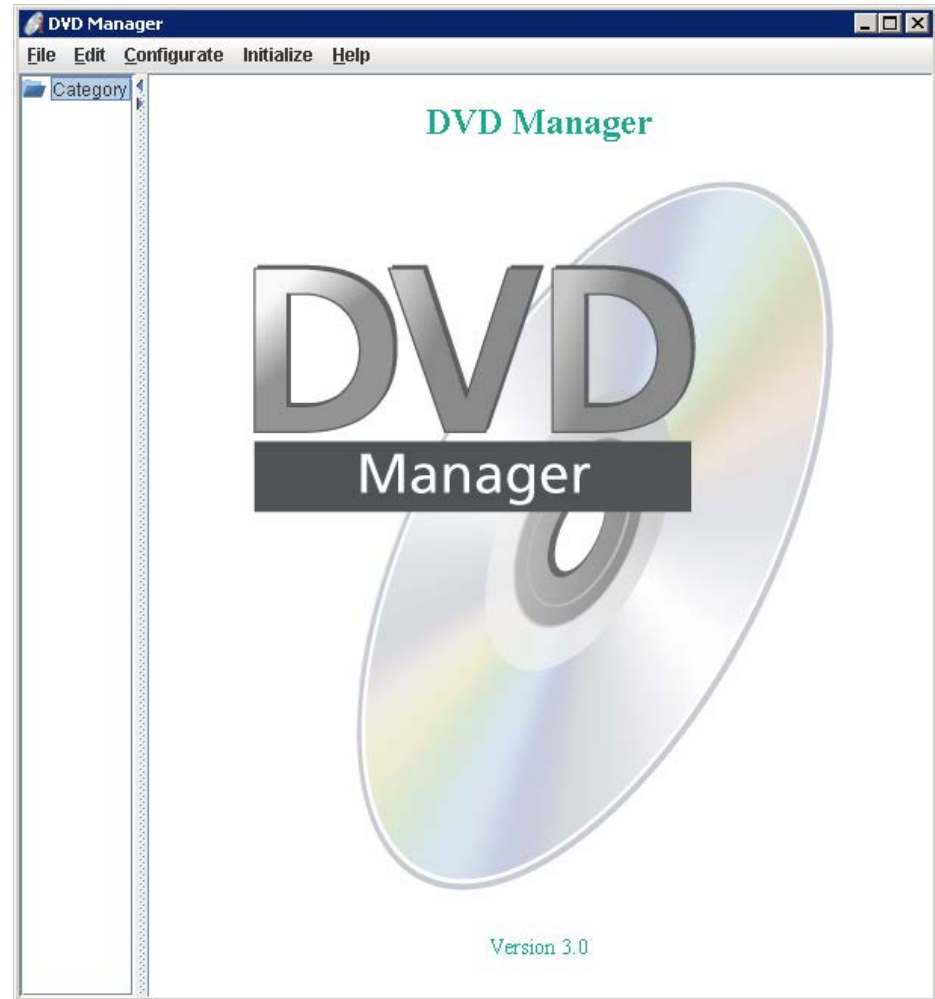
- Category names should be unique in the DVD Tool
- Open Test Case: Check uniqueness of category names
- Look at what we've got already
- Add final test actions:
  - Check enablement of OK button
  - Close dialog with cancel
- Do mapping and run FULLTEST

