

# Inspection

## used in various ways

[www.malotaux.nl/conferences](http://www.malotaux.nl/conferences)

[www.malotaux.nl/booklets](http://www.malotaux.nl/booklets)

[www.malotaux.nl/inspections](http://www.malotaux.nl/inspections)

**Niels Malotaux**

**N R Malotaux**  
Consultancy

+31-655 753 604

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

**Result Management**

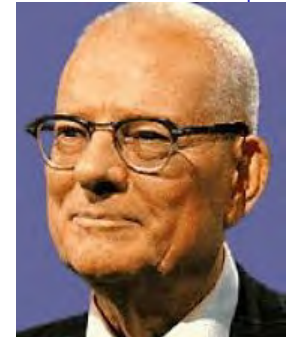
# Lean Quality Assurance

- **What is Lean ?** (better read the source: Taiichi Ohno)  
or [www.malotaux.nl/essenceoflean](http://www.malotaux.nl/essenceoflean)
- **What is Quality ?**
- **How do you get Quality ?**
- **What is the required Quality level ?**
- **How do you measure Quality ?**
- **How to assure Quality ?**
- **What is Quality Assurance ?**

# Who is the (main) customer of Testing and QA ?

- **Deming:**

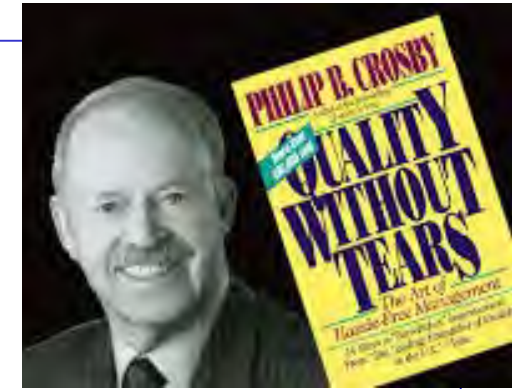
- Quality comes not from testing, but from *improvement of the development process*
- Testing does not improve quality, nor guarantee quality
- It's too late
- The quality, good or bad, is already in the product
- You cannot test quality into a product



**Deming**  
(1900-1993)

- **Who is the main customer of Testing and QA ?**
- **What do we have to deliver to these customers ?**  
*What are they waiting for ?*
- **Testers and QA are consultants to development**
- **Testing and QA shouldn't delay the delivery - How ?**

# Absolutes of Quality Crosby (1926-2001)



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

Philip Crosby, 1970

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

Added by Philip Crosby Associates, 2004



# Conformance to requirements

- **We meet the agreed requirements**

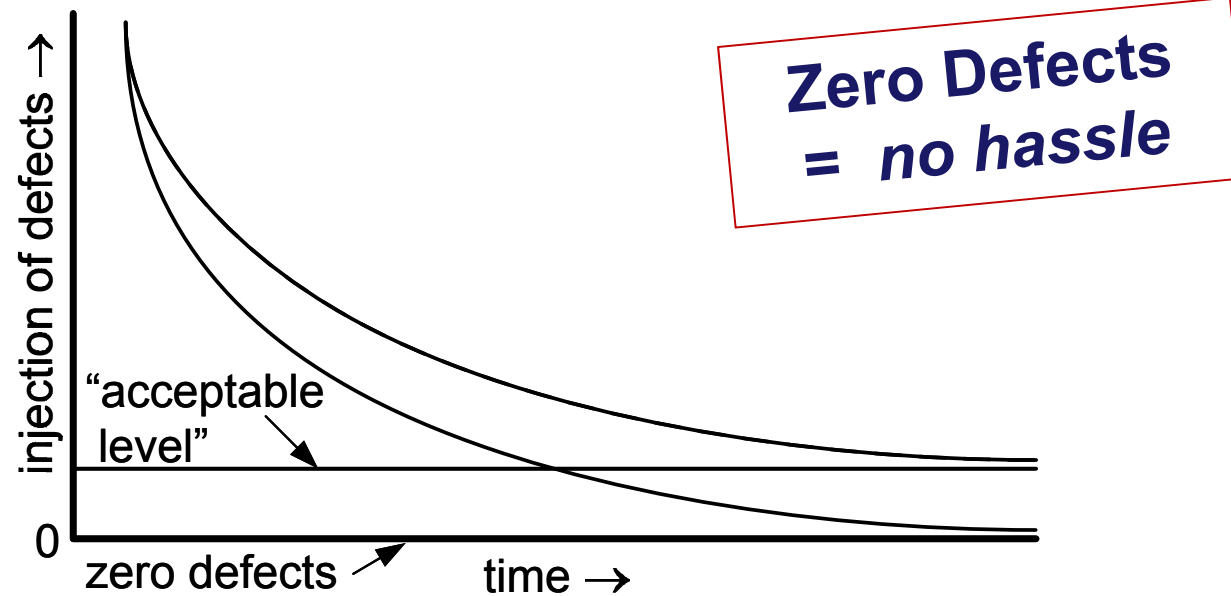
**or**

- **Have the requirements changed to what we and the customer really need**
- **We create requirements with care and we meet them with care**
- **Does your management take quality seriously ?**

