What has an Architect to do with planning ?

Niels Malotaux

N R Malotaux
Consultancy

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

See www.malotaux.nl/conferences

Niels Malotaux

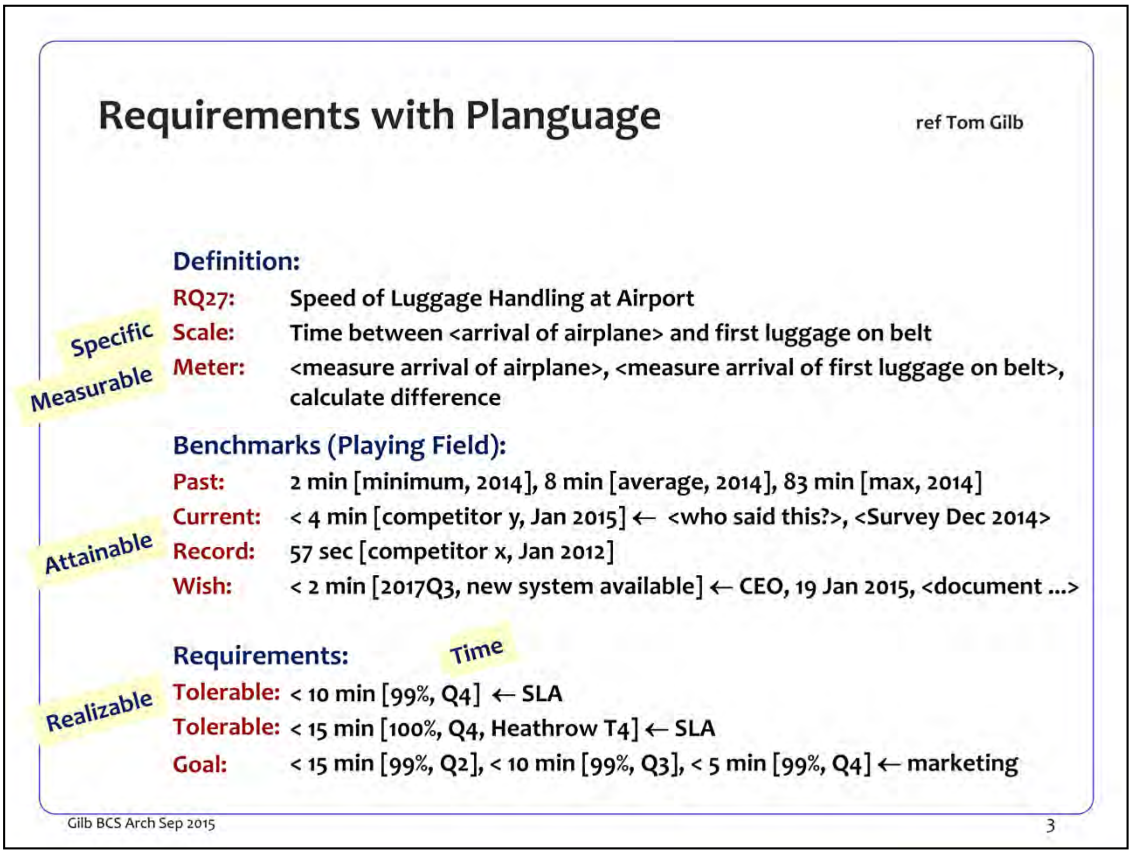
Graduated **Electronics** at Delft University of Technology in **1974**

Army service at the Dutch Laboratory for Electronic Developments for the Armed Forces, designing computer systems

Philips Electronics – Application support for microcomputer systems design

Malotaux - Electronic Systems Design - : developing electronic systems for clients products

Now: **N R Malotaux - Consultancy**



This is my standard simple Planguage example. It shows a bit more (Current, Record, Wish) than the bare minimum of a performance requirement.

Someone said that the ‘requirements should be SMART’. Of course Planguage provides SMART requirements.

See www.malotaux.nl/planguage for an explanation of the elements shown.

Impact Estimation principle

How much % of what we want to achieve do we achieve by this solution

At what cost ?

Possible solutions to achieve it

Could we get all, within the budgets of time and cost ?

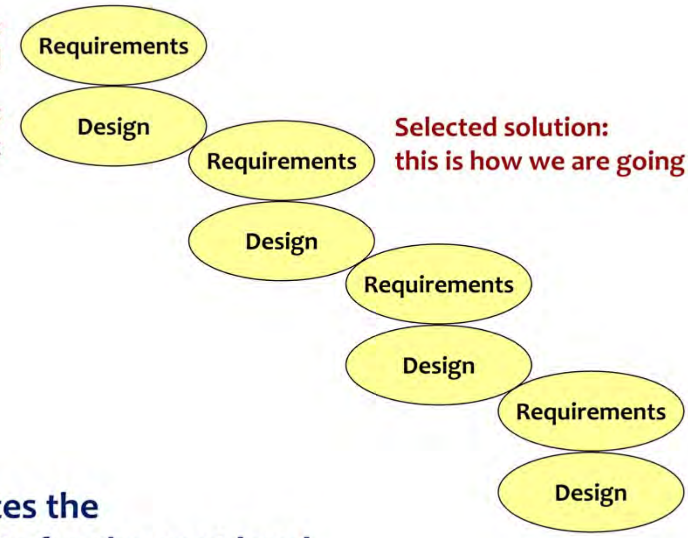
		Design Idea #1	Design Idea #2	Design Idea #3	Total Impact
What to achieve	Objectives	Impact on Objective	Impact on Objective	Impact on Objective	Sum of Impacts on Objectives
Cost to achieve it	Resources Time Money	Impact on Resources	Impact on Resources	Impact on Resources	Sum of Impact on Resources
Return on Investment	Benefits to Cost Ratio	$\frac{\text{Benefits}}{\text{Cost}}$	$\frac{\text{Benefits}}{\text{Cost}}$	$\frac{\text{Benefits}}{\text{Cost}}$	

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



Design creates the
Requirements for the next level

Kai Gilb's 3 levels of organisation

BUSINESS GOALS	Training Costs	User Speed
Profit	-10%	40% *
Market Share	50%	10%
Resources	20% **	10%

* = est. % of Profit value goal level delivered by User Speed goal

** = est. % of budgeted resources needed to implement Training Costs solution

STAKEHOLDER GOALS	Intuitiveness	Intelligibility
Training Costs	-10%	50 %
User Speed	10 %	10%
Resources	2 %	5 %

