June 2018

Niels Malotaux

Improving the Effectiveness of Reviews and Inspections

N R Malotaux - Consultancy
tel +49-5632 922 5132
mob +31-655 753 604
niels@malotaux.nl
www.malotaux.nl

Niels Malotaux

Niels Malotaux is an independent Project Coach and expert in optimizing project performance. He has some 40 years of experience in designing electronic and software systems, at Delft University, in the Dutch Army, at Philips Electronics and 20 years leading a systems design company. Since 1998 he has devoted his expertise to helping projects and organizations to deliver Quality On Time: delivering what the customer needs, when they need it, to enable customer success. To this effect, Niels developed an approach for effectively teaching Evolutionary Project Management (Evo) Methods, Requirements Engineering, Review and Inspection techniques, as well as Reliable Embedded Systems Design and how to achieve Zero Defects for the customer. Since 2001, he has taught and coached well over 400 projects in 40+ organizations in the Netherlands, Belgium, China, Germany, Ireland, India, Israel, Japan, Poland, Romania, Serbia, South Africa, the UK and the US, which has led to a wealth of experience in which approaches work better and which work less well in practice.

Niels puts development teams on the Quality On Time track and coaches them to stay there and deliver their quality 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
- Zero Defects delivery

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

Zum Anspel 5
59964 Düdinghausen
Germany
tel +49-5632-922 5132
mob +31-655 753 604
niels@malotaux.nl
www.malotaux.nl

Result Management

Niels Malotaux

Optimizing the Effectiveness of Reviews and Inspections

Improving the Effectiveness of Reviews and Inspections

www.malotaux.nl/conferences
www.malotaux.nl/booklets
www.malotaux.nl/inspections

SE-TRAINING
N R Malotaux Consultancy

Niels Malotaux

+31-655 753 604 niels@malotaux.nl www.malotaux.nl

Niels Malotaux

Project and Organizational Coach

Helping projects and organizations to quickly become

- More effective - doing the right things better
- More efficient - doing the right things better in less time
- Predictable - delivering as needed

Getting projects back on track

Helping with Architecture/Design/Review of electronics/firmware/software

Project types
electronic products, firmware, software,
space, railway, telecom,
industrial control, parking system

Result Management

Ultimate Goal of a What We Do
(for our salary)

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

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 3

Evolutionary Project Management elements (Evo) – Tom Gilb

- Plan-Do-Check-Act
  - The powerful ingredient for success
- Business Case
  - Why we are going to improve what
- Requirements Engineering
  - What we are going to improve and what not
  - How much we will improve: quantification
- Architecture and Design
  - Selecting the optimum compromise for the conflicting requirements
- Early Review & Inspection
  - Measuring quality while doing, learning to prevent doing the wrong things

Why
- What
- How much
- Are we done

How

Zero Defects Attitude

Check as early as possible

Right Result

Quality On Time

Right Time

Evo Project Planning - added

- Weekly TaskCycle
  - Short term planning
  - Optimizing estimation
  - Promising what we can achieve
  - Living up to our promises
- Bi-weekly DeliveryCycle
  - Optimizing the requirements and checking the assumptions
  - Soliciting feedback by delivering Real Results to *eagerly waiting* Stakeholders
- TimeLine
  - Getting and keeping control of Time: Predicting the future
  - Feeding program/portfolio/resource management

Efficiency of what we do

Effectiveness of what we do

What will happen and what will we do about it?

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 4

Is there a Quality On Time problem?

- What made you decide to attend ?
- Do your projects produce the Right Results ?
- Do your projects deliver the Right Results at the Right Time ?
- What can we do about it ?

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 5

Who is who ?

- Systems Engineer ?
- Architect ?
- QA ?
- Project Manager ?
- Product Owner ?
- Scrum Master ?
- Team Member ?
- Customer ?
- Manager ?
- Consultant ?
- Coach ?

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 6

Schedule, we'll try to keep ☺

9:00~10:45	1:45
break	0:15
11:00~12:45	1:45
lunch	1:00
13:45~15:15	1:30
break	0:15
15:30~17:00	1:30

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 7

Homework - did you bring with you:

- Three copies of one or two pages of a document
  - Preferably some requirements document
  - Preferably being used in your current project
  - Perhaps even already reviewed your usual way
- Not too confidential so that others can help you review
- Also bring these pages electronically
  then we can show it to all (if allowed), to discuss what you have been reviewing
- It's much more educational if you exercise on your own real actual document
- Warning: After the Inspection you may decide to discard the document as unacceptable !
  (better to know than being ignorant)

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 8

Who is regularly doing Reviews and/or Inspections?

•

Baseline: Let's check your document

- Take one page
- Would you invite others to review the same ?
- How much time shall we spend ?

- What did you find ?

Let's use some Rules

ref Tom Gilb

- Unambiguous
  Every word and phrase should be unambiguous to all potential intended readers
- Clear to test
  Every word and phrase should be clear enough to allow objective test
- Quantified quality
  All qualities (things we want to improve) shall be expressed quantitatively (element of unambiguousness)
- No design in requirements
  Objectives shall not be expressed in terms of solutions

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 11

Tom Gilb quote

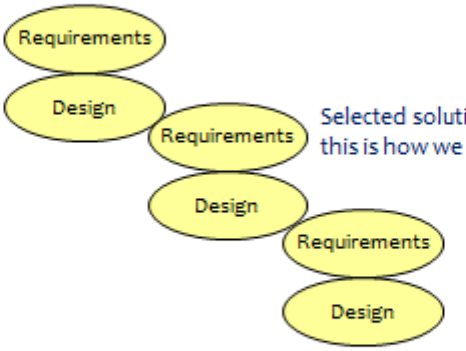
- The fact that we can set numeric objectives, and track them, is powerful; *but in fact it is not the main point*
- The main purpose of quantification is to force us to *think deeply*, and *debate exactly*, what we mean
- So that others, later, *cannot fail* to understand us

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 12

No Design in the requirements, but ...

Needs:
what do we need

Options:
how can we do it



Selected solution:
this is how we are going to do it

Requirement: What the acquirer cares about: 'how good it should be'
Design: Set of decisions made by development: 'how to be good'
Design provides the Requirements for the next level

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 13

Let's check again

Let's use some Rules

ref Tom Gilb

- Unambiguous
  Every word and phrase should be unambiguous to all potential intended readers
- Clear to test
  Every word and phrase should be clear enough to allow objective test
- Quantified quality
  All qualities (good things we want to improve) shall be expressed quantitatively
- No design in requirements
  Objectives shall not be expressed in terms of solutions

- Take the same page
- What did you find ?

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 14

Verification - Validation

- Verification - you built the thing right
  Checking the implementation against
  the requirements and the selected solution
- Validation - you built the right thing
  Checking the implementation against
  what the requirements *should have been*
- *Better make sure the requirements are right*

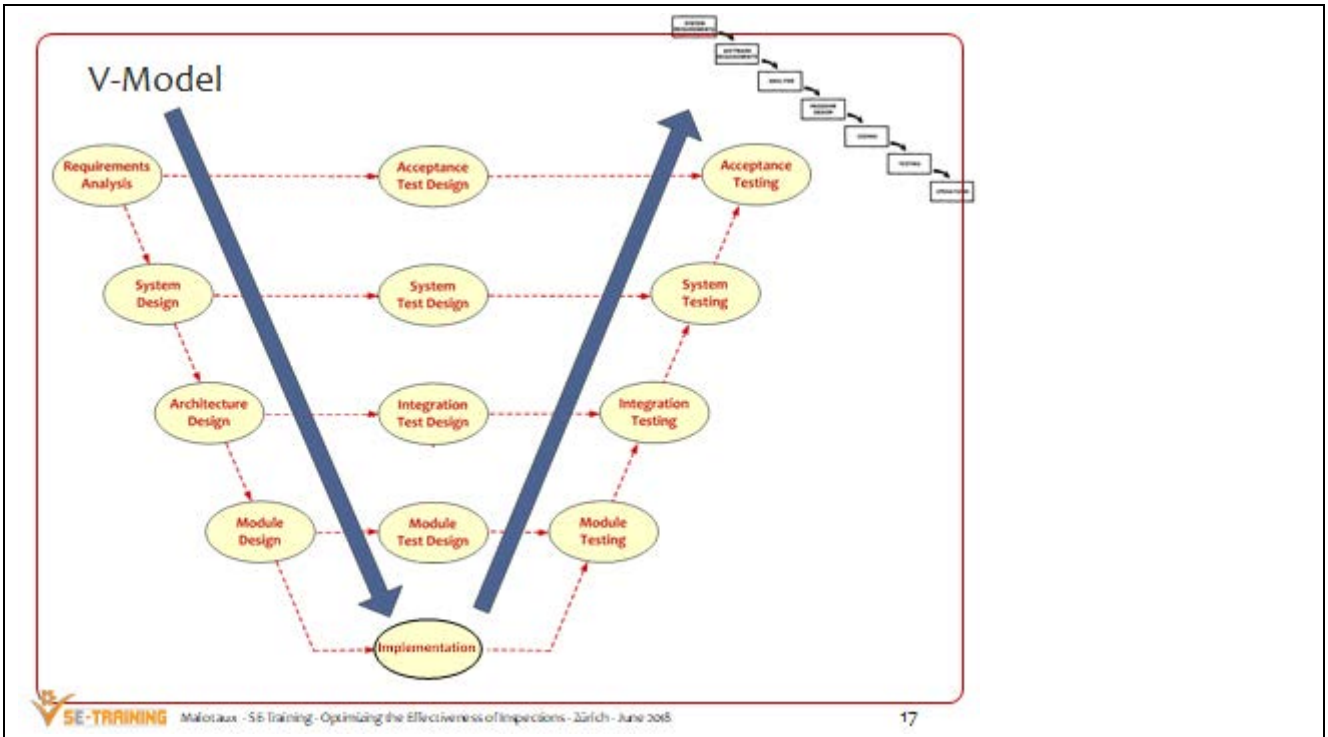
How good are our requirements ?

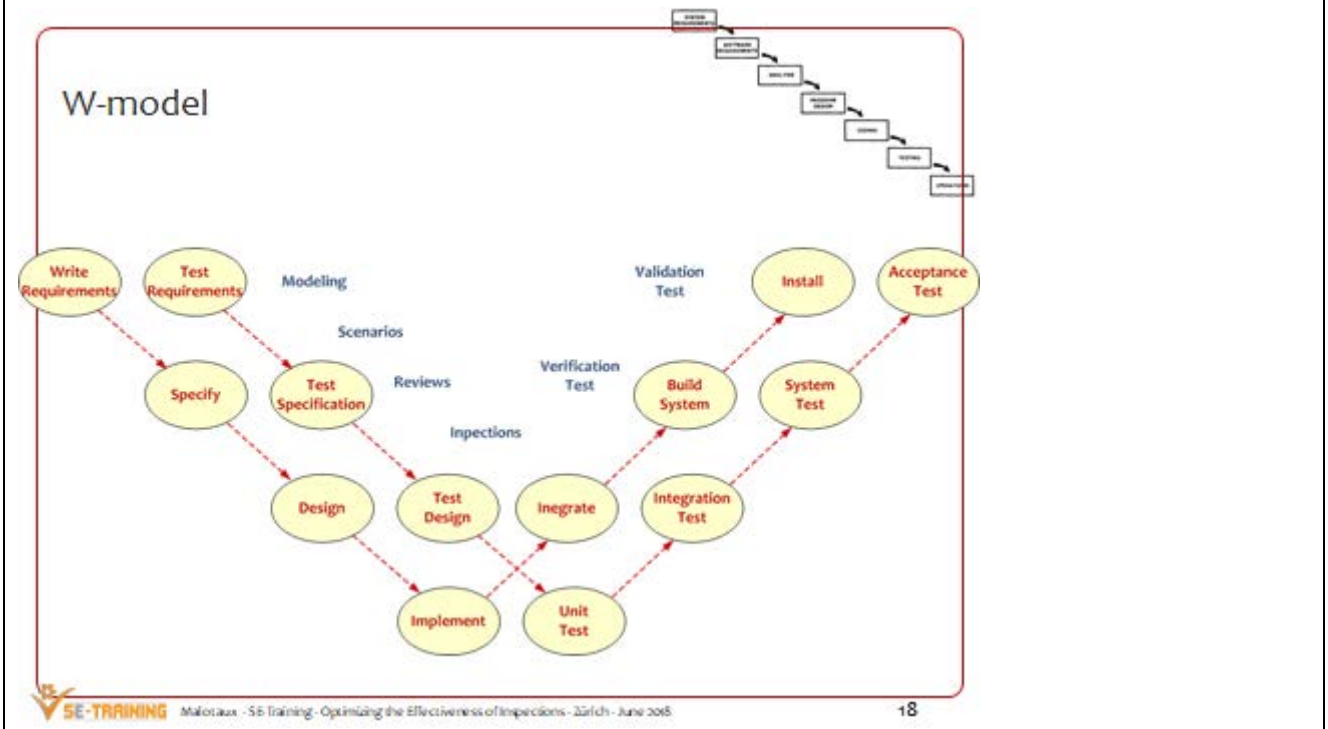
- We meet the agreed requirements

but ...

- Have the requirements changed to
  what we and the customer really need
- We create requirements with care and
  we meet them with care

Philip Crosby


V-Model
Requirements Analysis
Acceptance Test Design
Acceptance Testing
System Design
System Test Design
System Testing
Architecture Design
Integration Test Design
Integration Testing
Module Design
Module Test Design
Module Testing
Implementation
SE-TRAINING
17


W-model
Write Requirements
Test Requirements
Modeling
Scenarios
Validation Test
Install
Acceptance Test
Specify
Test Specification
Reviews
Verification Test
Build System
System Test
Inpections
Design
Test Design
Inegrate
Integration Test
Implement
Unit Test
SE-TRAINING
18

What do you review ?