## 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?

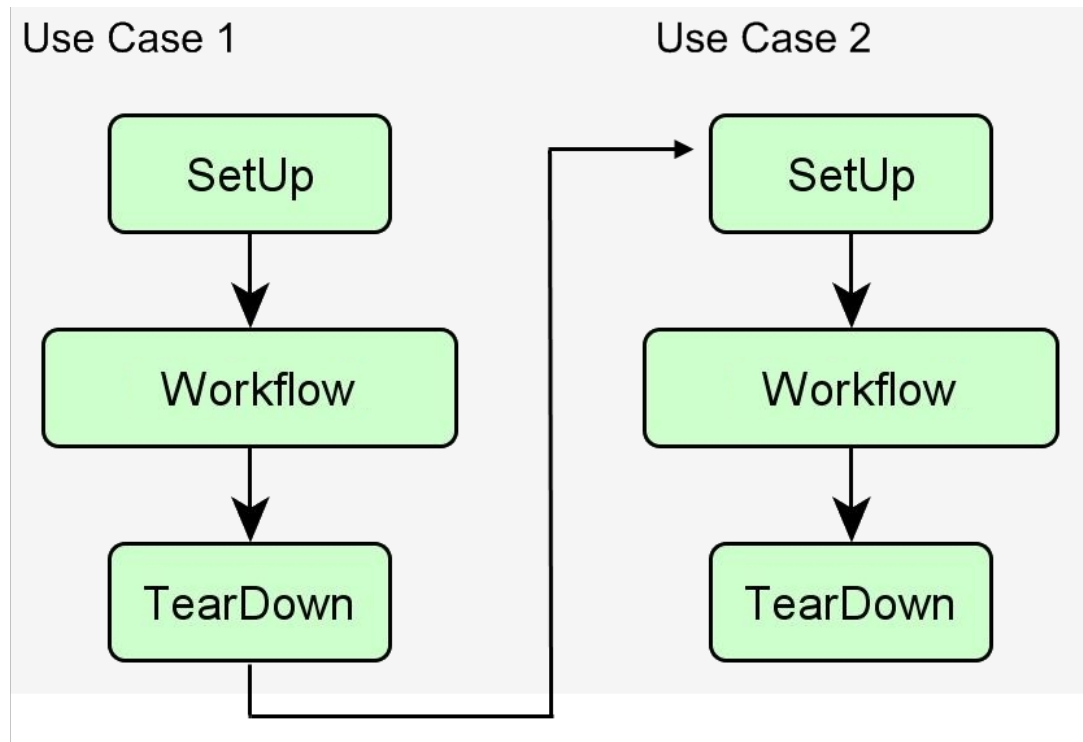
- Error in startup module → EXIT
  - (e.g. when the menu path changed for load standard data)
- Error within a Use Case in a Test Suite
  - Can we fix it?
    - Yes → Retry Event Handler
    - No / Don't know → Return Event Handler