Technical Design

Technical Requirements	3D Interface	Content Training
Intuitiveness	-10%	40%
Intelligibility	50%	80 %
Resources	1 %	2 %

Implementation				
Implementation	Estimated resources	Activities to achieve		
3D Interface	1% → … hr	task 1	task 2	…
Content Training	2% → … hr	task 11	task12	…

3D Interface		hrs
-	task 1	3
-	task 2	5
	…	
Content Training		
-	task 11	2
-	task 12	7
-	…	

Evolutionary Project Management (Evo)

- **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 *for whom 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

Zero Defects Attitude

Right product

Evo Planning

- **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

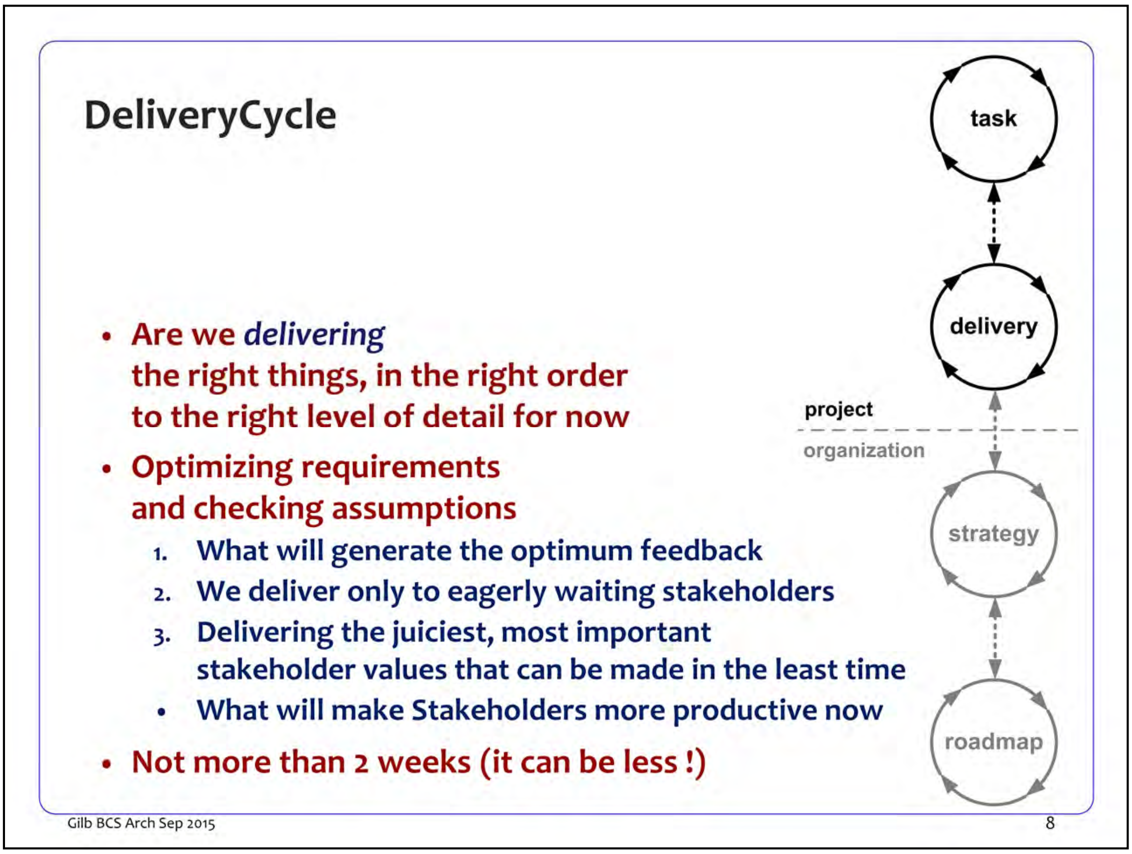
Efficiency of what we do

Effectiveness of what we do

Right time

What will happen and what will we do about it ?

See www.malotaux.nl/processes



In the **Delivery cycle** we are **Optimizing requirements and checking assumptions,** while **delivering real value to stakeholders**

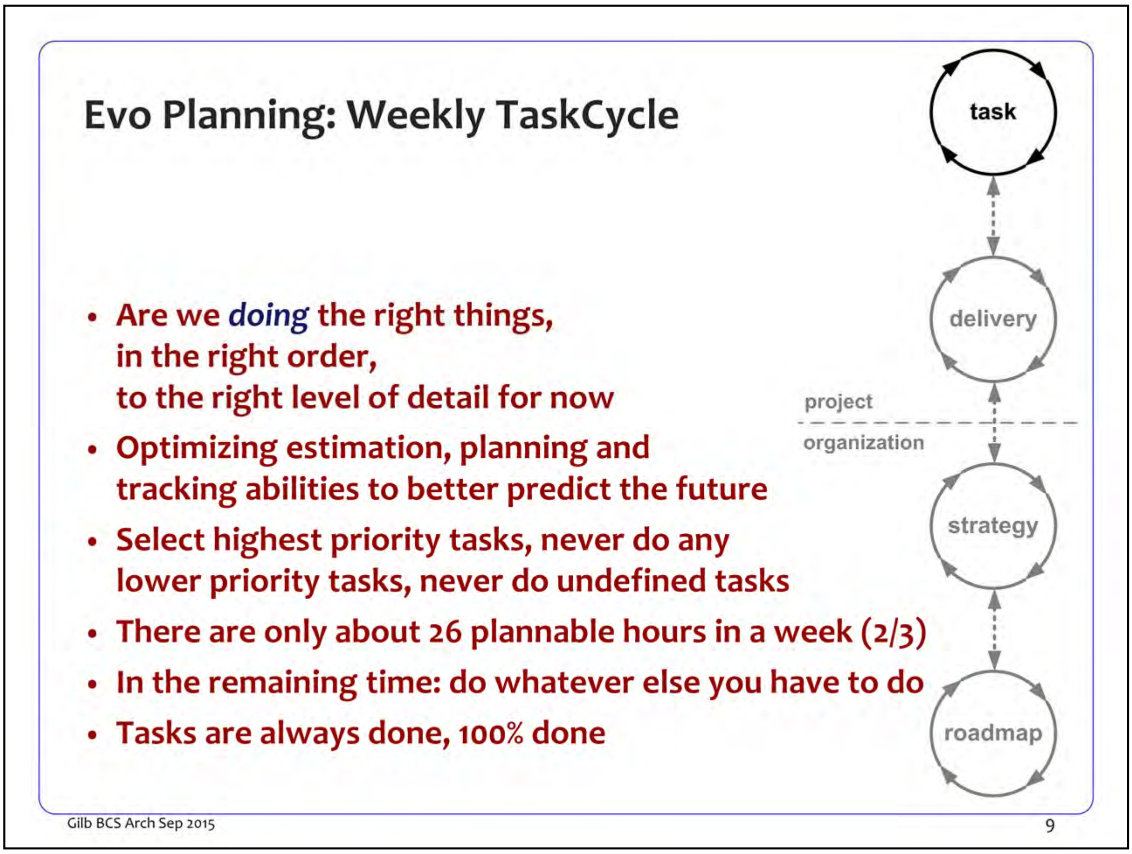
Are we DELIVERING the right things, in the right order, to the right level of detail?