**Phil Crosby**

# What is Zero Defects

- **Zero Defects is an *asymptote***



- **When Philip Crosby started with Zero Defects in 1961, errors dropped by 40% almost immediately**
- **AQL > Zero means that the organization has settled on a level of incompetence**
- **Causing a hassle other people have to live with**

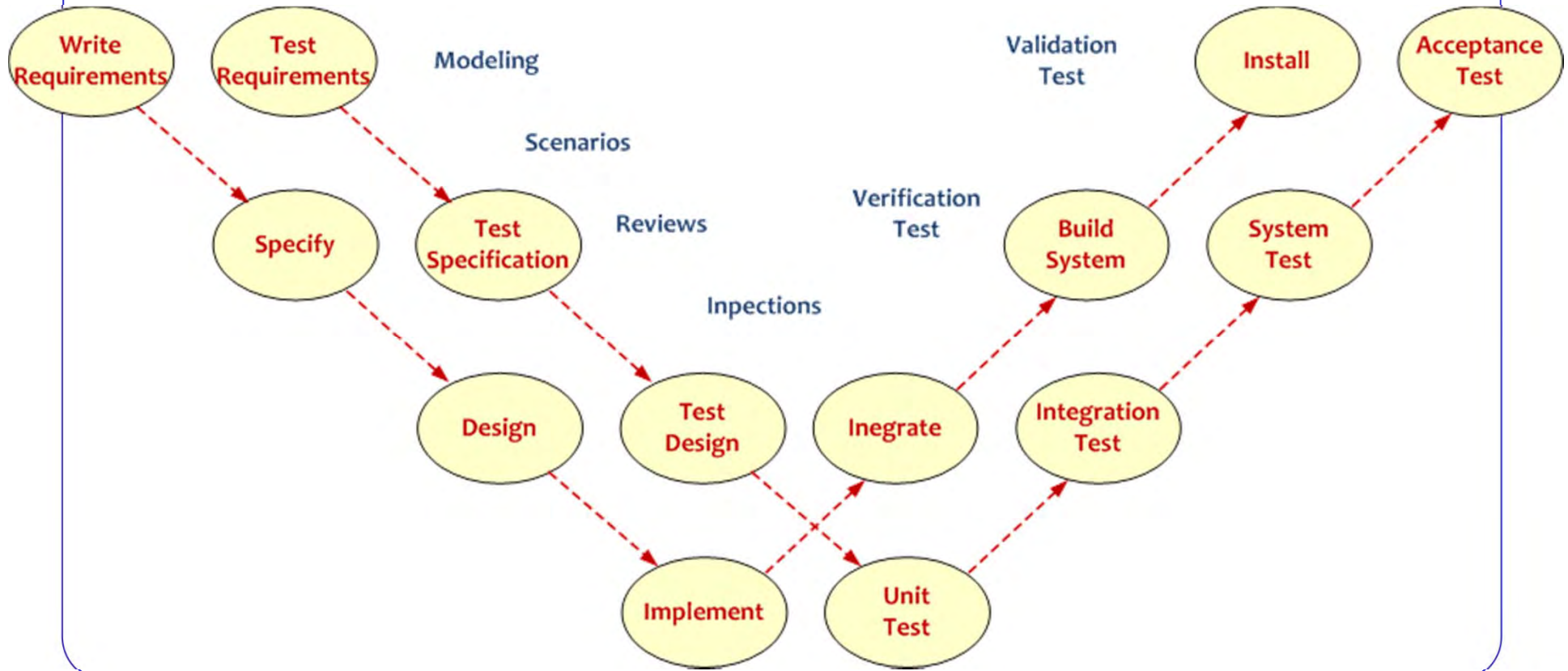
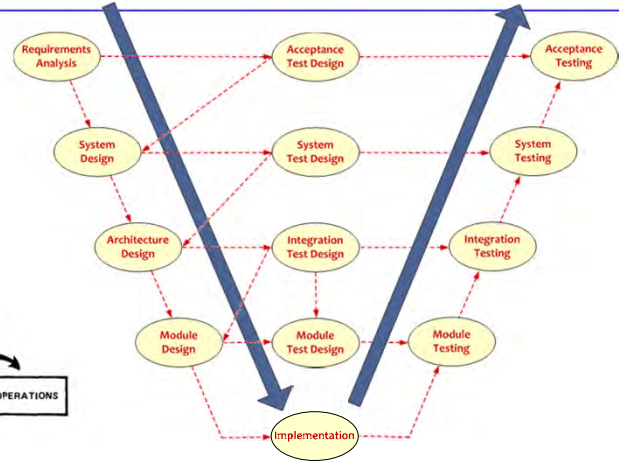
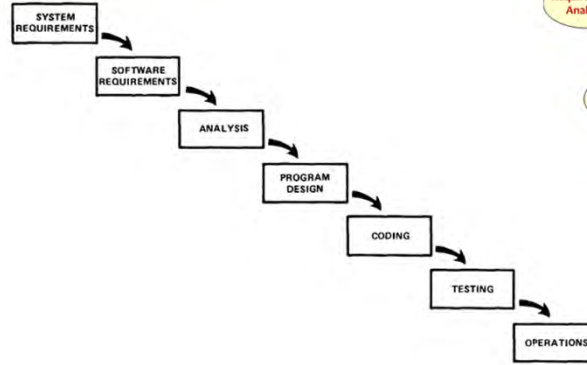
# Philip Crosby

