

What is the Essence of Being Lean and Agile ?

Niels Malotaux

N R Malotaux
Consultancy

+31-30-228 88 68

niels@malotaux.nl

www.malotaux.nl

1

Niels Malotaux

Result Management

- **Project Coach**
 - Evolutionary Project Management (Evo)
 - Requirements Engineering
 - Reviews and Inspections
 - **Dependability** (Systems that simply work)



Helping projects and organizations to become predictable

2

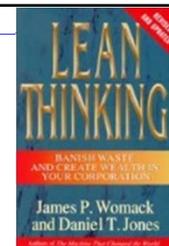
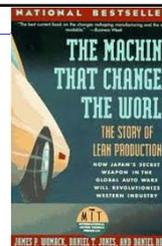
Ultimate Goal of a What We Do

Quality on Time

- **Delivering the Right Result at the Right Time, wasting as little time as possible (= efficiently)**
- **Providing the customer with**
 - what he needs
 - at the time he needs it
 - to be satisfied
 - to be more successful than he was without it
 - **Constrained by (win - win)**
 - what the customer can afford
 - what we mutually beneficially and satisfactorily can deliver
 - in a reasonable period of time

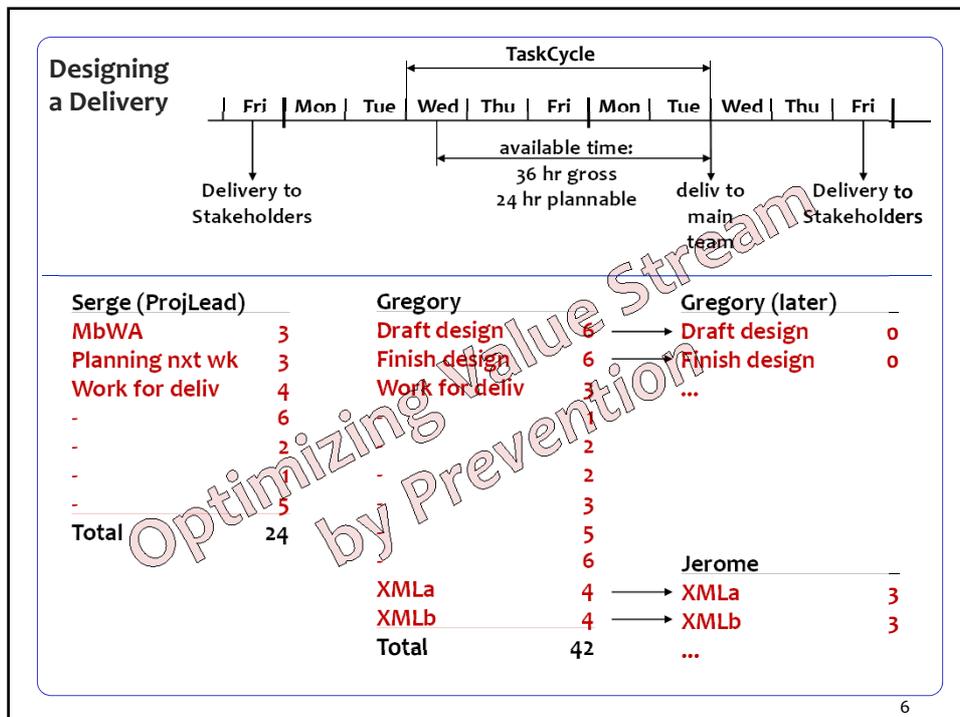
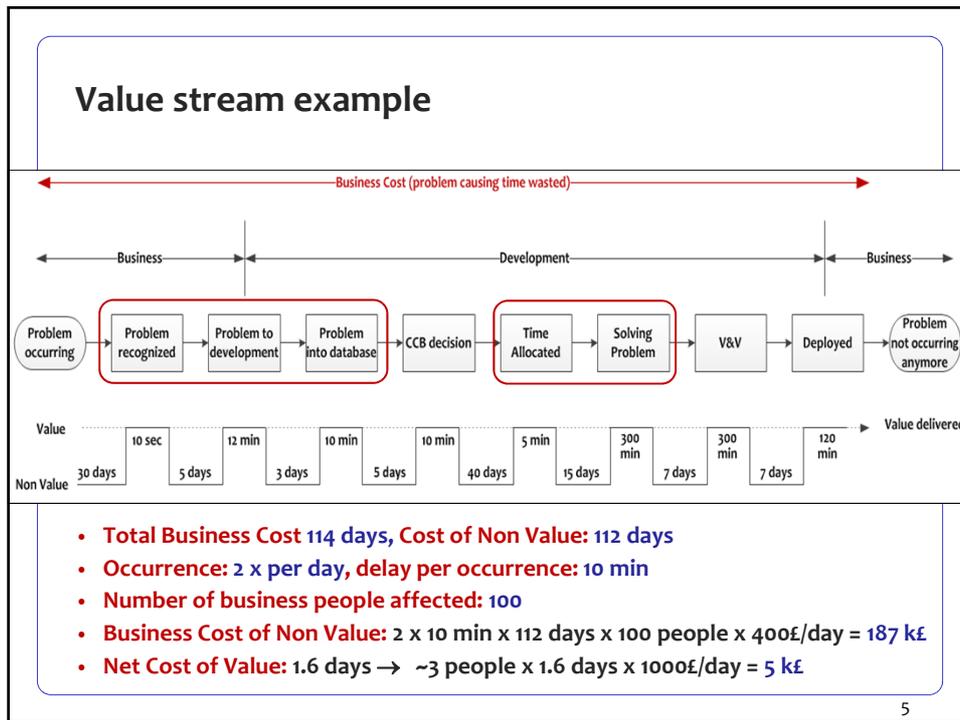
3

