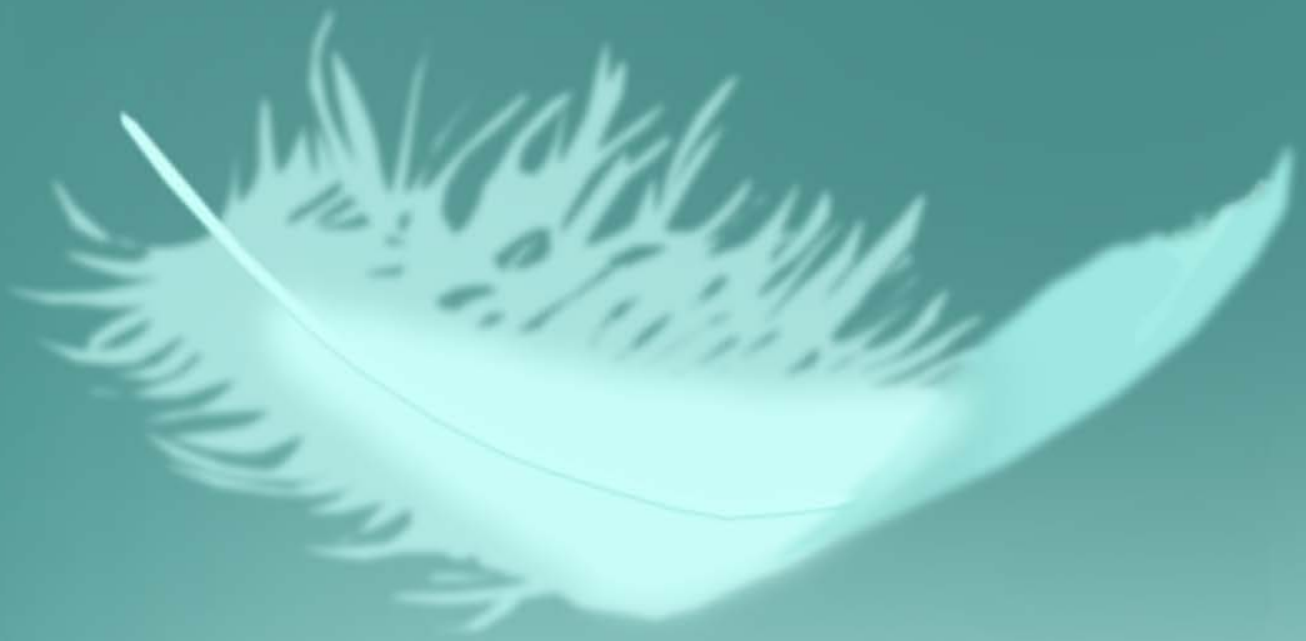
BREDEX
Software-Entwicklung und Beratung



Cutting the right corners

Balancing effort and payoff for GUI test automation

Alexandra Schladebeck, BREDEX GmbH

 Version

Agenda

Once upon a time

- Fixed price

But then one day…

- Time and materials

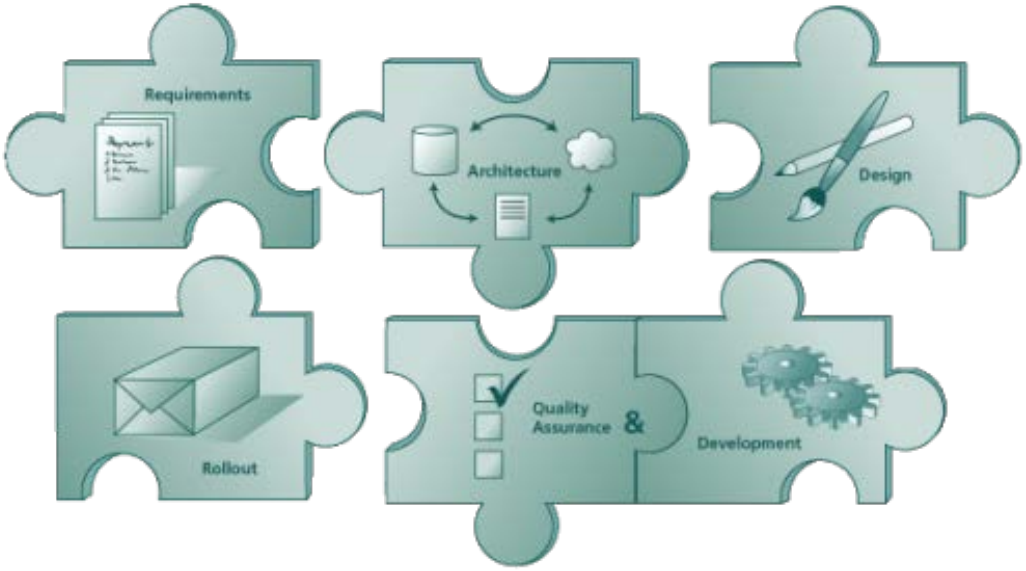
Happily ever after?

