

# Just Enough Measurement and Estimation

UKSMA Conference  
Thursday 15 October 2009

**Niels Malotaux**

N R Malotaux  
Consultancy

1

---

---

---

---

---

---

---

---

The Right Result at the Right Time

**Quality on Time**

- Do projects regularly deliver Quality on Time?
- How do you know?
- Why not?
- Is this normal?
- Can we do something about it?
  
- What is the Right Result?
- What is the Right Time?

2

---

---

---

---

---

---

---

---

## Optimum Project Result

- 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

Applies equally to  
developers, testers, and metrics evangelists

3

---

---

---

---

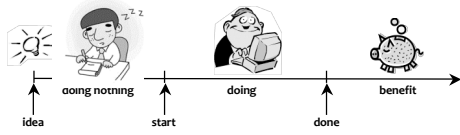
---

---

---

---

## Project ROI



### Return on Investment (ROI)

- + **Benefit of doing** - huge (otherwise other projects would be more rewarding)
- **Cost of doing** - project cost, usually minor compared with other costs
- **Cost of doing nothing** - every day we start later, we finish later
- **Cost of being late** - lost benefit
- **Loss of doing nothing at all** - diminishing benefit from legacy system

4

---

---

---

---

---

---

---

---

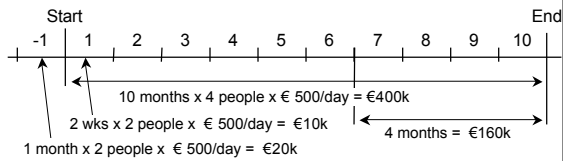
---

---

---

---

## The Cost of Time



- We can save 4 months by investing €200k → "That's too much !"
- It's a *nicer* solution - Let's do 2 weeks more research on the benefits
- What are the expected revenues when all is done? → €16M/yr<sup>(1.3M/mo)</sup>
- So 2 weeks extra doesn't cost €10k, but rather €16M/24 = €670k
- And saving 4 months brings €16M/3 = €5M extra
- Invest that €200k *NOW* and *don't waste time* !

5

---

---

---

---

---

---

---

---

---

---

---

---

- 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

## Evolutionary Project Management (Evo)

Zero Defects Attitude

- 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

### Evo planning

6

---

---

---

---

---

---

---

---

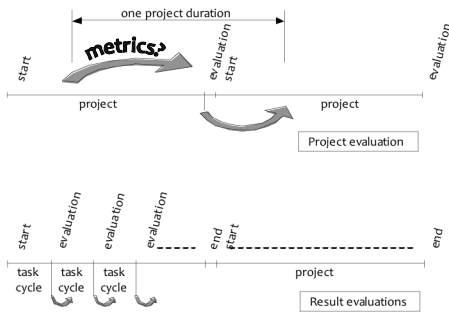
---

---

---

---

## Project evaluations



7

---

---

---

---

---

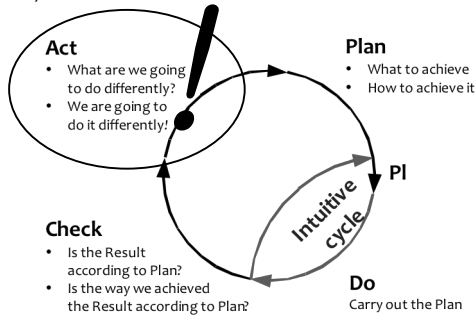
---

---

---

## The essential ingredient: the PDCA Cycle

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



8

---

---

---

---

---

---

---

---

## Every week we plan

- How much time do we have available
- 2/3 of available time is net plannable time
- What is most important to do
- Estimate effort needed to do these things
- Which most important things fit in the net available time (default 26 hr per week)
- What can, and are we going to do
- What are we *not* going to do

2/3 is default start value  
this value works well in development projects

Task a	2
Task b	5
Task c	3
Task d	6 do
Task e	1
Task f	4
Task g	5 26
Task h	4
Task j	3 do
Task k	1 not

9

---

---

---

---

---

---

---

---

## At the end of the week

**Immediate consumption of metrics**

- Was all planned work really done?