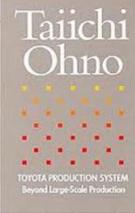
Lean



- **The goal is reduction of waste**
- **To achieve this, a company must look at what creates value and eliminate all other activities**
 - Understand and specify the **value** desired by the customer
 - Identify the **value stream** for each product providing that value
 - **Challenge** all of the wasted steps (generally nine out of ten) currently necessary to provide it
 - Make the product **flow continuously** through the remaining value-added steps
 - Introduce **pull** between all steps where continuous flow is possible
 - Manage toward **perfection** so that the number of steps and the amount of time and information needed to serve the customer continually falls

4





Toyota Production System (TPS)

1950

- Toyota almost collapsed
- Laying off 1/3 of workforce

Four specific aims:



- Deliver the highest possible quality and service to the customer
- Develop employee's potential based upon mutual respect and cooperation
- Reduce cost through eliminating waste in any given process
- Build a flexible production site that can respond to changes in the market

7



Taiichi Ohno - The Toyota Production System

- All we do is looking at the TimeLine from Order to Cash (p.ix)

order ←===== cash



Reducing the time by removing non-value-added wastes

- The Toyota Production System began when I challenged the old system (p11)
- Necessity is the mother of invention: improvements are made on clear purposes and need (p13)
- The TPS has been built on the practice of asking “Why?” 5 times (p17)
- The time that provides me with the most vital information about management is the time I spent in the plant, not in the office (p20)
- Toyota's top management watched the situation quietly and I admire the attitude they took (p31)

8

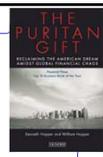
Building on quite some history

- **Benjamin Franklin** (1706-1790)
 - Waste nothing, cut off all unnecessary activities, plan before doing, be proactive, assess results and learn continuously to improve
- **Henry Ford** (1863-1947)
 - **My Life and Work** (1922)
 - We have eliminated a great number of wastes
 - **Today and Tomorrow** (1926)
 - Learning from waste, keeping things clean and safe, better treated people produce more
- **Toyoda's (Sakichi, Kiichiro, Eiji)** (1867-1930, 1894-1952, 1913-)
 - **Jidoka: Zero-Defects, stop the production line** (1926)
 - **Just-in-time – flow – pull**
- **W. Edwards Deming** (1900-1993)
 - **Shewart cycle: Design-Produce-Sell-Study-Redesign** (Japan – 1950)
 - **Becoming totally focused on quality improvement** (Japan – 1950)
 - **Management to take personal responsibility for quality of the product**
 - **Out of the Crisis** (1986) – Quality reduces waste
- **Joseph M. Juran** (1904-2008)
 - **Quality Control Handbook** (1951, Japan – 1954)
 - **Total Quality Management – TQM**
 - **Pareto Principle**
- **Philip Crosby** (1926-2001)
 - **Quality is Free** (1980)
 - **Zero-defects** (1961)
- **Taiichi Ohno** (1912-1990)
 - **(Implemented the) Toyota Production System (Beyond Lange-Scale Production** (1978, 1988))
 - **Absolute elimination of waste - Optimizing the TimeLine from order to cash**
- **Masaaki Imai** (1930-)
 - **Kaizen: The Key to Japan's Competitive Success** (1986)
 - **Gemba Kaizen: A Commonsense Low-Cost Approach to Management** (1997)

Why do we still have to talk about this ?

