



Acceptance testing for Eclipse Applications

Why? What? How?

Alexandra Imrie BREDEX GmbH

Agenda

- ▶ **Observations on test and development process**
- ▶ **The role of acceptance testing**
- ▶ **Methods of acceptance testing**
- ▶ **Introducing and improving acceptance testing**

BREDEX GmbH

- ▶ **Software developing and consulting**
Individual customer projects
Focus on Java since 1995
- ▶ **Agile processes**
- ▶ **Test enthusiasts**
Test tool *GUldancer*
- ▶ **Eclipse foundation members since 2006**



Observations

- ▶ Where does acceptance testing fit in?





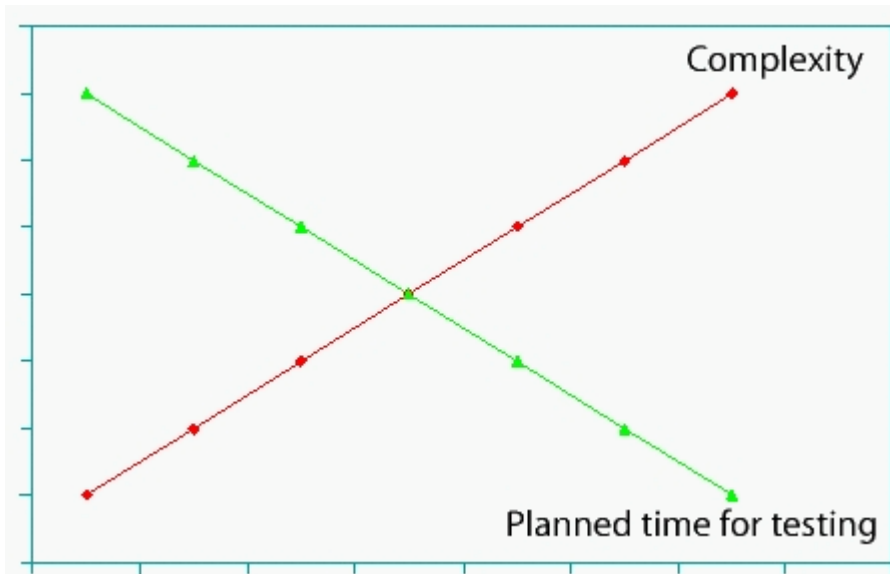
Observations

- ▶ Where does acceptance testing fit in?