# About Event Handlers

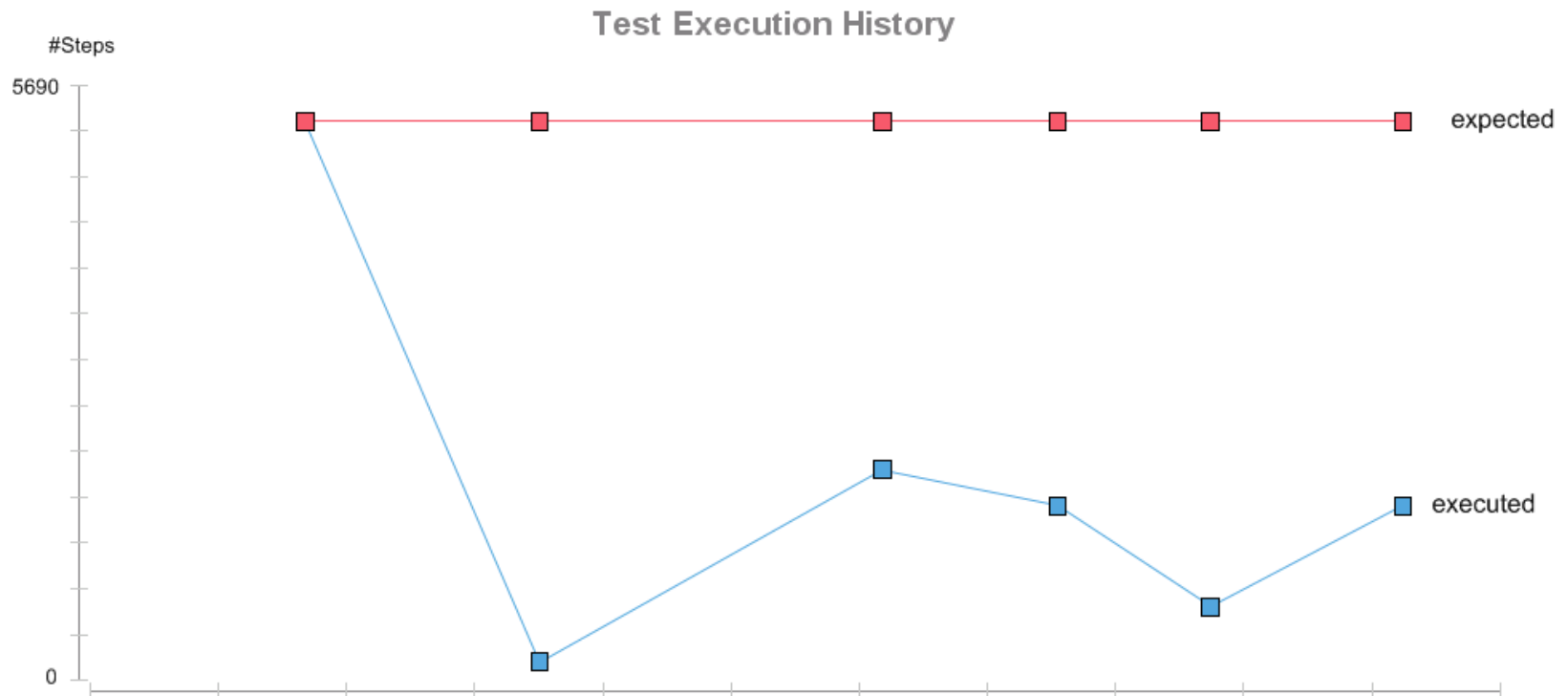
- React to errors
- Action error
- Component not found
- Check failed
- Configuration error
- Continue test
- Continue
- Exit
- Return
- Retry
- Break
- Pause

# Why do we need Event Handlers?

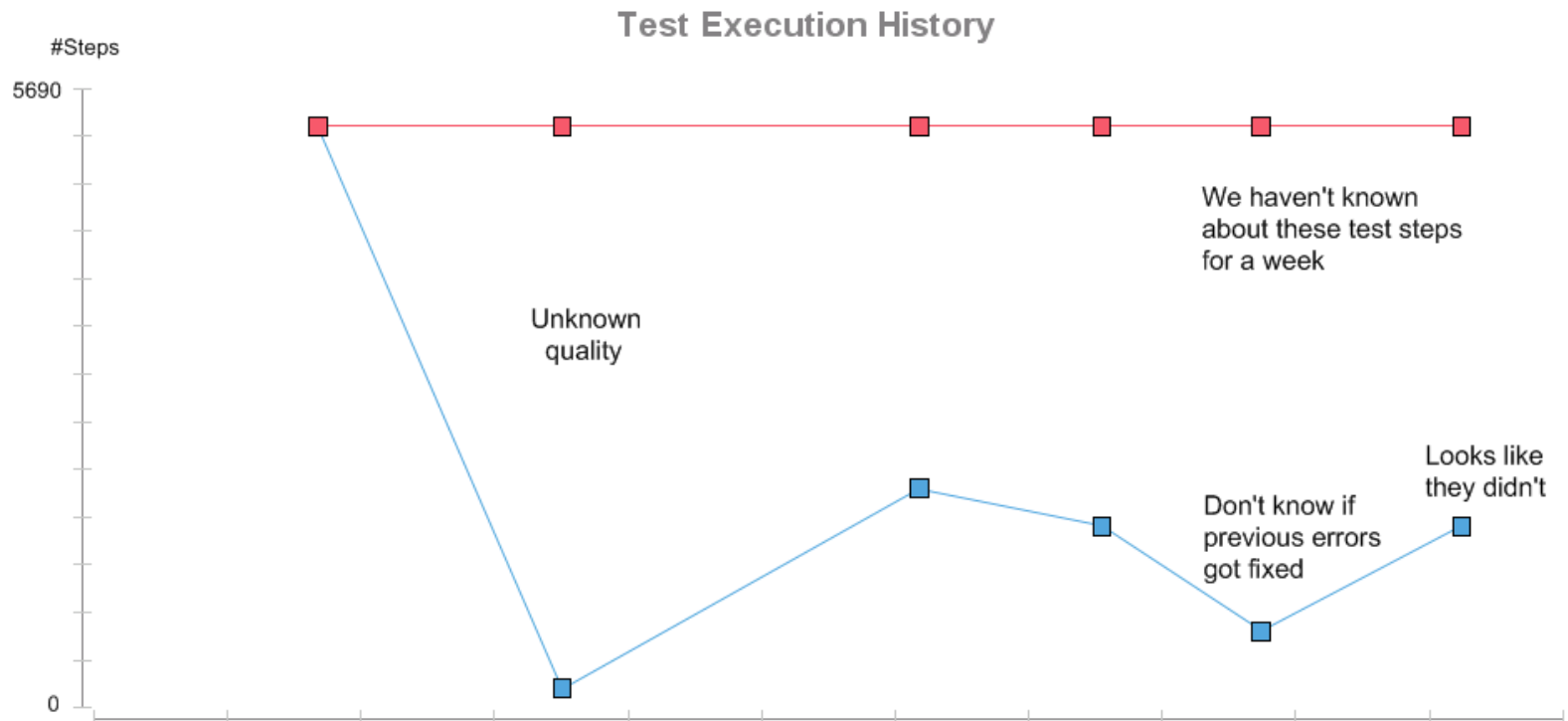
- What if an error occurs?