[Quality is Still Free]

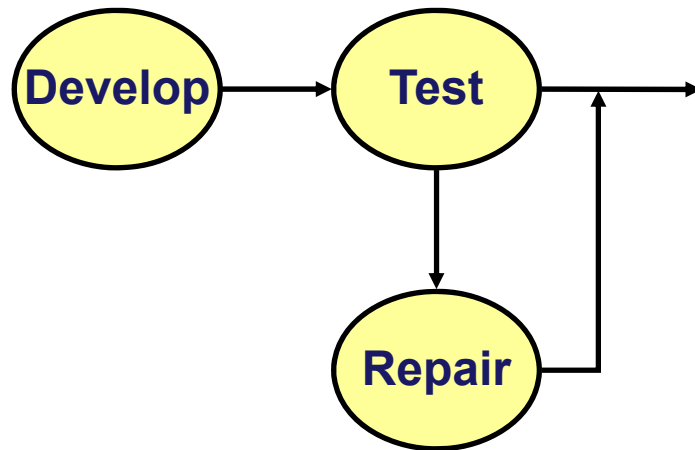
- **Conventional wisdom says that error is inevitable**
- **As long as the performance standard requires it, then this self-fulfilling prophecy will come true**
- **Most people will say: People are humans and humans make mistakes**
- **And people do make mistakes, particularly those who do not become upset when they happen**
- **Do people have a built-in defect ratio ?**
- **Mistakes are caused by two factors:  
lack of knowledge and lack of attention**
- **Lack of attention is an attitude problem**



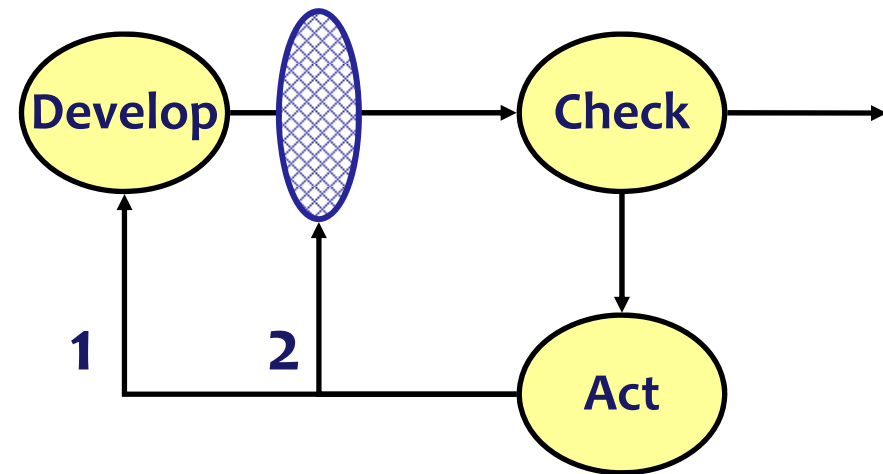
# W-model



# 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 ?

# Root Cause Analysis

## If a defect is found:

- **Is Root Cause Analysis routinely performed ?**
- **What is the Root Cause of a defect ?**
- **Cause:**  
**The error that caused the defect**
- **Root Cause:**  
**What caused us to make the error that caused the defect**
- **Without proper RCA, we're doomed to repeat the same errors**