- Wish specification — Thank you, nice input, to be taken seriously
- Contract — This is what I'll take you to court with
- Business Case — Why are we doing it
- Requirements — What the project agrees to satisfy
- Design — Selecting the 'optimum' compromise
- DesignLog — How we arrived at this decision
- Specification — This is how we are going to implement it
- Implementation — Models, schematics, plans, procedures, hardware, software, documentation, training

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 19

Are all of your documents always reviewed ?

- If your product is tested (V&V), how do you know it's correct ? (testing hardly proves anything)
- Reviews are for
  - early detection
  - quick learning
  - prevention
- Without proper education reviews are not very effective
- Inspections are a special kind of review

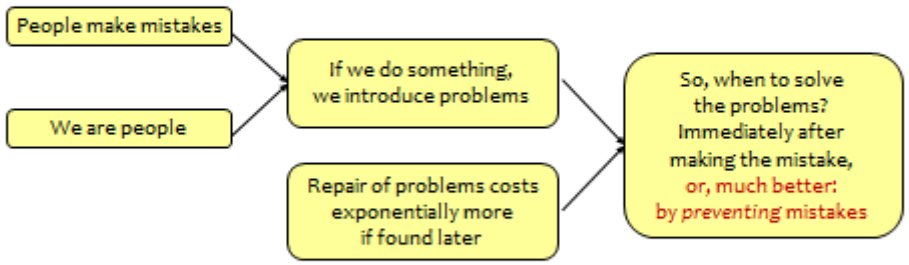
SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 20

Do you ever make a mistake?

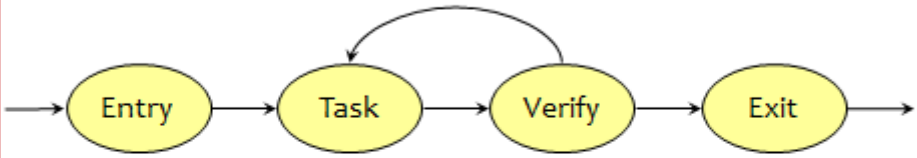
- People make mistakes
- We are people

If we think we are done
there are still defects

Inevitable consequence



Checking immediately after (ETVX)

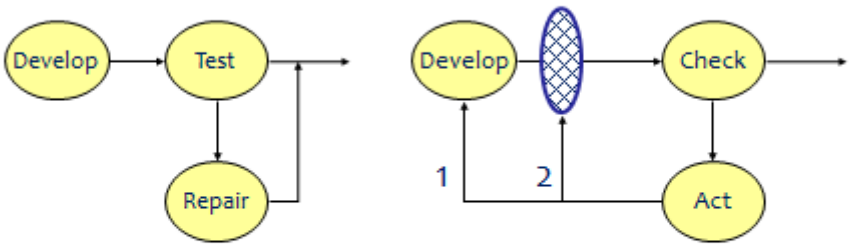


Are you already doing this?

Even better: checking *during* workproduct generation

- We know that errors are being made
- Why wait until all issues are injected ?

Testing (V&V) should find it's OK



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 ?

Can you find all issues in a document ?

- Let's do the F-test

How many times F, f can you find on this screen ?

Federal Funds are the result of years of scientific study combined with the experience of years

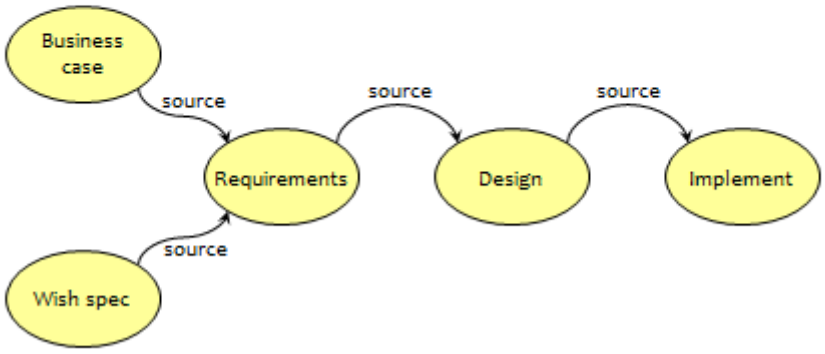
(Deming)

Rules

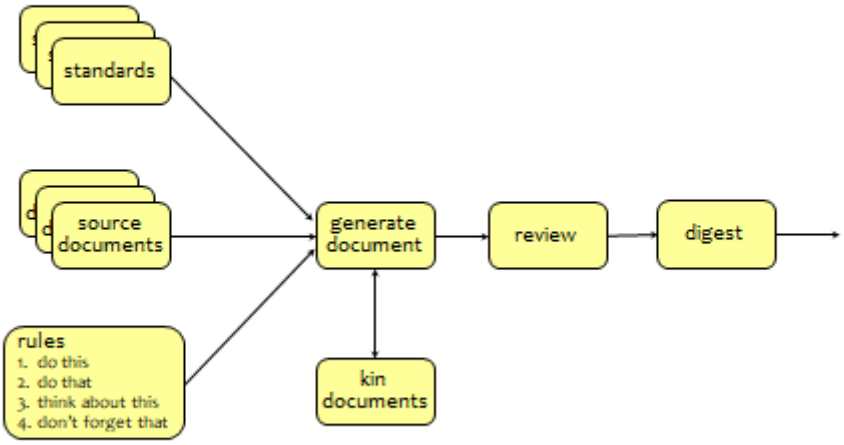
- Rules are the law for documents
- Defect = Violation of a Rule
  - Not "I think this is wrong"
  - If it's not a violation of a rule, then it's not a defect (perhaps an Improvement Suggestion ?)
- Rules:
  - All quality requirements must be expressed quantitatively
  - The document should be consistent with itself and with source documents

Simple Rule for Reviews

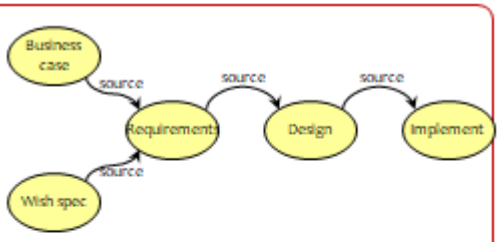
"We don't review unless there is a source document"



Document generation



Correlation



- Any workproduct will be reviewed against
  - Itself
  - Kin documents
  - Source documents

  If we don't have the source, how can we judge the workproduct?
- We always update the source document first before changing the workproduct(s)
  - First change the Requirement, then the Design, then the Implementation, otherwise ...

What is a Defect ?

What is a defect ?

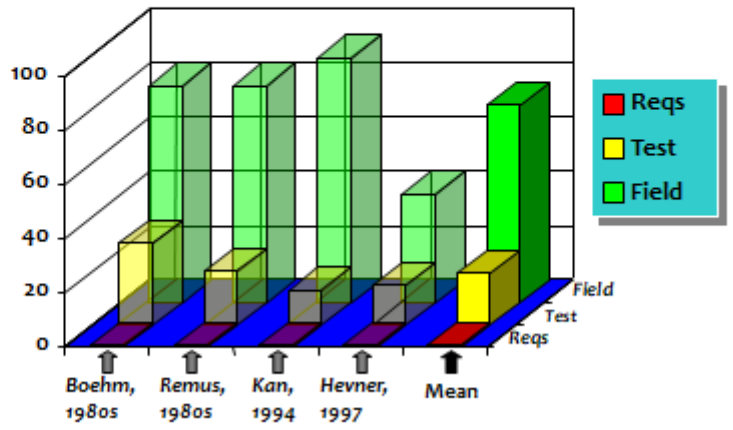
- A defect is the cause of a problem experienced by any of the stakeholders while relying on our results
- Making the customer more successful implies *no defects*

- All we have to do is delivering results without defects
- Do we?

- Is being late a defect ?
- Is being on time in your requirements ?

Cost of Requirements Defects

The longer a defect stays in the system,
the more it costs to repair



Defects found are symptoms of deeper lying problems

Repairing defects creates risks:

- Repair is done under pressure
- We think the problem is solved
- We introduce scars
- We keep repeating the same problems

→ Do Root Cause Analysis and make sure it *never happens again*

Prevention: Root Cause Analysis

- Is Root Cause Analysis routinely performed – *every time*?
- What is the Root Cause of a problem?

- Cause:
  The error that caused the problem
- Root Cause:
  What caused *us* to make the error that caused the problem

- Without proper Root Cause Analysis,
  we're *doomed to repeat the same errors*

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 35

What to look for ?

Let's focus on requirements

- Are your requirements clear ?
- What's the point in designing and implementing based on unclear requirements ?
- Working on a great solution for the wrong problem ?
- First develop the problem, then the requirements, then the design, only then the implementation
- What's your experience ?

Can you find these by V&V ?

- Fuzzy requirements (what to test against ?)
- Functions that won't be used (superfluous requirements)
  - Why repair defects in the implementation of these requirements ?
  - The only defect is that it has been implemented
- Nice things (not checked for real need, not paid for)
  Shouldn't be there in the first place
- Missing quality levels (should have been in requirements)
  Checking the implementation of the requirements won't help
- Missing constraints (should have been in requirements)
  Product could be illegal (if that's the purpose, you'd better tell)
- Unnecessary constraints (not required)
  What would testing say about these ?

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 38

Ever seen such requirements ?

- The system should be extremely user-friendly
- The system must work exactly as the predecessor
- The system must be better than before

- Do you know other examples ?

- It shall be possible to easily extend the system's functionality on a modular basis, to implement specific (e.g. local) functionality

- It shall be reasonably easy to recover the system from failures, e.g. without taking down the power

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 39

'Weak words' in requirements

- e.g. (is it a requirement or not?)
- etc. (could be anything)
- and (probably two requirements)
- or (can I choose whichever?)
- includes (what more?)
- such as (is it a requirement or not?)
- specific conditions (but not specified)
- essentially the same (how much is essentially?)
- information *may be* shown (may also be not shown?)
- all possible data (that's a lot!)

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 40

Basic Types of Requirements

- Functional *binary*
  - Determine the scope of the project:
  - What are we working on
- Quality/performance *scalar*
  - To enhance the performance of the selected functions
- Constraints *binary / scalar*
  - What should we not do, be aware of, be limited by

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 41

Rules for Performance objectives

1. Unambiguously clear to the intended reader
2. SCALE of measure
3. Complex concepts should be broken down into a set of measurable *elementary* concepts
4. To define 'relative' terms like 'higher' there should be at least two points of reference on the defined SCALE
5. Specify when a quality level is to be available
6. Not mixing design ideas in objectives/requirements
7. Specifying the source (like contract, standard, marketing plan)
8. Fuzzy unclear concepts shall be marked with <angle brackets> for improvement

TG SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 42

How Many Defects in a Statement?

The objective is to get higher adaptability using advanced architecture

1. Unambiguously clear to the intended reader
2. SCALE of measure
3. Complex concepts should be broken down into a set of measurable *elementary* concepts
4. To define 'relative' terms like 'higher' there should be at least two points of reference on the defined SCALE
5. Specify when a quality level is to be available
6. Not mixing design ideas in objectives/requirements
7. Specifying the source (like contract, standard, marketing plan)
8. Fuzzy unclear concepts shall be marked with <angle brackets> for improvement

TG SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 43

Requirements with Planguage

ref Tom Gilb

SMART

Specific

Measurable

Definition:

RQ27: Speed of Luggage Handling at Airport

Scale: Time between <arrival of airplane> and first luggage on belt

Meter: <measure arrival of airplane>, <measure arrival of first luggage on belt>, calculate difference

Benchmarks (Playing Field):

Past: 2 min [minimum, 2016], 8 min [average, 2016], 83 min [max, 2014]

Current: < 4 min [competitor y, Jan 2018] ← <who said this?>, <Survey April 2018>

Attainable

Record: 57 sec [competitor x, Jan 2016]

Wish: < 2 min [2020Q3, new system available] ← CEO, 19 Jan 2018, <document ...>

Requirements:

Time

Traceable

Realizable

Tolerable: < 10 min [99%, Q4] ← SLA

Tolerable: < 15 min [100%, Q4, Heathrow T4] ← SLA

Goal: < 15 min [99%, Q2], < 10 min [99%, Q3], < 5 min [99%, Q4] ← marketing

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 44

How many issues can you find ?

Unambiguous, Clear to Test, Quantified, No Design

- The system should be extremely user-friendly
- The system must work exactly as the predecessor
- The system must be better than before
- It shall be possible to easily extend the system's functionality on a modular basis, to implement specific (e.g. local) functionality
- It shall be reasonably easy to recover the system from failures, e.g. without taking down the power

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 45

Some requirements

- REQ 4010 - The storage of the [system] shall store diagnostic information, excluding sensor information, for a period of at least 4 months
- REQ 3776 - Recorded data shall be stored and available for transfer for at least 2 months
- REQ 1503 - The [system] shall record all diagnostic data in a non-volatile memory
- REQ 5037 - Deactivation of a failure by the [system] shall be only allowed when the [system] detects that the failed function is working correctly again in the same state as the failure was activated
- REQ 4758 - The [system] shall provide the other diagnostic data (sensor values, performance and usage counters and other possible data) to the [system] service interface for transmission to the wayside within other time intervals

SE-TRAINING Malotaux - SE Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 46

Unambiguous, Clear to Test, Quantified, No Design

(ref TG - Platform Rationalisation in a bank)

- Rationalize into a smaller number of core processing platforms. This cuts technology spend on duplicate platforms, and creates the opportunity for operational saves. Expected 60%-80% reduction in processing cost to Fixed Income Business lines
- International Securities on one platform, Fixed Income and Equities (Institutional and PB)
- Global Processing consistency with single Operations In-Tray and associated workflow
- Consistent financial processing on one Accounting engine, feeding a single sub-ledger across products
- First step towards evolution of "Big Ideas" for Securities
- Improved development environment, leading to increased capacity to enhance functionality in future
- Removes duplicative spend on two back office platforms in support of mandatory message changes, etc

SE-TRAINING Malotaux - SE Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 47

Can we develop based on Management Poetry ?