A **Delivery Cycle** takes usually not more than **two weeks.**

Cutting the work and deliveries into small pieces is one of the difficult issues in Evo. People, however, can learn this in a very short time, if coached well.

Most Project Managers say “Nice story. But MY project is different, so in MY project this is not possible”. We found that it is ALWAYS possible. It takes not more than an hour to make people find out themselves.

There are more cycles on the organization level (strategy, roadmap), but these are beyond the scope of this talk.



The basic working cycle in Evo is the task-cycle. Normally we work in weekly task-cycles, in which all we agree to will be finished.

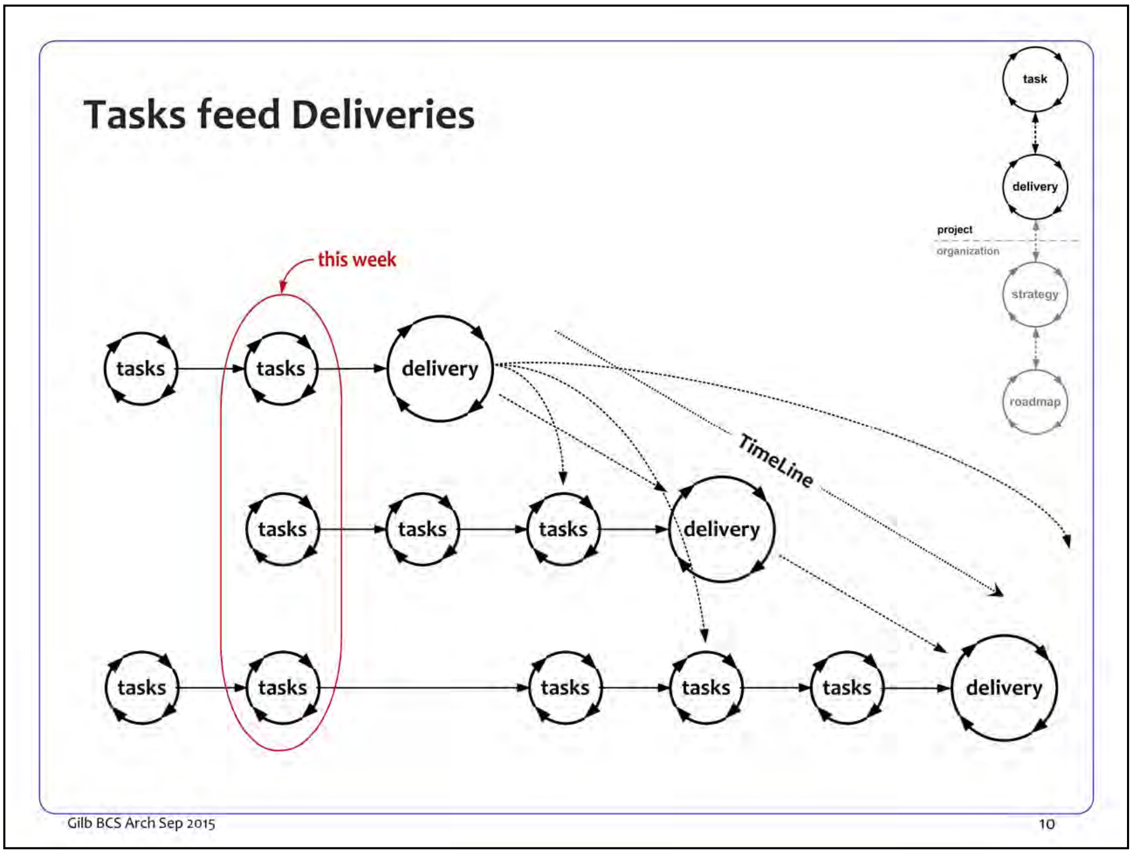
We learn estimation within a few weeks, which allows us to do better planning and prediction of the future.

We select the highest priority tasks only, we should never work on any lower priority tasks (by definition)

Are we *DOING* the right things, in the right order, to the right level of detail

• There are only about 26 real effort hours in a week
We select about two-thirds of the time for our project work. Be realistic: we have to go to the toilet, drink coffee, help each other, talk to each other, have meetings, etc. So, if we think we can do more real effort than about 2/3, we are fooling ourselves and we will fail anyway. Being realistic about this, allows us to succeed. People enjoy success. Success breeds motivation. Motivation is the Motor of Productivity. So, by being realistic, we are more productive. Practice proves this!

• Tasks are always done, 100% done
We use a time-box, rather than a feature-box.
This is to avoid the 90% syndrome: When 90% is done, people start the next 90%. Only by getting things 100% done, *no need to think about it any more*, we know what we have finished and can we learn to better estimate future work.



Every task-cycle, we are working on tasks for the current delivery. Some deliveries need preparation of more than just two weeks. So, we may also be working on tasks that make future deliveries possible.

Feedback from deliveries may generate new tasks, or change envisaged tasks.

Weekly 3-Step Process

- **Individual preparation**
  - Conclude current tasks
  - What to do next
  - Estimations
  - How much time available
- **Modulation with / coaching by Project Management (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

See www.malotaux.nl/threestepprocedure

Weekly TaskCycle

Task$_a$	2	do
Task$_b$	5	
Task$_c$	3	
Task$_d$	6	
Task$_e$	1	
Task$_f$	4	
Task$_g$	5	26
Task$_h$	4	do *not*
Task$_j$	3	
Task$_k$	1	

- **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**

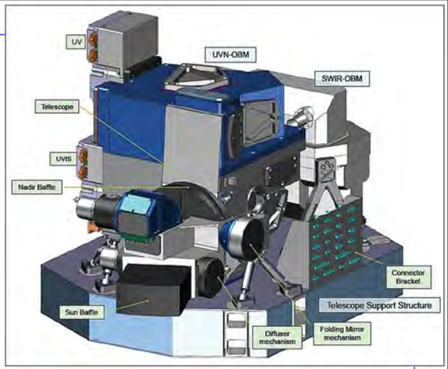
2/3 is default start value
this value works well in development projects

See also www.malotaux.nl/weeklyplanning

Why is this important for Architects ?

