Agile Testing Days
21 November 2012, Potsdam

Niels Malotaux

Testers are bearers of good news

N R Malotaux - Consultancy
The Netherlands
tel +31-30-2288868
fax +31-30-2288869
niels@malotaux.nl
www.malotaux.nl

Niels Malotaux

Niels Malotaux is an independent Project Coach and expert in optimizing project performance. He has well over 35 years experience in designing electronic and software systems, at Delft University, in the Dutch Army, at Philips Electronics and 20 years leading his own systems design company. Since 1998 he devotes his expertise to helping projects to deliver Quality On Time: delivering what the customer needs, when he needs it, to enable customer success. To this effect, Niels developed an approach for effectively teaching Evolutionary Project Management (Evo) Methods, Requirements Engineering, and Review and Inspection techniques. Since 2001, he taught and coached well over 150 projects in 30+ organizations in the Netherlands, Belgium, China, Germany, India, Ireland, Israel, Japan, Romania, South Africa and the US, which led to a wealth of experience in which approaches work better and which work less in real practice.

Niels puts development teams on the Quality On Time track and coaches them to stay there and deliver their quality software or systems on time, without overtime, without the need for excuses. Practical methods are developed, used, taught and continually optimized for:

- Evolutionary Project Management (Evo)
- Requirements Engineering and Management
- Reviews and Inspections.

Within a few weeks of turning a development project into an Evo project, the team has control and can tell the customer when the required features will all be done, or which features will be done at a certain date. Niels enjoys greatly the moments of enlightenment experienced by his clients when they find out that they can do it, that they are really in control, for the first time in their lives.

N R Malotaux
Consultancy

Niels Malotaux
project coach

Result Management

Bongerdlaan 53
3723 VB Bilthoven
The Netherlands
tel +31-30-228 88 68
fax +31-30-228 88 69
mob +31-6-5575 3604
niels@malotaux.nl
www.malotaux.nl

Niels Malotaux

Testers are bearers of good news

Testers are Bearers of Good News

Niels Malotaux

N R Malotaux Consultancy

+31-30-228 88 68 niels@malotaux.nl www.malotaux.nl

Niels Malotaux

Project Coach

Helping projects and organizations very quickly to become

- More effective – doing the *right* things *better*
- More efficient – doing the *right things* better in *less time*
- Predictable – delivering *as predicted*

Getting projects back on track

niels@malotaux.nl www.malotaux.nl

Heard at various Testing Conferences

Testers are Bearers of Bad News

- Do testers delay the delivery of the product ?
- Or is it the poor quality of the developers ?
- Is there a 'gap' between developers and testers ?

Who is the main customer of Testing and QA ?

- Deming:
  - Quality comes not from testing, but from *improvement of the development process*
  - Testing does not improve quality, nor guarantee quality
  - It's too late
  - The quality, good or bad, is already in the product
  - You cannot test quality into a product
- Who is the main customer of Testing and QA ?
- Developers are the main customer
- What do we have to deliver to these customers ?
  What are they *waiting for* ?

Ultimate Goal of a What We Do

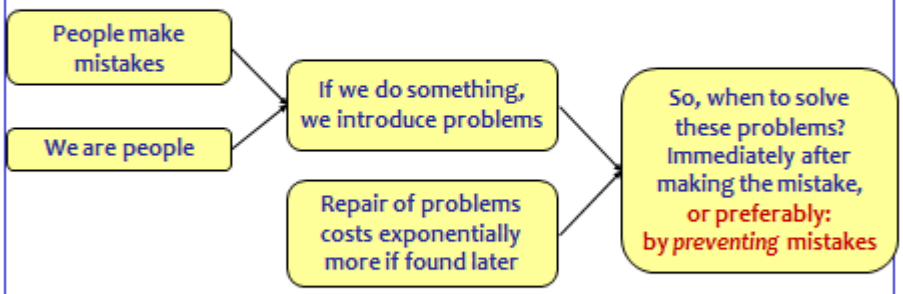
Delivering the Right Result at the Right Time,
wasting as little time as possible (= efficiently)

Quality on Time

- Providing the customer with
  - what he needs
  - at the time he needs it
  - to be satisfied
  - to be more successful than he was without it
- Constrained by (win - win)
  - what the customer can afford
  - what we mutually beneficially and satisfactorily can deliver
  - in a reasonable period of time

Malotaux – Agile Testing Days 2012 5

Inevitable consequence



Prevention costs much less than inject → find (?) → repair (?)