*Eliminating Waste
Not doing what doesn't yield value*





9

The essential ingredient: the PDCA Cycle

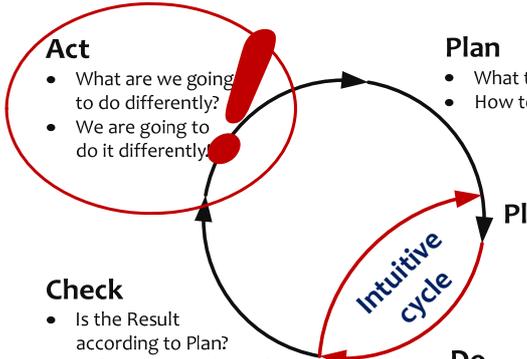
(Shewhart Cycle - Deming Cycle - Plan-Do-Study-Act Cycle - Kaizen)

Act

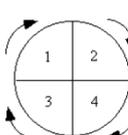
- What are we going to do differently?
- We are going to do it differently.

Plan

- What to achieve
- How to achieve it



Investigative Survey



Design

Manufacture

Check

- Is the Result according to Plan?
- Is the way we achieved the Result according to Plan?

Do

Carry out the Plan

10

Identifying waste

Manufacturing	Development	Possible Remedies
Overproduction	Extra features Unused documents	Real Requirements, prioritizing, deciding what not to do
Inventory	Partially done work	Synchronization, Just In Time
Transport	Handoffs	Synchronization if different people have to do it
Processing	Design inefficiency Wishful thinking	Knowledge, experience, reviews Preflection
Waiting	Delays	Process/Organization design Active synchronization
Movement	Task Switching Finding right files Number of clicks	Max 2 tasks in parallel Digital 5S Design
Defects	Defects	Prevention
Ignoring ingenuity of people	Ignoring ingenuity of people	Real management, coaching, empowerment, bottom-up responsibility, inviting whistle-blowing

11

What is value ?



- **Heathrow Terminal 5: “Great success !”**
 - Normal people aren’t interested in the technical details of a terminal
 - They only want to check-in their luggage as *easily* as possible
and
 - Get their luggage back as *quickly* as possible in *acceptable condition*
at their destination
 - They didn’t
- **One of the problems is to determine what the project (or our work in general) really is about**
- **A project doesn’t deliver value**
 - A project only delivers *the conditions* for value

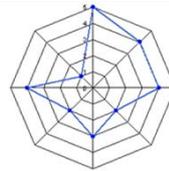
12

What is Agile ?

- A philosophy (Agile Manifesto)

13

The Agile Manifesto (2001)



We are uncovering better ways of **developing software** by **doing it** and **helping others do it**

Through this work we have come to value:

- **Individuals and interactions** **over** **processes and tools**
- **Working software** **over** **comprehensive documentation**
- **Customer collaboration** **over** **contract negotiation**
- **Responding to change** **over** **following a plan**

That is, while there is **value in the items on the right**, we value the items on the left more

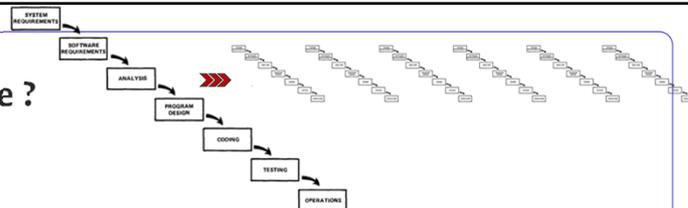
14

From the Principles behind the Agile Manifesto

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software
- We welcome changing requirements, even late in development
- We deliver working software frequently; Working software is the primary measure of progress
- Business people and developers must work together daily
- Simplicity - the art of maximizing the amount of work not done
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly

15

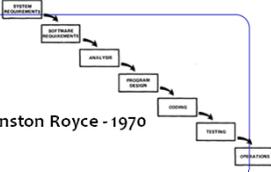
What is Agile ?