- Nice input, to be taken seriously
- We write back the requirements, don't we ?
- This is what we plan to do, if you let us continue

- Are we better at requirements ?
  - Unambiguous, Clear to Test, Quantified, No Design

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 48

Is this a Requirement ?

or 'nice input', to be taken seriously ?

"Create a new 'Price Sentinel' component that can detect if the bank's published customer quotations go off-market, and then to immediately cancel all current quotations."

Ref http://rsbatechnology.co.uk

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 49

Using 5 Whys

Why do you need a "Price Sentinel" ?

1. To prevent publishing off-market tradable prices
2. To prevent trading loss
   (having to buy at a higher price than the bank offered to the customer)
3. To demonstrate to senior management that e-trading business can safely (no unexpected loss) manage customer trading
4. To ensure that senior management will agree to expand e-trading business in the future, based on current business performance to other customer segments and business areas
5. To meet business medium / long-term financial targets

Ref http://rsbatechnology.co.uk

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 50

First try

New 'Price Sentinel' component:

- detect if the bank's customer quotations go off-market
- then immediately cancel all current quotations

- Off-market
  - Our margin less than 0.1%
- Immediately
  - Scale: seconds after <detection> (<happening>?)
  - Current: 600 sec (=10 min)
  - Goal: 1 sec

Prioritized solutions by Impact Estimation

	Kill button	Price Sentinel
Cancel 600 → 1 sec	10.5 sec (note) 98%	1 sec 100%
Cost	1 day	30 day (6 sprint)
Note: 10 sec human recognition time, 0.5 sec cancel time		

The Jet Case

The Jet Case

Introduce the following three rules for Inspecting a requirements document

Three Rules for Requirements:

1. Unambiguous to intended readership

   Two designers arrive at the same result

2. Clear enough to test

   Two testers get same result

3. No design mixed in requirements (mark as "D")
   - Requirements: What the acquirer cares about: 'how good to be'
   - Design: set of decisions made by the developer: 'how to be good'

Defect

Explain the definition of a Defect

- A Defect is a violation of a Rule
- Note: If there are 10 ambiguous terms in a single requirement then there are 10 defects!

Severity

Explain:
- the definition of Major Defect
- focus on finding Major Defects

- Major: a defect severity where there is potential of high loss later downstream (test, field)
- "10 lost engineering hours"

Exit?

Agree with the management team on a numeric exit condition:
Is 1000 Majors per page OK ? 100, 10, 1 ?

- Exit Conditions:
  (document can go to next stage with little risk)
  - Maximum 1 Major Defect / (Logical) Page ?
- Logical Page = 300 Non Commentary Words

The Job

- You have up to 15 minutes for checking
  one Requirements Logical page from the 82 pages document
- Count all Rule Violations → Defects
- Classify Major, minor, or Design

Report for Page 51

	Total	Major	Design
1.	24	15	5
2.	44	15	9
3.	55	20	4
4.	22	4	2

Inspection results on requirements document, 4 managers

Defect Density

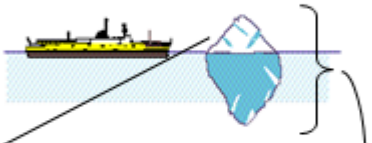
- Total for group 20 x 2 = 40 Majors assume are unique
- If 33% effective, total in page = 3 x 40 = 120
- Of which 2/3 or 80 were not yet found
- If we attempt to fix the 40 found, and correctly fix 5/6 then 7 are failed fixes, so:
- Total after Inspection and editing: 80 + 7 = 87 Majors per page remaining

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 59

Report for Page 52

	Total	Major	Design
1.	41	24	1
2.	33	15	5
3.	44	30	10
4.	24	3	5

Inspection results on requirements document, 4 other managers



Defect Density

- Total for group 30 x 2 = 60 Majors assume are unique
- If 33% effective, total in page = 3 x 60 = 180
- Of which 2/3 or 120 were not yet found
- If we attempt to fix the 60 found, and correctly fix 5/6 then 10 are failed fixes, so:
- Total after Inspection and editing: 10 + 120 = 130 Majors per page remaining

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 60

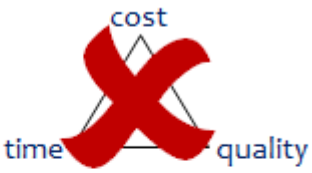
Extrapolation to Whole Document

- Page 51: 120 Majors/p
- Page 52: 180 Majors/p
- Average 150 Majors/page x 82 pages = 12300 Majors in the document.

- If a Major has 1/3 chance of causing loss (12300 / 3 = 4100)
- And each loss is avg 10 hours
- Then total project rework cost is about 41000 hours loss
- (The project was over a year late and expected one more year)
  - 1 year = 2000 hour x 10 people = 20000 hours

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 61

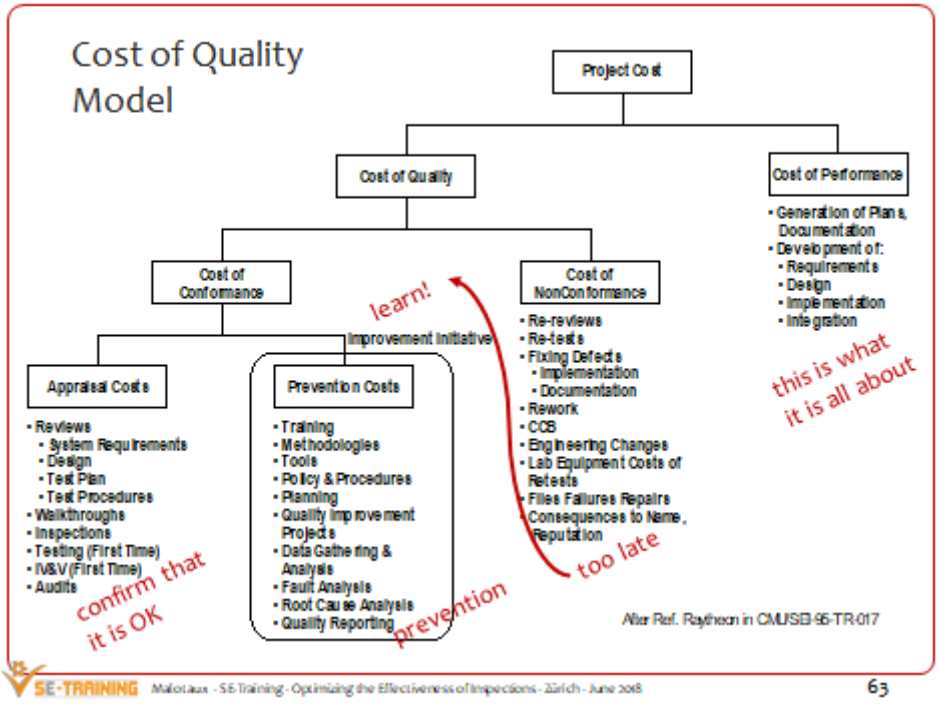
Does quality cost more ?

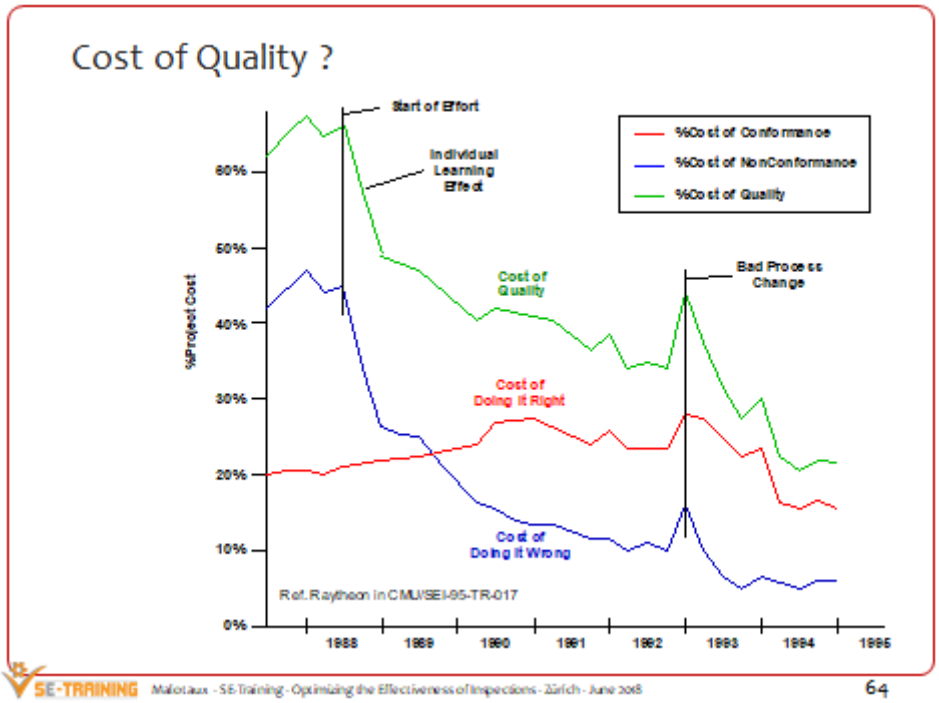


- The ~~cost is not~~ benefit is in the quality
- The cost is in the non-quality

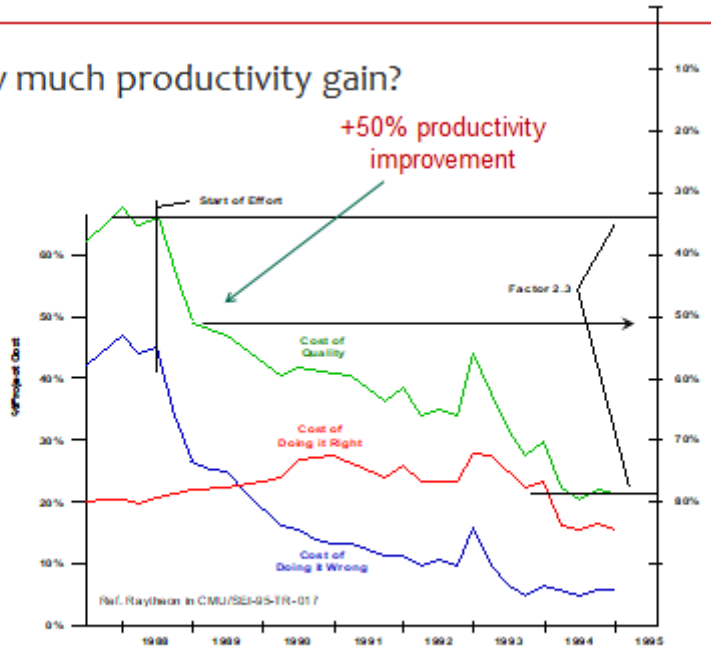
The *right* quality costs *less*

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 62


Cost of Quality Model
Project Cost
Cost of Quality
Cost of Performance
• Generation of Plans, Documentation
• Development of:
• Requirements
• Design
• Implementation
• Integration
Cost of Conformance
Cost of NonConformance
learn!
Improvement Initiative
Appraisal Costs
• Reviews
• System Requirements
• Design
• Test Plan
• Test Procedures
• Walkthroughs
• Inspections
• Testing (First Time)
• IV&V (First Time)
• Audits
Prevention Costs
• Training
• Methodologies
• Tools
• Policy & Procedures
• Planning
• Quality Improvement Projects
• Data Gathering & Analysis
• Fault Analysis
• Root Cause Analysis
• Quality Reporting
• Re-reviews
• Re-tests
• Fixing Defects
• Implementation
• Documentation
• Rework
• CCB
• Engineering Changes
• Lab Equipment Costs of Retests
• Files Failures Repairs Consequences to Name, Reputation
this is what it is all about
confirm that it is OK
prevention
too late
After Ref. Raytheon in CMU/SEI-95-TR-017
SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018
63


Cost of Quality ?
Start of Effort
Individual Learning Effect
%Cost of Conformance
%Cost of NonConformance
%Cost of Quality
Cost of Quality
Bad Process Change
Cost of Doing it Right
Cost of Doing it Wrong
%Project Cost
80%
60%
40%
30%
20%
10%
0%
1988
1989
1990
1991
1992
1993
1994
1995
Ref. Raytheon in CMU/SEI-95-TR-017
SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018
64

How much productivity gain?

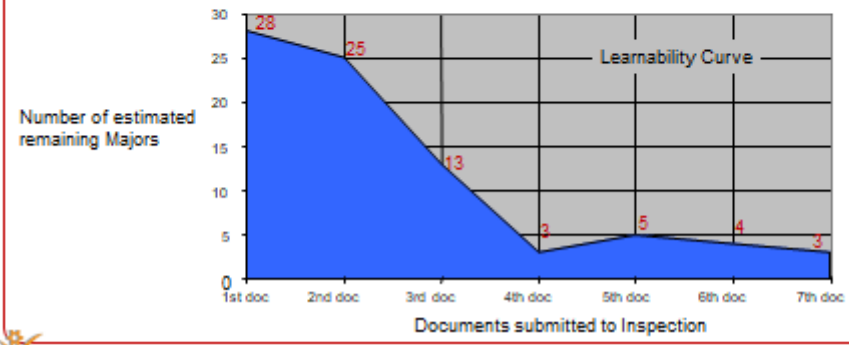


Inspection goals and effects

- Identify and correct major defects
  becoming even less important
- Most important:
  Identify and remove the *source* of defects
- Consequence:
  Education and interaction:
  How should we generate documents in the first place?
- Interesting side-effect:
  People get to know each other's documents efficiently

Individual learning Curve

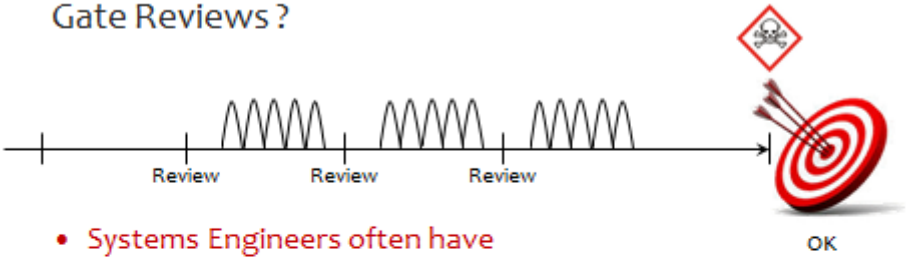
- The speed of individual learning to follow the Rules,
- Measured by reduced Major Defects found in Inspections
  - Faster, earlier and more dramatic than "process improvement"
  - Never mentioned in literature as a measurable



Many types of Review to choose from

- Which ones do you use?
  - Informal Review
  - Pair Programming (Pair Designing)
  - Technical Review
  - Walkthrough
  - Formal Inspection (Fagan type)
  - Cleanroom Inspection
  - Formal Inspection (Gilb/Graham type)
  - Agile/Extreme/Lean/Early Inspection (SQC)
  - Gate Review
  - Ritual Meeting

Gate Reviews ?



