Oak Café – 29st March 2023

Hosted by: Niels Malotaux

How to be on time (with system development projects)

How Space Systems Engineers learnt to meet all deadlines

or

How to be on time

Niels Malotaux

Cobb's Paradox (1995):
We know why projects fail
We know how to prevent their failure
So why do they still fail ?

niels@malotaux.eu
www.malotaux.eu/conferences

Niels Malotaux

- Independent Engineering and Team Coach
- Expert in helping projects and organizations to quickly become
  - More effective - doing the right things better
  - More efficient - doing the right things better in less time
  - More 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

Delivering
Quality On Time
the Right Results
at the Right Time

Happy customers

- From one happy customer to another one

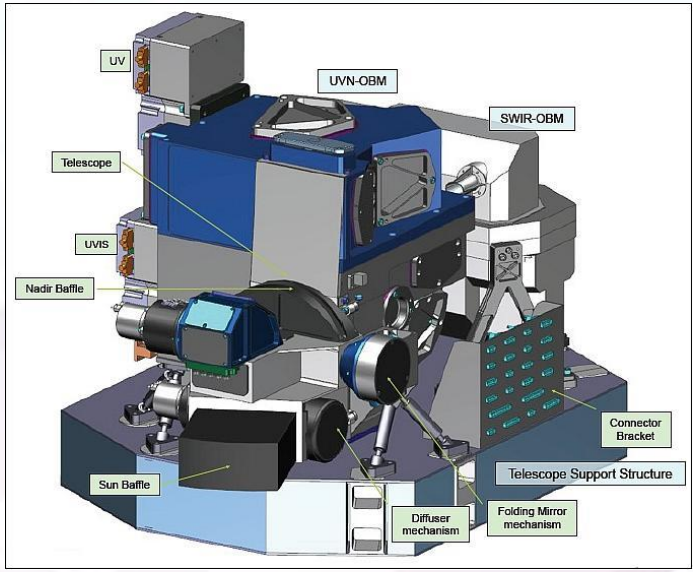
- We will be late and we don’t want to be late
- We cannot afford to be late
- When the money is used up, there is no more

Earth observation instrument

source: www.tropomi.eu

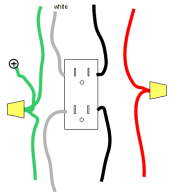
In short

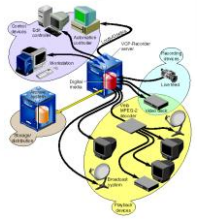
- Very experienced Systems Engineers
- Using quantified requirements routinely
- 6 year waterfall project (imposed by ESA process)
- Don’t know exactly where they’ll end up
- One problem: They missed all deadlines (can you help us)
- 9 weeks later: They haven’t missed any deadline since
- “Sorry, we delivered 1 day early” (instead of expected 1 year late)
- Savings: at least 40 man-year (about €6M ?)
- How did they do that ?

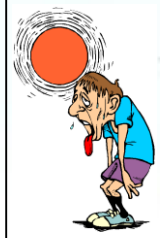


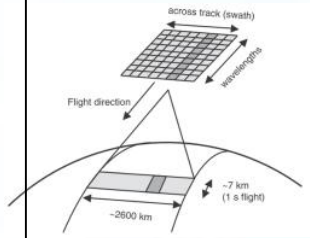
Project Manager

Systems Engineer









Mechanical

- engineer
- engineer
- engineer
- engineer
- engineer

Electrical

- engineer
- engineer
- engineer
- engineer
- engineer

Interface

- engineer
- engineer
- engineer
- engineer
- engineer

Thermal

- engineer
- engineer
- engineer
- engineer
- engineer

Optical

- engineer
- engineer
- engineer
- engineer
- engineer

QA

- engineer
- engineer
- engineer
- engineer
- engineer

Sub-contractors across Europe

Convincing the Project Manager

With CTO:

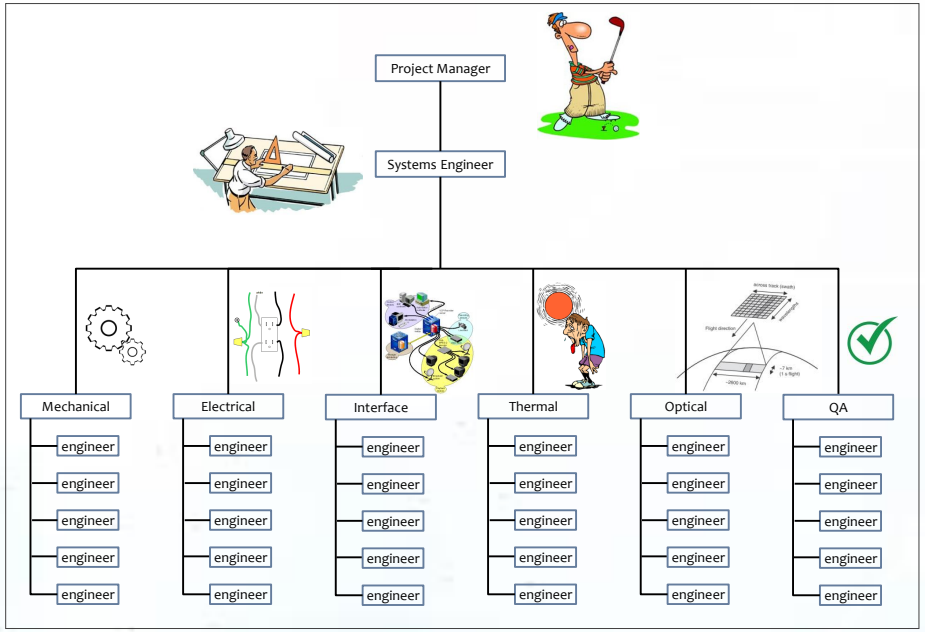
- Don't put me on the training budget
- Put me on the project budget

With Project Manager:

- 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 ?
- Anything to deliver by the end of the week ?
- A status report
- How much time do you need ?
- How much time do you have ?
- Does it fit ?

Time needed?	about 2 more hours
What still to do?	getting input from 6 people
How?	email
Always immediate reply?	no
Time per person?	email, reminder, going there, getting status, check again, compile in report
….	1.5 hr per person
6 people?	6 x 1.5 = 9 hr
How much available?	I'm very busy! Perhaps 4 hr left
Will we succeed?	You can coach the team (get off my back!)