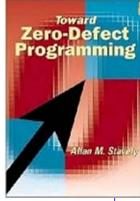
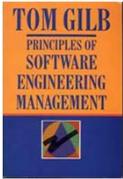
- A philosophy (Agile Manifesto)
- Agile = ability to move quick, easy and adaptable
- Short iterations – not one Waterfall
- Delivering value (not much notion how to define and measure real value)
- Retrospectives (no retrospectives on retrospectives: did it really work ?)
- Not a standard: You can make of it whatever you want
- XP - focus on software development techniques
- Scrum - very basic short term organization of development
- Are you Agile if you religiously focus on a 'method' ?

16

The past was already ahead



- **Managing the development of large software systems** - Winston Royce - 1970
 - Famous 'Waterfall document': figure 2 showed a 'waterfall'
 - Text and other figures showed that Waterfall doesn't work
 - Anyone promoting Waterfall doesn't know or didn't learn from history
- **Cleanroom software engineering** - Harlan Mills - 1970's
 - Incremental Development - Short Iterations
 - Defect prevention rather than defect removal
 - Inspections to feed prevention
 - No unit tests needed
 - Statistical testing
 - If final tests fail: no repair - start over again
 - 10-times less defects at lower cost
 - Quality is cheaper
- **Evolutionary Delivery - Evo** - Tom Gilb - 1974, 1976, 1988, 2005
 - Incremental + Iterative + *Learning and consequent adaptation*
 - Fast and Frequent Plan-Do-Check-Act
 - Quantifying Requirements - Real Requirements
 - Defect prevention rather than defect removal


17

XP – eXtreme Programming

- **Planning Game**
- **Metaphor**
- **Simple Design**
- **Testing (TDD)**
- **Refactoring**
- **Coding standards**
- **Small releases**
- **Pair programming**
- **Collective Ownership**
- **Continuous integration**
- **40-hour week**
- **On-site customer**

Original project was not successful
as soon as the writer of the book left the project

18

Scrum

- **Sprint**
 - 1 – 4 weeks
 - Sprint Planning meeting
 - Sprint Review meeting
 - Sprint Retrospective
- **Artefacts**
 - Product backlog
 - Sprint backlog
 - Sprint burn down chart
- **Roles**
 - Scrum Master (facilitates, coaches on rules)
 - Team – multifunctional (design, develop, test, etc)
 - Product Owner – voice of customer
- **Daily Scrum - Stand-up meeting**
 - a. What have you done since yesterday
 - b. What are you planning today
 - c. Impediments limiting achieving your goals ?

80% of Scrum projects are ScrumBut

a lot of ritual

19

It's not the method (which method do you use ?)

time →

now

If the previous method didn't work, the next won't work either

20

What's missing in Agile ?

Ref Tom Gilb

Stakeholder Focus

- Real projects have dozens of stakeholders
 - Not just a customer in the room, not just a user with a use case or story

Results Focus

- It is not about *programming*, it is about making *systems* work, for *real* people

Systems Focus

- It is not about coding, but rather: reuse, data, hardware, training, motivation, sub-contracting, outsourcing, help lines, user documentation, user interfaces, security, etc.
- So, a *systems engineering* scope is necessary to deliver results
- Systems Engineering needs *quantified performance and quality objectives*

Planning

Ref Niels Malotaux

- Retrospectives within the Sprint
- Retrospectives of retrospectives
- Planning what not to do → *preflection*
- Overall planning and prediction: when will what be done

21

Essence of being really Lean & Agile

Delivering the right stuff, the right way, at the right time, as efficiently as possible

- Understanding what *real* Value means
- Quickly and easily adapting to all Stakeholders (but only the Customer pays !)
- Total system focus - software is only an aid - only provides value when it is *used successfully*
- Continuous elimination of Waste
 - Doing what contributes the most value
 - Not doing what doesn't contribute value
 - Prevention rather than repair - relentless problem solving - root cause analysis
 - Perfection - Quality is *cheaper*
- Predictability: Continuously being able to tell what will be done when (doing something about it)
- Delivering in small steps to real Stakeholders doing real things - minimizing the waste of incorrect perceptions, assumptions and implementations, optimizing productivity of Stakeholders
- Continuously optimizing what we do, how we do it, and how we organize things using PDCA
- Empowerment - everybody feeling responsible for the Result (Goal of a Project)
- Assertiveness - actively removing impediments, no excuses
- Understanding that it's not about tools: a lot is craft (you cannot 'implement' Lean nor Agile)
- Management facilitating and coaching the workers to do the right things the right way at the right time
- Management to be personally responsible for continuous improvement (not just change)

22

What is the Essence of Being Lean and Agile ?

