



The Eclipse Jubula Project

Moving away from Capture-Replay

Alexandra Schladebeck, BREDEX GmbH





Agenda

- Introduction to Jubula
 - Concepts and comparisons
- Demonstration
- Jubula in the team and development process

Jubula

- "Born" in 2011 from core parts of *GUIDancer*

Jubula (open source)	<i>GUIDancer</i> (commercial)
<p>Test creation Test execution Test analysis</p> 	<p>Jubula features+ Code coverage Reporting Teststyle for test quality Dashboard in browser Mylyn for task-focussing Modelling</p> 



Jubula's aims

- Functional testing through the GUI

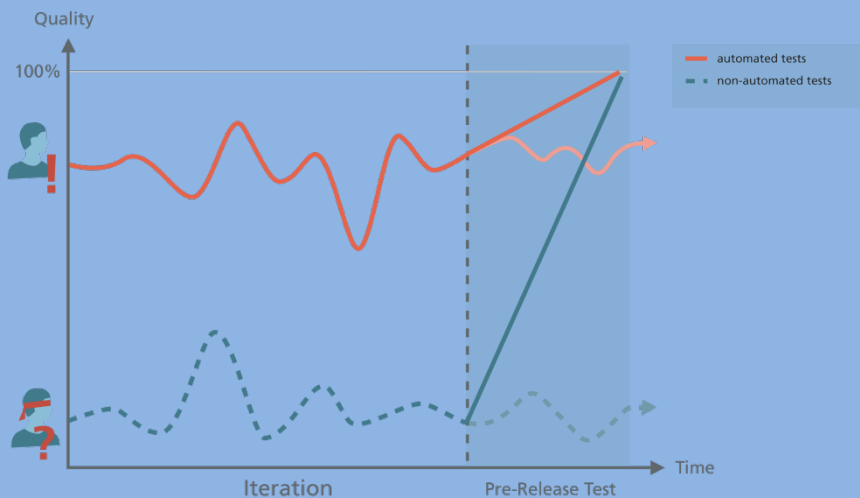


- Customer perspective
- User inputs
- All layers tested
- Business processes
- Workflows

Jubula's aims

- Acceptance testing → regression tests

- Acceptance testing
 - Whole team approach
 - Inclusion of testing from beginning (and ATDD)
 - Discussion about acceptance encouraged
- Regression testing
 - Constant quality information
 - Safety net
 - Shorter release cycles





Writing good tests through the GUI

Recurring actions
stored in libraries

Recurring
sequences saved
as reusable
modules

Readability
through natural
language naming

Data and object
map separate from
test flow

Abstract
definitions where
possible

Transparency for
testers and
customers

Test creation in
parallel with
development



Why not capture-replay?

Recurring actions
stored in libraries

Recurring
sequences saved
as reusable
modules

Readability
through natural
language naming

Data and object
map separate from
test flow

Abstract
definitions where
possible

Transparency for
testers and
customers

Test creation in
parallel with
development



Why not capture-replay?

These steps *must* be performed to make the test maintainable

Recorded tests are difficult to understand and can only be adapted by programmers

The effort ends up being more than if the test were written well from the outset



Why not tests in program code?

Recurring actions
stored in libraries

Recurring
sequences saved as
reusable modules

Readability through
natural language
naming

Data and object
map separate from
test flow

Abstract
definitions where
possible

Transparency for
~~testers and~~
customers

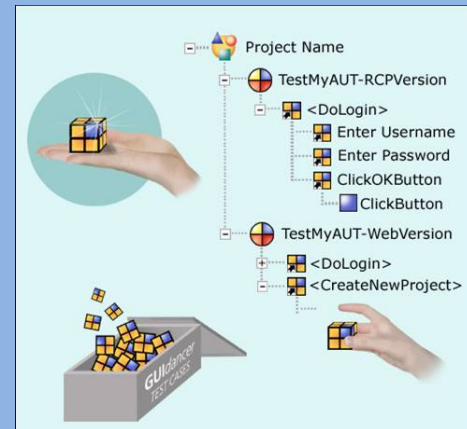
Test creation in
parallel with
development