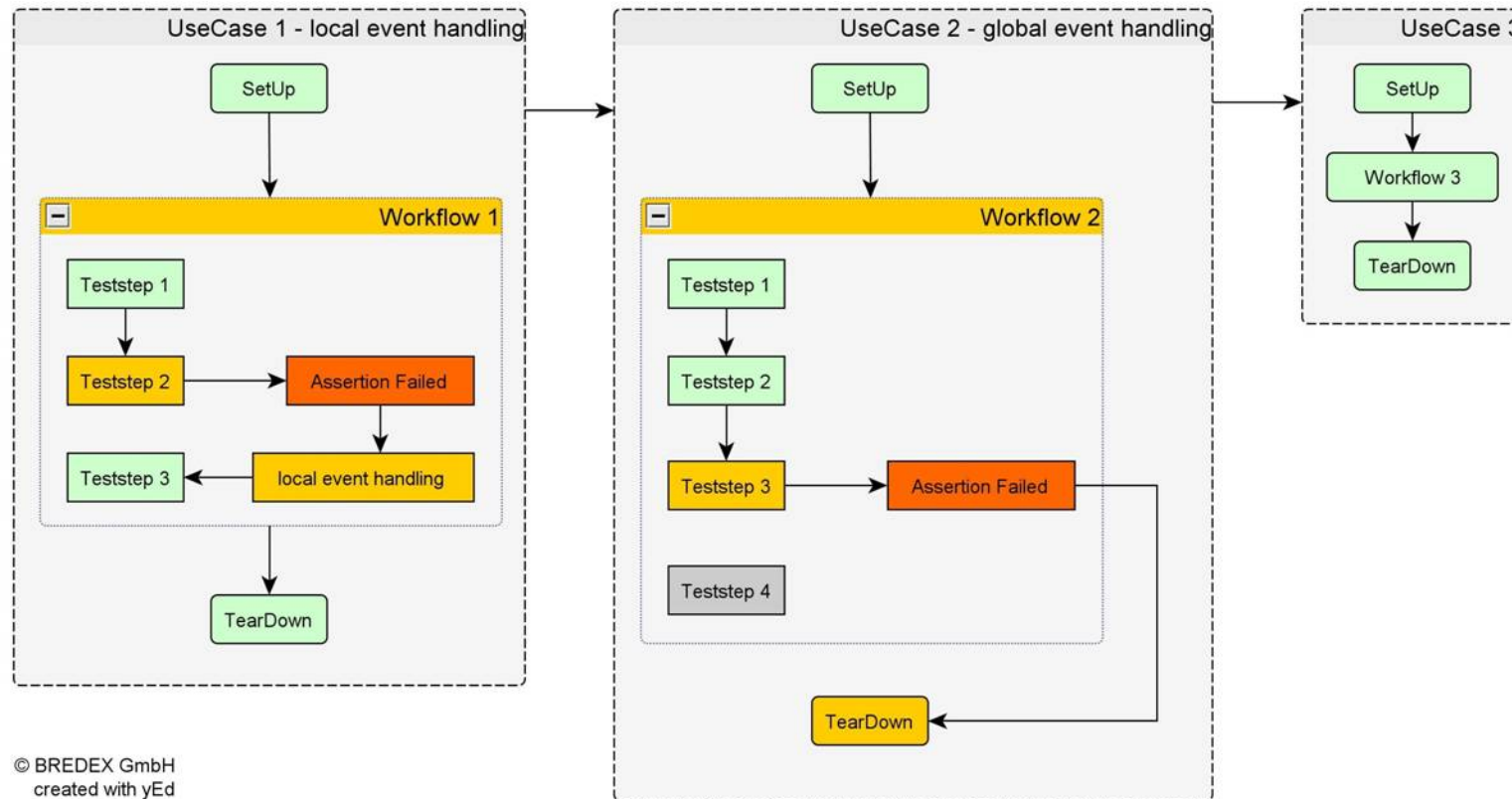
# Why do we need Event Handlers?