Niels Malotaux

N R Malotaux
Consultancy

+31-30-228 88 68

niels@malotaux.nl

www.malotaux.nl

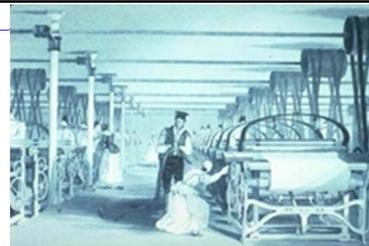
23

Pillars of the TPS

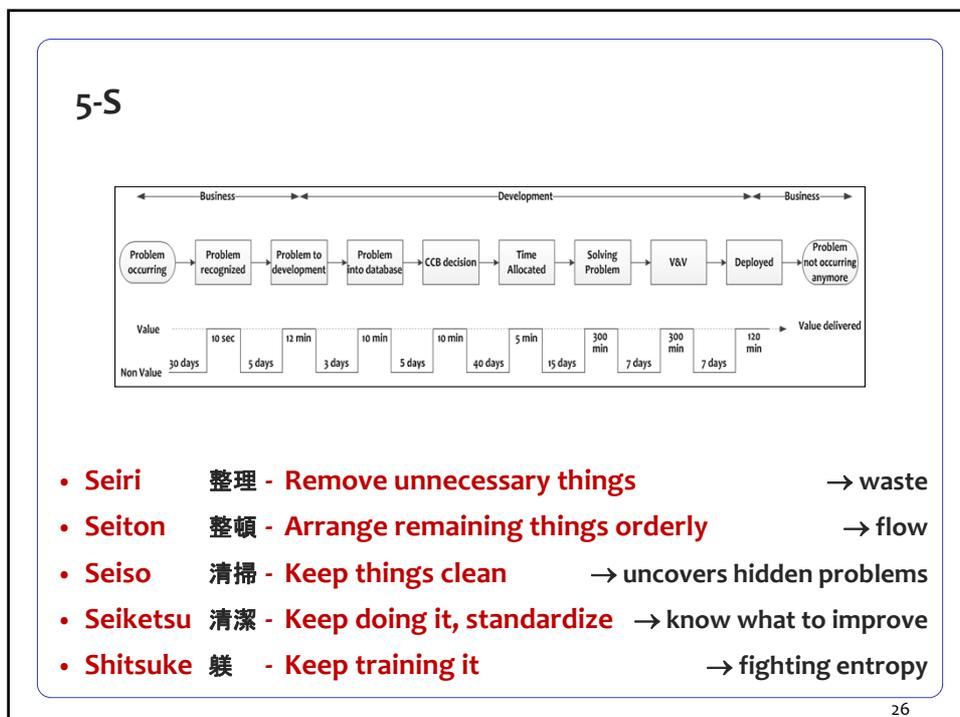
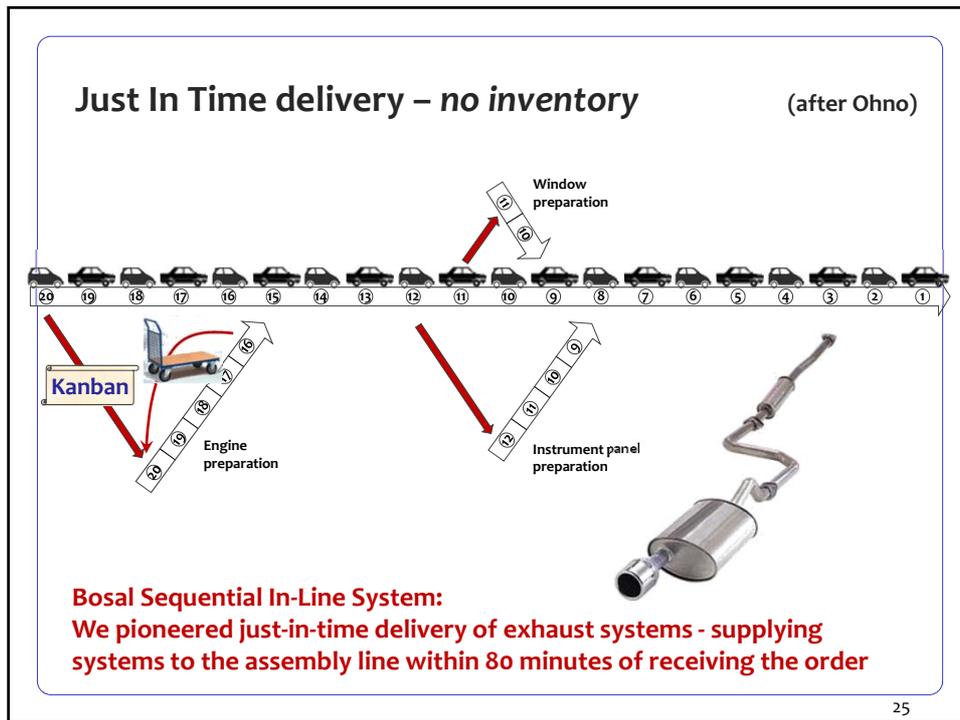
- **Just in Time**
 - No inventory
 - Doing the right things at the right time
- **Perfection**
 - Perfection is a condition for Just In Time to work
 - If a defect is found, stop the line, find root-cause, fix immediately
 - Continuous improvement of product, project and process
- **Autonomation**
 - The loom runs unattended until signalling it needs help