Jubula in the middle

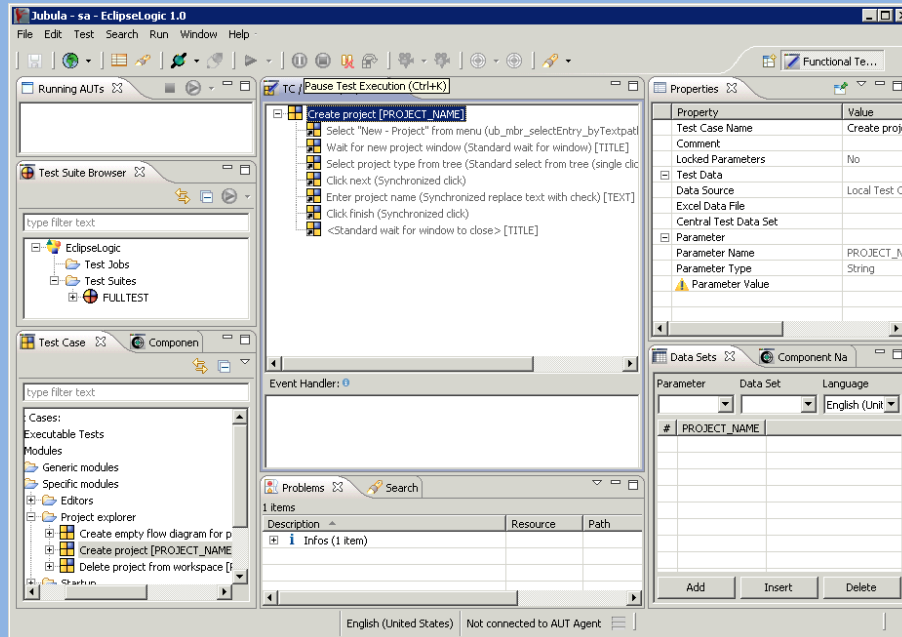
Capture-Replay	Jubula	Wiki-based (code)
Bad test structure	<i>Good test structure</i>	Good test structure
Testers write tests (at beginning)	<i>Testers write tests at all times</i>	Testers do not write tests
Transparency for customer	<i>Transparency for customer</i>	No transparency for customer

Jubula's concepts

- Keyword-driven testing
 - No recording, no programming
 - But using best practices from development
- Separation of AUT and test specification
 - Test creation in advance or in parallel
- "Directly" executable specifications
 - Transparency for testers and customer



Demonstration



The screenshot shows the Eclipse IDE interface with the JUnit GUI (Jubula - sa - EclipseLogic 1.0) open. The main window displays a test suite named 'FULLTEST' and a test case named 'Create project [PROJECT_NAME]'. The test case steps are:

- Select "New - Project" from menu (lb_mbr_selectEntry_byTextpad)
- Wait for new project window (Standard wait for window) [TITLE]
- Select project type from tree (Standard select from tree (single click))
- Click next (Synchronized click)
- Enter project name (Synchronized replace text with check) [TEXT]
- Click finish (Synchronized click)
- <Standard wait for window to close> [TITLE]

The Properties window on the right shows the following details for the test case:

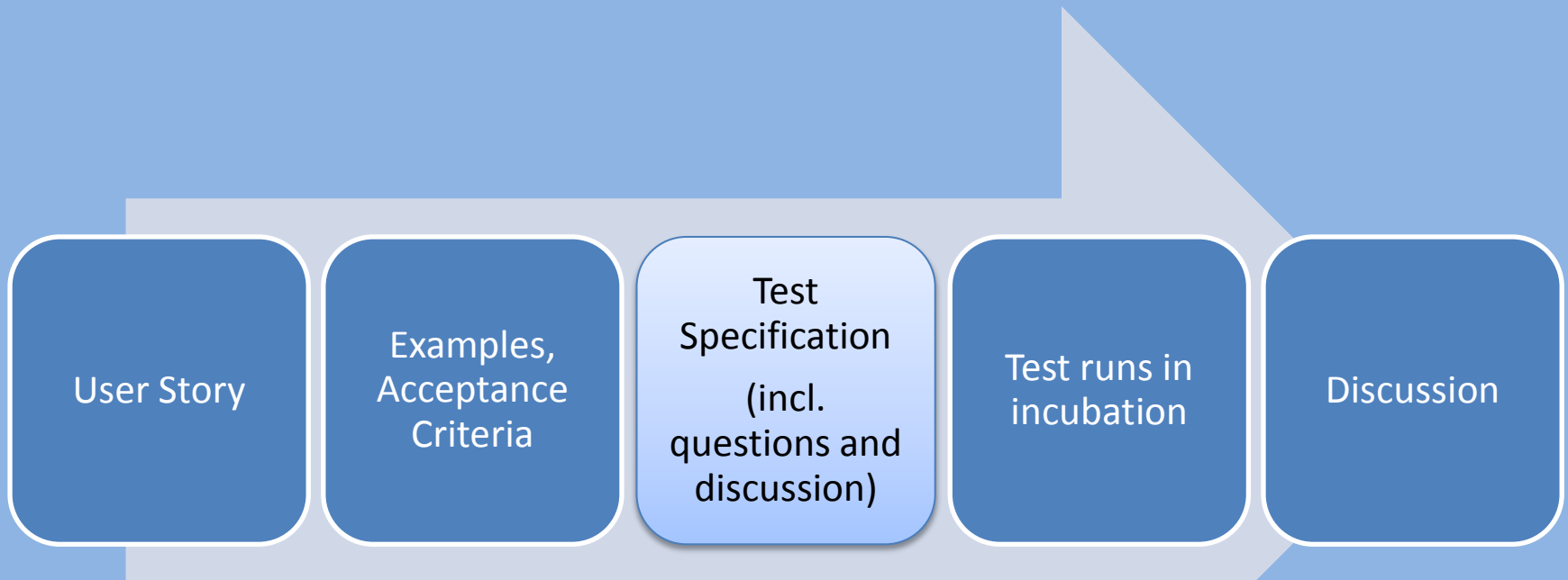
Property	Value
Test Case Name	Create project [PROJECT_NAME]
Comment	
Locked Parameters	No
Test Data	
Data Source	Local Test Case
Excel Data File	
Central Test Data Set	
Parameter	
Parameter Name	PROJECT_NAME
Parameter Type	String
Parameter Value	