Malotaux – Agile Testing Days 2012 6

Dijkstra (1972)

It is a usual technique to make a program and then to test it

However:

Program testing can be a very effective way to show the presence of defects

but it is hopelessly inadequate for showing their absence

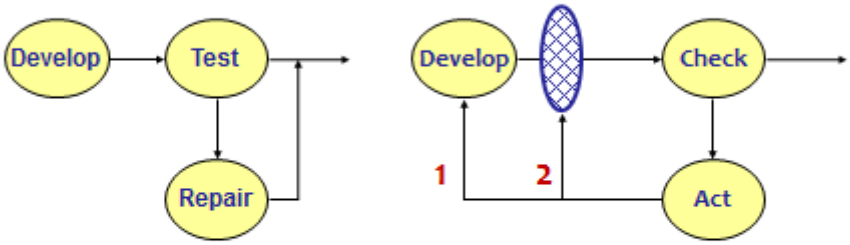
Conventional testing:

- Pursuing the very effective way to show the presence of defects

The challenge is, however:

- Making sure that there are no defects (development)
- How to show their absence if they're not there (testing?)

Testing is checking correctness



What we often see

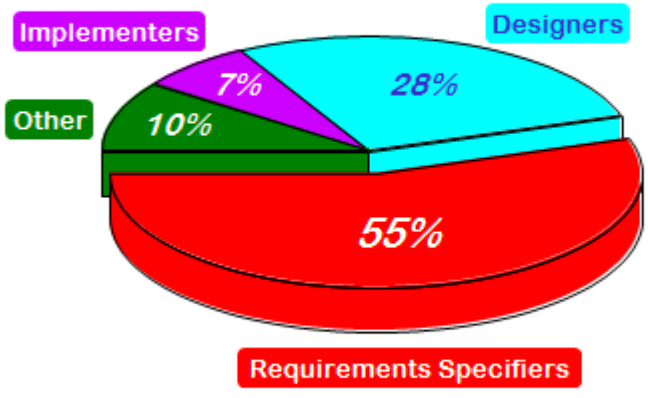
What we should expect

1. How can we prevent this ever happening again?
2. Why did our earliest sieve not catch this defect?

What techniques do testers have
to do their job properly ?

- What was their job ? (just checking)
- Is what Testers do right for doing their proper job ?
  - Focus on what's necessary to reach the goal
  - Even if that's not what you've been told before
  - Don't believe anything I say

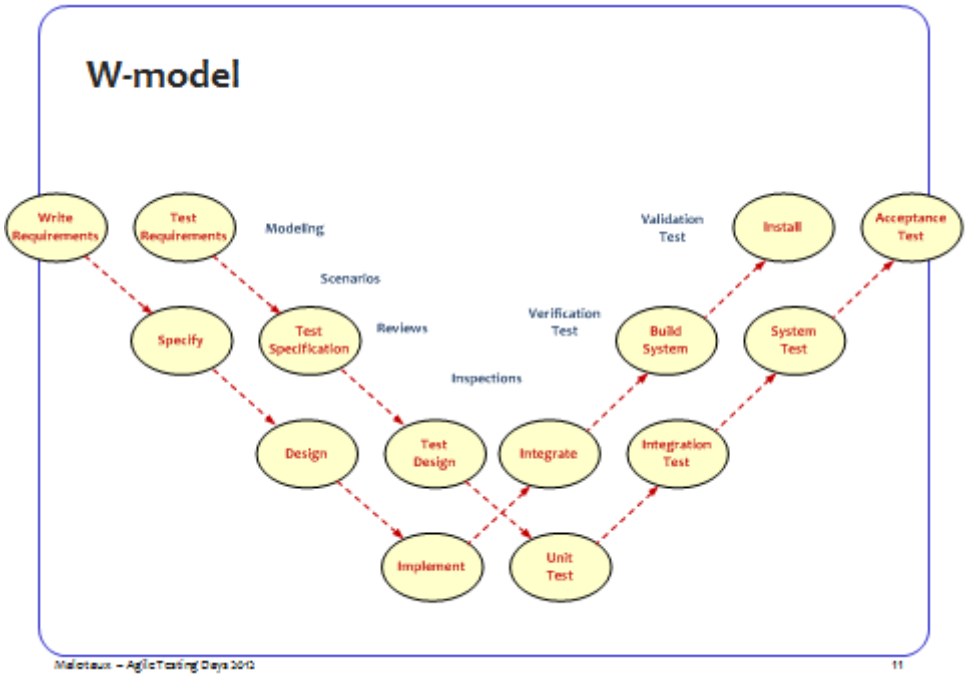
Typical Defect Injectors (cost breakdown)

