



SD Best Practices 2005 Boston

27 September 2005

Class

Why are you testing anyway?

Optimizing the Contribution of Testing to Project Success

Niels Malotaux

N R Malotaux - Consultancy
The Netherlands
+31-30-2288868
+31-30-2288869
niels@malotaux.nl
www.malotaux.nl/nrm/English

SD Best Practices - Boston 2005
Niels Malotaux
Optimizing the Contribution of Testing to Project Success


Niels Malotaux

Niels Malotaux is an independent consultant and project coach, teaching immediately applicable methods for delivering Quality On Time to projects and organizations. He has some 30 years experience in designing hardware 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 teaching and coaching projects to deliver Quality On Time. Since 2001 he coached some 30 projects at 11 different organizations in the Netherlands, Belgium and USA. He is a frequent speaker at conferences.

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 Generation 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	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/nrm/English
<i>Evolutionary Project Management</i> <i>Requirements Engineering</i> <i>Review & Inspection</i>	

Optimizing the Contribution of Testing to Project Success

Niels Malotaux

N R Malotaux
Consultancy

+31-30-228 88 68

niels@malotaux.nl

www.malotaux.nl/nrm/English

1

The Goal

- **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**
 - what the customer can afford
 - what we mutually beneficially and satisfactorily can deliver
 - in a reasonable period of time

2

Optimizing the Contribution of Testing to Project Success

The Problem

- Still too many defects experienced by users
- Apparently
- Still too many defects generated by developers
 - Still too many defects remain undiscovered

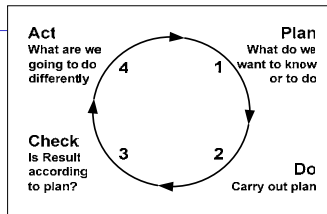
 - There is a lot of knowledge how to reduce the generation and proliferation of defects

There is a large budget to do something about it:

- Some 50% of project time is consumed by all kinds of testing
- About 50% of delivered software is never used
- (about 50% of developed software is never used)

3

Knowledge how to achieve the goal



- Using very short Plan-Do-Check-Act cycles
- Constantly selecting the most important things to do

then we can

- Most quickly learn what the real requirements are
- Learn how to most effectively and efficiently realize these requirements

and we can

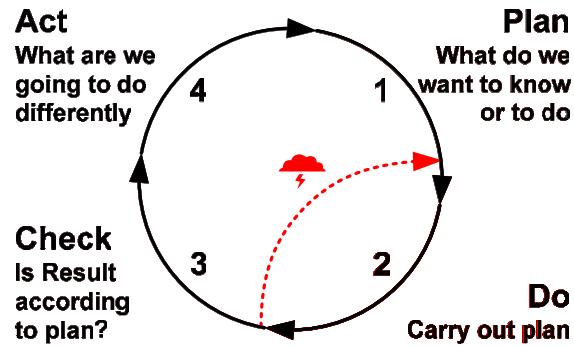
- Spot problems quicker, allowing more time to do something about them



4

Optimizing the Contribution of Testing to Project Success

The PDCA cycle



5

More knowledge

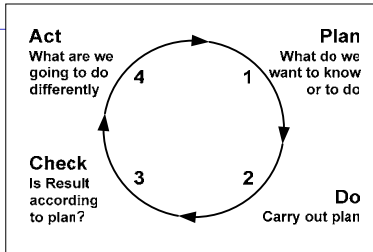
- Preventing defects is better than trying to find them
however
- Prevention requires a specific attitude that generally does not come naturally
- It requires more than doing our best

- Management should set a clear goal
- Then we can focus on achieving the goal
- If management does not set the goal, we should set it for ourselves