- Attempted solutions and their success

Introductions – BREDEX GmbH

- **Software development and consulting**
- **Focus on quality**

  Test tools: GUIdancer and Jubula
- **Eclipse Strategic Member**



Introductions – Context for Talk

- **Customer project which began in 2006**

  "Administration and control of measuring equipment"

  http://www.bredex.de/web/index.php/administration-and-control-of-measuring-equipment.html

- **Original pilot project for GUI test automation**
- **Long-running project**

  Progress and quality monitored throughout:

  lifecycle – different phases

  changing resources

  changing teams

Disclaimer

- **This talk is about being pragmatic!**

- **In an ideal situation, we have**

  a continuous focus on testing

  And a continuous tester presence

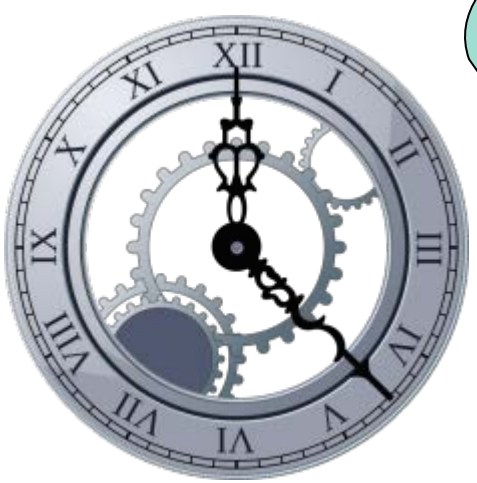
- **But some testing is better than**

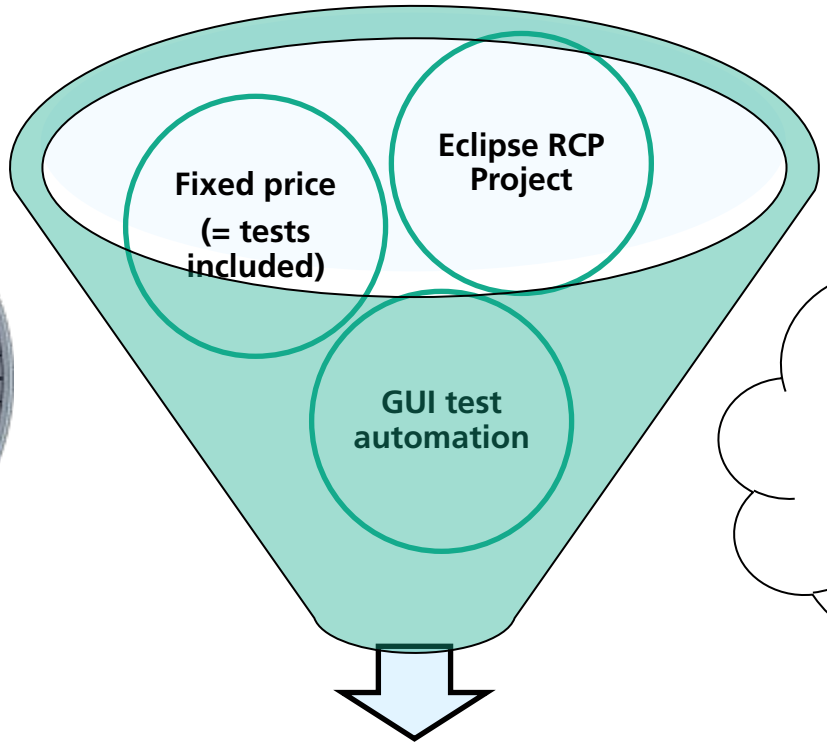
  no testing at all

  Aim for better, deal with less

- **And so our story begins…**

Once upon a time…





GUI test automation is economically viable and worthwhile

Extremely high-quality release

But then …

Move to time & materials

Fewer testing resources – money and people

Tests left to stagnate

More manual testing necessary

Quality suffers

Manual test effort increases

We need a hero!

- **The quest…**

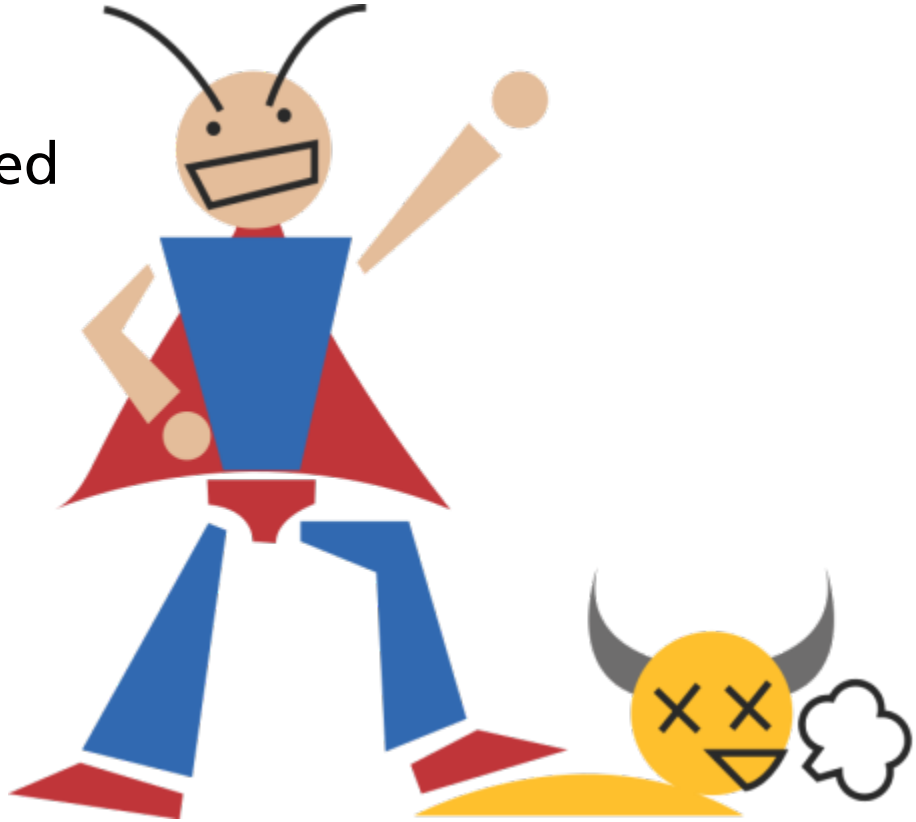
  To get the most information about quality from automated tests as possible