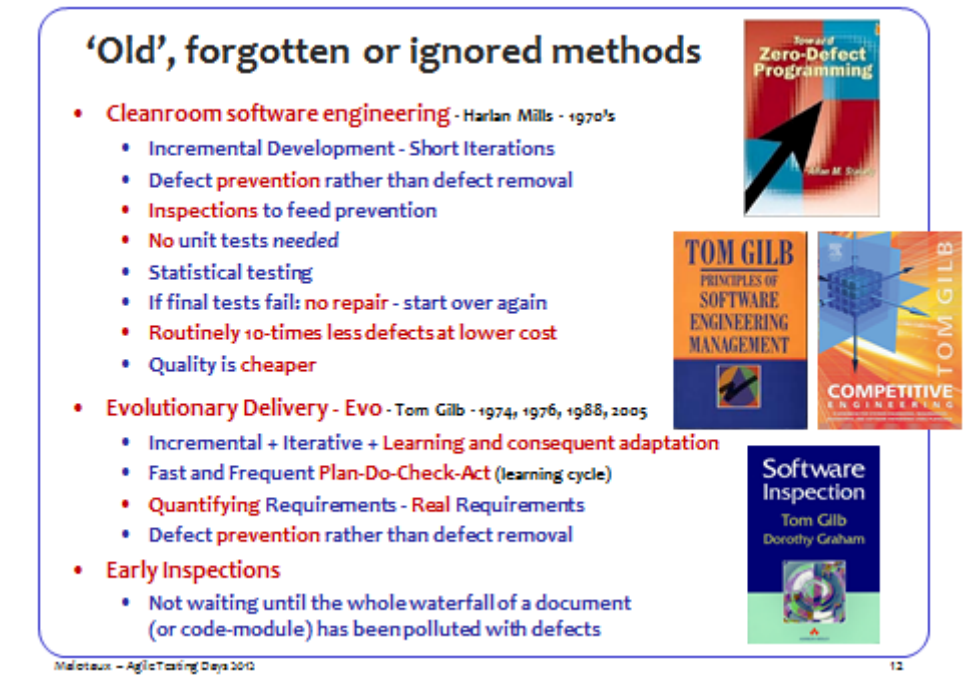


After Bender Associates, *1996*

- Where is our focus ?
- Where should our focus be ?

Testers are bearers of good news





Testing in Cleanroom (1970's)