6

Optimizing the Contribution of Testing to Project Success

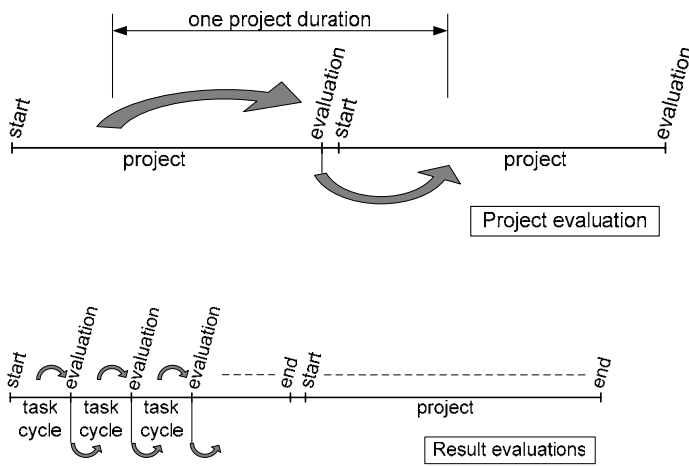
Evo



- **Evo (short for Evolutionary...)** uses this knowledge to the full
- **Combining Planning, Requirements- and Risk-Management into *Result Management***
- **Applying the PDCA-cycle actively, deliberately, rapidly and frequently, for *Product, Project and Process*, based on ROI**
- **A desire to Learning how to be better**
- **Projects seriously applying Evo, routinely conclude successfully on time, or earlier, *by design***
- **Proactively anticipating problems before they occur, working to prevent them**

7

Project evaluations



8

Optimizing the Contribution of Testing to Project Success

All we have to do ...

- **A defect is a problem caused to any of the stakeholders while relying on our results**
- **Making the customer more successful implies no defects**
- **All we have to do is delivering results without defects**
- **Do we?**

- **Is being late a defect?**

9

The process of defect injection

Conventional software development:

1. **Development phase: inject bugs**
2. **Debugging or Testing phase: find bugs and fix bugs**

Can't we do better?

10

Optimizing the Contribution of Testing to Project Success

Bugs are so important, are they really?

- “Software without bugs is impossible”
- Bugs are counted
- We try to predict the number of bugs we will find
- It is suspect if we don't find the expected number
- Bugs are normal
- What would we do if there were no bugs any more?

As long as we keep putting bugs in the center of the testing focus, there will be bugs

11

Defects found are symptoms of deeper problems

Repairing apparent defects creates several risks:

- Repair is done under pressure
- We think the problem is solved
- We introduce scars
- We keep repeating the same problems
- After finding the real cause, the redesign may make the repair redundant: time lost

12

SD Best Practices - Boston 2005
Niels Malotaux
Optimizing the Contribution of Testing to Project Success

Defects typically overlooked

- **Functions that won't be used** (superfluous requirements)
What's the use of repairing defects in the code of these requirements?
- **Nice things** (not checked, not paid for)
Shouldn't be there in the first place
- **Missing quality levels** (should have been in requirements)
Checking the implementation of the documented requirements won't help
- **Missing constraints** (should have been in requirements)
Product could be illegal
- **Unnecessary constraints** (not required)
What would testing say about these?

13

Testing is very expensive

- **You can prove the existence of a defect** (if you found one)
- **You cannot prove the absence of defects** (if you didn't find any)

- **Proving the absence of defects is difficult**
- **Proving the existence of defects is also difficult**

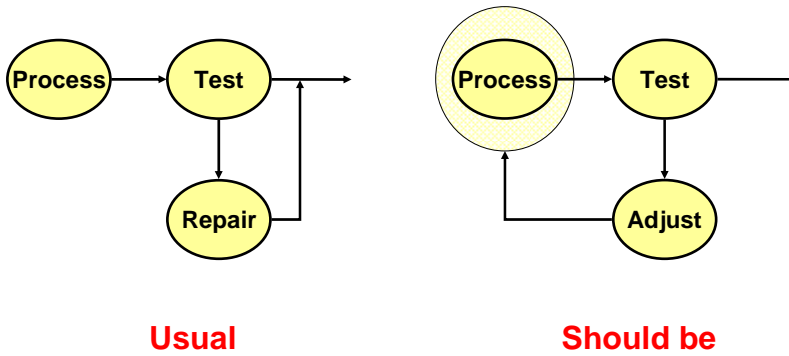
- **Why do we put so much emphasis on finding defects?**
- **While what we want is no defects**

