

A close-up, artistic photograph of a human eye, showing the eyelashes and the iris. The image is rendered in a monochromatic teal color, matching the background. The eye is looking slightly to the right.

Agile Acceptance Testing: A Case Study

Alexandra Imrie, BREDEX GmbH

Our story...

- ▶ **Adding agility in testing over series of stages**
- ▶ **Why?**
- ▶ **How?**
- ▶ **Lessons learned and benefits gained**

What we knew / what we know

- ▶ **Agile processes require agility in tests**
→ Automation is necessary
- ▶ **Acceptance tests must also be agile**
(and automated!)
- ▶ **Quality must be constantly monitored**
- ▶ **A test method compatible with agility is required**

Life isn't always that easy!

- ▶ **Project ran for 2 years before automated acceptance testing could be started**
 - Home grown solution didn't support the technology yet
- ▶ **Project uses increasingly agile processes**

Testing should also be agile
- ▶ **After 2 years, a HUGE test backlog**

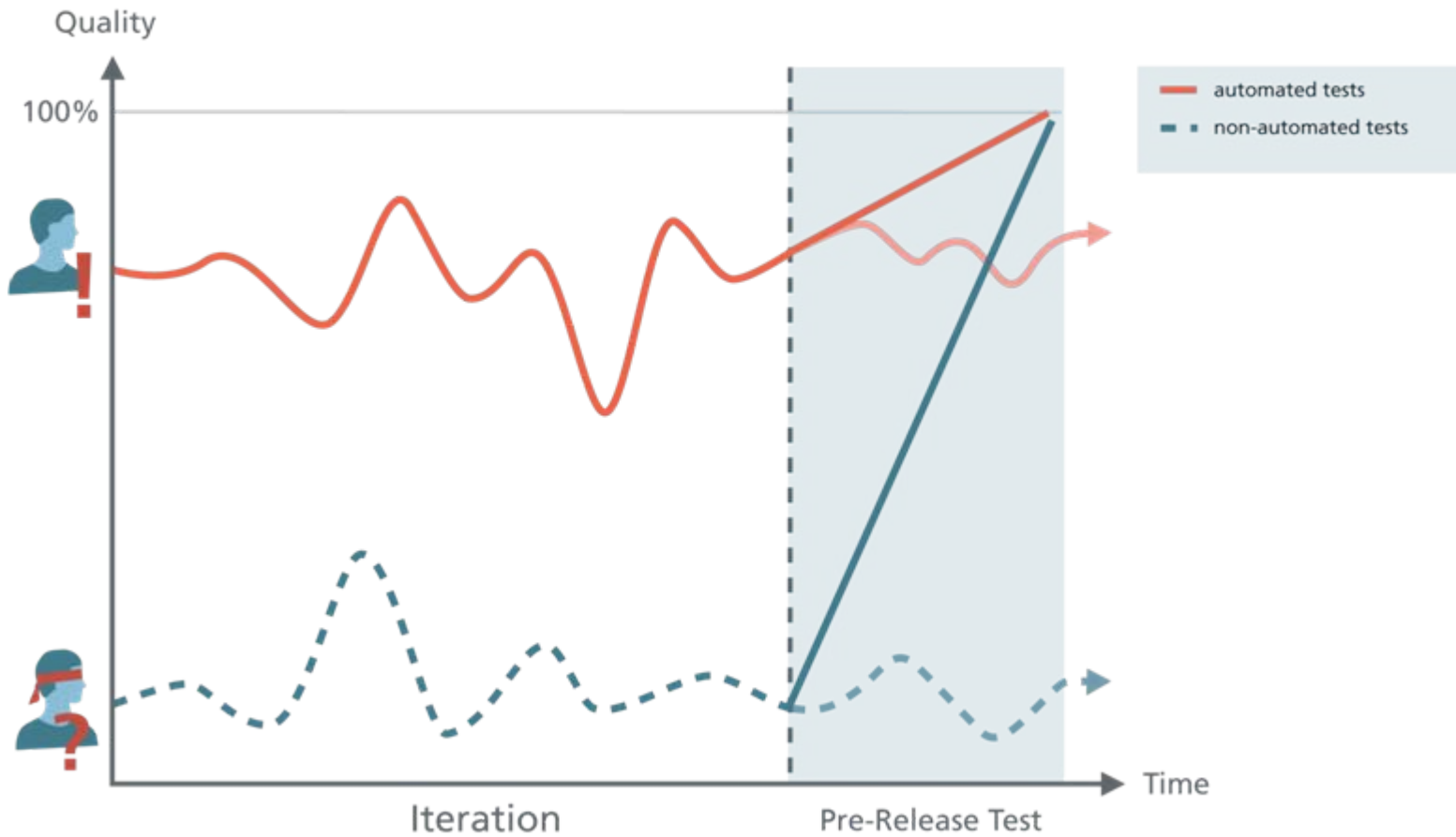
How can we become agile **and** have a safety net?