# Why do we need Event Handlers?



# Using Event Handlers



© BREDEX GmbH  
created with yEd



## Exercise 6: Add return Event Handler

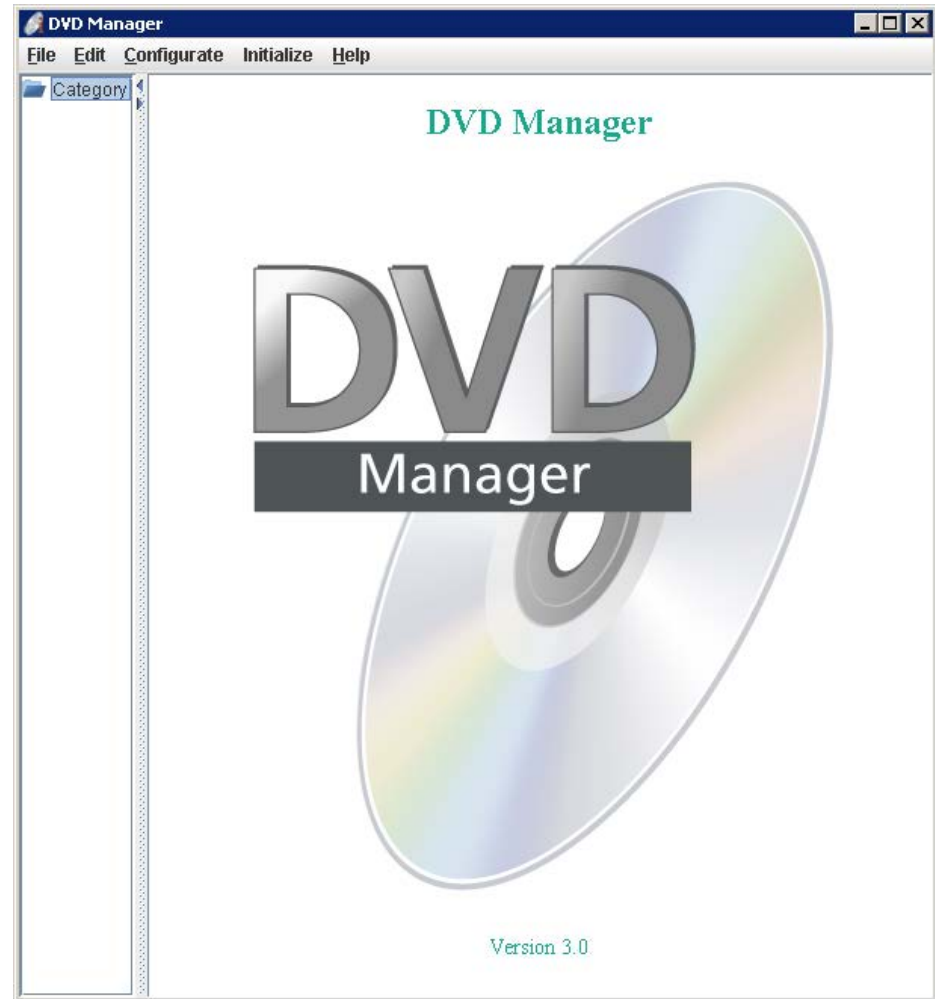
- Create empty Test Case
- Add the Test Case as an Event Handler to the Deletion Test Case
  - Error type: Action error
  - Reentry type: Return