Trends in testing

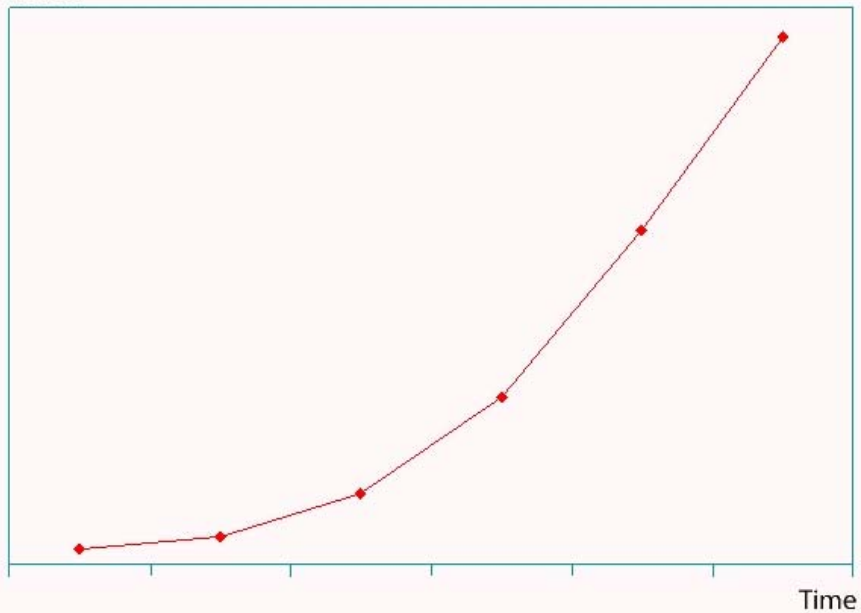
► Complexity and test phase



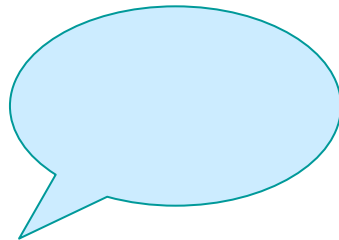
Trends in testing

► Stress towards release

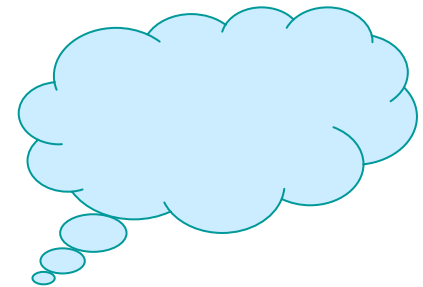
Stress levels



Responses



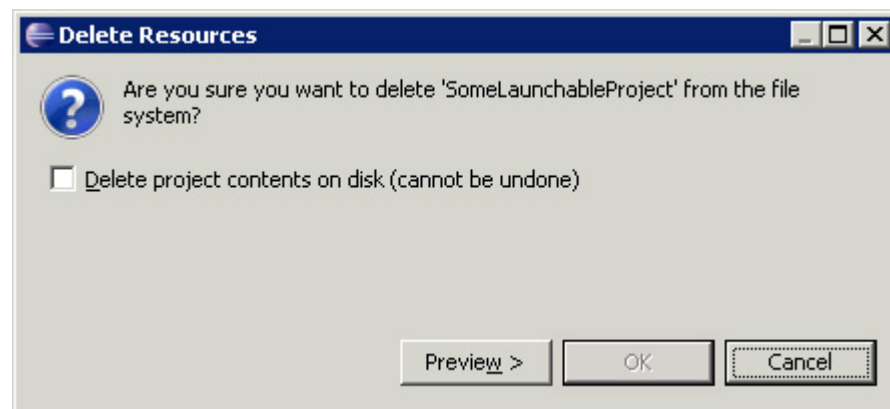
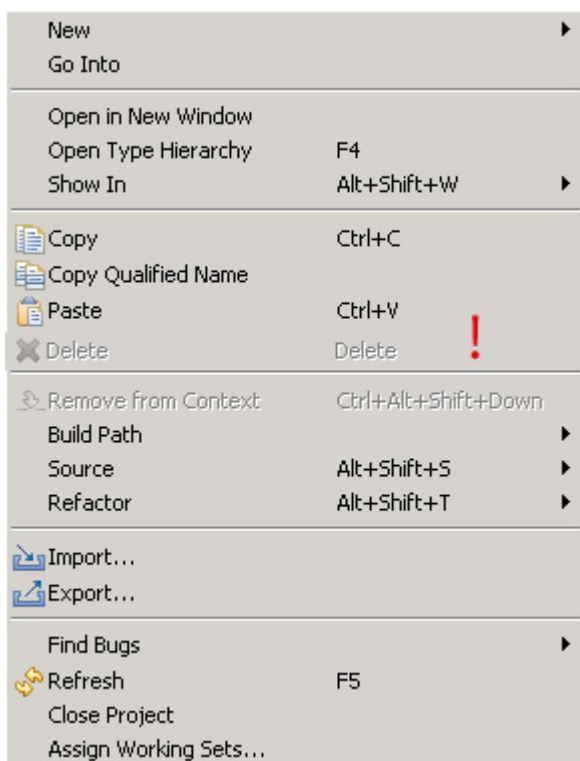
- ▶ **Where did that come from?**
- ▶ **How come this error wasn't noticed?**



- ▶ **Desire for quicker feedback on errors**
- ▶ **Move towards test first with JUnit**

But JUnit isn't an acceptance test

▶ JUnit test for delete project successful, but...

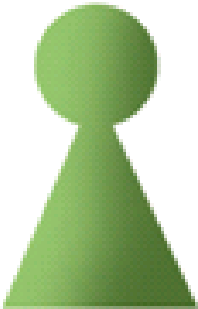




Paradox

▶ **Who do we develop software for?**

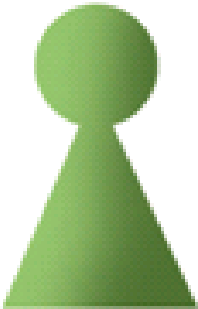
▶ **Whose perspective do we tend to neglect the most?**



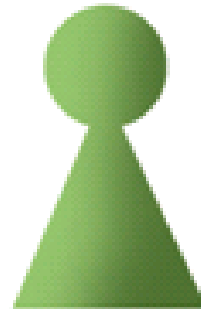


Paradox

▶ **Who do we develop software for?**



▶ **Whose perspective do we tend to neglect the most?**



The role of acceptance testing

- ▶ **Did we do what the customer asked for?**
- ▶ **Has the customer changed his mind?**
- ▶ **Did the customer have implicit assumptions?**
- ▶ **Are there errors or problems in the workflow?**