- Systems Engineers often have
  - SFR - System Functional Review
  - PDR - Preliminary Design Review
  - CDR - Critical Design Review
  - TRR - Test Readiness Review
  - SVR - System Verification Review
- What is the purpose of these reviews ?
- Really?
- Perhaps there's more work to do ?

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 69

Formal Reviews (vs Ad-Hoc)

- Defined, repeatable process
- Measures effectiveness
- Continuous improvement
- Rules/checklists
- Feeds prevention process

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 70

Inspection

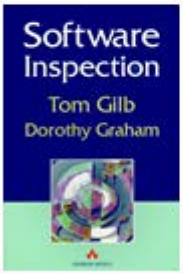
- Most rigorous form of review
- Pioneered by Fagan (IBM) (paper 1976)
  - Locating all the defects in a work product, focus on code
- Inspection economics: Gilb/Graham (Software Inspection, 1993)
  - Quantifying the defect density of a work product and preventing poor quality work from moving downstream
- Early Inspection
  - Not waiting until the whole waterfall of the document is completed
- Use:
  - Walkthroughs for training
  - Technical Reviews for consensus
  - Inspections to improve the quality of the document and its process
  - Gate Reviews to decide what to do with it
- Would you like to base further work or decisions on a document of unknown quality ?

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 71

Gilb / Graham Inspections

Not only software !





SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 72

Do you recognize this ?

- The document to be reviewed is given out in advance
- Typically dozens of pages to review
- Instructions are "please review this"
- Some people have time to look at it
- Review meeting often lasts for hours
- Typical comment: "I don't like this"
- Much discussion, some about technical approaches, some about trivia
- Don't really know if it was worthwhile, but we keep doing it
- Next document reviewed will be no better

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 73

Have you been looking at the document ?

Buying a second hand car

We checked your car at the bridge
(nachschauen auf der Brücke)

What I think

What they mean

Techniques

- Can you look at this ?
- Over the shoulder
- Pair Programming
- E-mail
- Tool
- On Screen
- Projector
- On Paper
- Formal process

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 76

Inspection is different

- The document to be reviewed is given out in advance
  not just product - rules to define defects, other docs to check against
- Typically dozens of pages to review
  chunk or sample
- Instructions are "please review this"
  training, roles
- Some people have time to look through it
  entry criteria to meeting, maybe not worth holding
- Review meeting often lasts for hours
  2 hr max
- Typical comment: "I don't like this"
  Best Practice rules - Rules are objective, not subjective
- Much discussion, some about technical approaches, some about trivia
  no discussion, highly focused, anti-trivia
- Don't really know if it was worthwhile, but we keep doing it
  exit criteria - continually measure costs and benefits
- Next document reviewed will be no better
  most important focus is improvement in processes and skills

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 77

16 page
Inspection
Manual

Inspection Manual
Procedures, rules, checklists and other texts
for use in Inspections

Version: 0.43 (Changed Plan into Goal)
Date: Oct 13, 2007
Owner: Niels Malotaux
Status: not inspected
Intended readership: anybody interested in or busy with inspections

Note: Most of these texts are originally taken from the book:
"Software Inspection" by Tom Gilb and Dorothy Graham
Addison Wesley, 1993, ISBN 0-201-63181-4, and from
web-sites, such as www.result-planning.com (Tom Gilbs web-site)
This is a starting point from which the procedures, rules, etc.
may be adapted to the local culture.

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 78

Generic Specification Rules

(see Inspection Manual)

GE0	(def)	Generic engineering specification rules apply to all engineering documents as required best practices
GE1	(relevant)	All statements should be relevant to the subject
GE2	(complete)	There should not be any significant omissions
GE3	(consistent)	Statements should be consistent with other statements in the same or related documents
GE4	(unambiguous)	All specifications should be unambiguous to the intended readership
GE5	(note)	Comments, notes, suggestions, not official part of document shall be clearly marked (“”, *ital*, /**/)
GE6	(brief)	All specifications shall be as brief as possible, to support their purpose, for the intended readership
GE7	(clarity)	All specifications shall result in clarity to the intended readership regarding it's purpose or intent (the burden is on author, not the reader) *Note: It is not enough that statements are unambiguous. They must contain clarity of purpose: why is it there?*
GE8	(elementary)	Statements shall be broken into their most elementary form *Note: This is so that they each can be cross-referenced externally (Traceability)*
GE9	(unique)	Specifications shall have a single instance in the entire project documentation
GE10	(source)	Statements shall have source info (spec ←source)
GE11	(risk)	The author should clearly indicate any information which is uncertain or poses any risk to the project, using indications like: {<vaguely defined>, ?, ??, 70% ±20, suitable comments or notes}
GE12	(verifiable)	All statements should be verifiable
GE13	(true)	(The statement is simply not true)

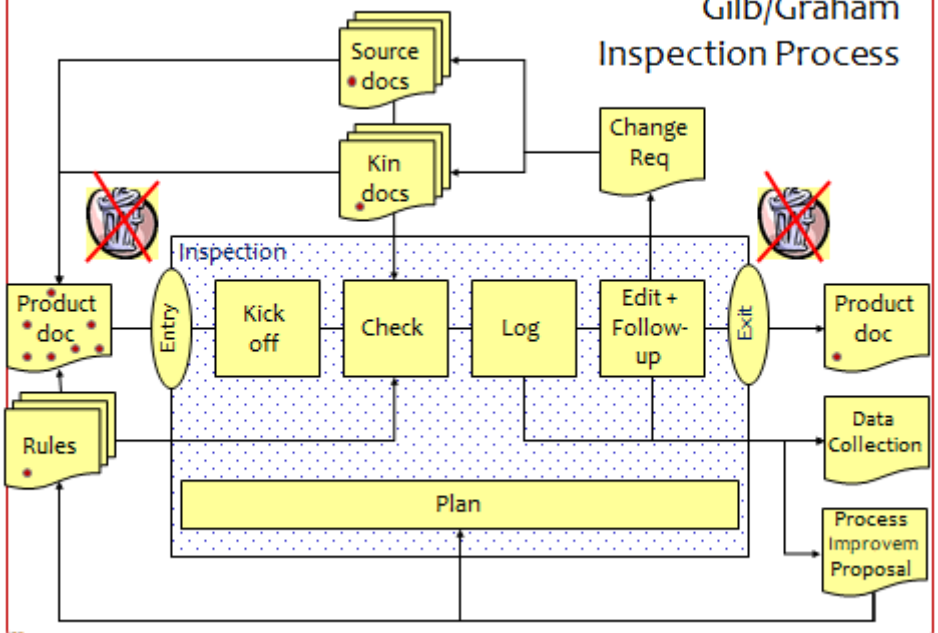
SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 79

Check Lists

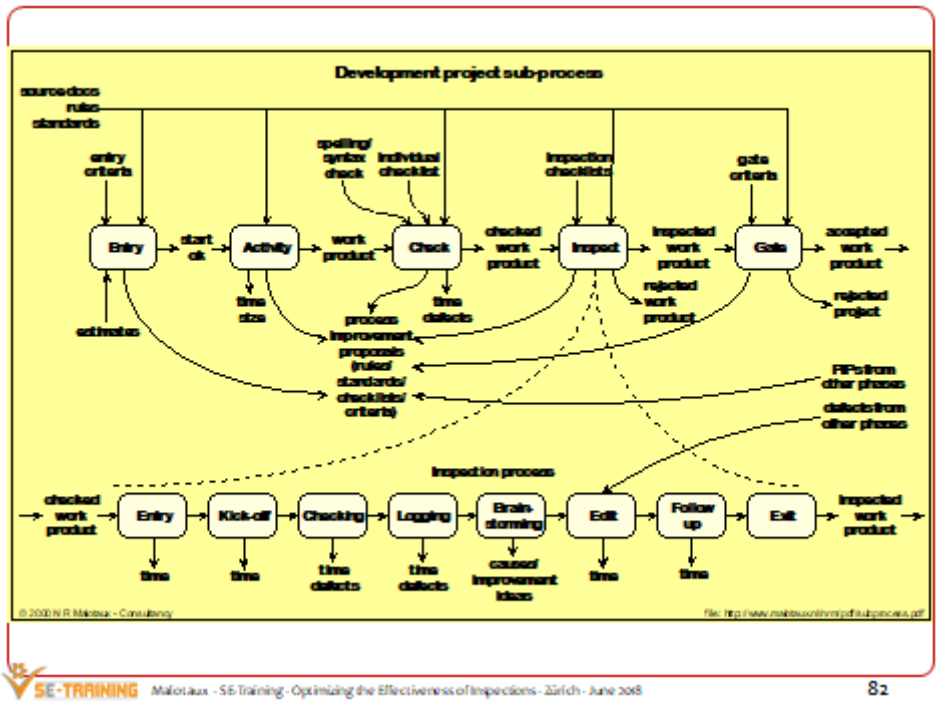
- Checklists contain interpretations of Rules to help reviewers to find more issues
- Rules are "The Law"
- Checklists provide "Jurisprudence"

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 80

Gilb/Graham Inspection Process



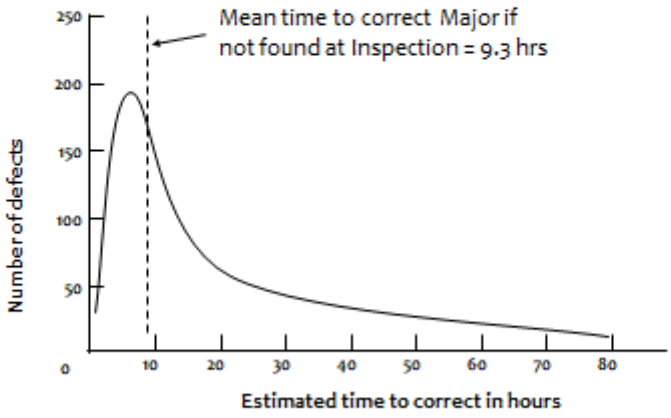
SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 81



SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 82

Cost of Repair

ref SI, fig 14.6, p315

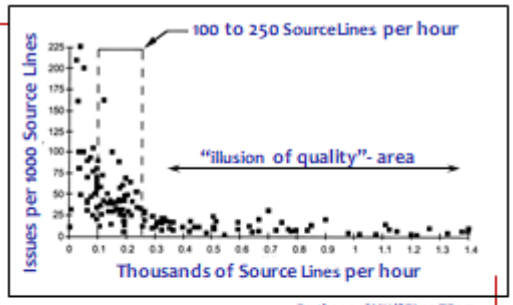


SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 83

How much time to spend per page ?

- What is the size of a typical document ?
- How much time do you spend per document ?

Optimum Checking Rate



Raytheon, CMU/SEI-95-TR017

- 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'

- 1 page of 300 words per hour ("logical page")
- 100~250 SLoC per hour
- More than we can afford. So … ? We must sample !

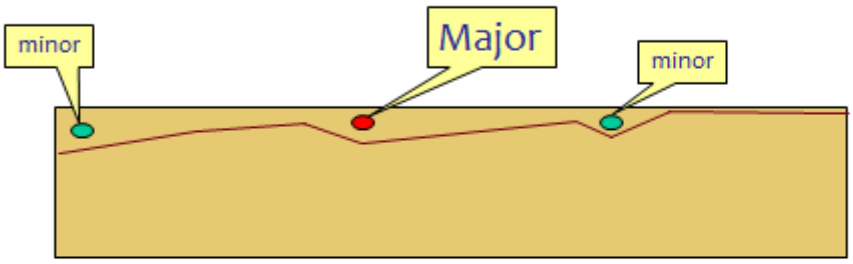
Optimum checking rate

Ref. Dorothy Graham

Here's a document: review this (or Inspect it)

Review "Thoroughness"?

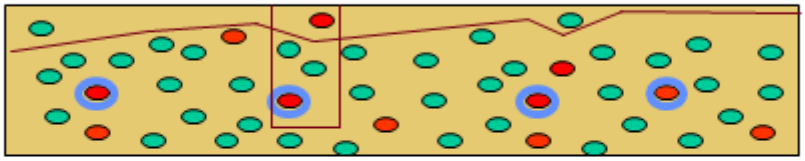
Ref. Dorothy Graham



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

Inspection Thoroughness

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 realize it!