- **The challenges**

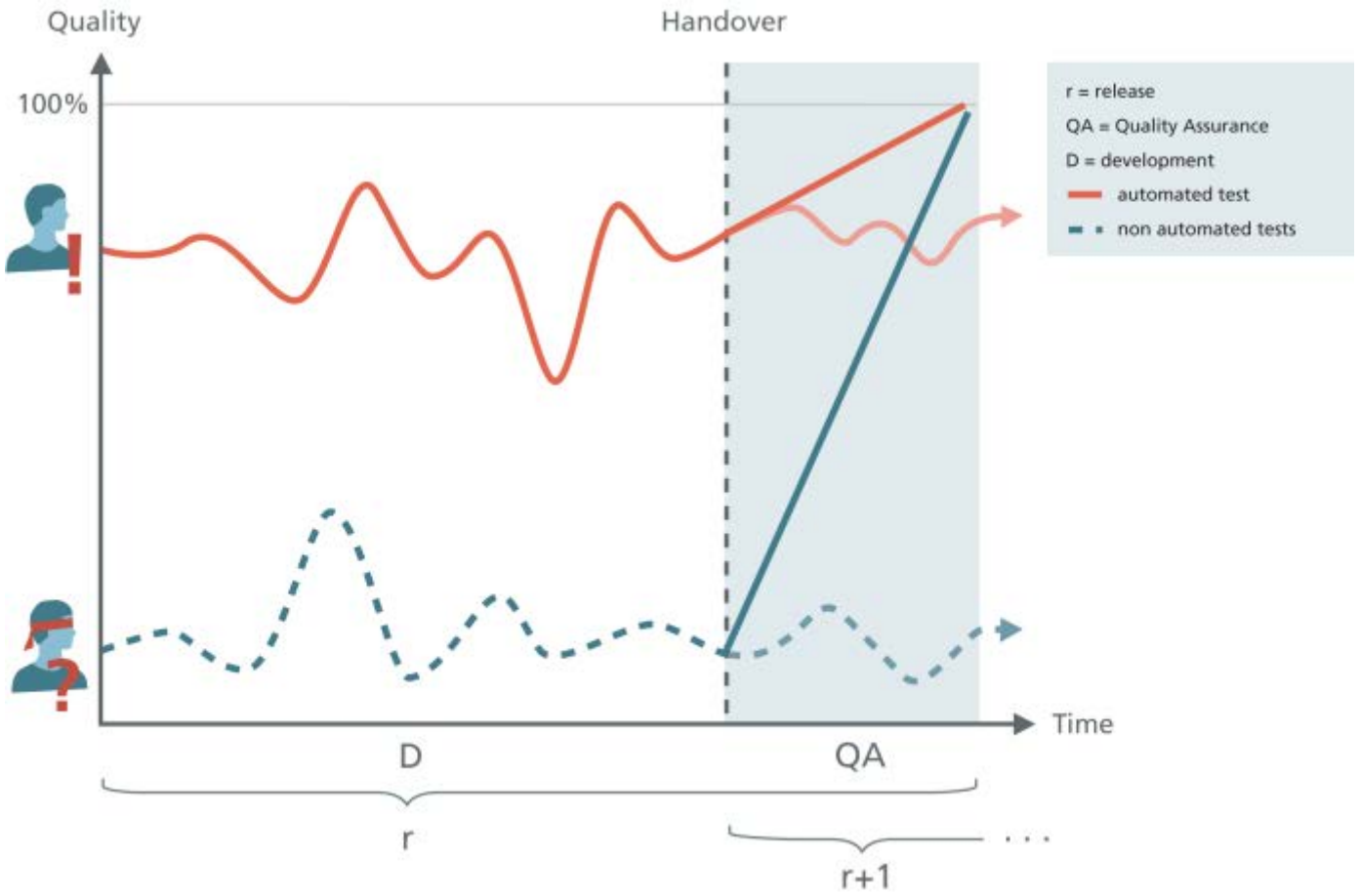
  Limited budget and time

  Backlog of tests

  Complex program



Quality information



Hero candidate number one

- **The guy from support**

  + people working in support make great testers

  - alongside other duties
  - too many other things to do
  - priority for tests too low

  - practically no introduction
  - no plan made to follow

- **Progress slow, practically no benefits**

Hero candidate number two

- **Student in Practical Semester (StiP)**

+ full week (40 h) for 1 – 2 months, 3x a year
+ no costs for team
- may not have testing / area experience

+ preparation
+ 3 day tool training incl. conventions, best practices
+ introduction to software under test by the team
+ systematic test plan created and followed

- had to write new tests (old tests not run now for > 2 years)

StiP does the trick?