For development:

 - The development team runs unattended until signalling they need help (caused by an issue beyond their control)
 - Management **observes** the team and **facilitates** them to become ever more efficient, to **prevent** issues delaying them beyond the teams control – *Education, Empowerment and Responsibility* of people
 - If an issue does occur, management helps to **remove obstacles** quickly, making sure it doesn't happen again



24



The 3 Mu's to remove

Because it costs nothing, eliminating waste is one of the easiest ways for an organization to improve its operations

- **Muda** 無駄 - **Waste** → minimize waste
 - **Mura** - **Irregularities** → optimize flow
 - **Muri** 無理 - **Stress** → sustainable pace
-
- **90 per cent of all corporate problems can be solved using common sense and improving quality while reducing cost through the elimination of waste**
Imai: *Gemba Kaizen - A Commonsense Low-Cost Approach to Management*
 - **Plan-Do-Check-Act cycle was by far the most important thing we did in hindsight** (Tom Harada)

27

Lean Requirements

- **CRM system, to replace 'old' CRM system**
- **Original plan: 6 months and € 1M**
- **Spent 1.5 years and € 5M: business hasn't seen any result**
- **New Project Manager, new System Integrator**

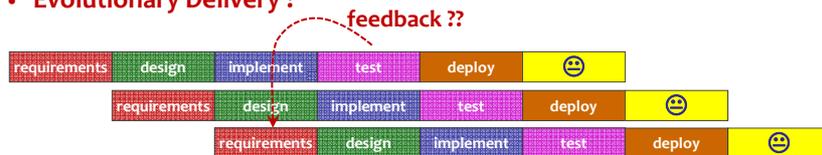
- **Who's project is this ?**

28

Delivery Requirements



- **Evolutionary Delivery ?**



- **Suggested Requirements:**

1. Within one week of any delivery, the business is not less efficient than before
2. The business decides whether they are satisfied

- **“Unacceptable” means supplier is saying:**

1. Within one week of a delivery, the business will be less efficient than before
2. The business will not be satisfied

Niels Malotaux

Niels Malotaux is an independent Project Coach and expert in optimizing project performance. He has some 35 years experience in designing electronic and software systems, at Delft University, in the Dutch Army, at Philips Electronics and 20 years leading his own systems design company. Since 1998 he devotes his expertise to helping projects to deliver Quality On Time: delivering what the customer needs, when he needs it, to enable customer success. Niels effectively teaches Evolutionary Project Management (Evo) Methods, Requirements Engineering, and Review and Inspection techniques. Since 2001, he taught and coached well over 100 projects in 25+ organizations in the Netherlands, Belgium, China, Germany, Ireland, India, Israel, Japan, Romania, South Africa and the US, which led to a wealth of experience in which approaches work better and which work less well in practice. He is a frequent speaker at conferences, and published several booklets, see www.malotaux.nl/Booklets .