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%

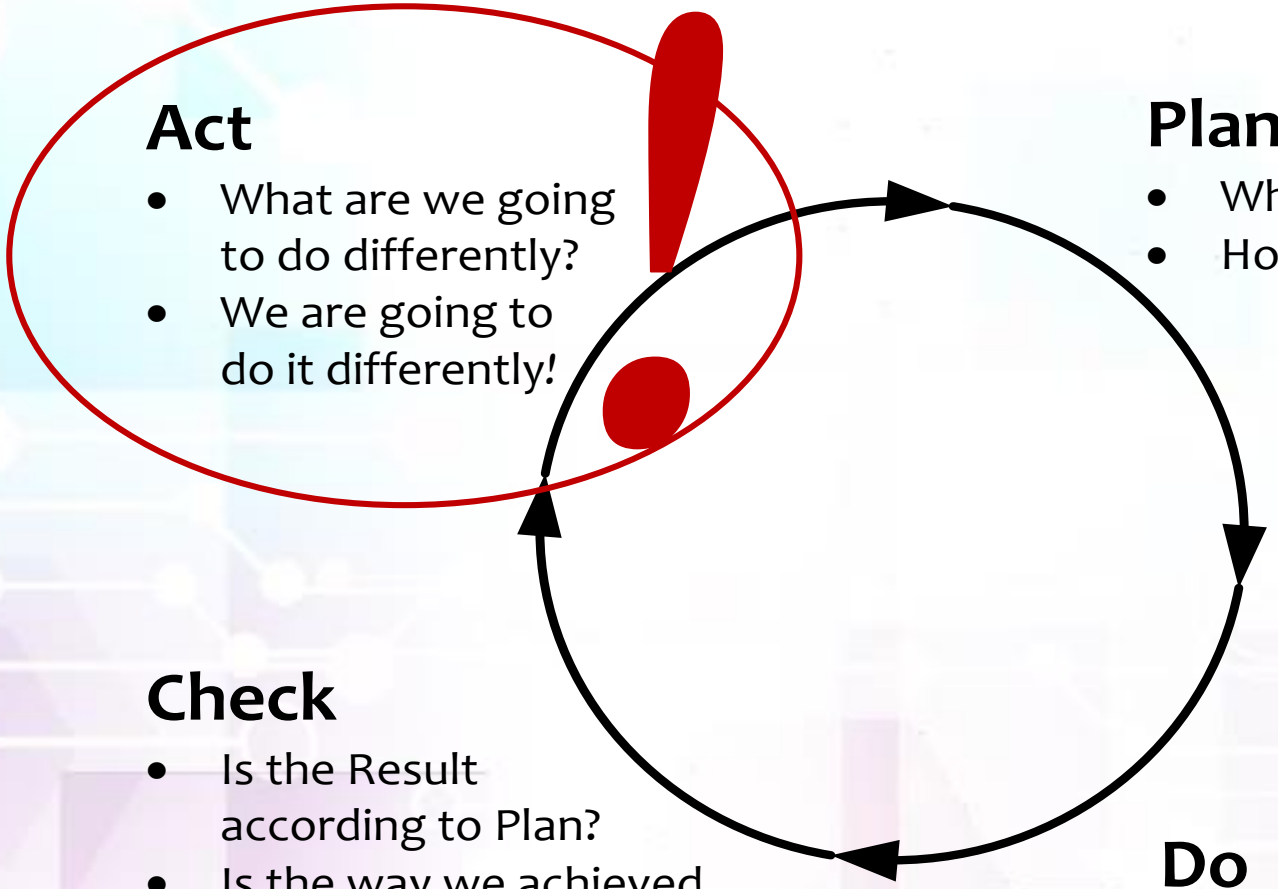
Ref Project Systems Engineer

The essential ingredient: the PDCA Cycle

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

www.malotaux.eu/?id=PDCA

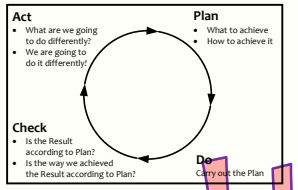
Deming



Evolutionary Project Management elements (Evo)

www.malotaux.eu/?id=processes – 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

Check and learn as early as possible

Zero Defects Attitude

https://tinyurl.com/y7u479lj

Right Result

Quality On Time

Right Time

- 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 Project Planning - Niels

Efficiency of what we do

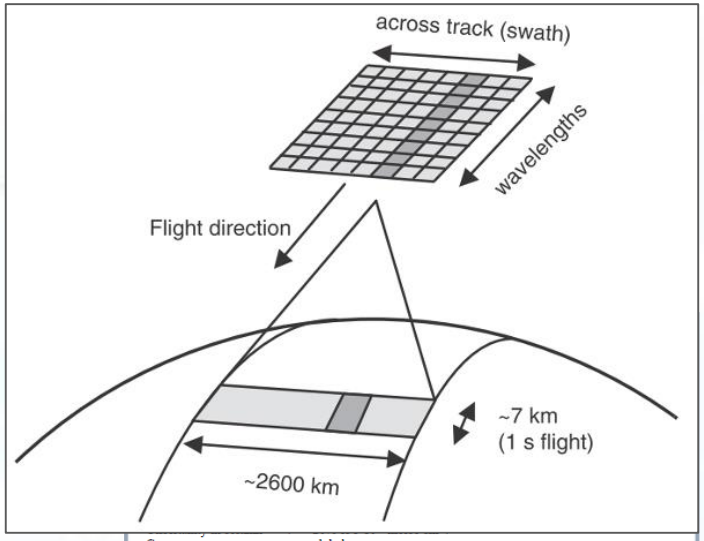
Effectiveness of what we do

What will happen, and what will we do about it ?

Requirements weren't the problem

- 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*)

source: www.tropomi.eu



Coverage : global
Frequency of observation : daily (threshold); multiple observations per day (target)

3.2.2 Requirements for tropospheric O_3 / O_3 profile and total O_3

The scientific requirements for tropospheric O_3 and the O_3 profile are identified from an overall assessment of the different roles of tropospheric O_3 in the TROPOMI science objectives (see Chapter 2). The uncertainty in tropospheric O_3 and the O_3 profile depend on the altitude domain and are summarised in Table 3.1. The given uncertainties include retrieval uncertainties. The target horizontal resolution for tropospheric O_3 is 5 × 5 km^2 (threshold 20 × 20 km^2). Tropospheric ozone is obtained on a horizontal resolution of 20 × 20 km^2 by subtracting the stratospheric column from the total column. For the threshold resolution of the ozone profile (40 × 40 km^2) the assumption is made that total column variations within the area of 40 × 40 km^2 are mainly related to tropospheric ozone variations. Because of the ~90% contribution of stratospheric ozone to the total ozone column the overall uncertainty in the total ozone column should not exceed 2%. Vertical profile uncertainties are altitude dependent and range between 3 – 5% above 12 km, 12% between 6 and 12 km, 20% below 6 km, and 60% for the PBL. The uncertainty requirements are also sufficient to monitor the ozone layer and the ozone hole. The temporal resolution shall be at least daily to be able to follow the course of the relevant meteorological processes. Because of the short mixing time scales in the boundary-layer information on the diurnal cycle of tropospheric O_3 would be beneficial. The vertical distribution within the troposphere could be better resolved in synergy with auxiliary thermal IR ozone observations.

