Quality on Time

Delivering the Right Results at the Right Time

Niels Malotaux

Cobb’s Paradox:
We know why projects fail
we know how to prevent their failure
so why do they still fail?
Niels Malotaux

Independent Project and Organizational Coach
Expert in helping optimizing performance

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

Embedded Systems architect (electronics/firmware)

Project types
electronic products, firmware, software,
space, road, rail, telecom,
industrial control, parking system
Happy customers

- From one happy customer to another one

- We will be late and we don’t want to be late
- We cannot afford to be late
- When the money is used up, there is no more
source: www.tropomi.eu
• Very experienced Systems Engineers
• Using quantified requirements routinely
• 6 year pure waterfall project (imposed by ESA process)
• Don’t know exactly where they’ll end up
• One problem: They missed all deadlines (can you help us)
• 9 weeks later: They haven’t missed any deadline since
• “Sorry, we delivered 1 day early” (instead of expected 1 year late)
• Savings: at least 40 man-year (about £6M ?)
• How did they do that?
Convincing the Project Manager

- Don’t put me on the training budget
- Put me on the project budget

- We’ve been doing this kind of projects for 27 years
- We’re very good at it
- What do you think you can contribute to that?

- Do you have to deliver anything by the end of the week?
  - A status report
  - How much time do you need?
  - How much time do you have?
  - Does it fit?
  - What are we going to do about it?
Many sub-contractors across Europe
Issues

- Many interdependent Deadlines
- Many unforeseen issues, resulting in significant changes
- Delay declared unacceptable by customer
  - Launch date fixed
  - Money fixed
- Team overstressed, no clear focus on tasks at hand
- Everything 80% complete, nothing 100%
• 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
• 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

**Evolutionary Project Management (Evo)**

• Why
  • Business Case
  • Why we are going to improve *what*

• What
  • Requirements Engineering
  • *What* we are going to improve and *what not*
  • *How much* we will improve: quantification

• How much
  • Architecture and Design
  • Selecting the optimum compromise for the conflicting requirements

• How
  • Early Review & Inspection
  • Measuring quality while doing, learning to prevent doing the wrong things

• Are we done
  • Weekly TaskCycle
  • Short term planning
  • Optimizing estimation
  • Promising what we can achieve
  • Living up to our promises

• Efficiency of what we do
  • Bi-weekly DeliveryCycle
  • Optimizing the requirements and checking the assumptions
  • Soliciting feedback by delivering Real Results to *eagerly waiting* Stakeholders

• Evo Project Planning
• Effectiveness of what we do
  • TimeLine
  • Getting and keeping control of Time: Predicting the future

• What will happen and what will we do about it?
  • Feeding program/portfolio/resource management
The essential ingredient: the PDCA Cycle
(Shewhart Cycle - Deming Cycle - Plan-Do-Study-Act Cycle - Kaizen)

**Plan**
- What to achieve
- How to achieve it

**Do**
Carry out the Plan

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

**Act**
- What are we going to do differently?
- We are going to do it differently!
The requirements weren't the problem at all

- **Requirements for tropospheric O₃**
  - Ground-pixel size: 20 × 20 km² (threshold); 5 × 5 km² (target)
  - Uncertainty in column: altitude-dependent
  - Coverage: global
  - Frequency of observation: daily (threshold); multiple observations per day (target)

- **Requirements for stratospheric O₃**
  - Ground-pixel size: 40 × 40 km² (threshold); 20 × 20 km² (target)
  - Uncertainty in column: altitude-dependent
  - Coverage: global
  - Frequency of observation: daily (threshold); multiple observations per day (target)

- **Requirements for total O₃**
  - Ground-pixel size: 10 × 10 km² (threshold); 5 × 5 km² (target)
  - Uncertainty in column: 2%
  - Coverage: global
  - Frequency of observation: daily (threshold); multiple observations per day (target)
7 x 7 km pixels

source: www.tropomi.eu
• 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
• Weekly Task Cycle
  • Short term planning
  • Optimizing estimation
  • Promising what we can achieve
  • Living up to our promises
• Bi-weekly Delivery Cycle
  • Optimizing the requirements and checking the assumptions
  • Soliciting feedback by delivering Real Results to eagerly waiting Stakeholders
• Time Line
  • Getting and keeping control of Time: Predicting the future
  • Feeding program/portfolio/resource management

Evolutionary Project Management (Evo)

Right Result
Quality On Time
Right Time

Zero Defects Attitude (27 June)
Tasks feed Deliveries

this week

Timeline
Weekly TaskCycle

- Are we *doing* the right things, in the right order, to the right level of detail for now

- Optimizing estimation, planning, and tracking abilities to better predict the future

- Select highest priority tasks, never do any lower priority tasks, never do undefined tasks

- There are only about 26 plannable hours in a week (2/3)