- **Testers should learn better how to prove the absence of defects while**
- **The developers should learn better how to avoid defects**
- **Testers can help them**

14

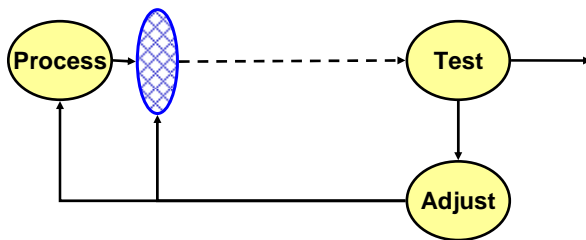
Optimizing the Contribution of Testing to Project Success

Testing is checking correctness



15

Testing is checking correctness



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

16

Optimizing the Contribution of Testing to Project Success

Let's move

Let's move from

- **Fixation to Fix**

to

- **Attention to Prevention**

- **If we don't deal with the root, we will keep making the same mistakes over and over**
- **Without feedback, we won't even know**
- **With quick feedback, we can put the repetition to a halt**

17

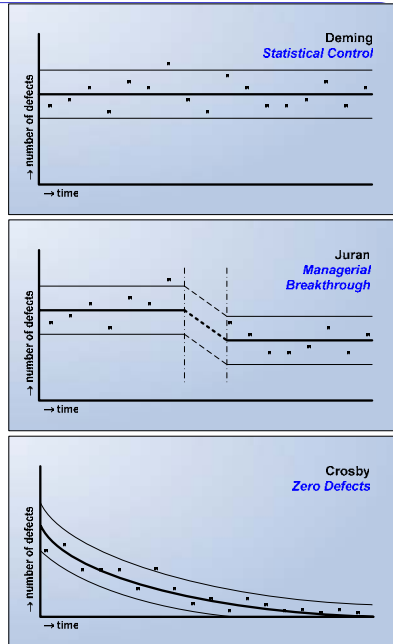
Hurray, it works!

- **If our result is supposed to make the customer more successful ...**
- **How do we make sure that success is going to happen?**
- **Do testers check this?**

18

Optimizing the Contribution of Testing to Project Success

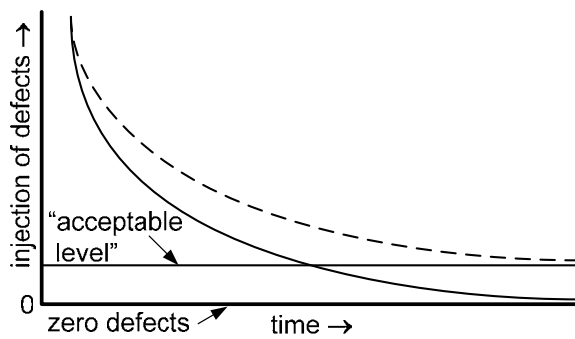
Deming - Juran - Crosby



19

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**

20

SD Best Practices - Boston 2005
Niels Malotaux
Optimizing the Contribution of Testing to Project Success

Absolutes of Quality

- **Conformance to requirements**
- **Obtained through prevention**
- **Performance standard is zero defects**
- **Measured by the price of non-conformance (PONC)**