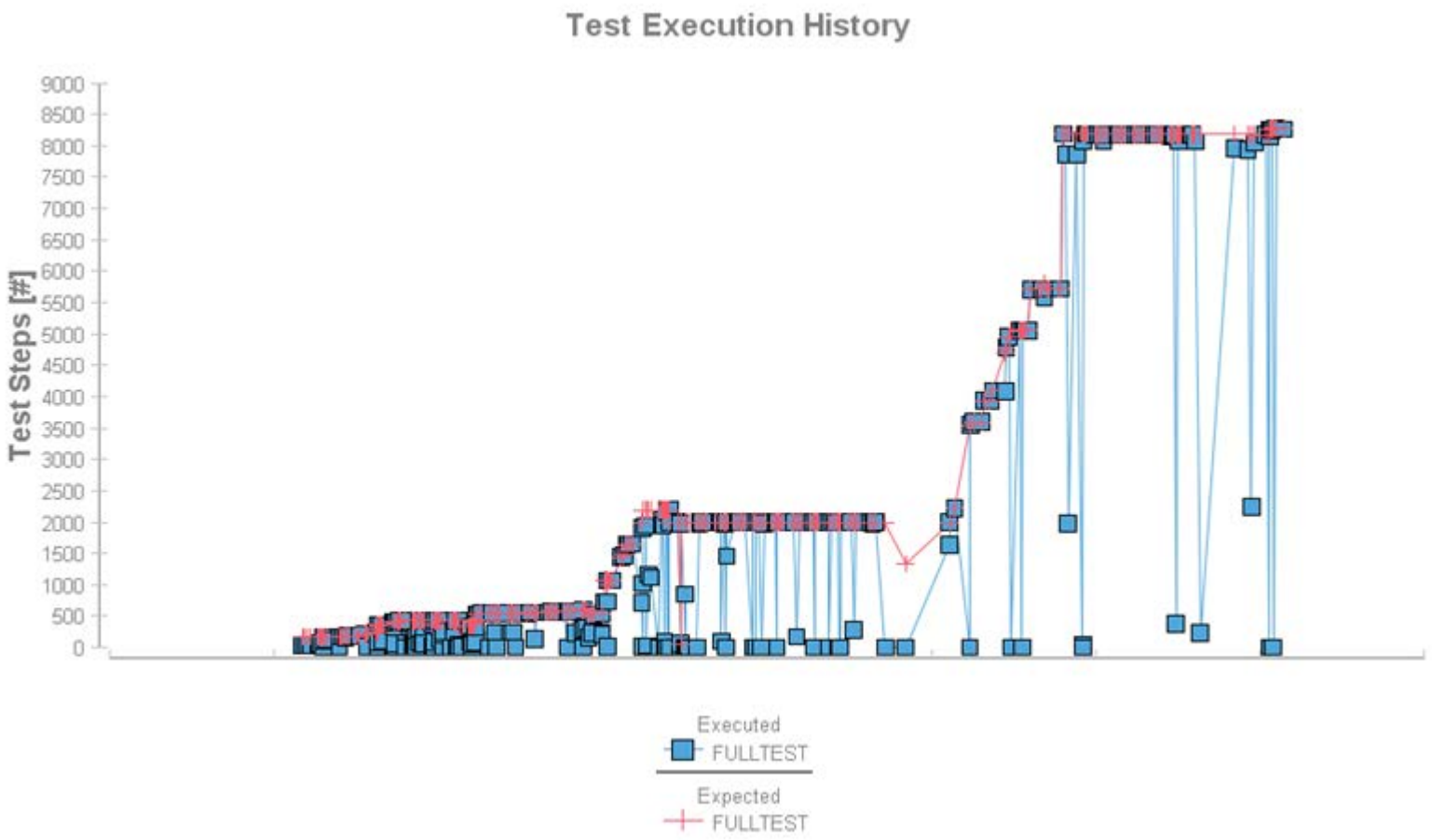
- **Progress after first StiP (1 month)**
  - 2000 test steps
  - Test runs for 70 minutes
- **Documentation for following StiP**
  - Current state
  - Use Cases
  - Software-specific conventions