The requirements for the ozone products can be summarised as:

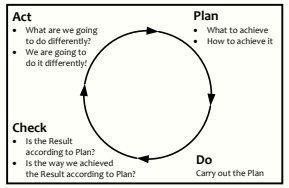
Requirements for tropospheric O_3
Ground-pixel size : 20 × 20 km^2 (threshold); 5 × 5 km^2 (target)
Uncertainty in column : altitude-dependent
Coverage : global
Frequency of observation : daily (threshold); multiple observations per day (target)

Requirements for stratospheric O_3
Ground-pixel size : 40 × 40 km^2 (threshold); 20 × 20 km^2 (target)
Uncertainty in column : altitude-dependent
Coverage : global
Frequency of observation : daily (threshold); multiple observations per day (target)

Requirements for total O_3
Ground-pixel size : 10 × 10 km^2 (threshold); 5 × 5 km^2 (target)
Uncertainty in column : 2%
Coverage : global
Frequency of observation : daily (threshold); multiple observations per day (target)

Evolutionary Project Management elements (Evo)

www.malotaux.eu/?id=processes – 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

Check and learn as early as possible

Zero Defects Attitude

Evo Project Planning - Niels

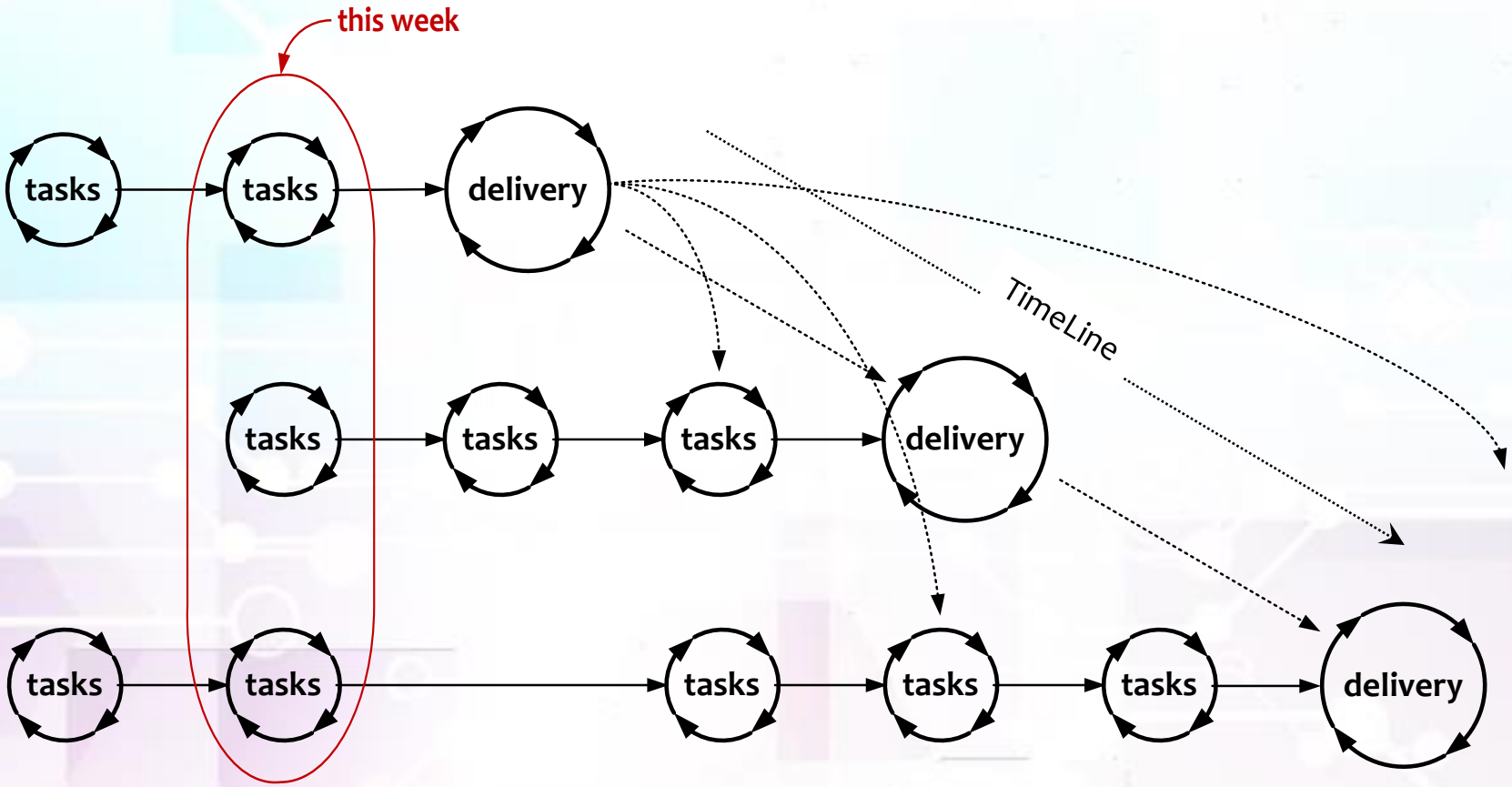
- Weekly TaskCycle
  - Short term planning
  - Optimizing estimation
  - Promising what we can achieve
  - Living up to our promises
- Bi-weekly DeliveryCycle
  - Optimizing the requirements and checking the assumptions
  - Soliciting feedback by delivering Real Results to *eagerly waiting Stakeholders*
- 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 ?