Philip Crosby, 1970

- **The purpose is customer success
(not customer satisfaction)**

Added by Philip Crosby Associates, 2004

21

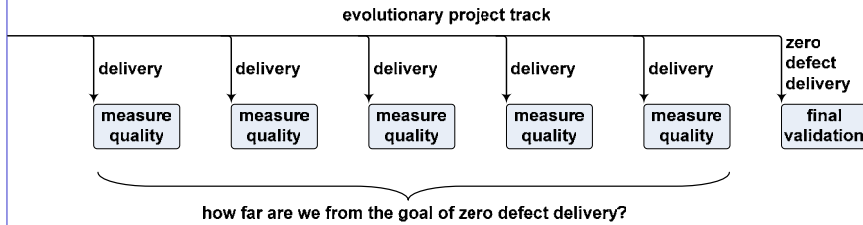
Attitude

- **As long as we think defect free software is impossible, we will keep producing defects**
- **From now on, we don't want to make mistakes any more**
- **We feel the failure (if we don't feel failure, we don't learn)**
- **If we deliver a result, we are sure it is OK and we are surprised when there proves to be a defect after all**
- **We do what we can to improve (continuous PDCA)**

22

Optimizing the Contribution of Testing to Project Success

Current Evo Testing



- **Final validation shouldn't find any problems**
- **Earlier verifications mirror quality level to developers: how far from goal and what still to learn**
- **Evo has *no debugging phase!***

23

Further Improvement

- **Testers focus on a clear goal**
- **Finding defects is not the goal**
- **Project Success is**
- **Tester's customer is "the developers"**
- **Testers select and use any method appropriate**
- **Testers check work in progress *even before* it is delivered**
- **Testers solve the Review and Inspection organizing problem**
- **Testing is organized the Evo way, entangling intimately with the development process**

24

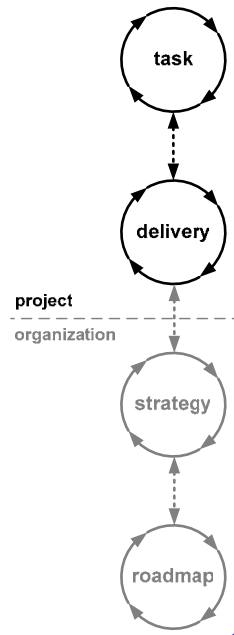
Optimizing the Contribution of Testing to Project Success

Cycles in Evo

- TaskCycle
- DeliveryCycle
- TimeLine

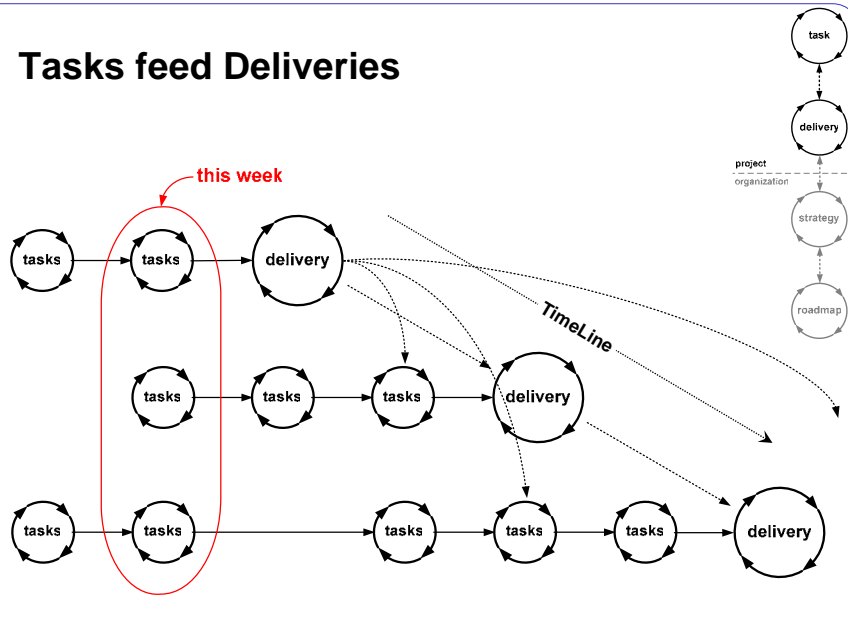
During these Cycles we are constantly optimizing