“Getting agile”

- ▶ **Can't happen overnight**
- ▶ **But where to start?**
 - What problems we had
 - How we overcame them
- ▶ **What we'd do differently in the future**



We have no automated tests!





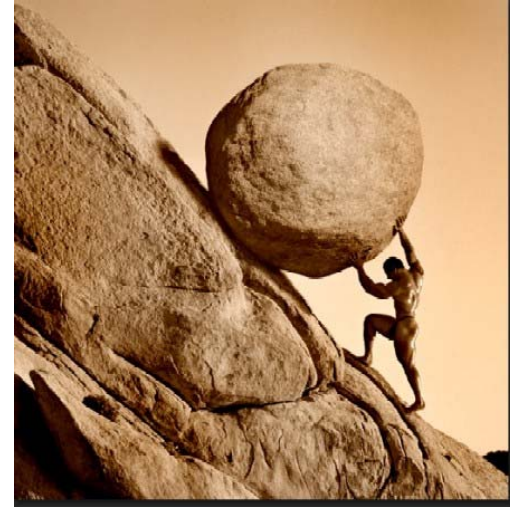
Where to start?

▶ **New features**

- + can be tested agilely
- lack of basis tests for workflows

▶ **Errors**

- + can ensure that errors don't occur again
- lack of basis tests for workflows
- "error driven" tests – uncoordinated, no test plan



The first steps

▶ Old features

- + easier to test, because behaviour well known
- + critical features, must work
- + create a set of modules for basic workflows
- + start working on backlog
- + “first 15 minutes”
- not agile yet

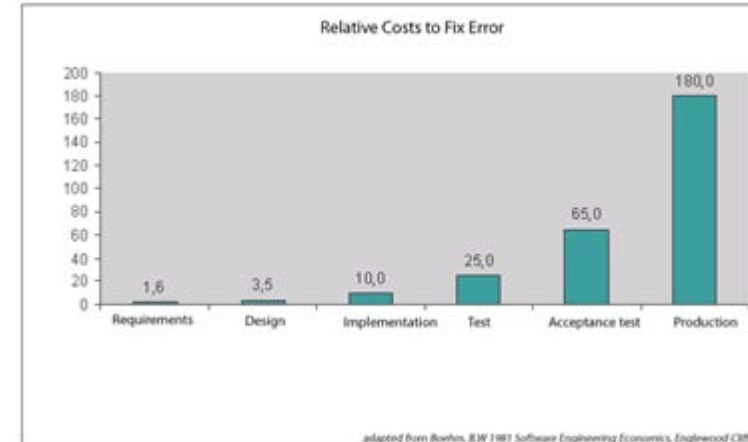
▶ Software being used and tested on daily basis

This bug has already been fixed!

- ▶ **Developers being distracted chasing up fixed bugs**
- ▶ **Testers wasting time reproducing old issues**
- ▶ **New builds manually started**
 - Too long before new features can be tested
 - Testers working with old version
 - Also a strain on dev team
- ▶ **→ Falling behind on development and quality**

This bug wasn't here yesterday!

- ▶ Automated build process introduced
- ▶ New version of software every day
- ▶ Tests run at night on build server
- ▶ Quicker feedback on errors
Manually or automatically discovered



The testers aren't in the loop!

- ▶ **Changes to old features aren't being communicated**

Tests not being updated in time

- ▶ **New features aren't discussed with tests in mind**

Usability & Testability

User acceptance, conformance to concepts

- ▶ **Misunderstandings**

Developer, customer, tester

Features not always as desired



Communication is the key!

▶ Invite testers

- to release & sprint planning meetings
- to stand up meetings
- to look at current developments

▶ Testers kept up to date on developments

▶ “Tester view” incorporated

- From beginning
- Throughout development of feature
- Better acceptance of new features

New areas are too wishy-washy!