- Half (±30%) of what people do in projects later proves not having been necessary
- During the TaskCycle planning we can very efficiently see
  - What people think they're going to do
  - Make sure they're going to work on the most important things
  - Not on unnecessary things
  - In line with our architecture and design
  - Leading most efficiently to the goal of the delivery
- We'll see two cases where the architect led the project to success in record time

Earth Observation Satellite



- Very experienced Systems Engineers
- They use quantified requirements routinely
- They don't know exactly where they'll end up
- 10 year pure waterfall project (imposed by ESA)
- Only problem: They missed all deadlines
- 9 weeks later: They haven't missed any deadline since
- Recently: delivered 1 day early (instead of 1 year late)
- Savings: some 40 man-year
- How did they do that ?

I was asked to coach a project where very clever Systems Engineers we developing an earth observation instrument to be launched this year.

These people said to me: "Niels. We are doing this kind of work already for 27 years. We're very good at it. What do you think you can add to that?", which of course was a relevant question.

Well, I didn't have to tell them much about Planguage because seasoned Systems Engineers know how to quantify requirements (example on next slide).

But there was one thing they still hadn't mastered: The missed every deadline and were complaining about impossible deadlines.

9 weeks later, and ever since they didn't miss any deadline.

I'll explain how they did this.

A few weeks ago I heard that they delivered one day before the expected deadline.

Normally, all their projects take at least one year more than the expected deadline.

Savings: 40 man year.

Summary of requirements for ozone measurements

- 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)

Example of requirements I found on Internet (but cannot find anymore ☹).

Awful schedule pressure !

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

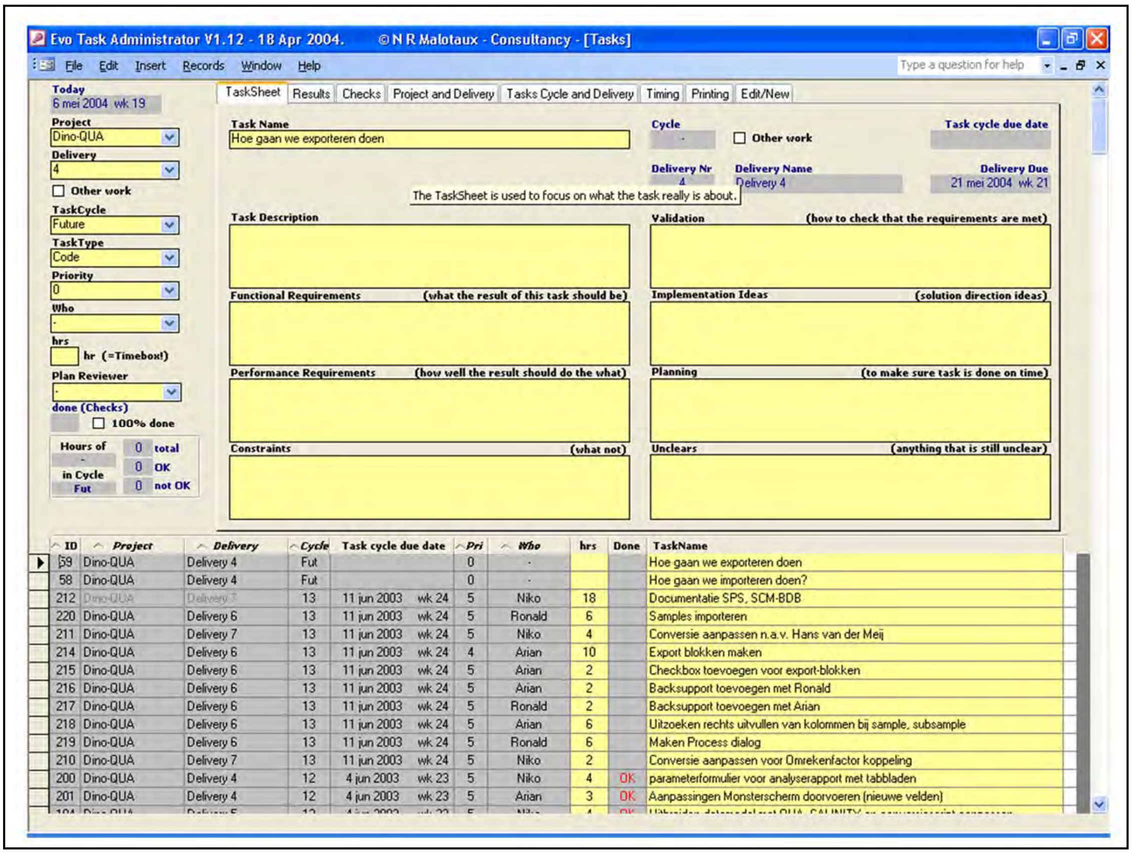
- How many documents to review ?
- How much time per document ?

	per doc	hr
4 heavy	15	60
3 easy	2	6
	total	66
other work		33
	total	99

available	2 x 26	52

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

Schedule pressure? What schedule pressure?



Download www.malotaux.nl/doc.php?id=20

Any Deadlines ?