Example

▶ Acceptance test to create and delete a Java Project in Eclipse

Prerequisite: The project does not already exist

Open new project dialog

Choose Java as the project type

Click Next

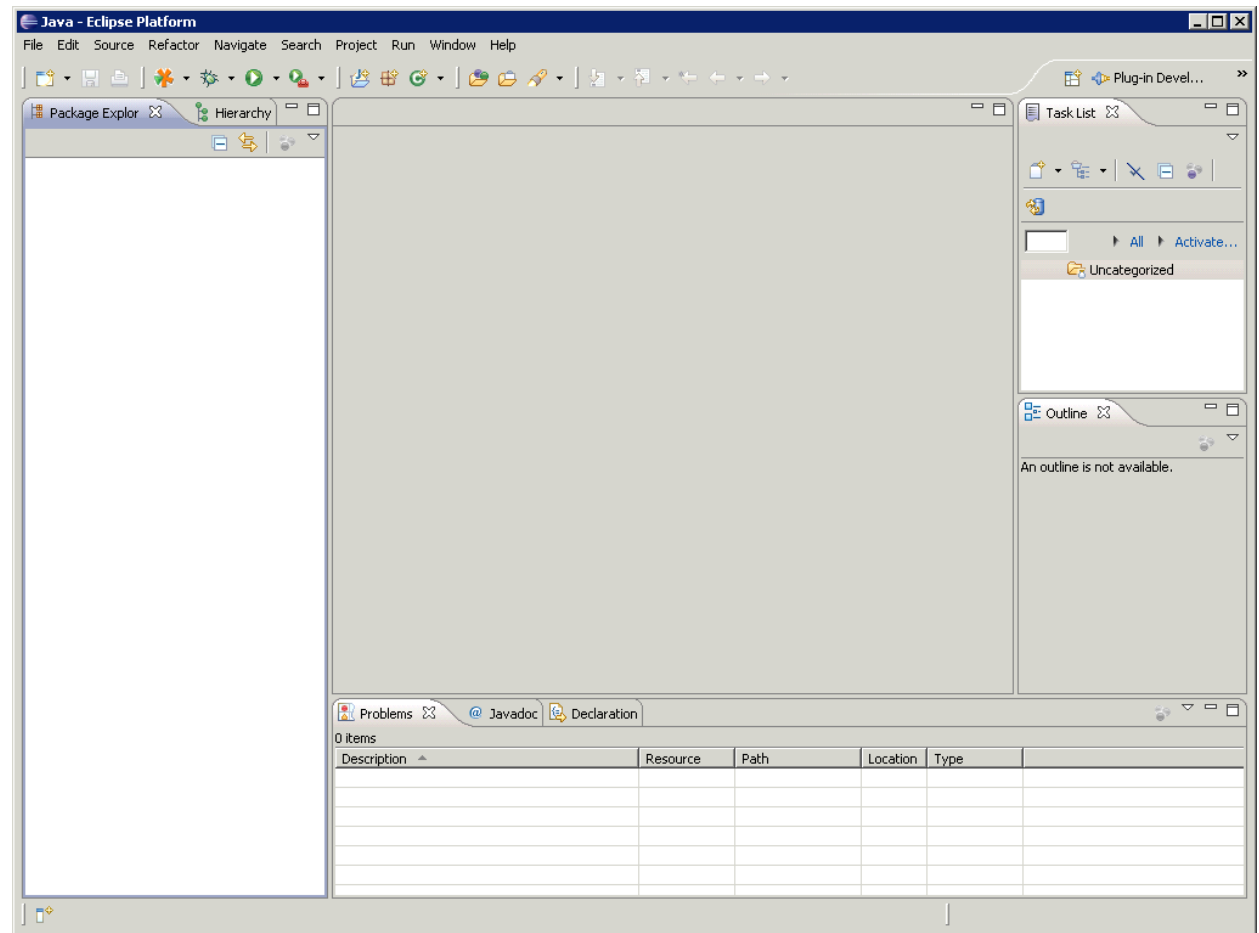
Enter the project name

Click Finish

Check that project was created

Delete project

Method 1 - Manual



Method 1 - Manual

▶ Positive aspects

Real person performs tests

▶ Negative aspects

Boring

Error prone

Not repeatable

Too much to test

Always at square one

Method 2 – Automated by recording

TC / Create and delete project

Create and delete project

- Select Menu Entry by Textpath on Menu Bar ("File","Edit","Source",...) [TYPE: Menu Bar; NAME: Menu; ACTION: Select Menu Entry by Textpath]
- Wait for Window [TYPE: Application; NAME: Application; ACTION: Wait for Window]
- Expand Node by Textpath on Tree [TYPE: Tree; NAME: NewProjWiz_ProjType_tre; ACTION: Expand Node by Textpath]
- Select Node by Textpath on Tree [TYPE: Tree; NAME: NewProjWiz_ProjType_tre; ACTION: Select Node by Textpath]
- Click on Button/Check Box/Radio Button ("Next >") [TYPE: Graphics Component; NAME: NewProjWiz_Next_btn; ACTION: Click]
- Replace Text on Text Field/Text Area/Editor Pane/Text Pane [TYPE: Component with Text Input; NAME: NewProjWiz_ProjName_cti; ACTION: Replace Text]
- Click on Button/Check Box/Radio Button ("Finish") [TYPE: Graphics Component; NAME: NewProjWiz_Finish_btn; ACTION: Click]
- Wait for Window to Close [TYPE: Application; NAME: Application; ACTION: Wait for Window to Close]
- Check Existence of Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Check Existence of Node by Textpath]
- Select Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Select Node by Textpath]
- Select Context Menu Entry by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Select Context Menu Entry by Textpath]
- Wait for Window [TYPE: Application; NAME: Application; ACTION: Wait for Window]
- Click on Button/Check Box/Radio Button ("Delete project contents on disk (cannot be undone)") [TYPE: Graphics Component; NAME: DeleteDialog_FromWorkspace_chbx; ACTION: Click]
- Click on Button/Check Box/Radio Button ("OK") [TYPE: Graphics Component; NAME: DeleteDialog_OK_btn; ACTION: Click]
- Wait for Window to Close [TYPE: Application; NAME: Application; ACTION: Wait for Window to Close]
- Check Existence of Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Check Existence of Node by Textpath]

```

if window('Java - Eclipse SDK'):
    select_menu('File>>New...>>Project...')
    if window('New Project'):
        select('org.eclipse.swt.widgets.Tree_1', '/Java/Java Project')
        click('Next')
    select('TextField', 'MyLovelyProject')
    click('Finish')
    close()
    assert('org.eclipse.swt.widgets.Tree_2', 'Text' 'MyLovelyProject', 'true')
    click('org.eclipse.swt.widgets.Tree_1', '/MyLovelyProject')
    select('org.eclipse.swt.widgets.Tree_1', '/MyLovelyProject')
    rightclick('org.eclipse.swt.widgets.Tree_1', '/MyLovelyProject')
    select_menu('Delete...')
    if window('Delete Confirmation'):
        select('org.eclipse.swt.widgets.Button_1', 'true')
        click('OK')
    close()
    assert('org.eclipse.swt.widgets.Tree_2', 'Text' 'MyLovelyProject', 'false')