- In the remaining time: do whatever else you have to do

- Tasks are always done, 100% done
Weekly Plan

- How much time do I have available
- \( \frac{2}{3} \) 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 am I going to do
- What am I not going to do

\[
\begin{array}{c|c}
\text{Task} & \text{Effort} \\
\hline
\text{Task}_a & 2 \\
\text{Task}_b & 5 \\
\text{Task}_c & 3 \\
\text{Task}_d & 6 \\
\text{Task}_e & 1 \\
\text{Task}_f & 4 \\
\text{Task}_g & 5 \\
\text{Task}_h & 4 \\
\text{Task}_i & 3 \\
\text{Task}_j & 1 \\
\hline
\end{array}
\]

2/3 of available time is default start value
this value works well in development projects
Weekly planning

- **Individual preparation**
  - Conclude current tasks
  - What to do next
  - Estimations
  - How much time available

- **Modulation with / coaching by coach (1-on-1)**
  - Status (all tasks done, completely done, not to think about it any more ?)
  - Priority check (are these really the most important things ?)
  - Feasibility (will it be done by the end of the week ?)
  - Commitment and decision

**Synchronization with group (team meeting)**

- Formal confirmation (this is what we plan to do)
- Concurrency (do we have to synchronize ?)
- Learning
- Helping
- Socializing
Awful schedule pressure!

- Meeting with sub-contractors in three weeks
- Many documents to review
- Impossible deadline

- How many documents to review?
- How much time per document?
- How much time available?

- Some suggestions …
- Result: well reviewed, great meeting, everyone satisfied
TimeLine

How do we know that we do, and get what is needed, when it’s needed?

- Better 80% 100% done, than 100% 80% done
- Let it be the most important 80%
Even more important: 
*Starting Deadlines*

- **Starting deadline**
  - Last day to start to make the finish deadline
  - Every day we start later, we will end later
What do we do if we see we won’t make it on time?

- Value Still to Earn
- Time Still Available

If the match is over, we cannot score a goal
Deceptive options

- Hoping for the best (fatalistic)
- Going for it (macho)
- Working overtime (fooling ourselves)
- Moving the deadline
  - Parkinson’s Law
    - Work expands to fill the time for its completion
  - Student Syndrome
    - Starting as late as possible, only when the pressure of the deadline is really felt

Intuition often guides us in the wrong direction
Adding people

Brooks’ Law (1975)
Adding people to a late project makes it later

Economic optimum?

intuition
people x time = constant
Man-Month Myth

lower cost
reality (Putnam)
shorter time

nine mothers area

project duration

number of people
Saving time

We don’t have enough time, but we can save time without negatively affecting the Result!

- **Efficiency in what (why, for whom) we do** - doing the right things
  - *Not doing what later proves to be superfluous*

- **Efficiency in how we do it** - doing things differently
  - **The product**
    - Using proper and most efficient solution, instead of the solution we always used
  - **The project**
    - Doing the same in less time, instead of immediately doing it the way we always did
  - **Continuous improvement and prevention processes**
    - Constantly learning doing things better and overcoming bad tendencies

- **Efficiency in when we do it** - right time, in the right order

- **TimeBoxing** - much more efficient than FeatureBoxing
Did it work for this project?

- 2 months needed to get the process in full swing
- All Engineering docs in PDR and CDR data packages on time
- Stress level in team greatly reduced
- More supervisory work for Systems Engineer - can effectively handle up to 8 people
- People not in the Evo swing lag behind
- So, we need everyone to follow
- Good enough to become company standard? I say YES
Why did it still take so long before actual launch?

- The launch was delayed caused by issues you cannot predict even with Evo:
  - The launch SW from the Ukraine, bought by ESA 5 years ago was to be used in Russia. Incomprehensibly, that was a bit more difficult than it was 5 years earlier
  - By now the problems seem to have been solved and the launch is planned for March/April …
  - New Deadline: August… (Finally launched 13th October 2017)

- Coincidentally I just today introduced our Evo way of working to a new team member of our current project (mapping the large-scale structure of the Universe over a cosmic time covering the last 10 billion years)

- I’m curious to find out how quickly she’ll really get the idea
Quality on Time
Delivering the Right Results at the Right Time

Niels Malotaux

Cobb’s Paradox:
We know why projects fail
we know how to prevent their failure
so why do they still fail?
www.malotaux.eu/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, except for the step-by-step TimeLine procedure

6 Human Behavior in Projects (APCOSE 2008)
   Human Behavioral aspects of Projects

7 How to Achieve the Most Important Requirement (2008)
   Planning of longer periods of time, what to do if you don’t have enough time

8 Help ! We have a QA Problem !(2009)
   Use of TimeLine technique: How we solved a 6 month backlog in 9 weeks

9 Predictable Projects (2012) - How to deliver the Right Results at the Right Time

RS Measurable Value with Agile (Ryan Shriver - 2009)
   Use of Evo Requirements and Prioritizing principles

www.malotaux.eu/inspections
   Inspection pages