The Data Sets window shows a table with the following data:

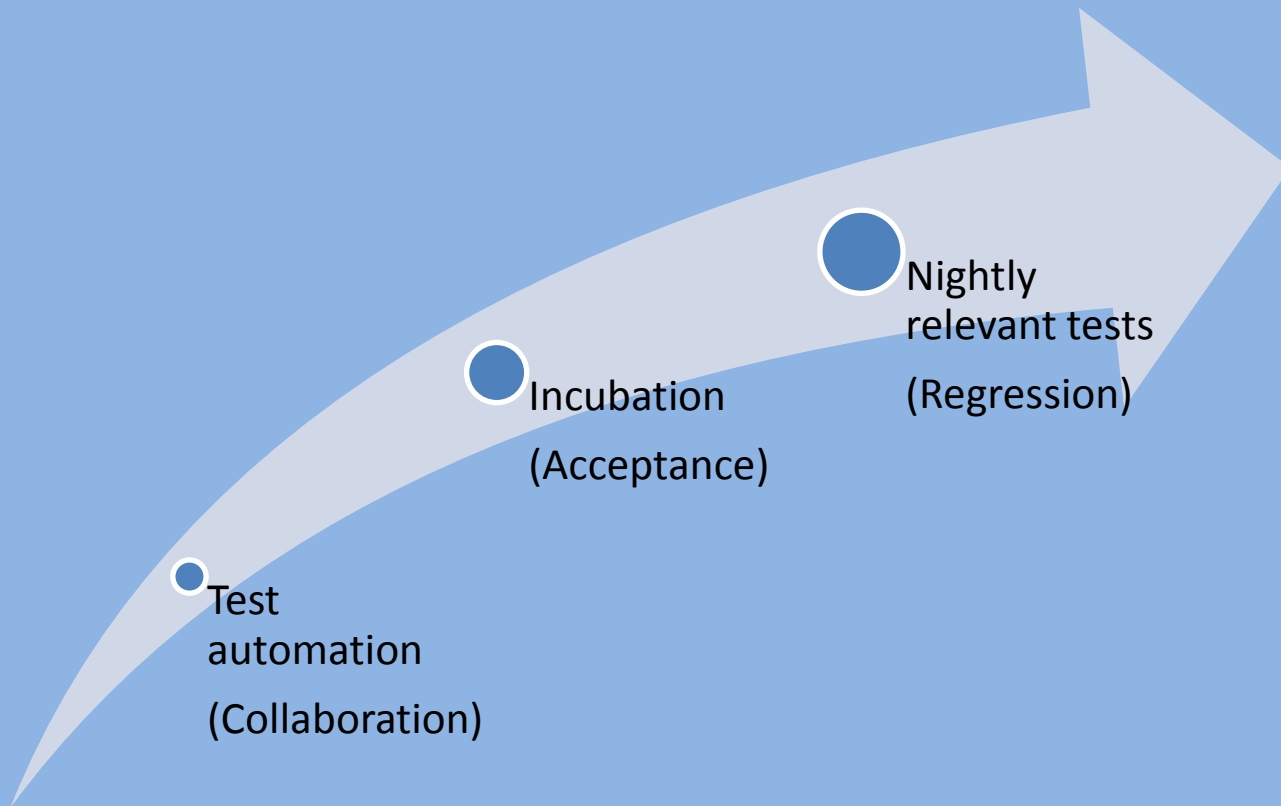
Parameter	Data Set	Language
# PROJECT_NAME		English (United States)

The bottom status bar indicates the language is set to English (United States) and the system is not connected to the AUT Agent.

Jubula in the team



Jubula in the development process





Summary

- Testing through the GUI is important
 - And it doesn't have to be a pain!
- Stick to same best practices as in other areas
- Transparency for testers and customers
 - Testers should write the tests
 - Test result reports should be understandable by all



Thank you!

- More information
 - www.eclipse.org/jubula
 - www.guidancer.com
- @EclipseJubula