Typical for Gilb/Graham Inspection

Gilb/Graham inspection differs from other types of inspection in some or all of these ways:

Purpose
Quantify quality, not search for all defects

Controlled reading rate
The material being inspected is read very thoroughly (1 hr per page) in order to identify as many defects as possible

Sampling
Samples are inspected to optimize time and effort investment

Entry/Exit Criteria
Quantified entry and exit criteria are used to guide the inspection effort

Rules
Written rule sets are used during the inspection to locate and classify defects

Gilb/Graham Inspection

Purpose	To quantify the *quality level* of a document
Why	Because low quality documents cannot be used as a basis for further development: the risk is too high
When	Upon author request, before document baselines and any go/no-go business decisions based on the document
Who	Small team (2-5), experienced in the Inspection process and able to devote the required time to it
What	Detailed review of samples of the document against document-type-specific rules, checklists, etc.
Cost	Typically, 4-6 hours per reviewer, plus a few additional hours for the Author and the Inspection leader

What do you need

- Trained Inspection leader
- Inspection Manual
  - Rules, Procedures
- Documents + owners
- Checkers
- Inspection Master Plan template
  - Who, What, Where
- Presentation for the Kick-off meeting
  - Why, How, What
- Inspection metrics template
  - Data collection
  - Issue collection
  - (Brainstorm - fruits collection)

Inspection Master Plan

Owner: NielsMalotaux – Version 1.01 – 23 Nov 2001

Inspection no. SW1 **Date requested:** Nov 19, 2001

who	name	init	tel	e-mail	role	scan	time	min/ page	check	time	min/ page	rule set
Leader	Niels Malotaux	nms	-	niels@malotaux.nl	Leader							
Author					Author		1 hr			1 hr		GE
Checker					-		1 hr			1 hr		GE
Checker					-		1 hr			1 hr		GE
Checker					-		1 hr			1 hr		GE
Checker					-		1 hr			1 hr		GE
Checker					-		1 hr			1 hr		GE

doc	owner	init	tel	e-mail	docname	date	ver	location	Insp status	maj/ page
Product									No Inspection	
Reference	Niels Malotaux	nms		niels@malotaux.nl	InspectorManual		0.4		Not Inspected	
Source									Not Inspected	
Source									Not Inspected	
Source									Not Inspected	
Source									Not Inspected	

meeting	date	location	start	end
KickOff				
Logging				

Individual checker data collection	Checker:	
	scan	check
Time spent (X.X hrs)		
Pages studied		
Majors		
Super majors (project threat)		
Minors		
Process Improvements		
Questions		

Instructions

Inspection goals: Getting the product edited
Learning Inspections

Strategy to meet goal: Do Inspection, find as many issues as possible
Note: The tabinator will initially be replaced by:
- 30 min. discussion about what you think of this Inspection process
- 30 min. Just In Time Training on the subject of the document

Optimum checking rate: 60 min per page
At first Inspections we will use about 30 min per logical page

Exit condition: < 2 major defects remaining per page

Assignment for this Inspection:

Please check the sheets against all source document and rule set GE. See Inspection Manual. In this manual you can also find the procedure for checking (Procedure for Checker during Checking : CC). Read this procedure to know what to do during checking.

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 92

Use these columns during individual checking (print these columns up to row 40 or 50)									for logging meeting		used during edit	

Inspection Issue Log

Item types:
S, M, m, Q, P, N

InspectionID 1 Date 11-apr-01

Item No	Doc ref	Doc page	Scan/ Check	Location on page	Type of item	Checklist or rule tag	Description If long explanation, remind 'Isa/It isT words' rule?	Number of occurr	time ref	who	Editor note	done
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												
25												
26												
27												
28												
29												
30												
31												
32												
33												
34												

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 93

Inspection statistics

Data summary

Owner: Niels Malotaux - Version 1.01 - 23 Nov 2009

prepare | fill in | changeable | calculated | assumed | results

Inspection ID	2	Date	29-nov-01	Leader	Niels Malotaux	e-mail	niels@malotaux.nl		
Product document	Eco Product Configuration x SDI / 784-RMU28					Pages	9	Chck	3

Individual checking data (to be reported during the entry process for logging meeting)

Checker report	Pages studied		Time spent (xx hrs)		Major + SM issues		minor issues		Improvements		Questions of intent		Check rate hr per page		Majors per hour		Majors per page	
	Scan	Chck	Scan	Chck	Scan	Chck	Scan	Chck	Scan	Chck	Scan	Chck	Scan	Chck	Scan	Chck	Scan	Chck
Author	9,0	3,0	0,5	1,0	9	4	4	1			2	1	0,05	0,33	20,0	4,0	1,0	1,3
Checker 1	9,0	3,0	0,5	1,5	2	0	1	4					0,06	0,50	4,0	0,0	0,2	0,0
Checker 2	9,0	3,0	0,5	1,0	3	4	1	2		1		1	0,06	0,33	6,0	4,0	0,3	1,3
Checker 3	9,0	3,0	0,5	1,3	1	1	19	2	0	1		1	0,06	0,42	2,0	0,8	0,1	0,3
Checker 4	9,0	3,0	1,0	2,0	19	30							0,11	0,67	19,0	15,0	2,1	10,0
Checker 5																		
Total checking hours			9,7	writes			Average item checking rate						0,07	0,45	10,2	4,8	0,8	2,6
							optimum checking rate						1,00	hr per page				

Logging meeting summary

	Major + SM issues		minor issues		Improvements		Questions of intent		Total items	
	Scan	Chck	Scan	Chck	Scan	Chck	Scan	Chck	Scan	Chck
Unique found during checking	21	21	13	12	2			1	36	34
New found in meeting									0	0
Total	21	21	13	12	2	0	0	1	36	34

Final findings as reported by editor

	Scan	Chck	Total
Major + SM issues	21	21	42
minor issues	13		
Change Reports	2		

Edit time		writes
Follow-up time		writes
Exit time		writes
Follow-up and exit time: author + leader		

Exit results

Did the Inspection Process meet the Exit Criteria? (yes/no)		date	
comment			

Assumptions

Average time to find and fix later	9,3	hrs major
% causing defects	50%	of found in inspection
Insp effective-ness	50%	% Maj found per page
Repair efficiency	5/6	(1 - fraction not repaired correctly)

Preparation

Planning time	2,0	writes
Entry time	1,0	writes
Kickoff, no of people	7	people
Kickoff, time	54	min
Planning and entry time: author + leader		

Logging meeting data (fill in at the end of logging meeting)

Number of people	7	people
Item logging time	90	min
Discussion time		min
Checking time		min
Pages chckd in meeting		pages
Brainstorming time		min
Items logged in meeting	36	
Logging time	1,0,5	writes
Item logging rate	0,40	items /min
Meeting checking rate	0,00	hr/page

Calculations

Total checking time	9,7	writes
Checking time before and in meeting		
Detection time	29,0	writes
Planning+Entry+Kickoff+Checking+Logging		
Control time	8,8	writes
Planning+Entry+Kickoff+Followup+Exit		
Defect removal time	29,0	writes
Detection+Kickoff+Followup+Exit		
Efficiency	1,4	Maj/writes

Time saved

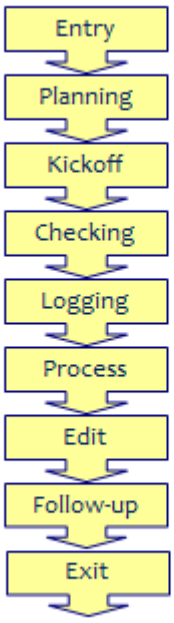
Net time saved	134	hrs saved
by using	29	hrs saved
Relative cost of inspecting	18%	vs without it

Results in document

Majors per page found	7,0	Maj/page
Maj per page remaining	8,2	Maj/page
Majors remaining in doc	73,5	Maj/p

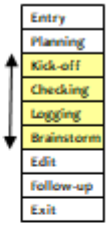
SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 94

Inspection Process Steps



SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 95

6 hour initial Inspection process



- 2 hr Kickoff
  - Why
  - How
  - What
- 2 hr Individual checking
  - 1 hr Whole document / relevant chapter
  - 1 hr 2 selected pages
- 2 hr Logging meeting
  - 1 hr Logging issues
  - ½ hr Discussion about Inspection process
  - ½ hr Discussion about what should have been in the document

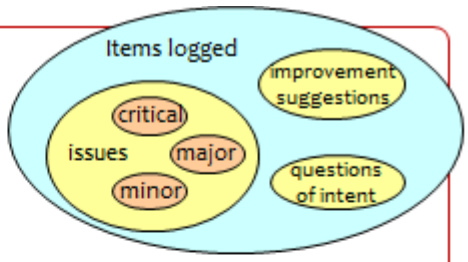
SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 96

4 hour mature Inspection process

- 1/2 hr Kickoff
  - Why
  - How
  - What
- 2 hr Individual checking
  - 1 hr Whole document / relevant chapter
  - 1 hr 2 selected pages
- 1 ½ hr Logging meeting
  - 1 hr Logging issues
  - ½ hr Discussion about Inspection process
  - ½ hr Brainstorm

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 97

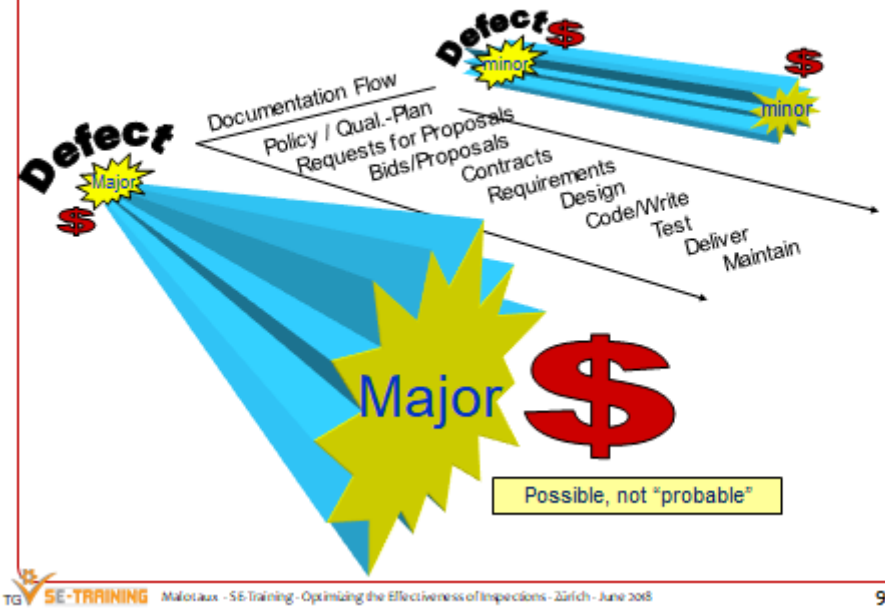
Defect classes



- Major defect
  - Defect probably has significantly increased costs to find and fix later (test, field)
    - 10 engineering hours lost extra
    - Average time in work-hours to find, log and fix a major defect by Inspection is 1 hour (observed by many sources)
- Minor defect
  - Not major (no significant impact on result)
- Super-major/critical
  - Order of magnitude more costly than major
  - Project threat

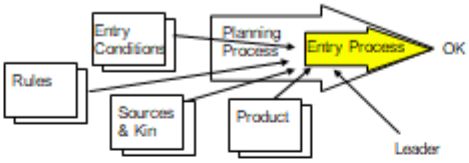
SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 98

Major ↔ minor Severity Concepts



TG SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 99

Entry Process



Entry
Planning
Kick-off
Checking
Logging
Brainstorm
Edit
Follow-up
Exit

- Purpose Why?
  - To avoid continuing a costly process which is doomed to failure (no exit, product not released)
  - To permit correction of fail-prone conditions
    - before using time and people fruitlessly
  - To permit continuous process improvement by learning which entry conditions are worth checking
- Organization How!
  - The Inspection Leader checks all conditions in the generic and product-specific entry Conditions
  - Anyone can suggest improved Entry Conditions to process owner (entry is part of process definition)
  - "Failed" entry conditions are dealt with (corrected, waivered) before entering

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 100

Entry: Generic entry criteria

Entry
Planning
Kick-off
Checking
Logging
Brainstorm
Edit
Follow-up
Exit

GEC1 The author can decide not to enter any substage of Inspection

GEC2 The leader can decide not to enter any substage of Inspection

GEC3 All source documents are in writing and successfully exited

GEC4 Generic and specific rule sets for the task are available in writing

GEC5 A master plan has been made with checking rate of one page per hour

GEC6 The leader has been trained and certified as Inspection leader

GEC7 A cursory (< 5 min) examination of a sample shows < 1 major/page

GEC8 Possible machine checks are done

GEC9 The author agrees to participate as checker

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 101

Checking roles

Entry
Planning
Kick-off
Checking
Logging
Brainstorm
Edit
Follow-up
Exit

- Different roles find different issues
  - User
  - Tester
  - System
  - Quality
  - Service
  - Source documents
  - Rules

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 102

Kickoff meeting

Entry
Planning
Kick-off
Checking
Logging
Brainstorm
Edit
Follow-up
Exit

- Instruction on Inspection
  - (why, how, what)
- Explain tasks to checkers
- General explanation about documents
- 0 ~ 2 hr

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 103

How about a general introduction ?

Entry
Planning
Kick-off
Checking
Logging
Brainstorm
Edit
Follow-up
Exit

- In the kick-off meeting some did not attend the general introduction
- This caused a rule:
  - You can only be a checker if you have been educated about the process

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 104

Individual checking

Entry
Planning
Kick-off
Checking
Logging
Brainstorm
Edit
Follow-up
Exit

CC1 Try to identify a maximum number of potential issues on behalf of your team, and to help the author

CC2 Your job is to help 'make the author a hero'

CC3 If you get a ridiculously high number of issues, consult with the leader

CC5 Don't be shy of noting any kind of issue you think you have found (you can later decide whether or not to report it)

CC6 You do not have to write a perfectly presented log. It is better to concentrate on finding more issues, but you may write any notes you like, any way you like. They are normally your private notes

CC7 If you have trouble finding issues, consult with the leader or another team member

CC8 If you have any time difficulty, consult with your Inspection leader

CC10 Focus on major (and super-major) issues, do not spend a lot of time and effort finding and noting minor issues

CC11 Classify as you go as S (super), M (major), m (minor), ? (question of intent), P (process improvement)

CC12 Fill in the section called Data Collection at the bottom of your master plan, with your personal checking data, so you can swiftly report your data at the beginning of the Logging Meeting.