*..if an action error occurs in this Test Case, then do nothing, but 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

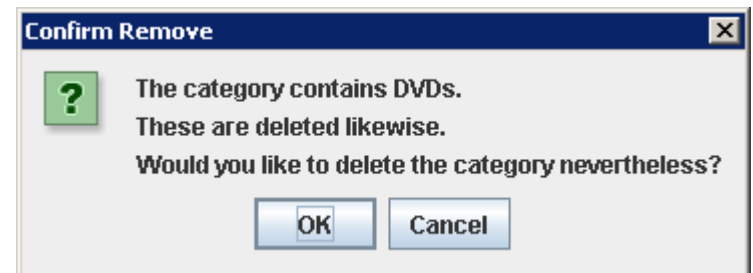


## Exercise 7: Run test on new version

- Stop version 2
- Start version 3
- Re-run FULLTEST on latest version

# Brief 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
- } Retry EH



## Exercise 7: 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

## Further information: From the command line

- Continuous integration
  - Check out sources
  - Build software
  - Prepare test environment
  - Install software (AUT) on test systems
  - Run tests – **Test Executor**
  - Check results
    - HTML / XML
    - In ITE

```
<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
- 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

## Jubula project status

- Current release:
  - BREDEX release: 1.2 Standalone, 6.0 GUI *dancer*
  - Next release: Juno (1.2 features also in Eclipse Jubula)
- Keep updated:  
@EclipseJubula





# Further information

## Community

- **<http://www.eclipse.org/jubula>**
- Forum
- Webinar
- FAQs
- Videos
- Documentation
- Mailing lists

## Professional services

- **<http://www.bredexsw.com>**
- Proof of Concept
- Appetizer packages
- Training
- Workshops
- Support
- Test reviews
- Best practices
- GUI *dancer* Demo Licenses
- Customized extensions

## And finally...

- Special offer on training for Eclipse Con Tutorial participants:
  - Leave your business card and get 10% off any Jubula or *GUIdancer* training for your team (valid until 30th June 2012)
- Eclipse Testing Days 2012
  - Call for Sponsors
  - Call for Papers
  - [http://wiki.eclipse.org/Eclipse\\_Testing\\_Day\\_2012](http://wiki.eclipse.org/Eclipse_Testing_Day_2012)
- Tell us how you test with Jubula and get a goodie!


# Give Feedback on the Sessions

1

Sign In: [www.eclipsecon.org](http://www.eclipsecon.org)

2

Select Session Evaluate

Making ALM Work -   
Transform your Application Lifecycle Management to Foster Innovation (presented by HP)  
Ronit [HP]

EVALUATE

3

Vote

+1

0

-1