If a Task was not completed, we have to learn:

- Time spent but the work not done? → effort estimation problem  
Discuss what the causes may be and decide how to change your estimation habits
- Time not spent? → time management problem
  - Too much distraction
  - Too much time spent on other (poorly-estimated) Tasks
  - Too much time spent on unplanned Tasks

Discuss what the causes may be and decide how to improve (Check and Act)

- Conclude unfinished Tasks after *having dealt with the consequences* ← immediate metrics consumption /
  - Feed the disappointment of the "failure" into your intuition mechanism
  - Define new Tasks, with estimates, and put on the Candidate Task List
  - Declare the Task finished after having taken the consequences
- Continue with planning the Tasks for the next week

10

---

---

---

---

---

---

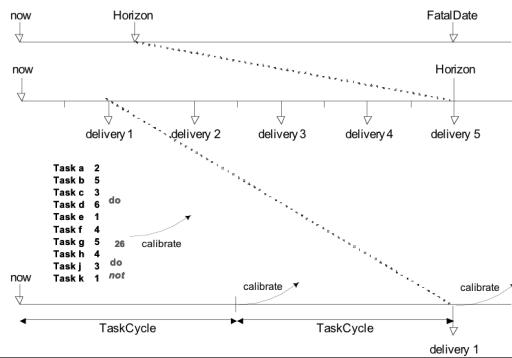
---

---

---

---

## Project Result to Tasks and back



11

---

---

---

---

---

---

---

---

---

---

Activity	Estimate	Real
Act1	Ae1	Ar1
Act2	Ae2	Ar2
Act3	Ae3	Ar3
Act4	Ae4	Ar4
Act5	Ae5	Ar5
Act6	Ae6	Ar6
Act7	Ae7	Ar7
Act8	Ae8	Ar8
Act9	Ae9	Ar9
Act10	Ae10	Ar10
Act11	Ae11	
Act12	Ae12	
Act13	Ae13	
Act14	Ae14	
Act15	Ae15	
Act16	Ae16	
Act17	Ae17	
Act18	Ae18	
Act19	Ae19	
Act20	Ae20	
Act21	Ae21	
Act...	Ae...	

## Calibration

ratio  $\Sigma Ar / \Sigma Ae$   
in the past

### Calibration Factor

$$\frac{\sum_{now-n}^{now-1} Ar}{\sum_{now-n}^{now-1} Ae}$$

predicted Value Still To Earn in the future

### Value Still To Earn

$$\text{Calibration Factor} * \sum_{now}^{then} Ae$$

← then2

12

---

---

---

---

---

---

---

---

---

---

### Predicting *what will be done when*

Line	Activity	Estim	Spent	Still to	Ratio real/es	Calibr factor	Calibr still to	Date done
1	Activity 1	2	2	0	1.0			
2	Activity 2	5	5	1	1.2	1.0	1	30 Mar 2009
3	Activity 3	1	3	0	3.0			
4	Activity 4	2	3	2	3.5	1.0	2	1 Apr 2009
5	Activity 5	5	4	1	1.0	1.0	1	2 Apr 2009
6	Activity 6	3				1.4	4.2	9 Apr 2009
7	Activity 7	1				1.4	1.4	10 Apr 2009
8	Activity 8	3				1.4	4.2	16 Apr 2009
↓	↓							
16	Activity 16	4				1.4	5.6	2 Jun 2009
17	Activity 17	5				1.4	7.0	11 Jun 2009
18	Activity 18	7				1.4	9.8	25 Jun 2009