```


Method 2 – Automated by recording

TC / Create and delete project

Create and delete project

- Select Menu Entry by Textpath on Menu Bar ("File", "Edit", "Source", ...) [TYPE: Menu Bar; NAME: Menu; ACTION: Select Menu Entry by Textpath]
- Wait for Window [TYPE: Application; NAME: Application; ACTION: Wait for Window]
- Expand Node by Textpath on Tree [TYPE: Tree; NAME: NewProjWiz_ProjType_tre; ACTION: Expand Node by Textpath]
- Select Node by Textpath on Tree [TYPE: Tree; NAME: NewProjWiz_ProjType_tre; ACTION: Select Node by Textpath]
- Click on Button/Check Box/Radio Button ("Next >") [TYPE: Graphics Component; NAME: NewProjWiz_Next_btn; ACTION: Click]
- Replace Text on Text Field/Text Area/Editor Pane/Text Pane [TYPE: Component with Text Input; NAME: NewProjWiz_ProjName_cti; ACTION: Replace Text]
- Click on Button/Check Box/Radio Button ("Finish") [TYPE: Graphics Component; NAME: NewProjWiz_Finish_btn; ACTION: Click]
- Wait for Window to Close [TYPE: Application; NAME: Application; ACTION: Wait for Window to Close]
- Check Existence of Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Check Existence of Node by Textpath]
- Select Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Select Node by Textpath]
- Select Context Menu Entry by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Select Context Menu Entry by Textpath]
- Wait for Window [TYPE: Application; NAME: Application; ACTION: Wait for Window]
- Click on Button/Check Box/Radio Button ("Delete project contents on disk (cannot be undone)") [TYPE: Graphics Component; NAME: DeleteDialog_FromWorkspace_chbx; ACTION: Click]
- Click on Button/Check Box/Radio Button ("OK") [TYPE: Graphics Component; NAME: DeleteDialog_OK_btn; ACTION: Click]
- Wait for Window to Close [TYPE: Application; NAME: Application; ACTION: Wait for Window to Close]
- Check Existence of Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Check Existence of Node by Textpath]

Method 2 – Automated by recording

TC / Create and delete project

Create and delete project

- Select Menu Entry by Textpath on Menu Bar ("File", "Edit", "Source", ...) [TYPE: Menu Bar; NAME: Menu; ACTION: Select Menu Entry by Textpath]
- Wait for Window [TYPE: Application; NAME: Application; ACTION: Wait for Window]
- Expand Node by Textpath on Tree [TYPE: Tree; NAME: NewProjWiz_ProjType_tre; ACTION: Expand Node by Textpath]
- Select Node by Textpath on Tree [TYPE: Tree; NAME: NewProjWiz_ProjType_tre; ACTION: Select Node by Textpath]
- Click on Button/Check Box/Radio Button ("Next >") [TYPE: Graphics Component; NAME: NewProjWiz_Next_btn; ACTION: Click]
- Replace Text on Text Field/Text Area/Editor Pane/Text Pane [TYPE: Component with Text Input; NAME: NewProjWiz_ProjName_cti; ACTION: Replace Text]
- Click on Button/Check Box/Radio Button ("Finish") [TYPE: Graphics Component; NAME: NewProjWiz_Finish_btn; ACTION: Click]
- Wait for Window to Close [TYPE: Application; NAME: Application; ACTION: Wait for Window to Close]
- Check Existence of Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Check Existence of Node by Textpath]
- Select Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Select Node by Textpath]
- Select Context Menu Entry by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Select Context Menu Entry by Textpath]
- Wait for Window [TYPE: Application; NAME: Application; ACTION: Wait for Window]
- Click on Button/Check Box/Radio Button ("Delete project contents on disk (cannot be undone)") [TYPE: Graphics Component; NAME: DeleteDialog_FromWorkspace_chbx; ACTION: Click]
- Click on Button/Check Box/Radio Button ("OK") [TYPE: Graphics Component; NAME: DeleteDialog_OK_btn; ACTION: Click]
- Wait for Window to Close [TYPE: Application; NAME: Application; ACTION: Wait for Window to Close]
- Check Existence of Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Check Existence of Node by Textpath]

Method 2 – Automated by recording

TC / Create and delete project

TC / Create and delete project

- Select Menu Entry by Textpath on Menu Bar ("File", "Edit", "Source", ...) [TYPE: Menu Bar; NAME: Menu; ACTION: Select Menu Entry by Textpath]
- Wait for Window [TYPE: Application; NAME: Application; ACTION: Wait for Window]
- Expand Node by Textpath on Tree [TYPE: Tree; NAME: NewProjWiz_ProjType_tre; ACTION: Expand Node by Textpath]
- Select Node by Textpath on Tree [TYPE: Tree; NAME: NewProjWiz_ProjType_tre; ACTION: Select Node by Textpath]
- Click on Button/Check Box/Radio Button ("Next >") [TYPE: Graphics Component; NAME: NewProjWiz_Next_btn; ACTION: Click]
- Replace Text on Text Field/Text Area/Editor Pane/Text Pane [TYPE: Component with Text Input; NAME: NewProjWiz_ProjName_cti; ACTION: Replace Text]
- Click on Button/Check Box/Radio Button ("Finish") [TYPE: Graphics Component; NAME: NewProjWiz_Finish_btn; ACTION: Click]
- Wait for Window to Close [TYPE: Application; NAME: Application; ACTION: Wait for Window to Close]
- Check Existence of Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Check Existence of Node by Textpath]
- Select Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Select Node by Textpath]
- Select Context Menu Entry by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Select Context Menu Entry by Textpath]
- Wait for Window [TYPE: Application; NAME: Application; ACTION: Wait for Window]
- Click on Button/Check Box/Radio Button ("Delete project contents on disk (cannot be undone)") [TYPE: Graphics Component; NAME: DeleteDialog_FromWorkspace_chbx; ACTION: Click]
- Click on Button/Check Box/Radio Button ("OK") [TYPE: Graphics Component; NAME: DeleteDialog_OK_btn; ACTION: Click]
- Wait for Window to Close [TYPE: Application; NAME: Application; ACTION: Wait for Window to Close]
- Check Existence of Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Check Existence of Node by Textpath]