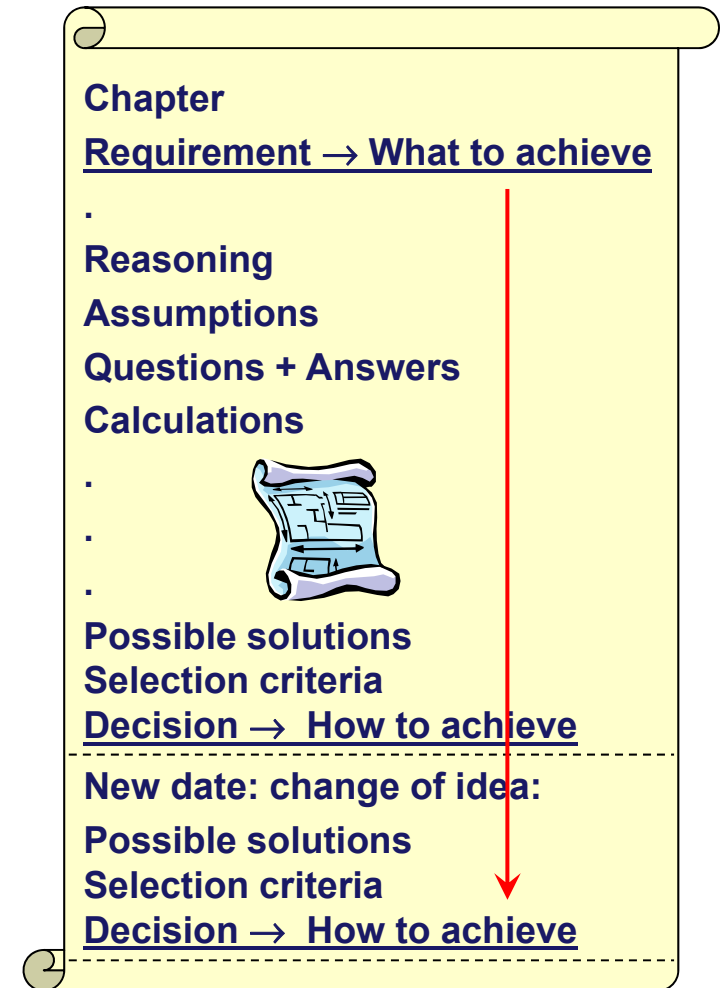
## Case: Can you teach Inspections ?

- **Short intro**
- **Are you regularly reviewing ?**
- **Let's do it: baseline**
  - Take a document
  - Reproduce one page
  - Do review
  - No issues
- **One rule ('source')**
  - Many issues

# **Datalog function improvement**

# DesignLog

- **In computer, not loose notes, not in e-mails, not handwritten**
  - Text
  - Drawings!
  - Chapter per subject
  - Initially free-format
  - For all to see
- **All concepts contemplated**
  - Requirement
  - Reasoning
  - Assumptions
  - Questions
  - Calculations
  - Possible solutions
  - Selection criteria
  - Choices:
    - If rejected: why?
    - If chosen: why?
- **Implementation specification**



# Results

- **No code until design-log reviewed**
- **You're delaying my project !**
- **Example**
- **Solution**
- **Thanks, you saved my project**
- **Now we can review to check the design before implementation**
- **Did I do the same ?**
- **Telling people to change: resistance**
- **How to let people change themselves ...**