- The product
- The project
- The process



25

Tasks feed Deliveries

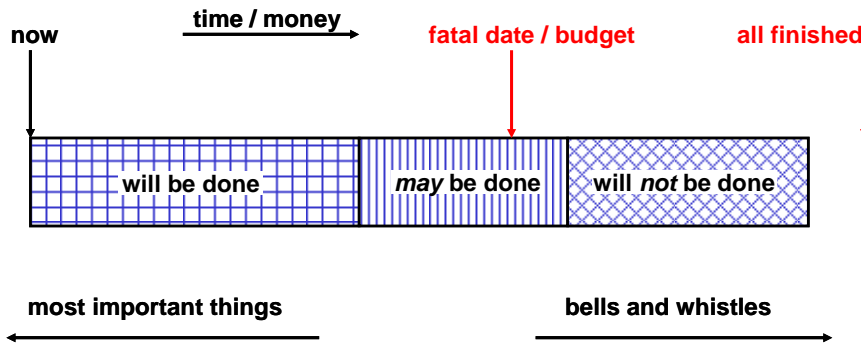


26

Optimizing the Contribution of Testing to Project Success

TimeLine

What the customer wants, he cannot afford



27

Weekly 3-Step Procedure

- 1. Individual preparation**
 - Conclude current tasks
 - What to do next
 - Estimations
 - How much time available
- 2. Modulation with / coaching by Project Management**
 - Status
 - Priority check
 - Feasibility
 - Commitment and decision
- 3. Synchronization with group (team meeting)**
 - Formal confirmation
 - Concurrency
 - Learning
 - Helping
 - Socializing

28

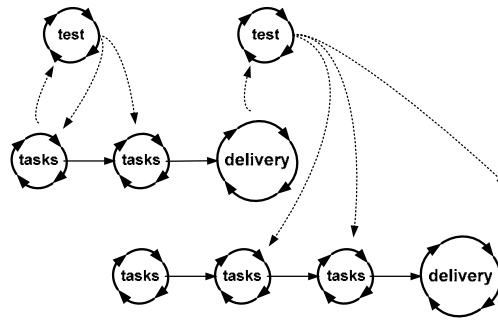
Optimizing the Contribution of Testing to Project Success

Developers are constantly optimizing

- **The product**
how to arrive at the most effective product (goal !)
- **The project**
how to arrive at the most effective product effectively and efficiently
- **The process**
 - Finding ways to do better
 - Learning from other methods
 - Absorbing those methods that work better
 - Shelving those methods that currently work less

29

Evo cycles for Testing



- **Testers organize their work in weekly TaskCycles**
- **DeliveryCycle is the Test-Feedback cycle**
- **Testers use their own TimeLine, synchronized with the developers TimeLine**
- **Testers conclude their work in sync with developers**
- **Testers check work in progress *even before* it is delivered**

30

Optimizing the Contribution of Testing to Project Success

Testers are constantly optimizing

- **The product**
how to arrive at the most effective product (goal !)
- **The project**
how to arrive at the most effective product effectively and efficiently
- **The process**
 - Finding ways to do better
 - Learning from other methods
 - Absorbing those methods that work better
 - Shelving those methods that currently work less

31

Testers are part of the project

- **Participating in the weekly 3-step process:**
 1. Individual Preparation
 2. 1-to-1's with project management
 - Project Manager (project issues)
 - Architect (product issues)
 - Test Manager (testing issues)
 3. Team meeting: Synchronization and synergy with the team
- **Testers see what developers are doing:**
 - No ambiguity with what the developers are doing
 - To which degree requirements are implemented

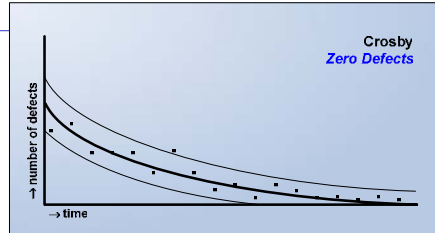
32

CRPR Database

- **Change Requests**
CR: customer pays
- **Problem Reports**
PR: you pay
- **Where, what, when, who**
- **Urgency, severity**
- **Classification**
- **Status**
- **Where caused and root cause**
- **Where should it have been found earlier**
- **Prevention plan**
- **Analysis tasks defined and put on Candidate Task List**
- **Prevention tasks defined and put on Candidate Task List**
- **Check lists updated for finding issues easier, in case prevention doesn't work yet**