Method 2 – Automated by recording

TC / Create and delete project

Create and delete project

- Select Menu Entry by Textpath on Menu Bar ("File", "Edit", "Source", ...) [TYPE: Menu Bar; NAME: Menu; ACTION: Select Menu Entry by Textpath]
- Wait for Window [TYPE: Application; NAME: Application; ACTION: Wait for Window]
- Expand Node by Textpath on Tree [TYPE: Tree; NAME: NewProjWiz_ProjType_tre; ACTION: Expand Node by Textpath]
- Select Node by Textpath on Tree [TYPE: Tree; NAME: NewProjWiz_ProjType_tre; ACTION: Select Node by Textpath]
- Click on Button/Check Box/Radio Button ("Next >") [TYPE: Graphics Component; NAME: NewProjWiz_Next_btn; ACTION: Click]
- Replace Text on Text Field/Text Area/Editor Pane/Text Pane [TYPE: Component with Text Input; NAME: NewProjWiz_ProjName_cti; ACTION: Replace Text]
- Click on Button/Check Box/Radio Button ("Finish") [TYPE: Graphics Component; NAME: NewProjWiz_Finish_btn; ACTION: Click]
- Wait for Window to Close [TYPE: Application; NAME: Application; ACTION: Wait for Window to Close]
- Check Existence of Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Check Existence of Node by Textpath]
- Select Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Select Node by Textpath]
- Select Context Menu Entry by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Select Context Menu Entry by Textpath]
- Wait for Window [TYPE: Application; NAME: Application; ACTION: Wait for Window]
- Click on Button/Check Box/Radio Button ("Delete project contents on disk (cannot be undone)") [TYPE: Graphics Component; NAME: DeleteDialog_FromWorkspace_chbx; ACTION: Click]
- Click on Button/Check Box/Radio Button ("OK") [TYPE: Graphics Component; NAME: DeleteDialog_OK_btn; ACTION: Click]
- Wait for Window to Close [TYPE: Application; NAME: Application; ACTION: Wait for Window to Close]
- Check Existence of Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Check Existence of Node by Textpath]

Method 2 – Automated by recording

TC / Create and delete project

Create and delete project

- Select Menu Entry by Textpath on Menu Bar ("File", "Edit", "Source", ...) [TYPE: Menu Bar; NAME: Menu; ACTION: Select Menu Entry by Textpath]
- Wait for Window [TYPE: Application; NAME: Application; ACTION: Wait for Window]
- Expand Node by Textpath on Tree [TYPE: Tree; NAME: NewProjWiz_ProjType_tre; ACTION: Expand Node by Textpath]
- Select Node by Textpath on Tree [TYPE: Tree; NAME: NewProjWiz_ProjType_tre; ACTION: Select Node by Textpath]
- Click on Button/Check Box/Radio Button ("Next >") [TYPE: Graphics Component; NAME: NewProjWiz_Next_btn; ACTION: Click]
- Replace Text on Text Field/Text Area/Editor Pane/Text Pane [TYPE: Component with Text Input; NAME: NewProjWiz_ProjName_cti; ACTION: Replace Text]
- Click on Button/Check Box/Radio Button ("Finish") [TYPE: Graphics Component; NAME: NewProjWiz_Finish_btn; ACTION: Click]
- Wait for Window to Close [TYPE: Application; NAME: Application; ACTION: Wait for Window to Close]
- Check Existence of Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Check Existence of Node by Textpath]
- Select Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Select Node by Textpath]
- Select Context Menu Entry by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Select Context Menu Entry by Textpath]
- Wait for Window [TYPE: Application; NAME: Application; ACTION: Wait for Window]
- Click on Button/Check Box/Radio Button "Delete project contents on disk (cannot be undone)" [TYPE: Graphics Component; NAME: DeleteDialog_FromWorkspace_chbx; ACTION: Click]
- Click on Button/Check Box/Radio Button ("OK") [TYPE: Graphics Component; NAME: DeleteDialog_OK_btn; ACTION: Click]
- Wait for Window to Close [TYPE: Application; NAME: Application; ACTION: Wait for Window to Close]
- Check Existence of Node by Textpath on Tree [TYPE: Tree; NAME: PackageExplorer_tre; ACTION: Check Existence of Node by Textpath]

Method 2 – Automated by recording

▶ Positive aspects

Quick start

▶ Negative aspects

Wait for functioning feature

Raw form:

→ unstructured

→ redundancies

→ hard to read

Effort to make maintainable

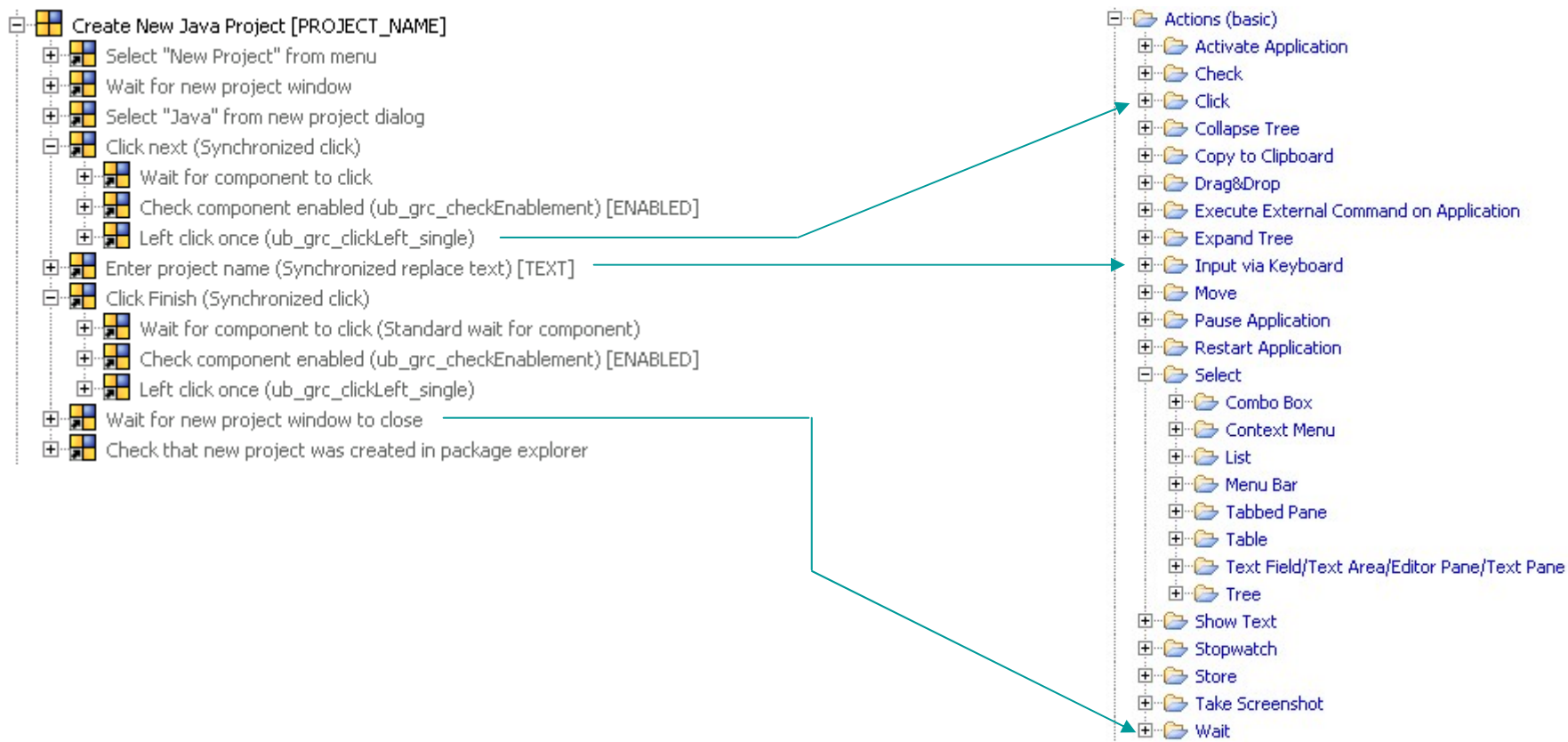
Tests *as-is* not *as-should-be*

No explicit intelligence in test
















Method 3 – Automated with keywords

- [-] Create New Java Project [PROJECT_NAME]
 - [+] Select "New Project" from menu
 - [+] Wait for new project window
 - [+] Select "Java" from new project dialog
 - [-] Click next (Synchronized click)
 - [+] Wait for component to click
 - [+] Check component enabled (ub_grc_checkEnablement) [ENABLED]
 - [+] Left click once (ub_grc_clickLeft_single)
 - [+] Enter project name (Synchronized replace text) [TEXT]
 - [-] Click Finish (Synchronized click)
 - [+] Wait for component to click (Standard wait for component)
 - [+] Check component enabled (ub_grc_checkEnablement) [ENABLED]
 - [+] Left click once (ub_grc_clickLeft_single)
 - [+] Wait for new project window to close
 - [+] Check that new project was created in package explorer

Method 3 – Automated with keywords



Method 3 – Automated with keywords

- [-]  Create New Java Project [PROJECT_NAME]
 - [+]  Select "New Project" from menu
 - [+]  Wait for new project window
 - [+]  Select "Java" from new project dialog
 - [-]  Click next (Synchronized click)
 - [+]  Wait for component to click
 - [+]  Check component enabled (ub_grc_checkEnablement) [ENABLED]
 - [+]  Left click once (ub_grc_clickLeft_single)
 - [+]  Enter project name (Synchronized replace text) [TEXT]
 - [-]  Click Finish (Synchronized click)
 - [+]  Wait for component to click (Standard wait for component)
 - [+]  Check component enabled (ub_grc_checkEnablement) [ENABLED]
 - [+]  Left click once (ub_grc_clickLeft_single)
 - [+]  Wait for new project window to close
 - [+]  Check that new project was created in package explorer