Waiting for next StiP

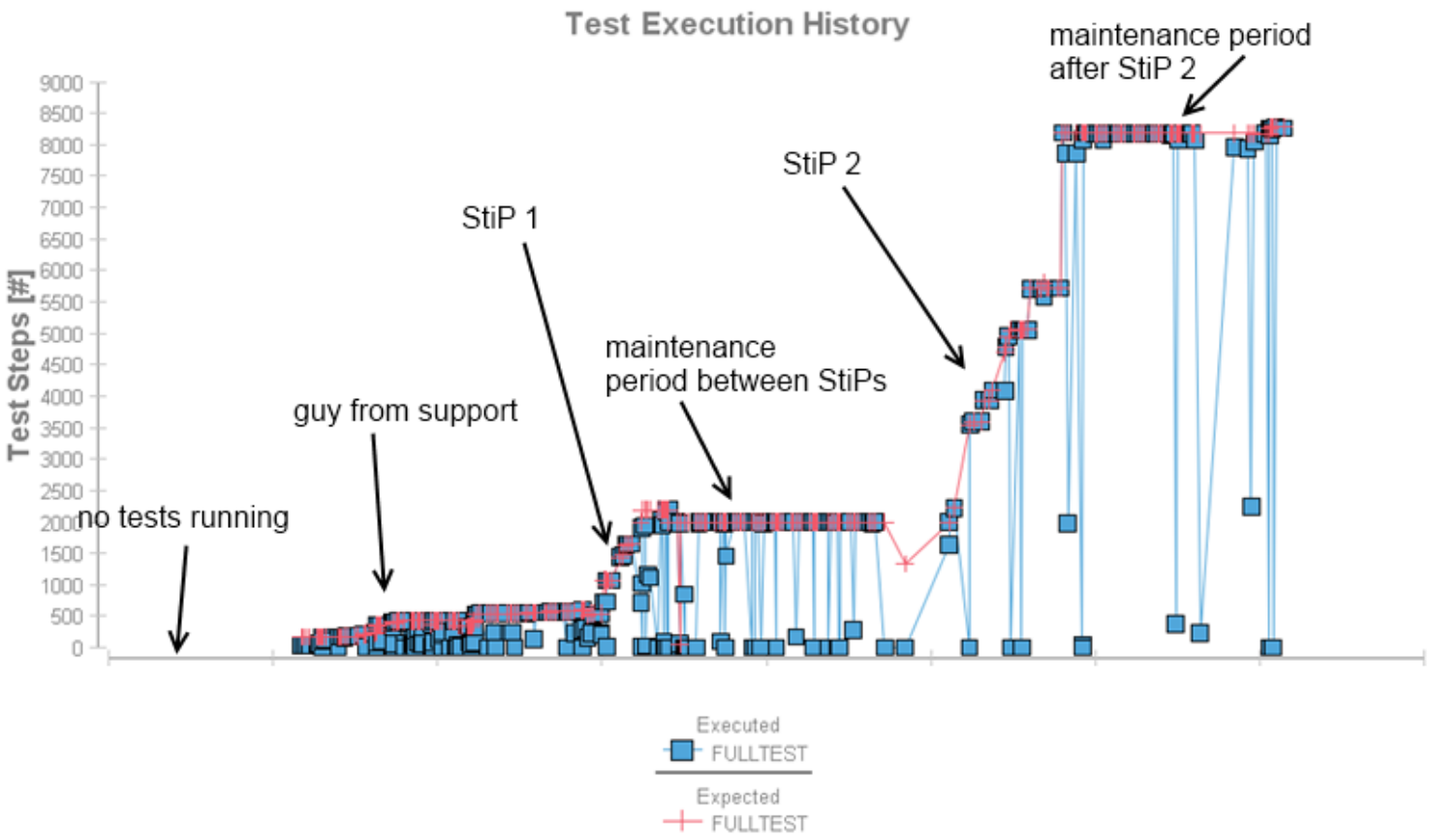
StiP number two

- **Preparation**
  - Training
  - Introduction to software and existing tests
  - Documentation from previous StiP
  - Discussion of test plan
- **Progress**
  - Addition of new tests, expansion of existing tests: 8000 steps
  - Test runs for 3 hours

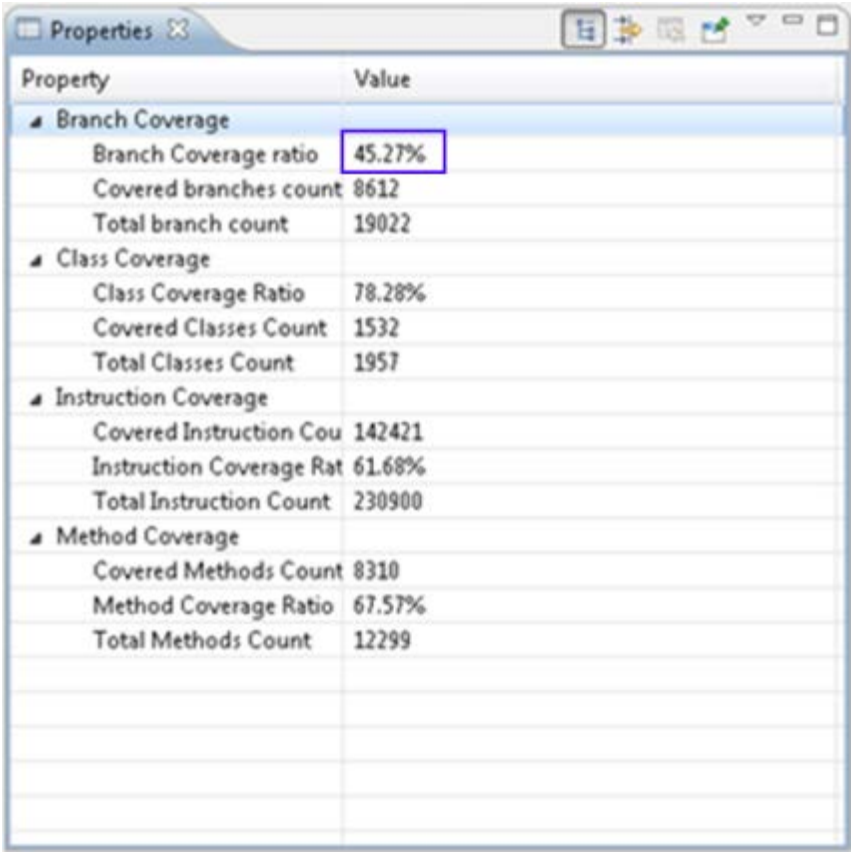
Progress


Test Execution History
Test Steps [#]
9000
8500
8000
7500
7000
6500
6000
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
Executed
FULLTEST
Expected
FULLTEST