now

FatalDate

Earned Value → Value Still to Earn

needed time << available time : OK for now

needed time = available time : not OK

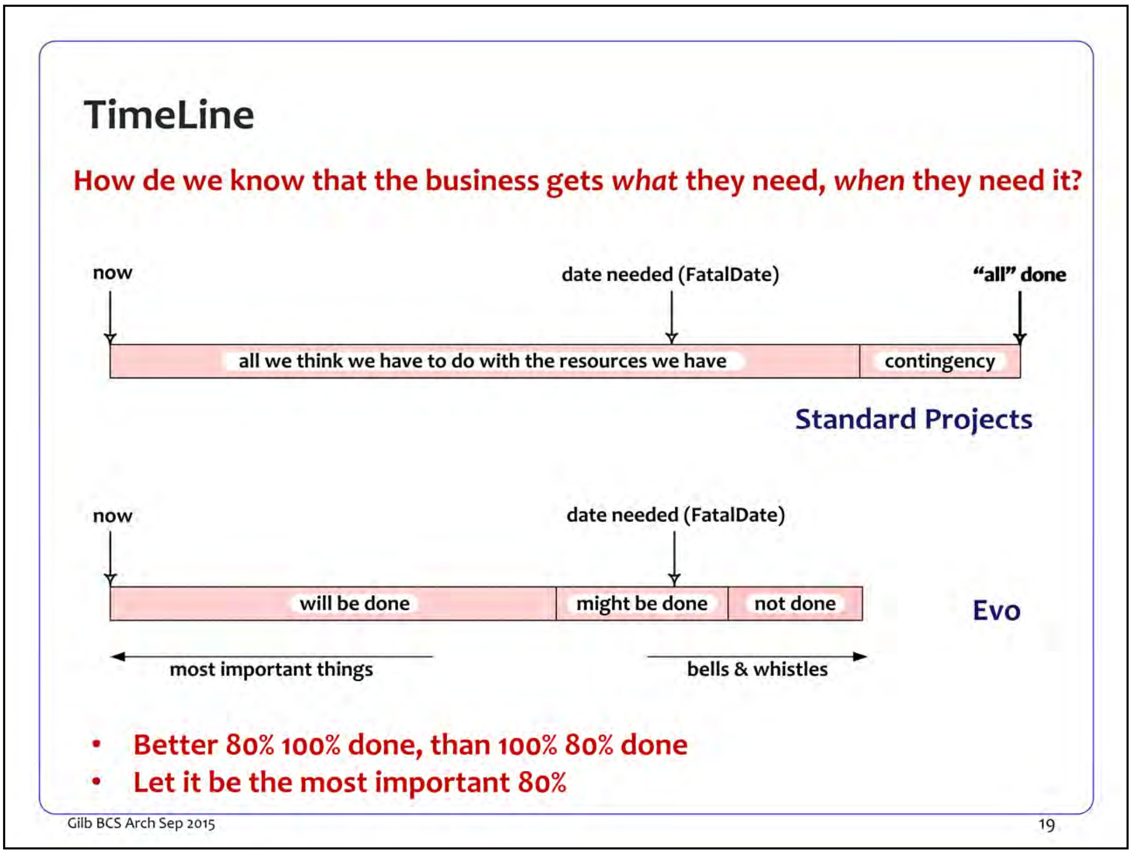
needed time > available time : not OK

- **Value Still to Earn**

versus

- **Time Still Available**

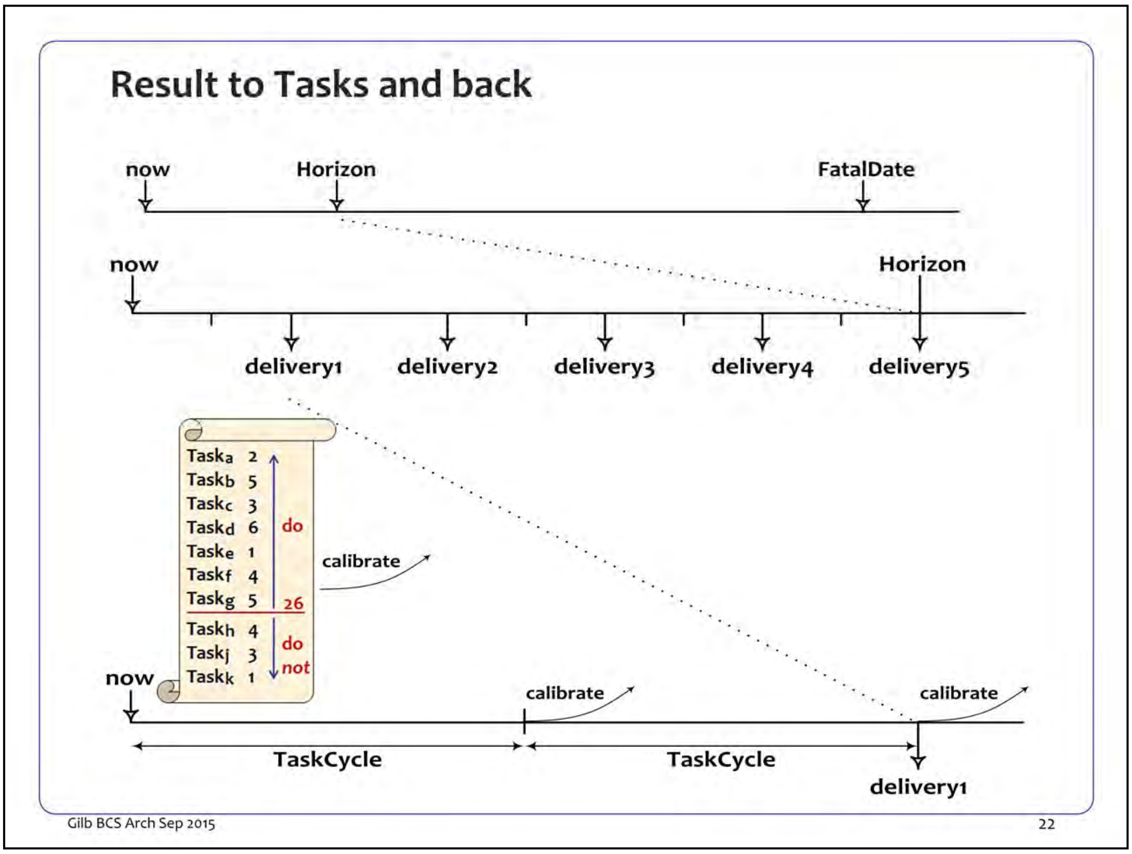
If the match is over, you cannot score a goal



See www.malotaux.nl/timeline

Page removed for confidentiality

Page removed for confidentiality



See www.malotaux.nl/timeline

Developing a new oscilloscope

- 4 teams of 10 people, 8 more people in Bangalore
- Introduced first in one team
- Other teams followed once convinced
- One team lagged because of 'micro-management'