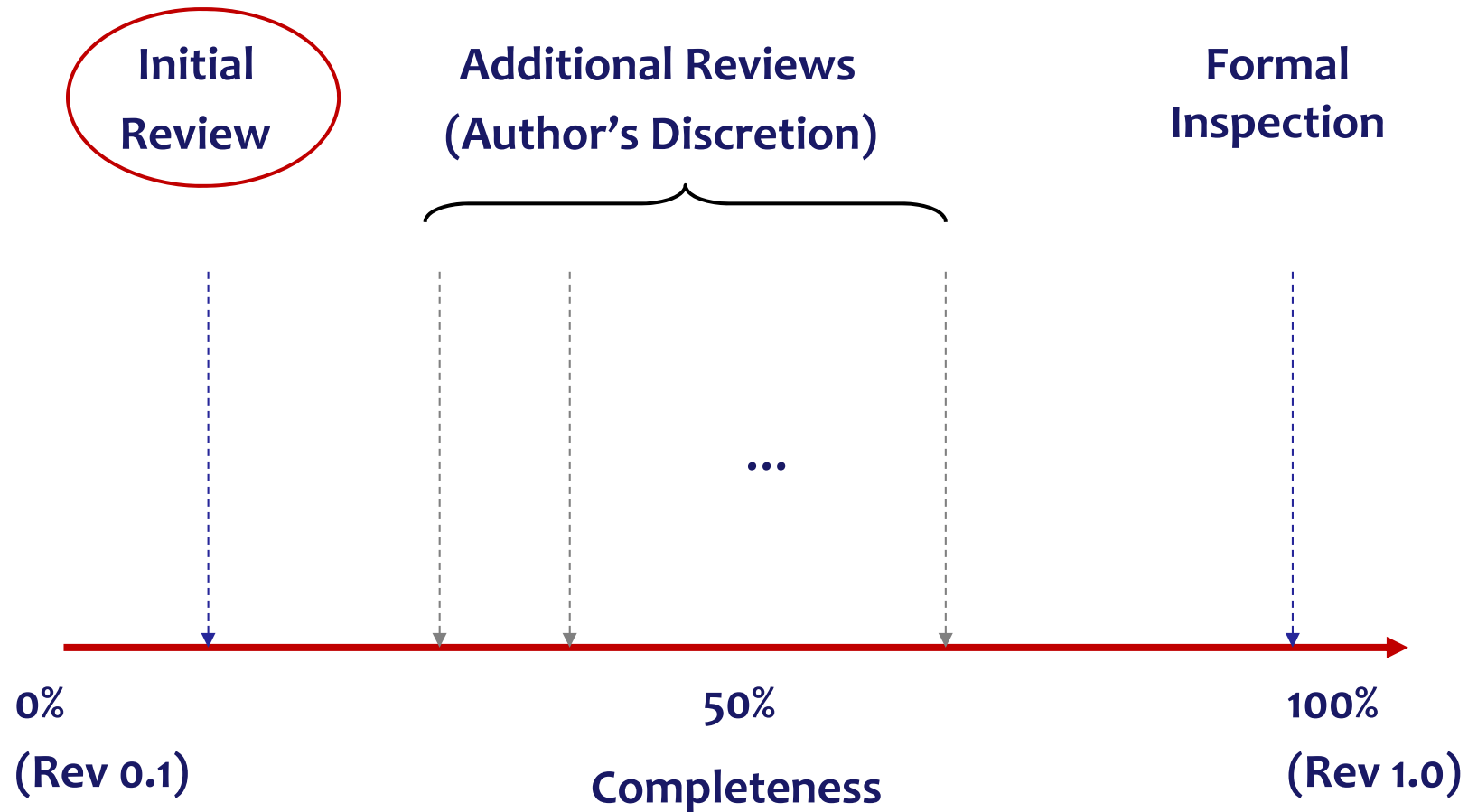
## Case: City of Amsterdam

- **Can you teach Inspections ?**
- **You'll ditch the document after the course !**
- **Ha ha**
- **Of course they did**



# Early Inspection

Prevention costs less than Repair



# Case: Early Inspection on Requirements

## Large e-business application with 8 requirements authors

- Each sent the first 8-10 requirements of estimated 100 requirements per author  
(table format, about 2 requirements per page including all data)
- Initial reviews completed within a few hours of submission
- Authors integrated the suggestions and corrections, then continued to work
- Some authors chose additional reviews others did not
- Inspection performed on document to assess final quality level



# Results



<b>Average major defects per requirement in initial review</b>	<b>8</b>
<b>Average major defects per requirement in final document</b>	<b>3</b>

## **Time investment: 26 hr**

- 12 hours in initial review (1.5 hrs per author)
- About 8 hours in additional reviews
- 6 hours in final inspection (2 hrs, 2 checkers, plus prep and debrief)

**Major defects prevented: 5 per requirement in ~750 total**

**Saved  $5 \times 750 \times 10 \text{ hr} = 37500 \text{ hr} / 3 = 12500 \times \$50 = \$625000$**

# Early Inspection

Prevention costs less than Repair



# Case: Test Cases

## **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



# Results

<b>First round</b>	<b>6 major defects per test case</b>
<b>Second round</b>	<b>0.5 major defects per test case</b>