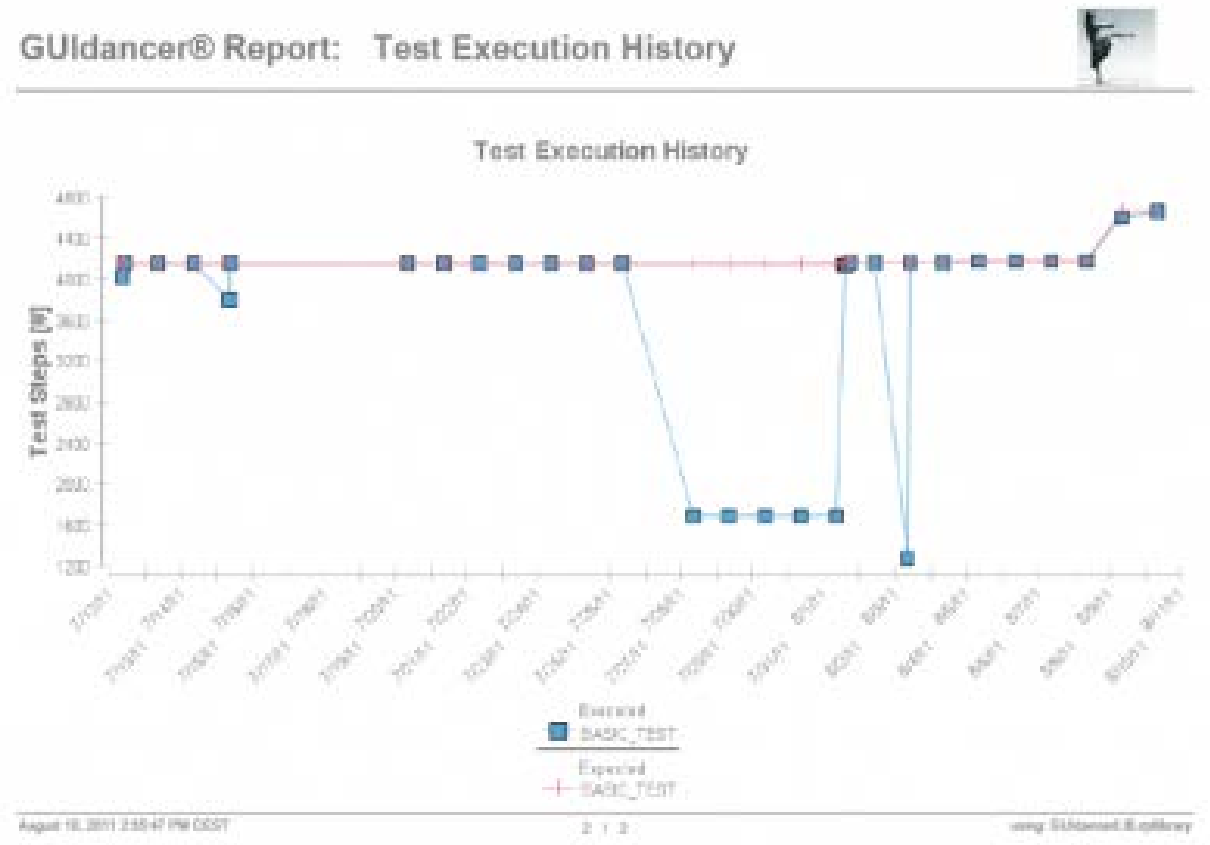
Progress


Test Execution History
Test Steps [#]
9000
8500
8000
7500
7000
6500
6000
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
no tests running
guy from support
StiP 1
maintenance
period between StiPs
StiP 2
maintenance period
after StiP 2
Executed
FULLTEST
Expected
FULLTEST