See Procedure for Checker during Checking: IN.PR.CC

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 105

The Logging Meeting

Entry
Planning
Kick-off
Checking
Logging
Brainstorm
Edit
Follow-up
Exit

- The sole purpose of the Logging meeting is to record for the Editor:
  - the highest possible number of unique issues in the time available
  - with sufficient clarity that the Editor can understand what the problem is
- Discussing issues is not the purpose
- Fixing issues is not the purpose
- Discovering additional issues is part of the purpose…

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 106

Logging meeting

Entry
Planning
Kick-off
Checking
Logging
Brainstorm
Edit
Follow-up
Exit

- Logging
  - no discussion allowed
  - no suggestions, no solutions
  - mostly majors
  - any issue is a violation of a rule
  - 0.5 ~ 2 issues per minute logged
- What did you think of the Inspection process
- How should the document have looked like
- 2 hours maximum

Inspectors are consultants, helping the author to be a hero

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 107

Checking and Logging

Entry
Planning
Kick-off
Checking
Logging
Brainstorm
Edit
Follow-up
Exit

- Follow Inspection Master Plan
- Use the time assigned
- Check according to CC procedures, by GE rules
- Product document is checked against sources
- No emphasis on checking source documents
- Fill in Individual data in Inspection Master Plan
- Read CL procedures (author also AL)
- Be in time
- Don't cheat

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 108

Edit

Entry
Planning
Kick-off
Checking
Logging
Brainstorm
Edit
Follow-up
Exit

- Author is document owner
- Author decides what to do with issues
- Author decides on minor, Major, Super
- All issues must be acted upon
- Improvement suggestions sent to owners

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 109

Exit

Entry
Planning
Kick-off
Checking
Logging
Brainstorm
Edit
Follow-up
Exit

- All editing completed
- All change requests sent
- Data summary completed and in database
- No more than 0.25 (2 for beginners) major defects per page remaining
- Author or Leader can veto exit
- Can we release this document for further use?

Not zero defects, but *economically defensible quality*, *not worth looking further* at this stage

Inspection basics

Entry
Planning
Kick-off
Checking
Logging
Brainstorm
Edit
Follow-up
Exit

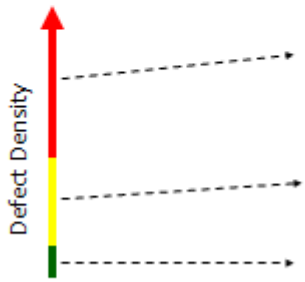
- The Inspection leader is trained and certified
- The leader is responsible for managing the process
- First objective is to identify and correct major defects
- Second, but most important, objective is to identify and remove the source of defects
- Fundamental measure of success is the quality-to-cost ratio of the total design life-cycle
- Short term measures include majors found per work-hour (efficiency) and percentage of defects found and treated compared with total defects (effectiveness)
- Productivity measure is the net hours saved due to defects found and removed earlier than they otherwise would be

(see One-page Inspection handbook)

Exit Criteria

Entry
Planning
Kick-off
Checking
Logging
Brainstorm
Edit
Follow-up
Exit

Once the quality level of a document is known, there are three possible paths forward:



Well above exit criteria: Process failure! Recreate after training or process improvement

Somewhat above exit criteria: Rework or enlarge inspection sample

Meets exit criteria: Success! Exit

Summary

- Rules are the laws for documents
- Optimum checking rate
- Sampling
- Types of defects
- Exit criteria
- Measuring the benefit

- Next: exercise

Exercise

Preparation: 15 mins in groups of 3

- Which document(s) are you Inspecting ?
  - Are there any source documents ?
- Which Rules are you checking against?
  - Generic Rule set or just top 3 ?
  - Any specific Rule sets for this document ?
    - e.g. requirements ? new ones for today ?
- Which page(s) will each of you be checking ?
  - All checkers check the same (most important) page ?
    - "logical" page, not necessarily one physical page (300 words text, 100 lines of code)
- Exit criterion?
  - How many Defects remaining ?

Exit criteria: estimating remaining majors (after fixing)

- You are about to Inspect your own document
- What is acceptable exit level?
  - 1000 estimated Major defects remaining per page ?
  - 100 ?
  - 10 ?
  - 1 ?
- What exit criteria will you use today?
  - I will accept no more than _____ estimated remaining major defects per page
- How much %% of defects do you think you'll find?
  - I will find _____ % of the defects

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 116

Checking

Individual Checking
Working alone
(tends to be very quiet)

- Check against your chosen Rules
- Check against source documents (if available)
- Look for Major defects
  - Rule violations with potentially large impact
- Note down what you have found (use issue log)
  - Majors only

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 117

Analysis

Are these reasonable for you ?
Any you wish to change ?
Why ?

- Overlap of defects
  - Assume total = double maximum found by one
- Number fixed correctly
  - Assume 5 out of 6 will be fixed correctly
- Defects missed?
  - Assume we have found one third
    (based on observed effectiveness of new Inspectors)
- Chance of a defect causing a problem
  - Assume one third of defects will cause loss
- Average loss from a major defect
  - Assume ten hours

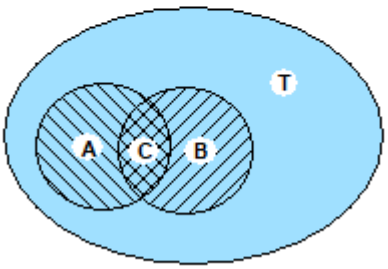
Capture - recapture

- How many fish are there in the lake?
  - Catch 20 fish
  - Mark them and let them swim again
  - Wait for good mixture
  - Catch another 20 fish
  - If 4 of these are marked, how many fish are there in the lake?

How many fish in the lake?

$$\frac{FoundMarked(C)}{Found(B)} = \frac{TotalMarked(A)}{Total(T)}$$

$$T = \frac{A * B}{C}$$



Report results

- Information from each group:
  - Type of document _____
    (e.g. requirements, functional specification, test plan, code)
  - Total size of document (in pages) _____
  - Number of pages Inspected (main focus) _____
    (i.e. number of words divided by 300)
  - Number of major issues found
    - By each individual checker _____
    - Total unique major issues _____
  - Major issues remaining _____
  - Potential time saved _____
  - Potential money saved _____

Early Inspections

SQC - Specification Quality Control

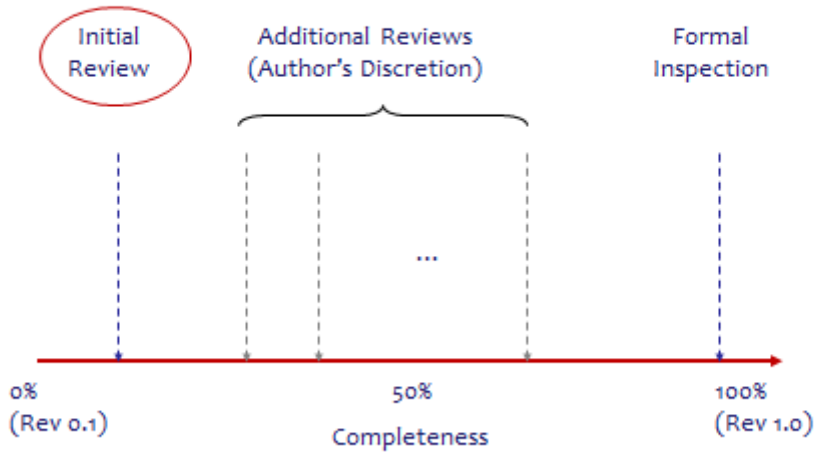
Extreme Inspections

Agile Inspections

Lean QA

Early Inspection

Prevention costs less than Repair



Initial Review

Purpose: Locating mistakes and tendencies that could lead to injecting major defects if not corrected

When: As soon as the author has completed a small representative portion of the specification, typically a few pages or 600-1200 words (e.g. few requirements)

Who: Individual or small team (1 or 2)
- Expertise in the subject matter
- Expertise in generic principles (such as requirements engineering, design, specific language)

What: Detailed review of the specification against rules and checklists for known error conditions and dangerous tendencies; formal inspection may be used

Duration: Because the sample is small, the initial review takes only 1-2 hr

The earlier it's reviewed, the more defects we can *prevent*

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

Average major defects per requirement in initial review	8
Average major defects per requirement in final document	3

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 x 750 x 10 hr = 37500 hr / 3 = 12500 x $50 = $625000

Why Early Inspection Works

- Many defects are repetitive and can be prevented
  - Early review allows an author to get independent feedback on individual tendencies and errors
  - By applying early learning to the rest (~90%) of the writing process, many defects are prevented before they occur
  - Reducing rework in both the document under review and all downstream derivative work products

Case: Test Cases

A tester 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

First round	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

Early Detection vs. Prevention

Denise Leigh (Sema group, UK), British Computer Society address, 1992:

An eight-work-year development, delivered in five increments over nine months for Sema Group (UK), found:

- 3512 defects through inspection
- 90 through testing
- and 35 (including enhancement requests) through product field use

After two evolutionary deliveries, *unit testing of programs was discontinued because it was no longer cost-effective*

Nice job! Early detection has big benefits - BUT...

How many of the 3512 defects found in end-of-line inspections could have been completely prevented by Early Inspection?

Cost-effective defect prevention is the bottom line

Outcome limitation of Extreme Inspection

- The outcome is to determine 'specification exit'
  - by measuring and estimating Major defect density
- The outcome is NOT (as with conventional inspection) to 'clean up' bad work
- Many of conventions of Inspections are NOT necessary or desirable
  Only need to focus on ***determining that the specification is exit-able or NOT***
- No need to get maximum effectiveness
  - by a large team or by using one hour per page or by looking at all pages
  - we can sample in 10-40 minutes and use one or 2 people
- If a checker detects one or more Majors in a page, it is NOT exit-able
  - Because the estimated actual quantity of majors exceeds the Exit limit of 'one per page'
  - If finding less than one Major on 4 pages, exit may be economic
- Economic is the key word
  - We are trying to determine if it pays off to exit now
  - Or to rewrite the spec to a cleaner level now

True Measure of Inspection Progress

- The measure Inspection effectiveness is NOT the quantity of Major defects found and fixed
  - In fact we strongly recommend that this measure is well hidden from public view!
- The true measure is the average level of Major defects/Page which we can consistently release
  - Moving from about 100 Majors/Page down towards about less than one per page
  - This cannot be achieved by finding and fixing defects (because we cannot find a large percentage at all)
  - It can only be achieved by motivating writers to reduce defects actually injected and move them down towards one maximum injected/page
  - This is the 'individual defect injection learning rate'
  - Individuals seem capable of reducing their own defect injection by about half for each cycle of learning
- The measure of real progress is the *defect density released*
- This measure most closely correlates with later statistics on quality and productivity of projects.

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 132

Example of Inspection Progress

Rev.	# of Defects	# of Pages	Defects per Page (DPP)	% Change in DPP
0.3	312	31	10.06	
0.5	209	44	4.75	-53%
0.6	247	60	4.12	-13%
0.7	114	33	3.45	-16%
0.8	45	38	1.18	-66%
1.0	10	45	0.22	-81%
Overall % change in DPP revision 0.3 to 1.0:				-98%

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 133

SPC - Specification Quality Control

- Human defect removal by Inspections/reviews/SQC is
  a hopeless cause: not worth it
- Spec QC can be used, in spite of imperfect effectiveness,
  to accurately estimate major defect level density
- This measurement can be used to motivate engineers to
  dramatically (50x! over about 5 learning cycles) reduce their defect insertion (rule violation) to a practical exit level
  (like less than 1.0 Majors/page)

Suggested SQC Policy

- All critical specifications will be measured for defects
- Defective work over the exit level,
  will not be released for others to use

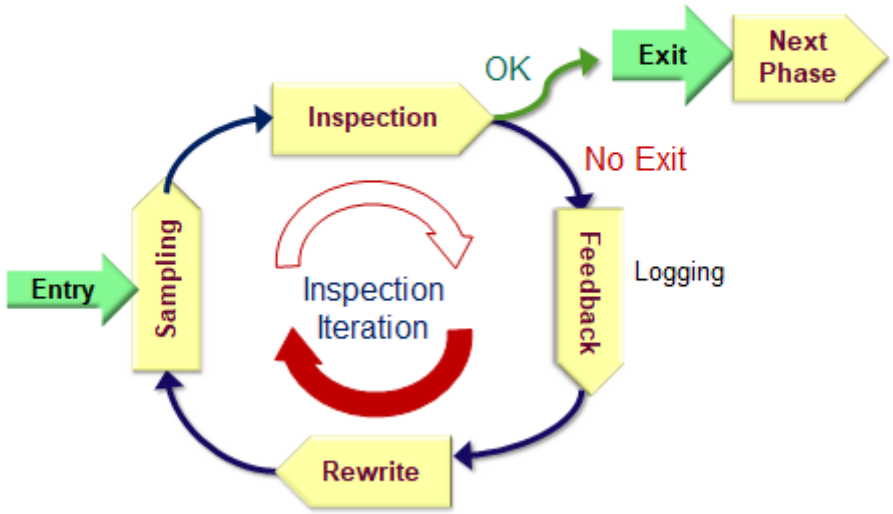
Expectations: Exit

- If you apply numeric quality level exit conditions you can expect:
  - Drastic reduction of Major Defects
  - Developers will bother to learn, and to apply correctly basic 'rules' (like 'clear', 'unambiguous')
  - Each individual will have to experience a gradual process of learning the 'rules'
  - Getting 50% better each time they try to meet the exit level

Brainstorming Root Causes

- After an Inspection, Major defects are examined by the checking team, half an hour sessions
  - They are looking at a colleagues work, colleague is there (the source of defects: knows why)
- Select one of a small group of recurrent types of defects, to work on: 10 in 30 minutes, 3 min each
- They brainstorm root causes (organizational, not personal)
  - Like: misleading training course information
- They brainstorm possible 'cures'
  - Like: enhance slides, and tests to make the point clearer
- They may themselves, carry out the proposed changes and try them to see if they work. Keep it simple – prove concept works
- Successful changes are picked up at corporate quality level and instituted more widely and more properly

Quick Learning



Improving the Effectiveness of Reviews and Inspections

www.malotaux.nl/conferences
www.malotaux.nl/booklets
www.malotaux.nl/inspections

SE-TRAINING
N R Malotaux Consultancy

Niels Malotaux

+31-655 753 604 niels@malotaux.nl www.malotaux.nl

Inspections

Used in Various Ways

Case: Can you teach Inspections ?