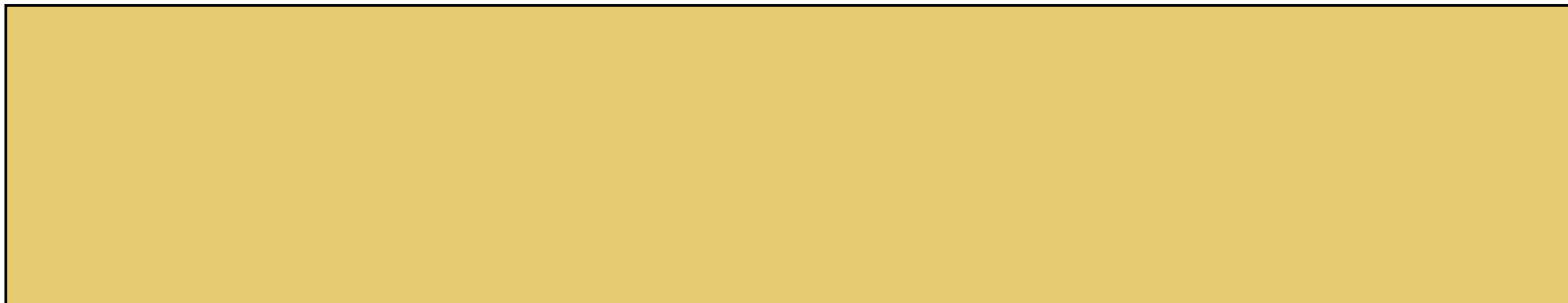
- **Time investment: 2 hours in initial review, 36 hours total in final formal inspection, excluding rework**  
(2 inspections, 4 hrs each, 4 checkers, plus preparation and debrief)
- **Historically about 25% of all defects found by testing were closed as “functions as designed”, still 2-4 hrs spent 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**

# Optimum Checking Rate

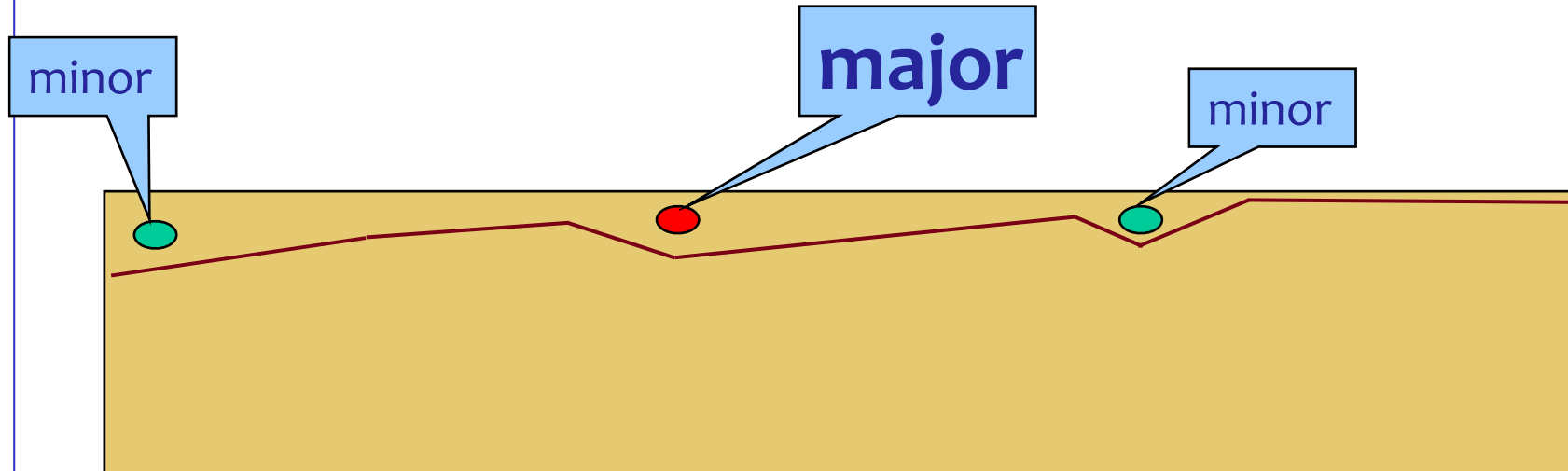
- The most **effective** individual speed for ‘checking a document against all related documents’ in page/hr
- Not ‘reading’ speed, but rather **correlation** speed
- Failure to use it, gives ‘bad estimate’ for ‘Remaining defects’
  