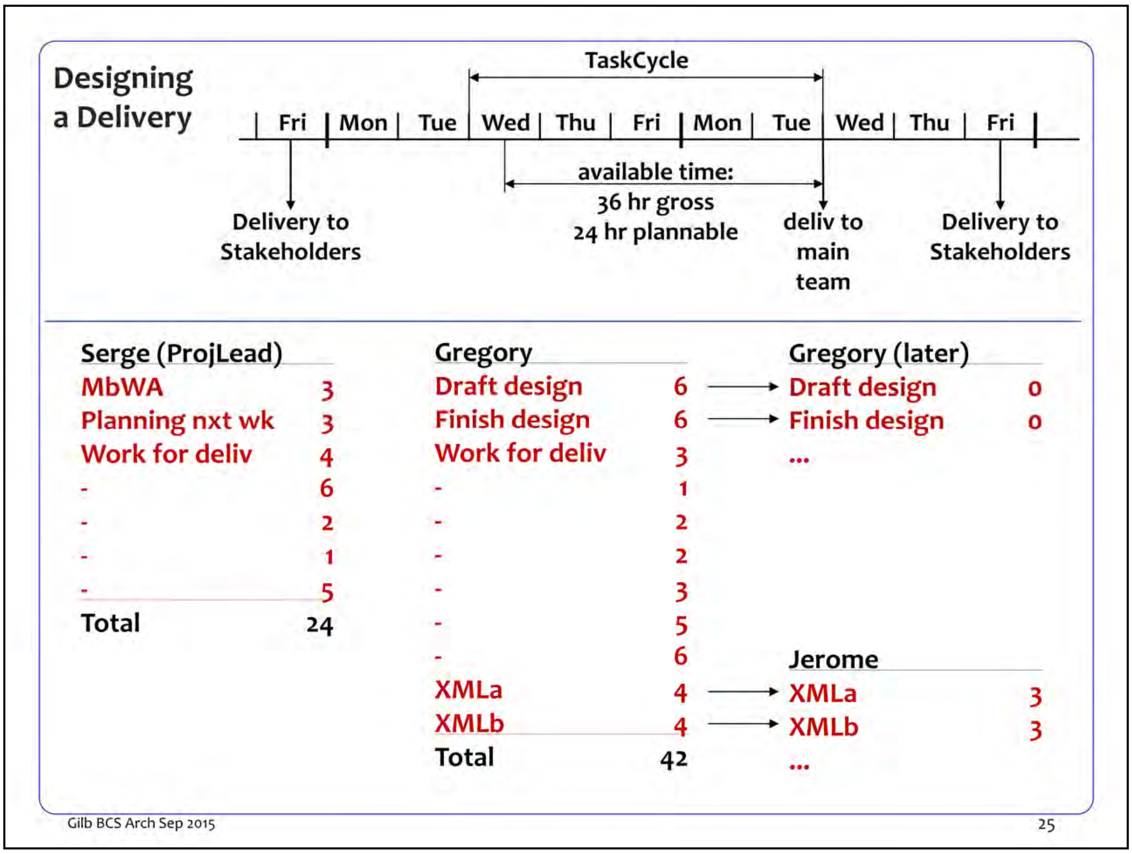
- Even if we would drop all you suggested, the 1-on-1's will be kept, because so powerful:
  - Before we did something and afterwards found out it wasn't what it should be
  - Now we find out before we do it what it actually should be

Conclusion

- Schedule accuracy for this platform development was 50% better than the program average (as measured by program schedule overrun) over the last 5 years
- This product was the fastest time-to-market with the highest quality at introduction of any platform in our group in more than 10 years
- The team also won a prestigious Team Award as part of the company’s Technical Excellence recognition program

http://www.malotaux.nl/doc.php?id=19 chapter 4.7.1, page 70

See www.malotaux.nl/doc.php?id=19 chapter 4.7.1, page 70



Click: Start a TaskCycle by drawing a line. Put ticks on the line for days or weeks, depending on the scale. Weeks for several deliveries. Days to detail actual work.

Click: In this case we show the days for the TaskCycle we are planning.

Click: We add the project Deliveries (In this project Friday 11:00)

Click: The TaskCycle we are planning is for a sub-team, delivering a Result to the main-team for integration into the Main-team Delivery. We decided that the sub-team will deliver end of Tuesday, so that their result is available to the main-team on Wednesday morning.

Click: So, the sub-team DeliveryCycle ends Tuesday evening. Therefore the TaskCycle for the sub-team starts Wednesday morning.

Click: This is a team starting with Evo planning, so we take a bit more time for Eco coaching, so we decide the gross available time for the team to be 36 hrs. If people work full time, this means 24 hr plannable time.

Click: We checked the Tasks already prepared by Serge and Gregory. Serge is Project Lead for 4 more people in his sub-team, so he needs to plan time for Management by Walking Around (MbWA), and planning the next week for the team. Then he planned several work tasks. Project Management easily deteriorates when the PM is also working. Rule: First manage. If you still have time left: manage better. If you really have time left, you may do some work. In order to get the Delivery on time Serge still had to do a lot of work himself, knowing that he'd have to catch up on management the next week. His 24 hrs just fit.

Gregory had planned 42 hours of "necessary" work. This didn't fit the 24 hrs, so even if he'd try, he wouldn't succeed and the Delivery would fail. Deliveries may not fail.

Click: First we found out that the Design Tasks were not for this Delivery and that the information to do the design wasn't even sufficiently available from the main team. So we moved these Tasks to the future. Note that the team automatically puts Repair Tasks on the planning after a delivery, with time = 0. The idea is that based on the result of the Delivery, an estimate for the repair time will be made.

Click: The team embraced the ZeroDefects idea by stating that from now on their goal would be to keep the Repair Task at 0 hr. This attitude alone already prevents defects and the team wanted to reap this benefit immediately.

Click: Now we were at 30 hr. Still 6 hours to move. The past week, Gregory had been struggling with XML, actually spending more time than he liked on it. Jerome knew XML better and had some time to spare. So we decided to move the two XML tasks to Jerome. Because Jerome would be quicker with XML, he said he'd need 3 hrs per Task to do the work. This didn't relieve Gregory completely from these Tasks, because he would have to explain the Tasks to Jerome and integrate Jeromes result in his own result. So Gregory planned 1 hr each for both Tasks, in stead of the original 4 hr each. Now he had 24 planned hrs and he accepted the responsibility to deliver.