- ▶ **Stories defined from dev perspective**
 - Internal changes
 - Technology and architecture
- ▶ **No testable requirements**
- ▶ **Stories often not thought through well enough**
 - Concept mish-mash
 - Wasted time on wrong path
- ▶ **Non-deliverable developments**

Tell me a story

▶ For each story, acceptance test criteria defined

Manual test

Automated test

“Four-eyes” test

→ Encourages thinking through

→ Results in well-rounded, successful features

▶ Story board

Test criteria written on story card

Including column for “to test”



Storyboard



Show me a story

- ▶ **Show-and-tell for new features**
 - Focus back on deliverable benefits
 - Better usability in first version
 - Missing features identified and story cards created
- ▶ **Encouragement to show other team members**

Test results are being ignored!

- ▶ **Tests are running every night**
- ▶ **Errors in software → less test coverage**
Often over multiple nightly runs
Start to lose safety net

Listening to the tests

▶ **Presentation of test results each morning**

Extra work for tester – analysis

Whole team aware of current quality status

Quality starts to become important

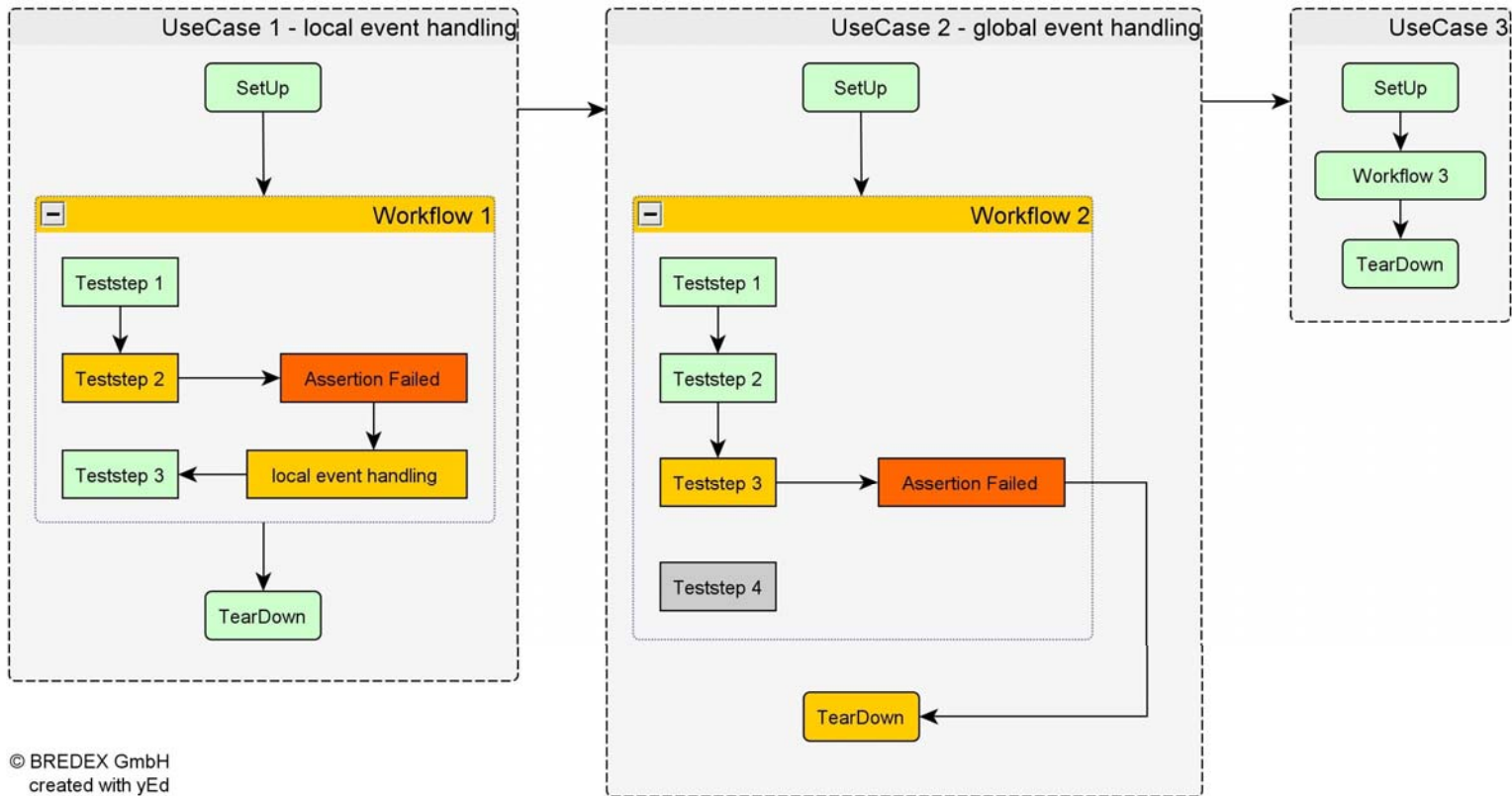
Encourages communication about new features or errors

▶ **Decision for each error**

Fix immediately?