Method 3 – Automated with keywords

- [-] Create New Java Project [PROJECT_NAME]
 - [+] Select "New Project" from menu
 - [+] Wait for new project window
 - [+] Select "Java" from new project dialog
 - [-] Click next (Synchronized click)
 - [+] Wait for component to click
 - [+] Check component enabled (ub_grc_checkEnablement) [ENABLED]
 - [+] Left click once (ub_grc_clickLeft_single)
 - [+] Enter project name (Synchronized replace text) [TEXT]
 - [-] Click Finish (Synchronized click)
 - [+] Wait for component to click (Standard wait for component)
 - [+] Check component enabled (ub_grc_checkEnablement) [ENABLED]
 - [+] Left click once (ub_grc_clickLeft_single)
 - [+] Wait for new project window to close
 - [+] Check that new project was created in package explorer

Method 3 – Automated with keywords

▶ Positive aspects

- High-level actions
- Independence from app
- Earlier focus on acceptance
- Readable tests
- No programming effort
- Modularity almost for free
- Tester in full control

▶ Negative aspects

- Requires planning & design
- Requires discipline:
 - naming conventions
 - any restructuring work
 - management of keywords

Method 3 – Automated with keywords

▶ Positive aspects

- High-level actions
- Independence from app
- Earlier focus on acceptance
- Readable tests
- No programming effort
- Modularity almost for free

▶ “Negative” aspects

- Requires planning & design
- Requires discipline:
 - naming conventions
 - any restructuring work
 - management of keywords

▶ No different to development!

Automatic tests are as important as code!

▶ **Must be planned**

Which areas should be tested

Which areas should be automated

▶ **Must be structured**

Based on clear use case

Test design

Reusable, flexible modules

Readable and maintainable

▶ **Contain explicit intelligence**

Synchronization

Checks and fall backs

Knowledge of application

Understanding of controls

Understanding of workflows



Automatic tests are as important as code!