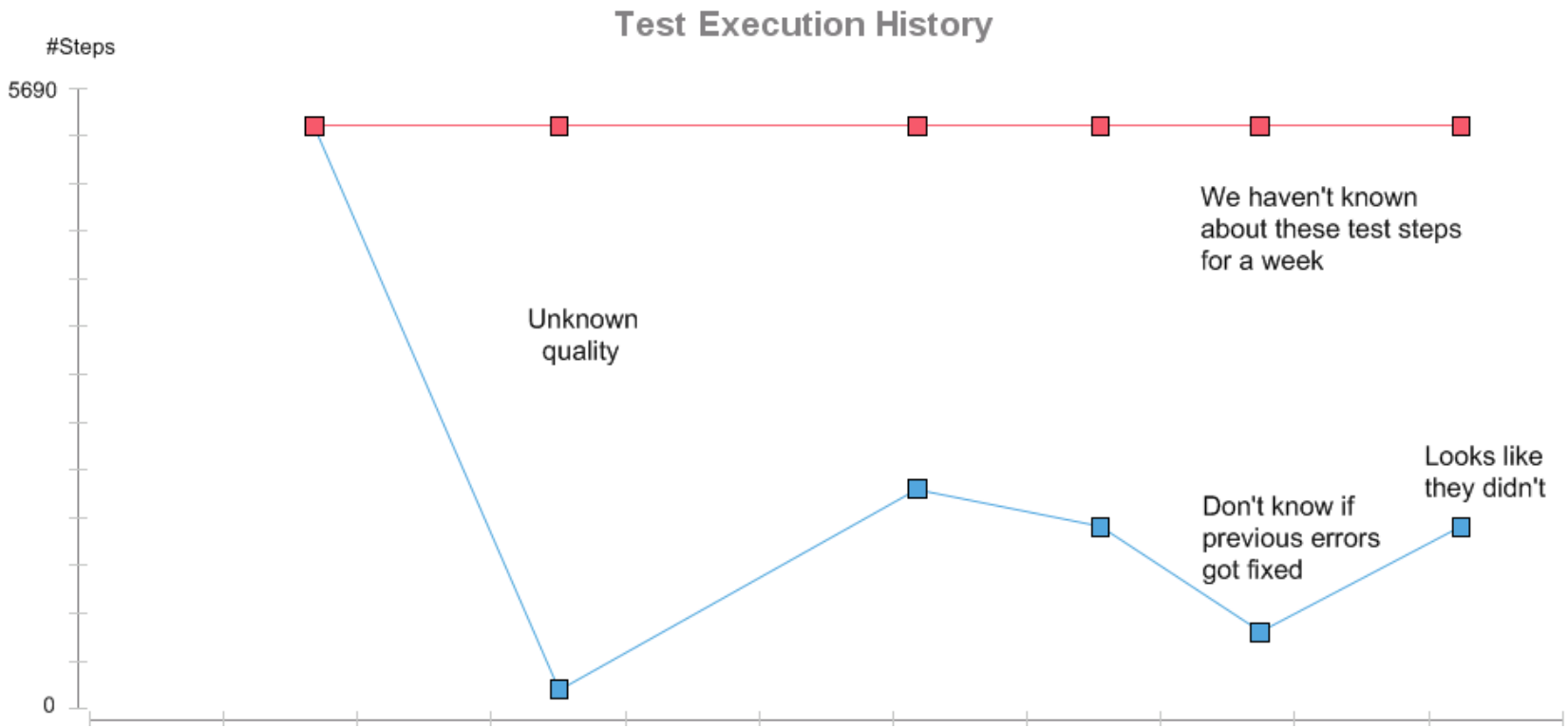
Progress


Properties
Property
Value
Branch Coverage
Branch Coverage ratio
45.27%
Covered branches count
8612
Total branch count
19022
Class Coverage
Class Coverage Ratio
78.28%
Covered Classes Count
1532
Total Classes Count
1957
Instruction Coverage
Covered Instruction Cou
142421
Instruction Coverage Rat
61.68%
Total Instruction Count
230900
Method Coverage
Covered Methods Count
8310
Method Coverage Ratio
67.57%
Total Methods Count
12299