Bug-tracking?

Making the tests more stable



The tests are taking too long!

▶ Tests run on build server, but run sequentially

First Windows, then Linux

In-between, test environment cleaned up

Tests are running over to morning (7am)

Want to add more supported platforms

Time to rebuild the build!

- ▶ **Build process changed**

Hudson job to build software

- ▶ **Current build installed on central server**

- ▶ **In parallel, for 5 systems**

Job to set up test environment on virtual machine

Job to install software on virtual machine

Job to run tests

- ▶ **Job to collect test results**

- ▶ **Job to clean-up test environment**

The challenges

- ▶ **Test environment must be well-defined**

Virtualization best way to achieve this

Memory and performance of virtual systems

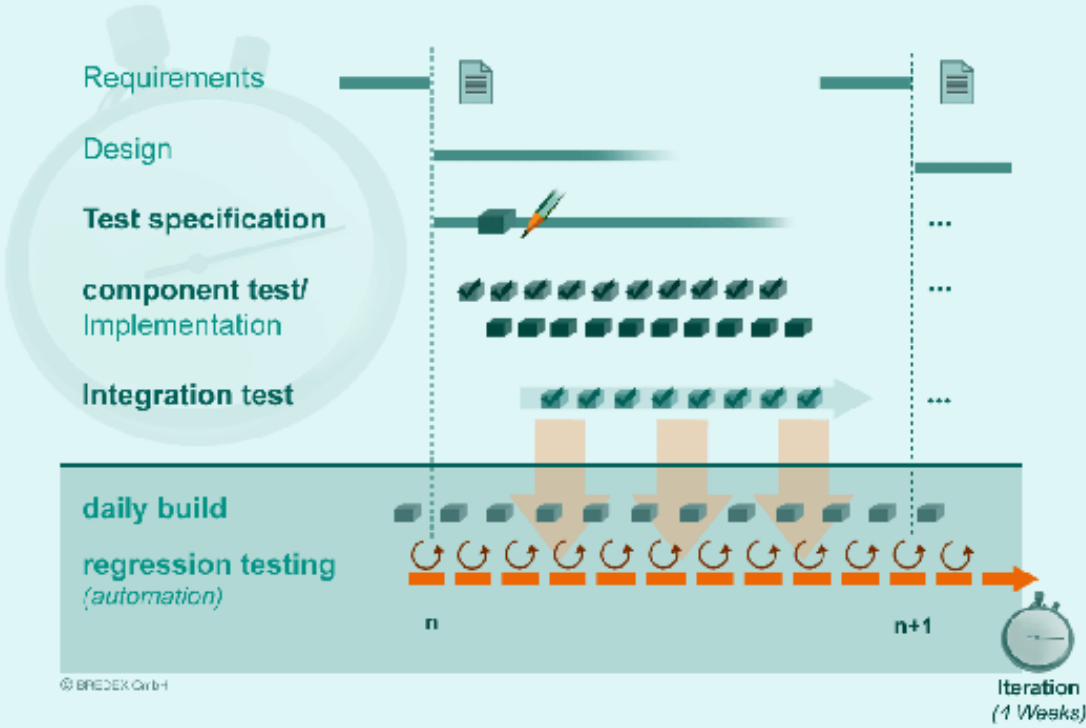
→ but solves the “works for me” issue

→ stops tests being disturbed by updates on central machine

- ▶ **Test data must be platform independent**

Where do we go from here?

Testing in agile processes



Where do we go from here?

- ▶ **Increase test coverage for critical areas**
Introduction of metrics planned
- ▶ **Agile introduction of tests for new areas**
Improves development process through discussion
Better acceptance
- ▶ **Back to beta?**

What agility has brought us

- ▶ **Better communication and teamwork**

Wrong or misunderstood requirements quickly cleared up

- ▶ **Flexibility to react to changes**

- ▶ **Regular priority and process reviews**

- ▶ **Working software every day**

- ▶ **Reduced manual test phase**

- ▶ **Manageable release effort**

- ▶ **Concentration on value for features**

In future projects...

- ▶ **Automated build & test from outset**
On well-defined virtual systems
- ▶ **Always include testers**
- ▶ **Make test results an important issue to whole team**
Display results in prominent place



Thank you!

▶ **Any questions?**