One week later, Serge and Gregory delivered. No stress. They even had a few hours left to implement something extra that they decided was actually forgotten in the design. (Tomorrow, Friday, Serge is going to present his delivery in Eindhoven. I'll be there to see whether he'll really make a smile on the Stakeholders faces)

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**

I think this Gilb-quote is important to remember.

www.malotaux.nl/booklets

More

1 **Evolutionary Project Management Methods (2001)**
Issues to solve, and first experience with the Evo Planning approach

2 **How Quality is Assured by Evolutionary Methods (2004)**
After a lot more experience: rather mature Evo Planning process

3 **Optimizing the Contribution of Testing to Project Success (2005)**
How Testing fits in

3a **Optimizing Quality Assurance for Better Results (2005)**
Same as Booklet 3, but for non-software projects

4 **Controlling Project Risk by Design (2006)**
How the Evo approach solves Risk by Design (by process)

5 **TimeLine: How to Get and Keep Control over Longer Periods of Time (2007)**
Replaced by Booklet 7, except for the step-by-step TimeLine procedure

6 **Human Behaviour in Projects (APCOSE 2008)**
Human Behavioural aspects of Projects

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

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

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

www.malotaux.nl/inspections

Inspection pages

What has an Architect to do with planning ?

www.malotaux.nl/?id=conferences

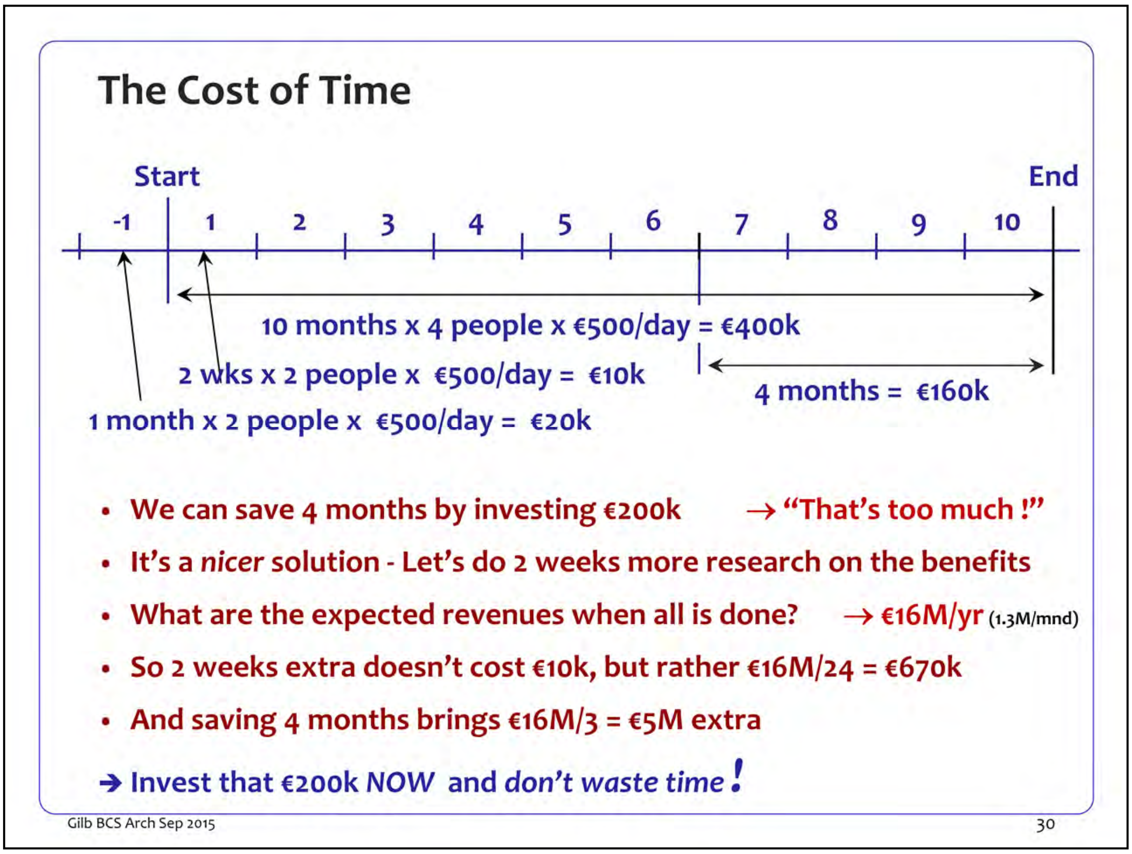
Niels Malotaux

N R Malotaux
Consultancy

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

Making best use of limited available time

- If the work is done, the time is already spent
- If we still have to do the work, we can decide
  - What is really important
  - What is less important
  - What we must do
  - What we can do
  - What we are going to do
  - What we are not going to do
- Therefore we plan first, instead of finding out later
- We cannot work in history



See www.malotaux.nl/importanceoftime

Do we learn from what happened ?

Insanity is doing the same things over and over again and hoping the outcome to be different ***(let alone better** - Niels**)***

Albert Einstein 1879-1955, Benjamin Franklin 1706-1790, it seems Franklin was first

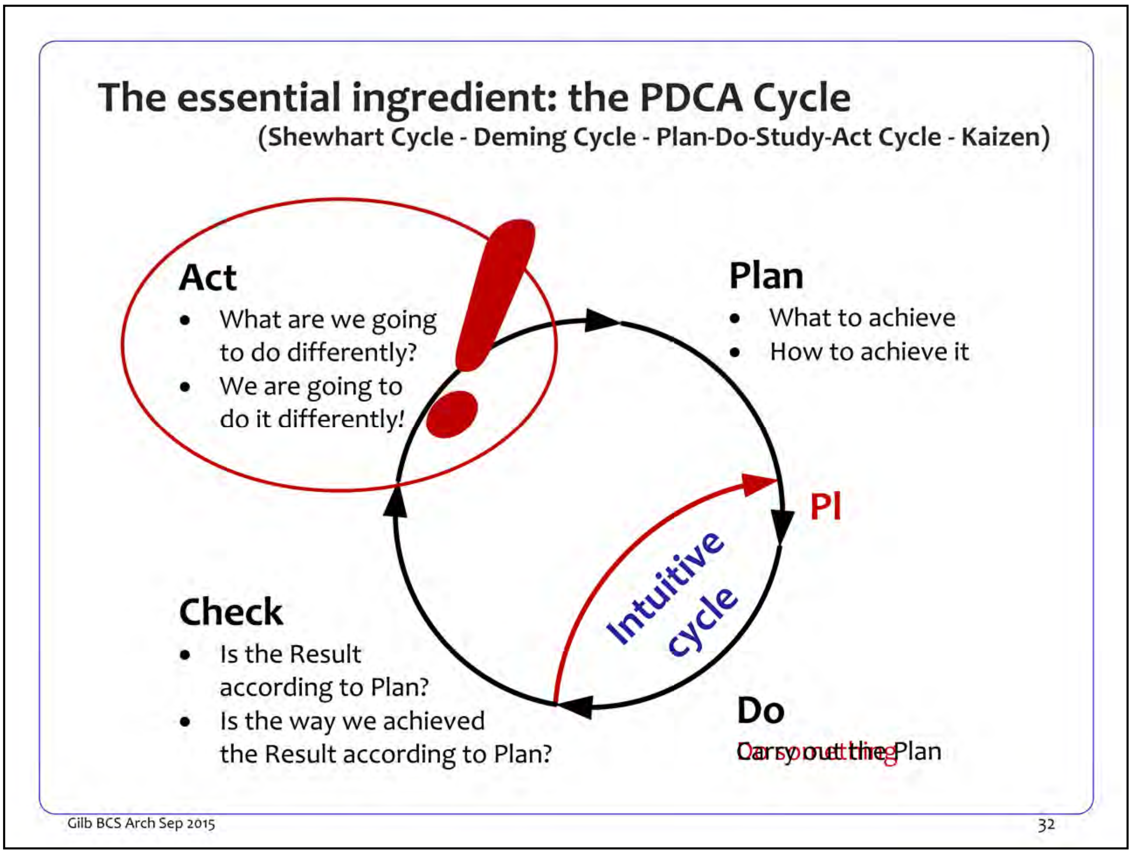
Only if we *change* our way of working, the result may be different