13

---

---

---

---

---

---

---

---

---

---

---

---

### Metrics used

- Metrics for the project
  - Ratio Real time used / Estimated time
  - Calibration factor  
 $\Sigma \text{ Realized} / \Sigma \text{ Estimations}$
  - Predicted date of what will be done when  
Today plus Sum of Calibrated Estimations
  - Ratio plannable / unplannable hours (default: 2/1)
  - Available time, available budget (less is better)
  - Cost of one day of delay  
Cost-Of-Doing-Nothing + Lost Benefit + Project Cost
- Metrics for the product
  - Quantified requirements (ref Planguage - Tom Gilb)
  - Rate of improvement on quantified requirements

14

---

---

---

---

---

---

---

---

---

---

---

---

### Metrics techniques used

- Just-enough metrics (don't do unnecessary things)
- Maximizing Return-on-Investment and Value Delivered
- Consuming the metrics immediately
  - Not putting them in Databases
  - Using immediately for learning and improving
  - Feeding intuition to come up with better estimations
  - Preventing failure
- Time-boxing (not Feature-boxing)
  - Minimizes the need for tracking
- Calibration
  - Coarse metrics provide accurate predictions (Law of Large Numbers)
  - Moving Sense of Urgency from the end towards now

15

---

---

---

---

---

---

---

---

---

---

---

---

## Estimation techniques used

- **Just-enough estimation** (don't do unnecessary things)
  - Maximizing Return-on-Investment and Value Delivered
- **Changing from optimistic to realistic predictions**
  - Estimation of Tasks in the TaskCycle
  - Prediction what will be done when in TimeLine
- **0<sup>th</sup> order estimations** (ball-park figures)
  - For decision-making in Business Case and Design
- **Simple Delphi**
  - For estimating longer periods of time in TimeLine
  - For duration of several (15 or more) elements of work
- **Simpler Delphi**
  - Same, but for quicker insight
  - Recently added by practice
- **Doing something about it** (if we don't like what we see)
  - Taking the consequence
  - Saving time



16

---

---

---

---

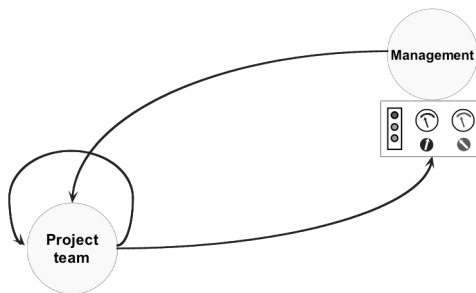
---

---

---

---

## Local loop principle



17

---

---

---

---

---

---

---

---

## Just Enough Measurement and Estimation

UKSMA Conference  
Thursday 15 October 2009

Niels Malotau

N R Malotau  
Consultancy

18

---

---

---

---

---

---

---

---

**More**

- [www.malotaux.nl/Booklets](http://www.malotaux.nl/Booklets)
  - 1 **Evolutionary Project Management Methods (2001)**  
Issues to solve, and first experience with the Evo Planning approach
  - 2 **How Quality is Assured by Evolutionary Methods (2004)**  
After a lot more experience: rather mature Evo Planning process
  - 3 **Optimizing the Contribution of Testing to Project Success (2005)**  
How Testing fits in
  - 3a **Optimizing Quality Assurance for Better Results (2005)**  
Same as Booklet 3, but for non-software projects
  - 4 **Controlling Project Risk by Design (2006)**  
How the Evo approach solves Risk by Design (by process)
  - 5 **TimeLine: How to Get and Keep Control over Longer Periods of Time (2007)**  
Replaced by Booklet 7
  - 6 **Human Behavior in Projects (2008)**  
Human Behavioural aspects of Projects
  - 7 **How to Achieve the Most Important Requirement (2008)**  
Planning of longer periods of time, what to do if you don't have enough time
  - 8 **Help ! We have a QA Problem ! (2009)**  
The story of this presentation
- [www.malotaux.nl/nrm/Insp](http://www.malotaux.nl/nrm/Insp)  
Inspection pages

---

---

---

---

---

---

---

---

---

---