- 100~250 SLoC per hour
- 1 page of 300 words per hour (“logical page”)

# Here's a document: review it





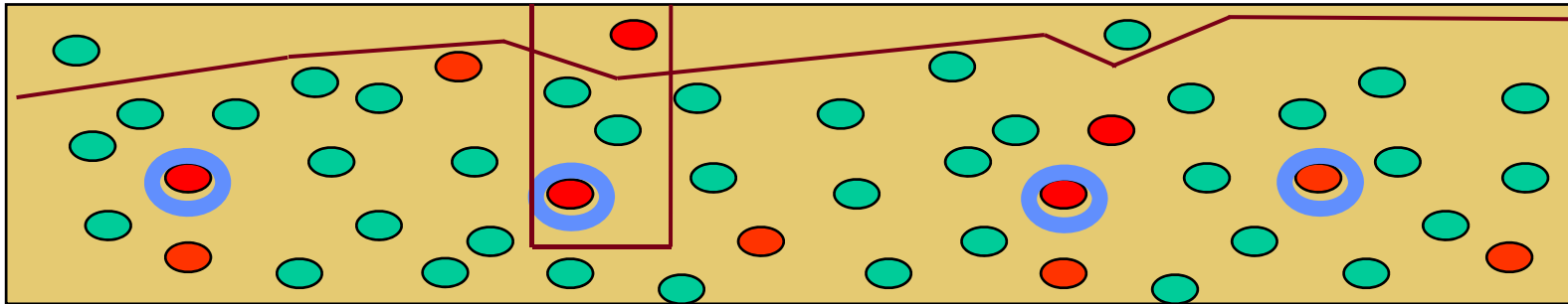
# Typical Review



- Find some defects, one Major
- Fix them
- Consider the document now corrected and OK ...

# Taking a sample

Ref. Dorothy Graham



- Inspection can find deep-seated defects
- All of that type can be corrected
- Needs optimum checking rate
- In the above case we are clearly taking a sample
- In the “shallow” case we were also taking a sample, however, *we didn't feel it!*

## Ultimate Goal of a What We Do

**Quality on Time**

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

- **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

# Inspection

## used in various ways

[www.malotaux.nl/conferences](http://www.malotaux.nl/conferences)

[www.malotaux.nl/booklets](http://www.malotaux.nl/booklets)

[www.malotaux.nl/inspections](http://www.malotaux.nl/inspections)