Another example



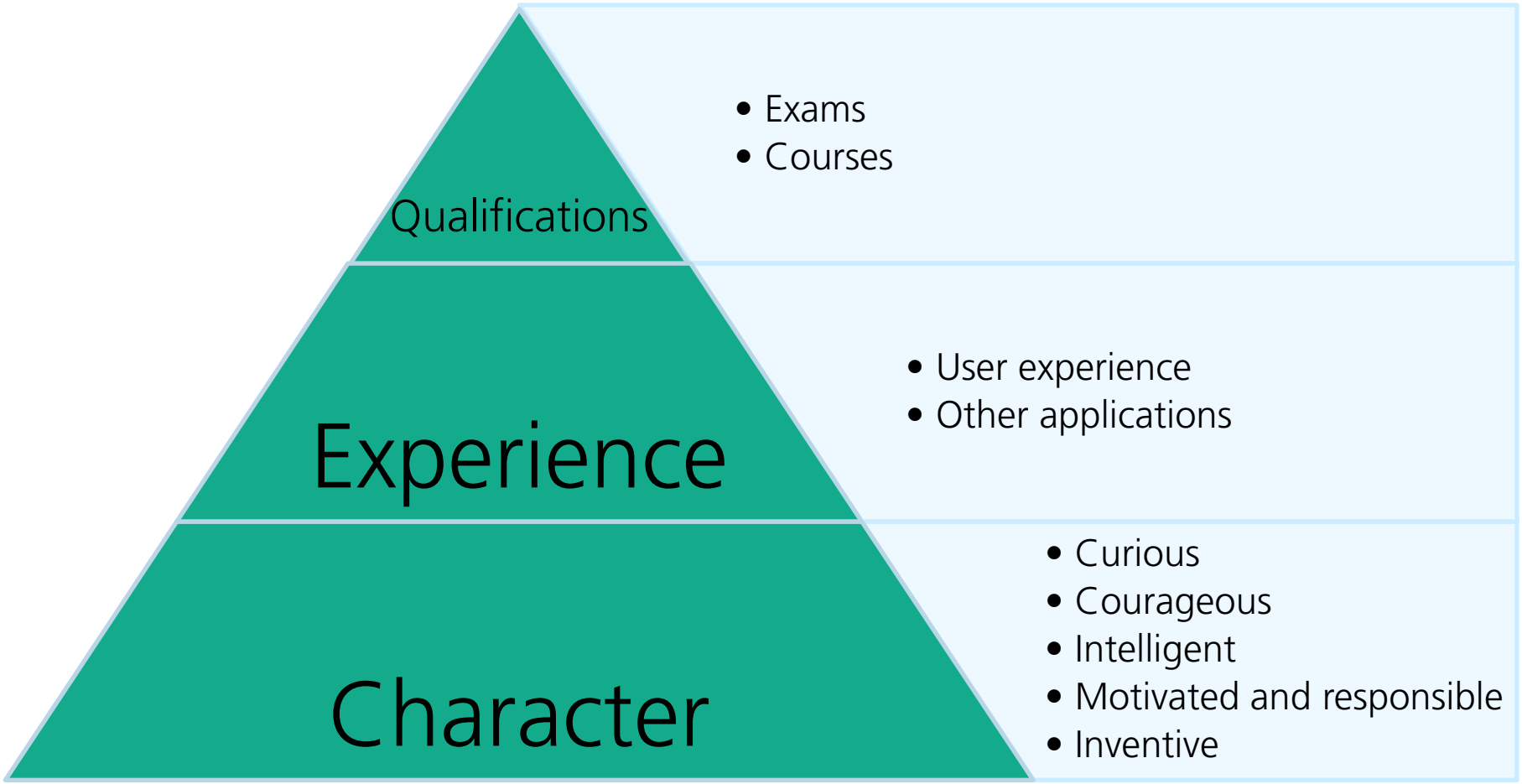
Counter example…



The moral of the story

- **Nice-to-Haves**
  - o Full time tester
  - o Qualified tester
  - o Constant test growth
- **Required**
  - + Good testers (…)
  - + Preparation
    - Training in tool
    - Good test design
    - Intro to software
    - Intro to tests
  - + Closeness to dev team
  - + Maintenance during "quiet" phases
  - + Continuous Integration (…)

What is a good tester?



Who makes a good tester?

User support

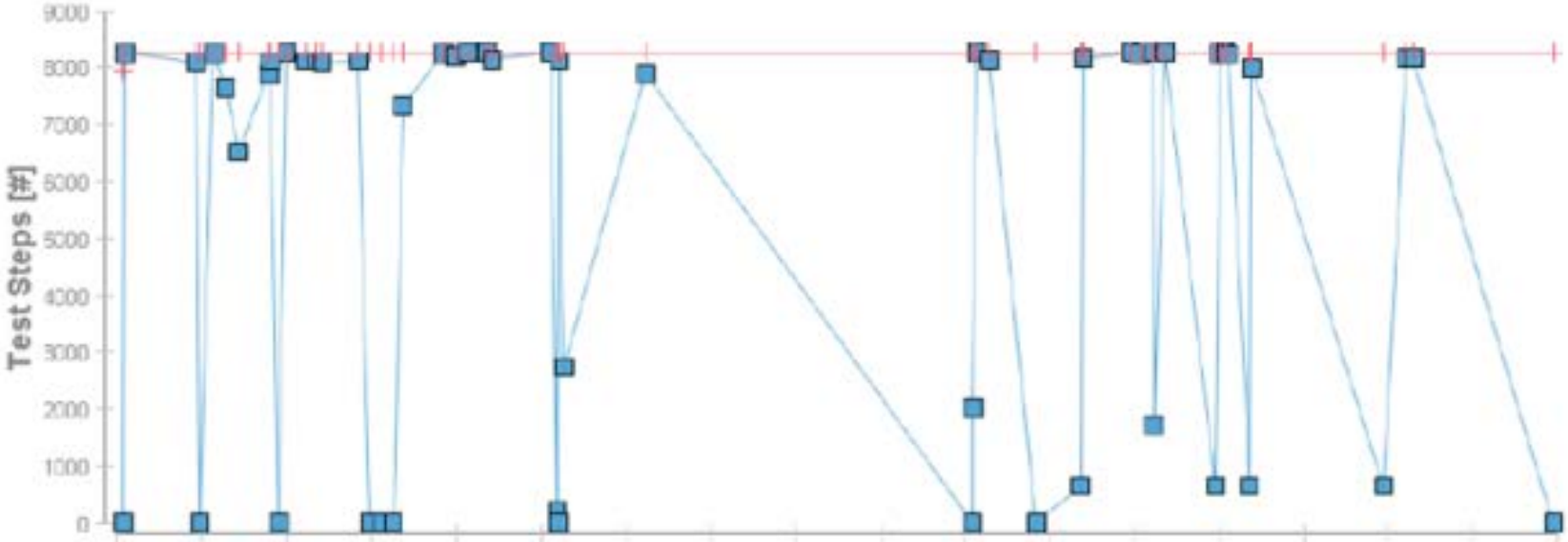
Customer liaison

Documentation

Students*

Trainers

The continuous integration hurdle



The continuous integration hurdle

- **CI must be in place**
  - Quick feedback from new tests
  - Monitoring of quality during maintenance phase
  - → Regression tests must run daily
- **Test environment must be stable**
  - Dedicated machines
  - Known status
- **Test results require reaction!**
  - Daily analysis
  - Fix bugs / alter tests

Strategies for cutting the right corners

1. Set up a continuous build and test process

2. Have “bursts“ of new test additions

Choose your tester carefully

Provide sufficient familiarization

3. React to regression test results in quiet phases

And they all lived happily ever after