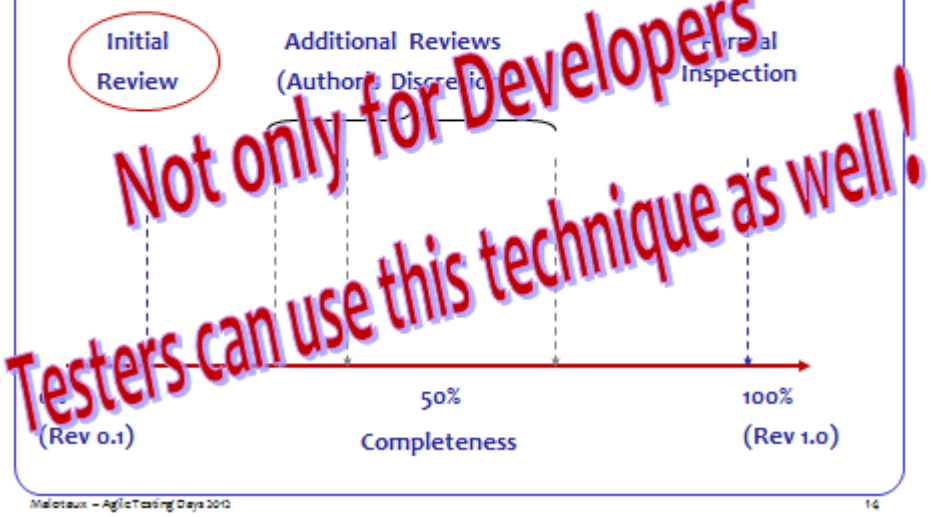
- Testing is an important part of the process, but it is done only after verification is successfully completed
- Testing is done:
  - Primarily to *measure quality*
  - Secondarily to find defects that escaped detection during verification
- Number of defects per thousand lines of code <10 after verification, compilation and syntax checking (= *before* test)
- Very good teams produce 2.3 defects per kLoC and reject code with 4 or 5 defects per kLoC
- No attempt is done to try to salvage rejected code by debugging
  - The code is sent back to the developers to be *rewritten* and reverified
  - Then it is tested as a completely new product
- Usage based testing
- Risk based testing

} Statistical testing

Malotaux – Agile Testing Days 2012 13

Early Inspection

Prevention costs less than Repair



Malotaux – Agile Testing Days 2012 14

E8

Case Study - Situation

- A tester's improvement writing successive test plans
  - Early Inspection used on an existing project to improve test plan quality
  - Test plan nearly "complete", so we simulated Early Inspection
  - First round: inspected 6 randomly-selected test cases
  - Author notes systematic defects in the results, reworks the document accordingly (~32 hrs)
  - Second round: inspected 6 more test cases; quality vastly improved
  - Test plan exits the process and goes into production
  - The author goes on to write another test plan

Case Study - Results

First round inspection	6 major defects per test case
Second round	0.5 major defects per test case

- Time investment: 2 hours in initial review, 36 hours total in final formal inspection, excluding rework
  (2 inspections, 4 hrs each, 4 checkers, plus preparation and debrief)
- Historically about 25% of all defects found by testing were closed as "functions as designed", still 2-4 hrs spent on each to find out
- This test plan yielded over 1100 software defects with only 1 defect (0.1 %) closed as "functions as designed"
- Time saved on the project: 500 - 1000 hrs (25% x 1100 x 2-4 hrs)

Defect Prevention in action: First inspection of this tester's next test plan: 0.2 major defects per test case

Let's move

Let's move from

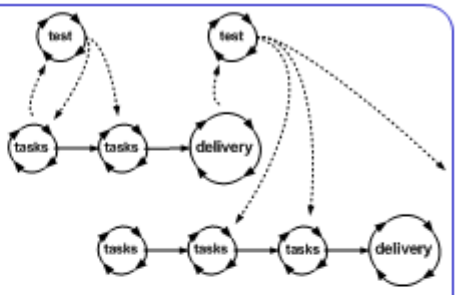
Fixation to Fix

to

Attention to Prevention

- If we don't deal with the root, we (and 'they') will keep making the same mistakes over and over
- Toyota Production System: "Stop the Line"
- Without feedback, we won't even know
- Only with *quick feedback* we can put the repetition to a halt

Organizing it



- Developers organize their work in weekly TaskCycles
- Testers organize their work in weekly TaskCycles
- Testers *know* what they are supposed to test
  Because they *know* what the developers are doing and will deliver
- Testers conclude their work in sync with developers
- Testers check work in progress *even before* it is finished

The aim of Testing

- Being done as soon as the development is done
- Well, almost
- Excuses, excuses, excuses
  - The developers are always late
    (Developers should live up to their promises)
  - The developers don't take testers seriously
    (Developers should listen to testers for help)
  - The developers don't inject enough defects
    (Now testing becomes a real challenge)
  - We are the bearers of bad news
    (find out what we're *really* supposed to do)
- Helping development to be successful

No excuses needed!