**Niels Malotaux**

**N R Malotaux**  
Consultancy

+31-655 753 604

niels@malotaux.nl

www.malotaux.nl

## [www.malotaux.nl/booklets](http://www.malotaux.nl/booklets)

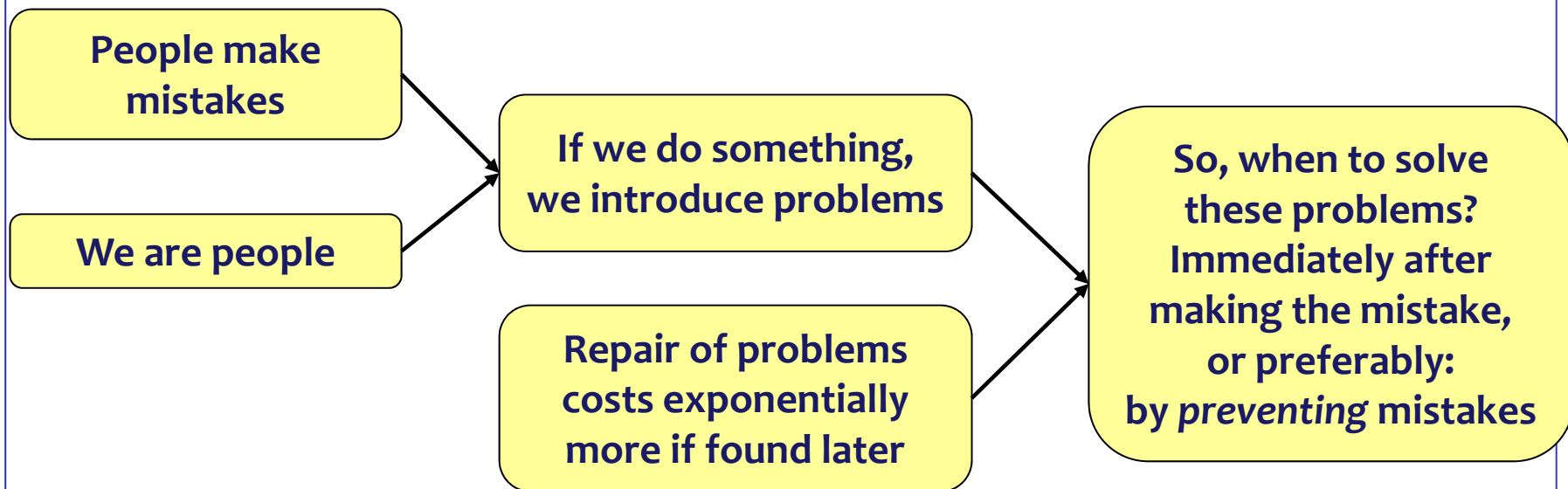
## More

- 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 Behaviour in Projects (APCOSE 2008)**  
Human Behavioural aspects of Projects
- 7 **How to Achieve the Most Important Requirement (2008)**  
Planning of longer periods of time, what to do if 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/inspections](http://www.malotaux.nl/inspections)

### Inspection pages

# Inevitable consequence



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

## 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**

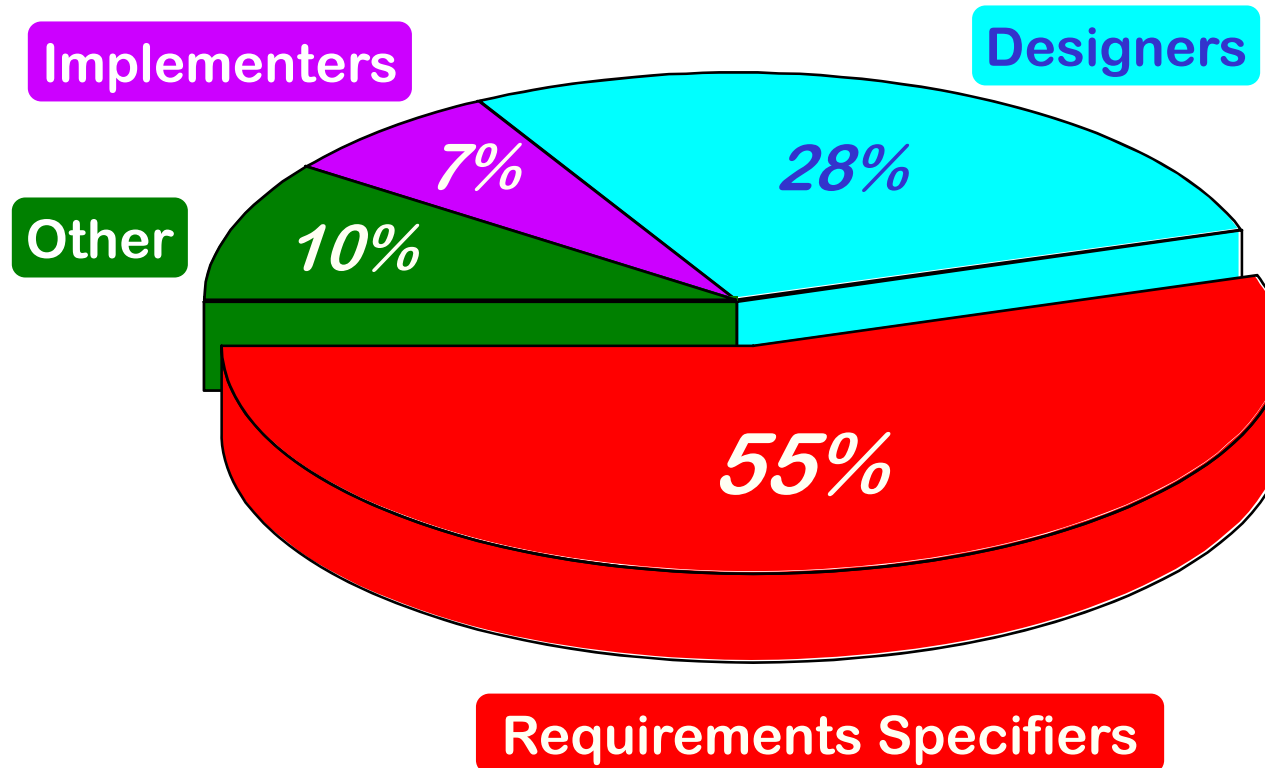
**Conventional testing:**

- 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?)

## Typical Defect Injectors (cost breakdown)



After Bender Associates, 1996

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



# The essential ingredient: the PDCA Cycle

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