- **Hindsight is easy, but reactive**
- **Foresight is less easy, but proactive**
- **Reflection is for hindsight and learning**
- **Preflection is for foresight and prevention**

Only with *prevention* we can save precious time

This is used in the Deming or Plan-Do-Check-Act cycle

See www.malotaux.nl/preflection

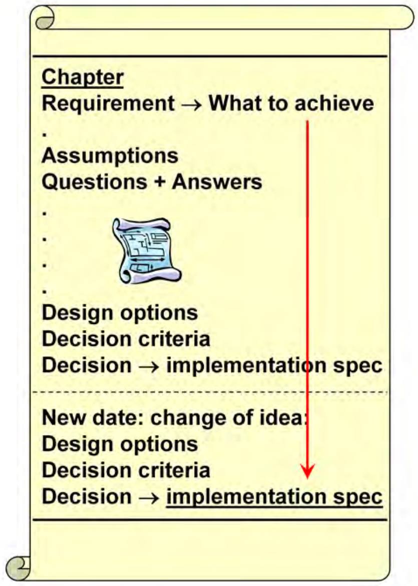


See www.malotaux.nl/pdca

DesignLog

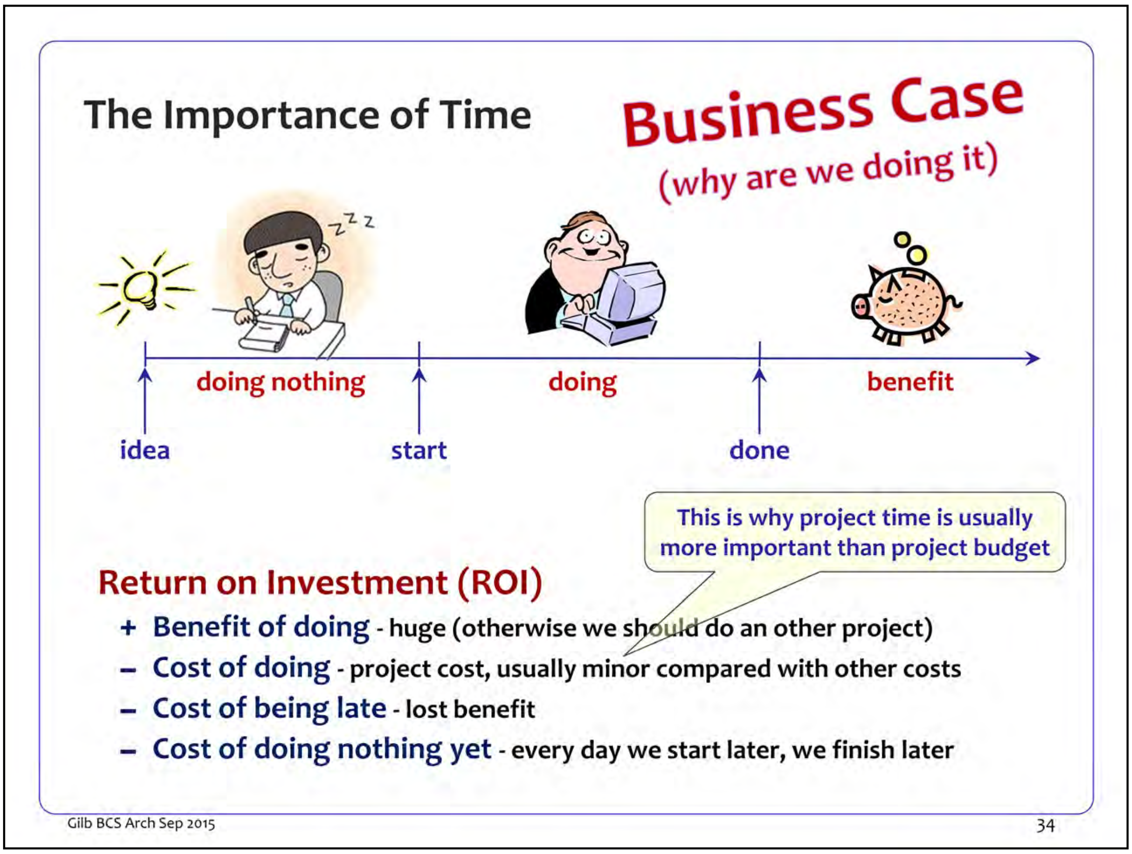
(project level)

- 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



Evo - Keio-SDM - Sep 2013 33

http://www.malotaux.nl/designlog



See www.malotaux.nl/importanceoftime

Can you do this yourself ?

- What is the cost of one day of (unnecessary) delay ?
- What is the benefit of your project
  - for your customer (who's waiting for it ?)
  - for you - your organization
- What is the cost of your project per day ?
- What is your cost per day ?
  - Note: that's not what you get !

- If you don't know these things, how can you make decisions ?
- It usually doesn't take more than a few minutes
- Ballpark figures are good enough to make decisions

Effort and Lead Time

- Days estimation → lead time (calendar time)
- Hours estimation → effort

- Effort variations and lead time variations have different causes
- Treat them differently and keep them separate
  - Effort: complexity
  - Lead Time: time-management
    - (effort / lead-time ratio)

See www.malotaux.nl/diffeffortleadtime