Malotaux – Agile Testing Days 2012 19

Testers are Bearers of Good News

Niels Malotaux

N R Malotaux Consultancy

+31-30-228 88 68 niels@malotaux.nl www.malotaux.nl

Malotaux – Agile Testing Days 2012 20

More

www.malotaux.nl/?id=booklets

1 Evolutionary Project Management Methods (2001)
Issues to solve, and first experience with the Evo Planning approach

2 How Quality is Assured by Evolutionary Methods (2004)
After a lot more experience: rather mature Evo Planning process

3 Optimizing the Contribution of Testing to Project Success (2005)
How Testing fits in

3a Optimizing Quality Assurance for Better Results (2005)
Same as Booklet 3, but for non-software projects

4 Controlling Project Risk by Design (2006)
How the Evo approach solves Risk by Design (by process)

5 TimeLine: How to Get and Keep Control over Longer Periods of Time (2007)
Replaced by Booklet 7, except for the step-by-step TimeLine procedure

6 Human Behavior in Projects (APCOSE 2008)
Human Behavioral aspects of Projects

7 Evolutionary Planning, or How to Achieve the Most Important Requirement (2008)
Planning of longer periods of time, what to do if you see you don't have enough time

8 Help! We have a QA Problem! (2009)
Use of TimeLine technique: How we solved a 6 month backlog in 9 weeks

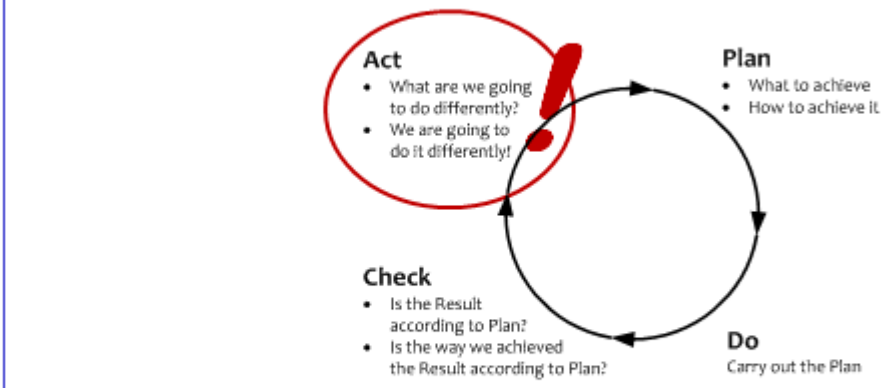
RS Measurable Value with Agile (Ryan Shriver - 2009)
Use of Evo Requirements and Prioritizing principles

www.malotaux.nl/?id=inspections
Inspection pages

Malotaux – Agile Testing Days 2012 21

The essential ingredient: the PDCA Cycle

(Shewhart Cycle - Deming Cycle - Plan-Do-Study-Act Cycle - Kaizen)



←The actual Shewhart Cycle was like this!

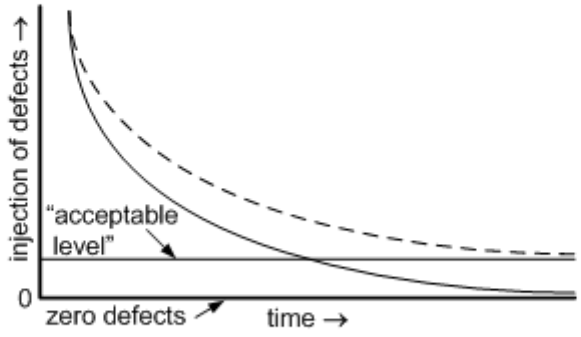
Evo - Kaizen SDM - Oct 2011 22

Do we deliver Zero Defect products ?

- How many defects do you think is acceptable ?

Is defect free software possible?

- Zero Defects is an asymptote



- When Philip Crosby started with Zero Defects in 1961, errors dropped by 40% almost immediately