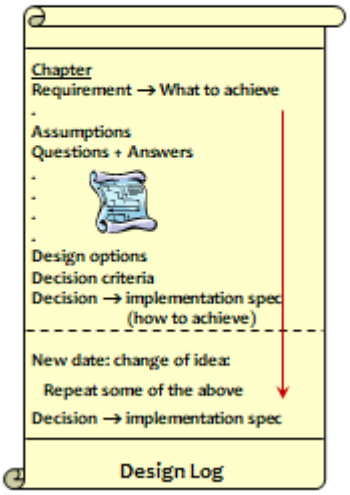
- Short intro
- Are you regularly reviewing ?
- Let's do it: baseline
  - Take a document
  - Reproduce one page
  - Do review
  - No issues
- One rule ('source')
  - Many issues

Sorry, picture removed for confidentiality reasons

Datalog
function
improvement

DesignLog

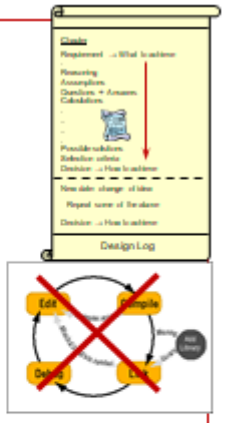
- In computer, not loose notes, not in e-mails, not handwritten
  - Text
  - Drawings!
  - On subject order
  - Initially free-format
  - For all to see
- All concepts contemplated
  - Requirement
  - Assumptions
  - Questions
  - Available techniques
  - Calculations
  - Choices + reasoning:
    - If rejected: why?
    - If chosen: why?
- Rejected choices
- Final (current) choices
- Implementation



Results

- No code until DesignLog 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: In the pub



James:

Niels, this is Louise
Louise, this is Niels, who taught me about DesignLogging - Tell what happened

Louise:

- *We had only 7 days to finish some software*
- *We were working hard, coding, testing, coding, testing*
- *James said we should stop coding and go back to the design*
- *"We don't have time !" - "We've only 7 days !"*
- *James insisted*
- *We designed, found the problem, corrected it, cleaned up the mess*
- *Done in less than 7 days*
- *Thank you!*

What James told me recently

- I gave the design to two colleagues for review
- Louise corrected some minor issues
- It went into a 'final' review, with another colleague
- Based in his expertise, *the solution was completely reworked*
- Actually, two features were delivered and deployed
  - One that was design and code reviewed had no issues after deployment
  - Other one, was the source of quite some defects
- In summary, this success has proved instrumental in buy-in for DesignLogs which are now embedded in the development process

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 146

Case: City of Amsterdam

- Can you teach Inspections ?
- Using a tender document that was already 3 weeks late
  (please can you come tomorrow ?)
- You'll ditch the document after the course !
- Ha ha
- Of course they did
- The project was ditched a few weeks later
- Why ?
- Saved a lot of tax-payers money

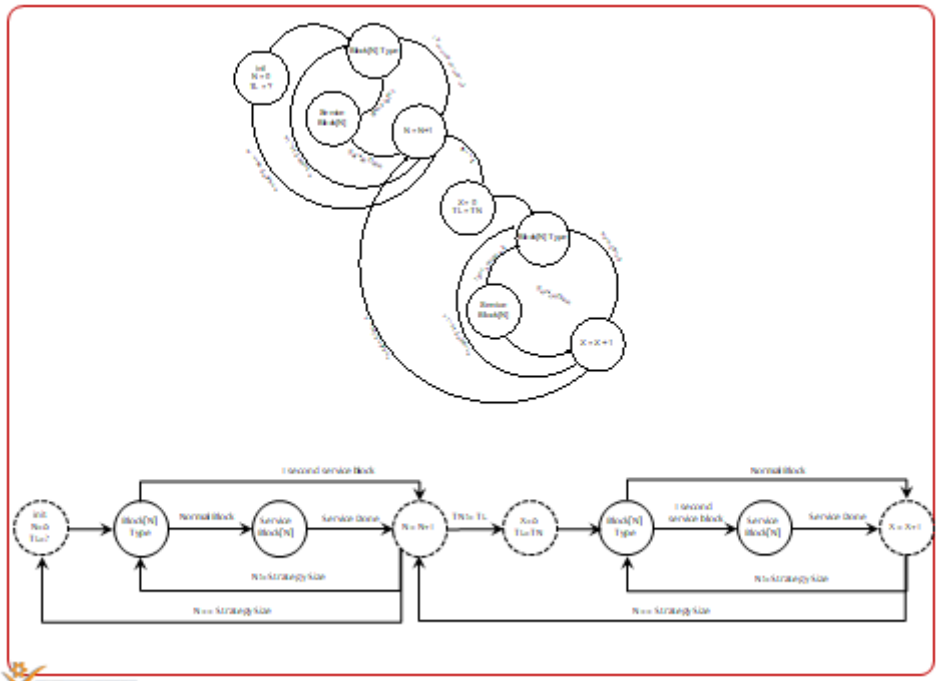
SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 147

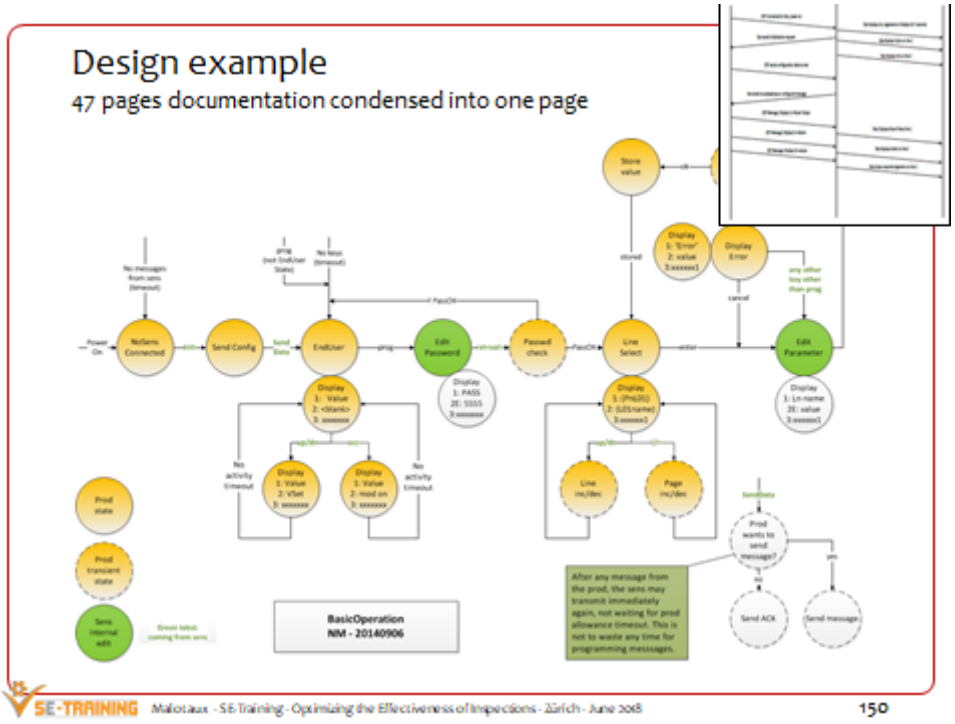
Make Documents Reviewable

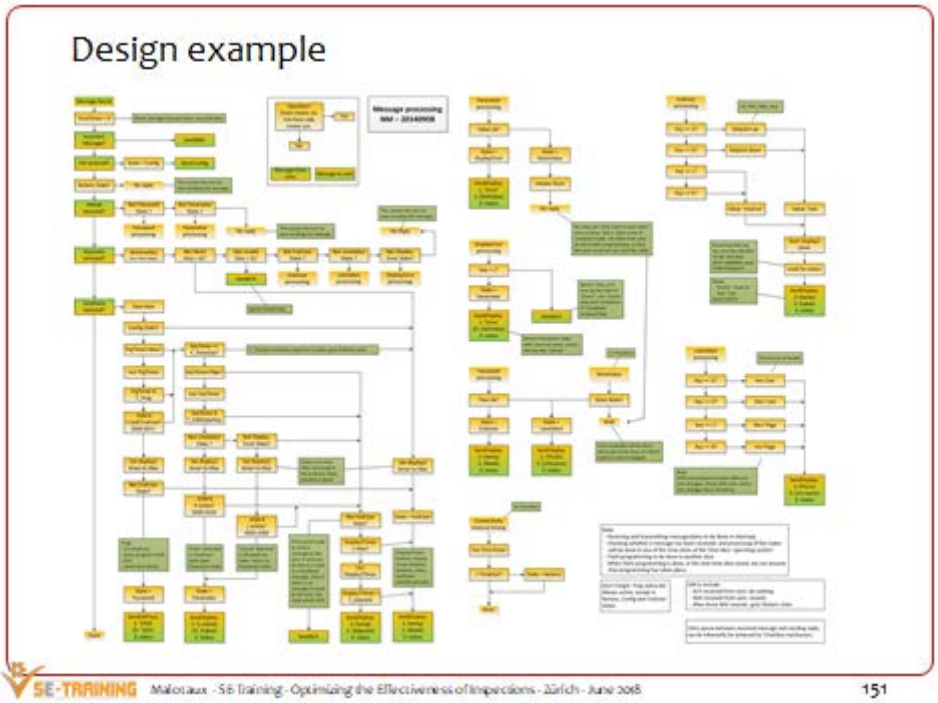
If not, they're probably not very useful

Unambiguous, Clear to Test, ...

Mantra: "Where are the pictures ?"



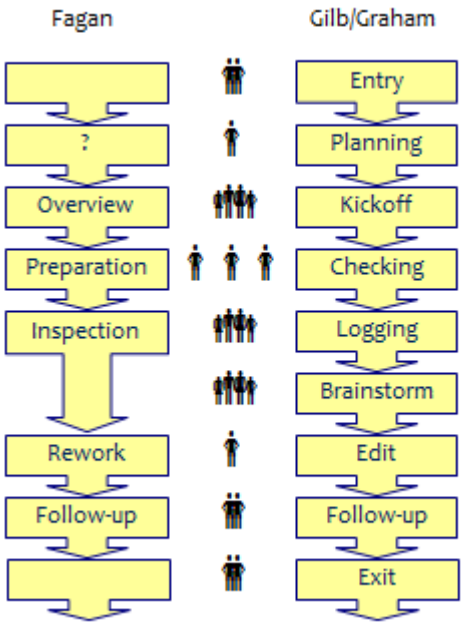

Design example
47 pages documentation condensed into one page
BasicOperation
NM - 20140906
After any message from the prod, the sens may transmit immediately again, not waiting for prod allowance timeout. This is not to waste any time for programming messages.
SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018
150


Design example
SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018
151

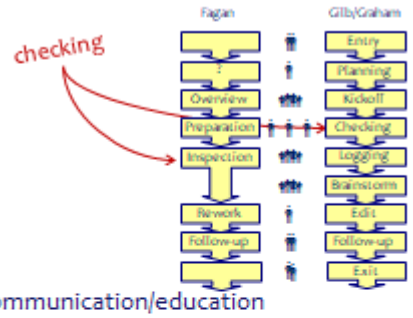
Fagan Inspections

http://extras.springer.com/2002/978-3-642-59413-7/4/rom/pdf/Fagan_hist.pdf

Inspection Process Steps



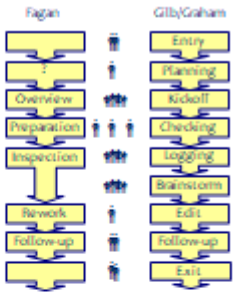
Fagan Process



- Steps
  - Overview team Communication/education
  - Preparation individual Education
  - Inspection team Finding errors (no discussion)
  - Rework author Resolving errors and problems
  - Follow-up moderator Decision - analyse - process
- *What* to look for in Inspection
  Errors classified by type, ranked by frequency,
- *How* to look for presence of errors (education!)
- Analyse results for prevention

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 154

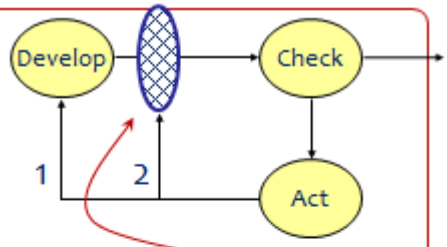
Fagan Inspections



- Objective: finding errors
- Based on publication in IBM Journal
- Emphasis on inspecting code
- If more than 5% reworked: 100% re-inspection
- If less than 5%: moderator decides
- *All* modifications better be inspected (even 1 line change)
- Most defects found during the meeting
- Typical defect list obtained used for prevention
- Typical defect list obtained used for next inspection
- Learn *how* to look for defects

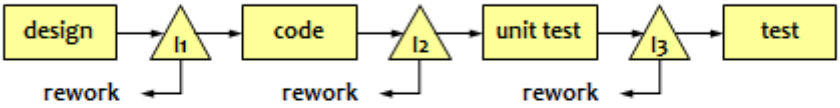
SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 155