▶ Must be planned

Which areas should be tested

Which areas should be automated

▶ Must be structured

Test design

Reusable, flexible modules

Readable and maintainable

▶ Contain explicit intelligence

Synchronization

Checks and fall backs

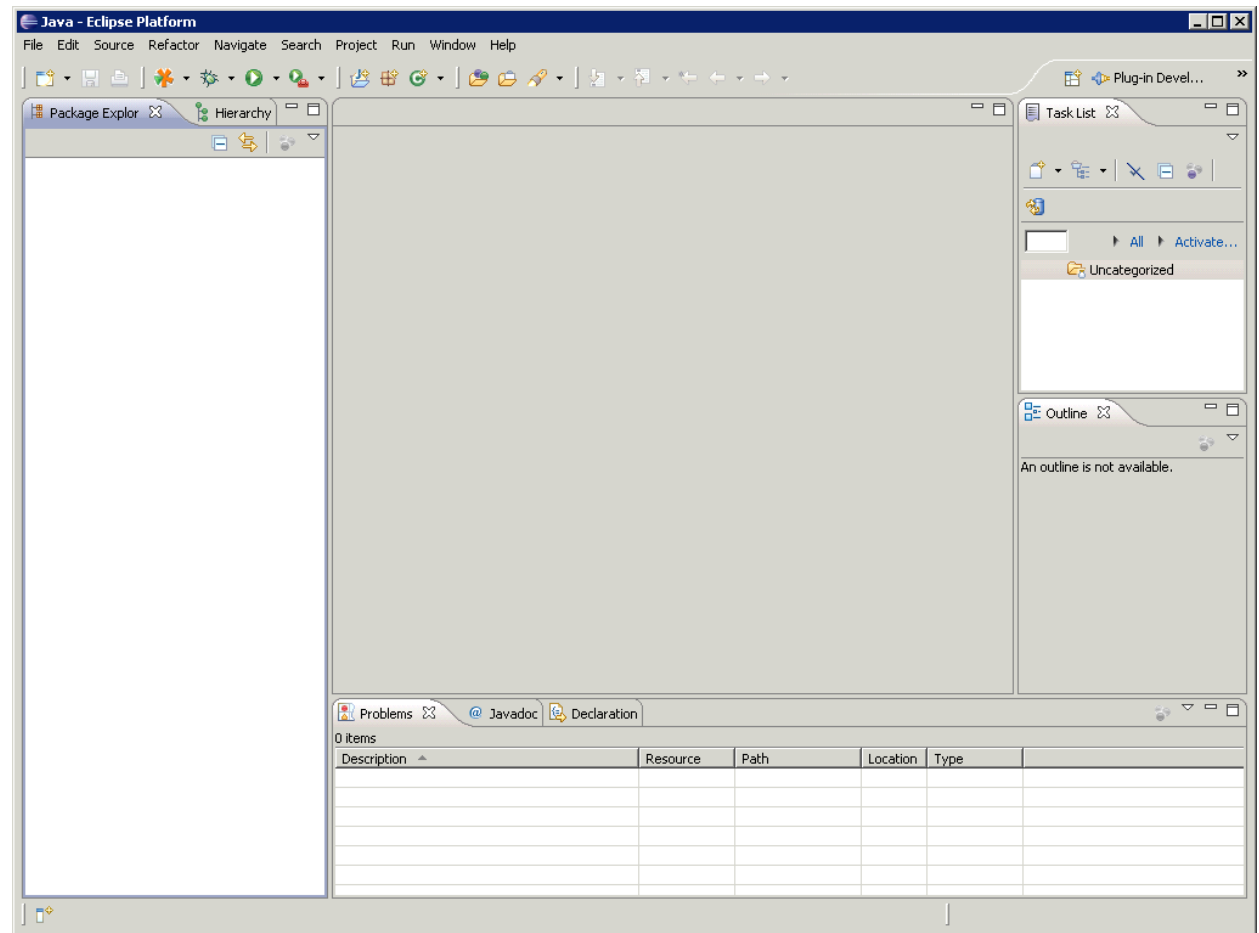
Knowledge of application

Understanding of controls

Understanding of workflows

▶ **Keywords → well structured tests without coding effort**

Keyword-driven test



Getting the mix right – automated tests

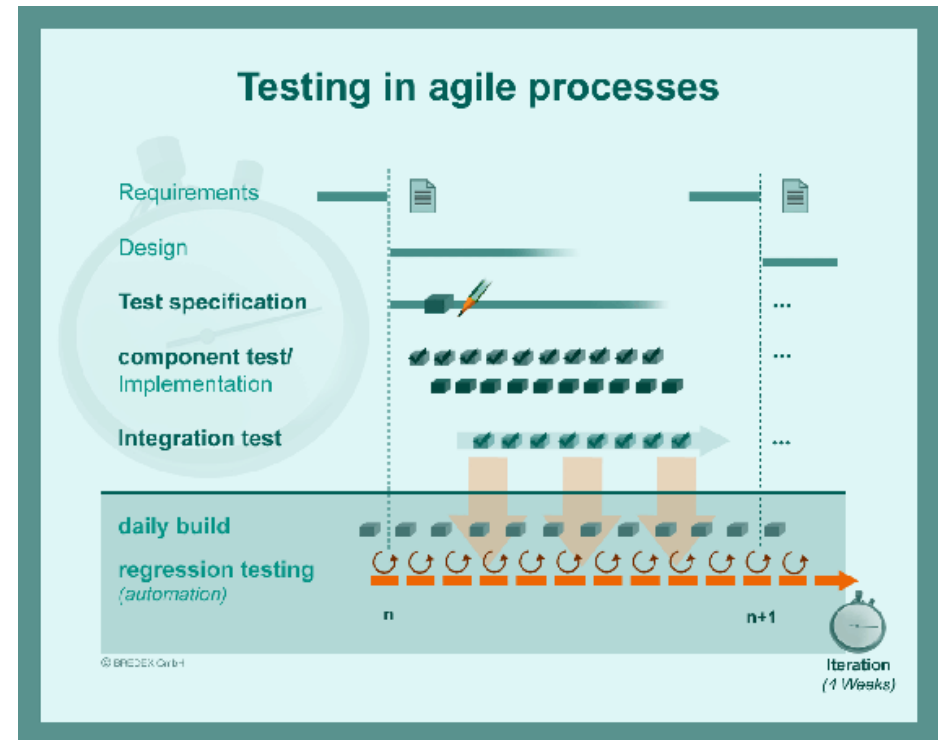


▶ Daily

Daily build – smoke test

▶ Weekly

Complete automated test



Getting the mix right – automated tests



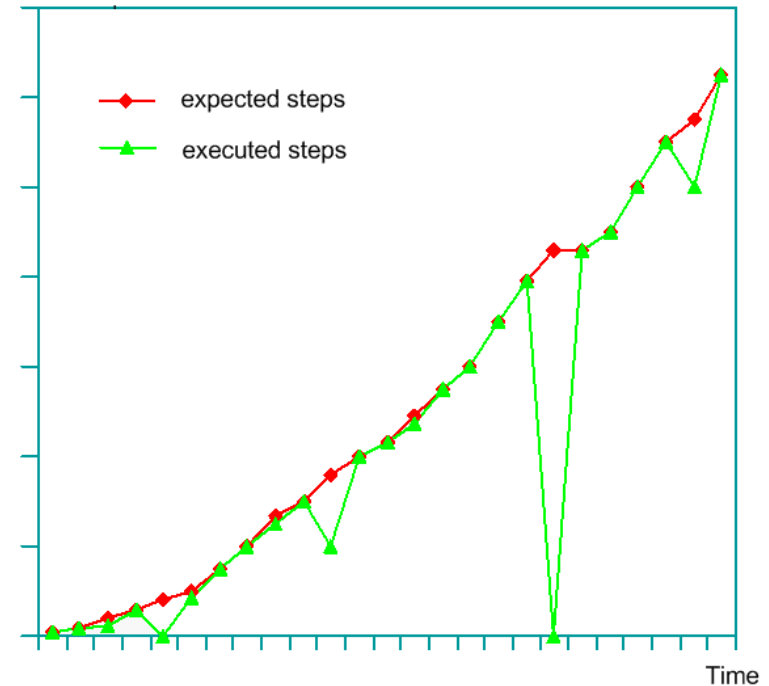
▶ Daily

Daily build – smoke test

▶ Weekly

Complete automated test

Test Steps



Getting the mix right – manual tests



- ▶ **Daily**

“Four-eye” test

- ▶ **Weekly**

Show and tell

- ▶ **Per feature (timely)**

Manual acceptance test

- ▶ **Before release**

Full manual test phase

Results

- ▶ **Acceptance tests at the forefront of development**
- ▶ **Timely discussion and time to react**
- ▶ **Customer satisfaction**