33

Metrics



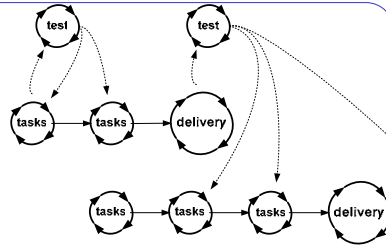
Don't improve non-value-adding activities. Better eliminate them

- **Estimation - planning - tracking**
 - If estimation is a TimeBox, tracking is a "zero activity"
- **Defects per kLoC or Defects per Page**
Stop counting defects, it conveys a bad message
- **Incoming defects per month (by test, by user)**
Don't count. Do something. Users shouldn't experience problems.
- **Defect detection effectiveness or Inspection yield**
 - Yield is 30% ~ 80%; testers are human after all
 - Zero defects at user means zero defects before final test
 - Whether that is difficult is beside the point

34

Optimizing the Contribution of Testing to Project Success

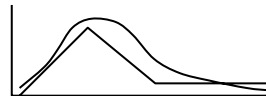
More metrics



- **Cost to find and fix a defect**
 - The less defects the higher the cost per defect
 - This was a bad metric anyway
- **Closed defects per month**
 - Closing depends on prioritizing process, through Candidate Task List
- **Age of open customer found defects**
 - Purpose of many metrics seems to be *policing*: not trusting people to take appropriate action
 - In Evo we take appropriate action
- **Remaining defects**
 - Still useful as measure of Prevention success

35

When are we done with testing?



- **Conventional:**
 - Number of bugs found per day less than n
 - Defect backlog decreased to zero
 - Prediction by curve fitting based on early found defect numbers
 - Using historical data
 - Other?
- **Evo:**
 - The project is ready at the agreed date, or earlier
 - That includes testing

36

Useful Evo metric

- **Size of the smile on the customers face**
- **In many cases, the Evo attitude and techniques replace the need for metrics**
- **I did not say always**

37

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 bugs*
 - *but it is hopelessly inadequate for showing their absence*
 - **Conventional testing is pursuing the very effective way to show the presence of bugs**
 - **The challenge is, however, to solve the hopeless inadequacy of showing their absence**
 - **And working towards their absence**

38

Optimizing the Contribution of Testing to Project Success

Links

- <http://www.gilb.com>
Tom Gilb's website: Evo guru
- <http://www.malotaux.nl/nrm/English>
Niels' activities: Evo evangelist
- <http://www.malotaux.nl/nrm/Evo>
Evo pages
- <http://www.malotaux.nl/nrm/pdf/MxEvo.pdf>
Evolutionary Project Management Methods
(issues and 2001 experience)
- <http://www.malotaux.nl/nrm/pdf/Booklet2.pdf>
How Quality is Assured by Evolutionary Methods
(more recent practical implementation experience)
- <http://www.malotaux.nl/nrm/pdf/EvoTesting.pdf>
Optimizing the Contribution of Testing to Project Success
- <http://www.malotaux.nl/nrm/Evo/ETAF.htm>
Download the Evo Task Administrator (ETA) tool
(expects MSAccess2000-2003)

39

More on Evo

Wednesday 1:45 ~ 3:15

Niels Malotaux

**Routinely assuring project success
you can do it too**

Wednesday 1:45 ~ 3:15 + 3:45 ~ 5:15

Debra Schratz

**Clear, Concise and Measurable Requirements
Are They Possible?**

Using Planguage (Tom Gilb) to specify requirements to design your path to
customer's success

40

20050815