Purposes of Inspection



- Producing defect-free products with high productivity
- Reduce total defect rework
- Reduce the schedule impact of defects
- Find defects immediately after injection
- Provide the author with the quickest feedback on defects, how to recognize and avoid them in the future
- Without immediate feedback and learning, we will keep making the same mistakes

Fagan experiment



Productivity change by Inspections:

- No Inspection: 100% (baseline)
- I1 only: 112% (9/10 people can do the same)
- I1 and I2: 123% (8/10 people can do the same)
- I3 had negative ROI, it was discarded

M.E. Fagan: *Design and Code Inspections to reduce errors in program development*
IBM Systems Journal, Vol15, No3, 1976

Perseverance and results

- I did not receive much support

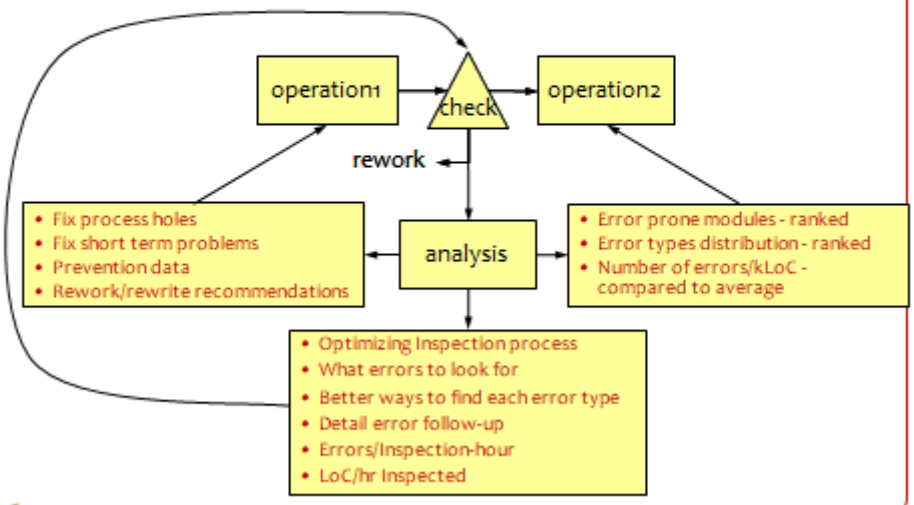
  in fact, I was ridiculed, counselled and otherwise told to stop the nonsense and get on with the job of managing projects the way everyone else was doing it

- Applied and executed as intended

  it produces significant improvements to the software development process, including
  - schedule and cost reduction
  - productivity improvements
  - fewer customer-reported defects

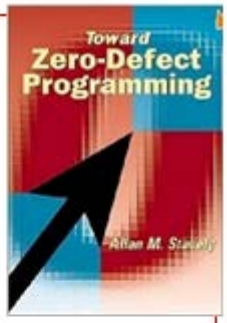
Prevention and knowledge building

(ref Fagan)



Cleanroom Inspections

Cleanroom Software Development



- Design (Mathematical proof)
- Verification (review of design by others)
- Implementation
- Verification (review of code by others)
- No unit test
- Only Integration Test (by others)
  (Test is Running Code)

- *Verification* is for finding defects
- *Testing* is for not finding defects

Cleanroom fundamentals

- Design principle
  - Designers can and should produce systems free of defects *before* testing
- Testing principle
  - The purpose of testing is to *measure* quality
- Main development model
  - Incremental (Cleanroom) / Evolutionary (Gilb) / Cyclic (TSP) / Agile
    - Each increment is a working subset of the final product
    - Stable requirements for each increment
    - No eleventh hour integration

Cleanroom Principles

- Incremental development
  - User verifiable increments
- Team organisation
  - 4~8 people
- Formal methods of specification and design
  - Level of formalism varies even within project
- Intense review
  - Mathematical proof of correctness
  - Verifying individual control structures
- No unit test
  - Not testing infinite number of paths, infinite combination of data
- Statistical testing as reliability measurement
  - Testing is not suitable for bug-hunting

Cleanroom Inspections

- The purpose of Inspection is to eliminate defects
- Exit criterion for design:
  - One design statement materializes as 3 to 10 code statements
- Checklists of typical errors we make
  - Listed in order of frequency
- No Unit Test - Developer does not 'try' software !

- Testing:
  - Finding as many of the remaining defects as possible
  - Too many errors discovered
    - → previous steps are not being done properly
    - → redo previous steps (do not "repair")

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 164

Cleanroom: 'Slowest reviewer sets the pace'

- Wrong: Does anyone consider this incorrect?
  (dreamers won't answer)
- Better: Does everybody agree that this is correct?
  (attention is required)
- A team does not consider a verification condition proven until the slowest person to respond has expressed agreement

It is important to resist taking shortcuts here

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 165

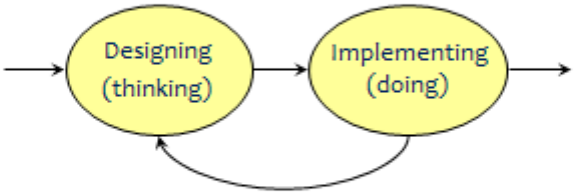
Cleanroom benefits

- Short development cycles
- Zero failures in field use
- Long product life

Quality costs less

Getting stuck somewhere ?

- Getting stuck in implementation? Back to the design !



- Getting stuck in Inspection? Back to the design !
- Getting stuck in Testing? Back to the design !

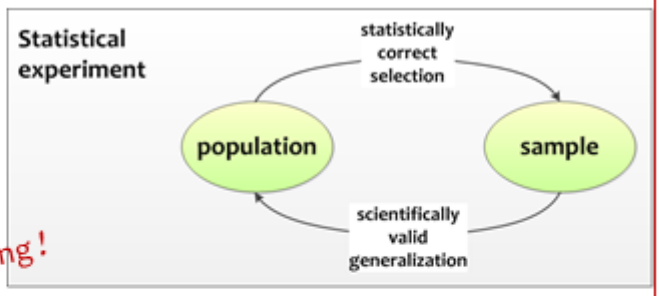
- Why do we get stuck ?
- Root cause analysis !

Testing in Cleanroom

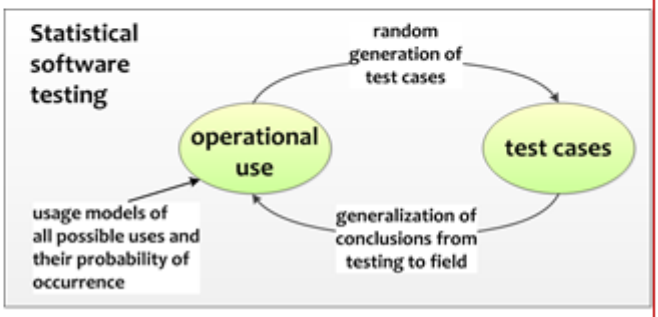
- Testing is an important part of the process, but it is done only after verification (by Inspection) is successfully completed
- Testing is done:
  - Primarily to *measure* quality
  - Secondarily to find defects that escaped detection during verification
- Number of bugs per thousand lines of code <10 after verification, compilation and syntax checking
- 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 – statistical testing
- Risk based testing – high risk, low probability will still be checked!

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 168

Statistical Testing



You need also other forms of testing!



SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 169

Rules for Code

Tick the Code Rule Set

(Miska Hiltunen, 2007)

Extra baggage rules

DEAD	Avoid unreachable code
DRY	A comment must not repeat code
INTENT	A comment must either describe the intent of the code or summarize it
ONE	Each line shall contain at most one statement
UNIQUE	Code fragments must be unique

Tick the Code Rule Set

(Miska Hiltunen, 2007)

Missing info rules

DEFAULT	A 'switch' must always have a 'default' clause
ELSE	An 'if' always has an 'else'
MAGIC	Do not hardcode values
PTHESES	Parenthesize amply
TAG	Forbidden: marker comments
ACCESS	Variables must have access routines
HIDE	Direct access to global and member variables is forbidden

Tick the Code Rule Set

(Miska Hiltunen, 2007)

Chaos-inducers

CALL	Call subroutines where feasible
NAME	Bad names make code bad
RETURN	Each routine shall contain exactly one 'return'
SIMPLE	Code must be simple
FAR	Keep related actions together
DEEP	Avoid deep nesting
FOCUS	A routine shall do one and only one thing

Tick the Code Rule Set

(Miska Hiltunen, 2007)

Risky assumptions

CHECK-IN	Each routine shall check its input data
NEVERNULL	Never access a 'NULL' pointer or reference
NULL	Set freed or invalid pointers to 'NULL'
CONST 1ST	Put constants on the left side in comparisons
ZERO	Never divide by zero
PLOCAL	Never return a reference or pointer to local data
ARRAY	Array accesses shall be within the array
VERIFY	Setter must check the value for validity

Tick the Code Speed

(Miska Hiltunen, 2007)

Rule	Call	Check-In	Dead	Deep	Default	Dry	Else
Ticks/hr	46	82	45	76	11	53	322
Rule	**Hide**	**Magic**	**Name**	**NeverNull**	**Tag**	**Unique**	
Ticks/hr	186	516	93	90	18	20	

- Average number of ticks found per hour per rule
- Software developers could find this many violations in one hour in the code they produce
- 144 developers checked for 108h to create the data

Draft Rule Set for Java (Sybren Stüvel, 2007)

SIMPLE	Code should be as simple as possible, but not simpler
DOCUMENT	Documentation should be such that a developer who's unfamiliar with the code can still understand the reasoning behind it
CORRECT	Naming and documentation must be correct
CONDITIONAL CORE	Core functionality of a method should be outside any conditional block
EARLY RETURN	Return as soon as you can from a method. Assigning to a temporary variable and returning that variable usually results in overly complex code
EXCEPTIONS	Use exceptions to signal an error condition Don't return null to signify an error

Draft Rule Set for Java (Sybren Stüvel, 2007)

REUSE	Use common library functions where applicable At least take a look at StringUtils and ListUtils (Spring framework) and ArrayUtils (Apache Commons) Use XStream for parsing and generating XML
EQUALS	To compare objects use their equals method
MAGIC	Define constants in one place, and use them
REFER	Use @see and @link in JavaDoc to refer readers to relevant other locations
READABLE	Ensure the code is easily readable
SENSIBLE TEST VALUES	Test values should be sensible
EARLY JAVADOC	Write a method's JavaDoc before writing actual code. This gives a method its scope
REVIEW TESTS	Start by reviewing the unit tests

MISRA C

- MISRA: Motor Industry Software Reliability Association
- MISRA C (1998) has 127 rules
- Providing a set of guidelines to restrict features in the ISO C language of known undefined or otherwise dangerous behaviour
- Of these, 93 are required and the remaining 34 are advisory
  - Rule 104 (required): Non-constant pointers to functions shall not be used

Version	Rules	Sections	Pages
MISRA C 1998	127	17	69
MISRA C 2004	141	21	111

MISRA C

Rule 59 (required): The statement forming the body of an "if", "else if", "else", "while", "do ... while", or "for" statement shall always be enclosed in braces

```
if (x == 0)
{
y = 10;
z = 0;
}
else
y = 20;
z = 1;
```

MISRA C

Rule 33 (required):
The right hand side of a "&&" or "||" operator shall not contain side effects

```
if ((x == y) || (*p++ == z))
{
/* do something */
}
```

```
if (x == y)
{
doSomething = 1;
}
else if (*p++ == z)
{
doSomething = 1;
}

if (doSomething)
{
/* do something */
}
```


MISRA C

Motor Industry Software Reliability Association

```
a[i] = ++i;
```
happens once in every 7,000 lines in C

```
c == d;

if (c=d)
{
}
```

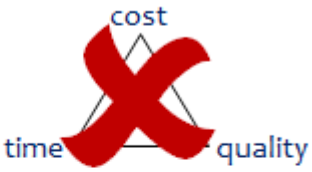
Put on checklist

What is Quality ?

What is Quality ?

- I know it when I see it … ?
- Should be *predictable* before it is there
- Should be *measurable* whether it is there
- But …
  ultimately they must like it when they see it

Does quality cost more ?



benefit is

- The ~~cost is no~~t in the quality
- The cost is in the non-quality

The *right* quality costs *less*

Deming

- Quality comes not from inspection (Verification & Validation), but from *improvement of the production process*
- Inspection does not improve quality, nor guarantee quality
- It's too late
- The quality, good or bad, *is already in the product*
- *You cannot inspect quality into a product*

→ *People who do the work put the quality in, good or bad*

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)

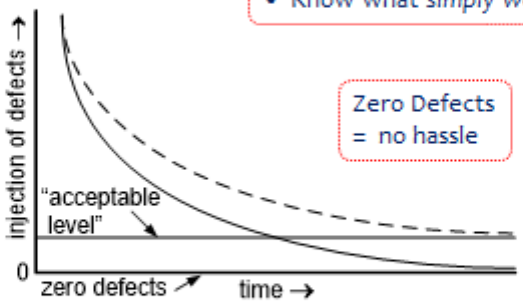
Added by Philip Crosby Associates, 2004

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 186

What is Zero Defects

- We aren't perfect, but the customer shouldn't find out
- What we deliver *simply works*
- Know what *simply works* means !

- Zero Defects is an *asymptote*



Zero Defects = no hassle

- 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

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 187

Is Zero Defects possible ?

- As long as we think Zero Defects is impossible, we will keep producing defects
- From now on, we don't want to make mistakes any more
- We feel the failure (no pain - no gain)
- If we deliver a result, we are sure it is OK and we'll be highly surprised when there proves to be a defect after all
- We do what we can to improve (continuous improvement)

SE-TRAINING Malotaux - SE-Training - Optimizing the Effectiveness of Inspections - Zürich - June 2018 188

Malotaux - SE-Training - June 2018

SE-TRAINING

Niels Malotaux

Improving the Effectiveness of Reviews and Inspections

N R Malotaux - Consultancy
tel +49-5632 922 5132
mob +31-655 753 604
niels@malotaux.nl
www.malotaux.nl