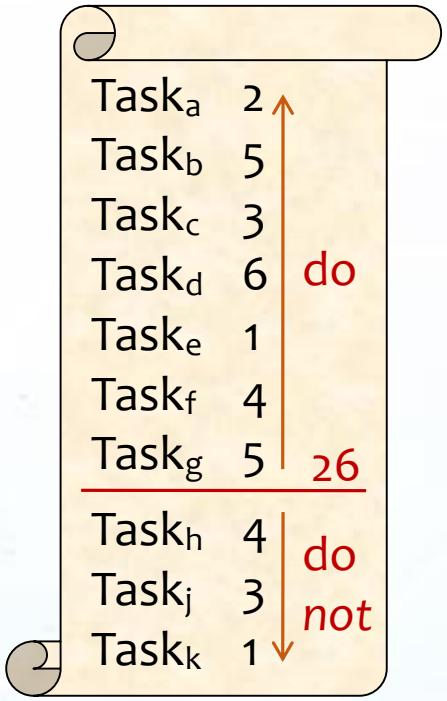
Tasks feed Deliveries



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 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

Weekly planning

- Individual preparation
  - Conclude current tasks
  - What to do next
  - Estimations
  - How much time available
- Modulation with / coaching by Coach / Team Lead / Peer(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

Why is this important?

- TaskCycle Planning is not just planning the work for the coming week
- Half (±30%) of what people do in projects later proves not having been necessary
  → using ~~Retrospectives~~ Prespectives
- During the TaskCycle planning we can very efficiently see
  - What our colleagues think they're going to do
  - Make sure they're going to work on the most important things, in the right order
  - Not on unnecessary things, or wrong order
  - In line with the architecture and design
  - Leading most efficiently to the goal of the delivery
- Everyone in the project-team knows what the others will do

We see issues before they become a problem, saving time

Awful schedule pressure !

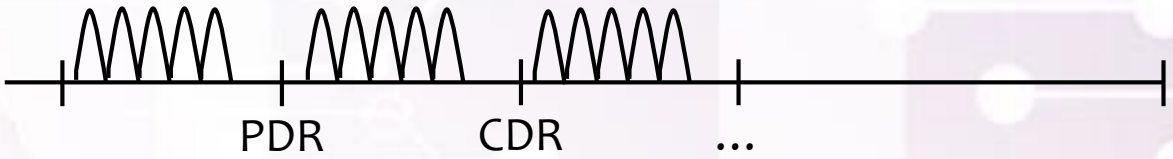
	per doc	hr
4 heavy	15	60
3 easy	2	6
	total	66
other work		33
	total	99

available	2 x 26	52

	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6	Doc 7
John	x		x	x	x	x	
Samuel	x	x		x		x	x
Paul	x	x	x	x	x	x	x
Michael	x			x	x		
Marc			x	x		x	x

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

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



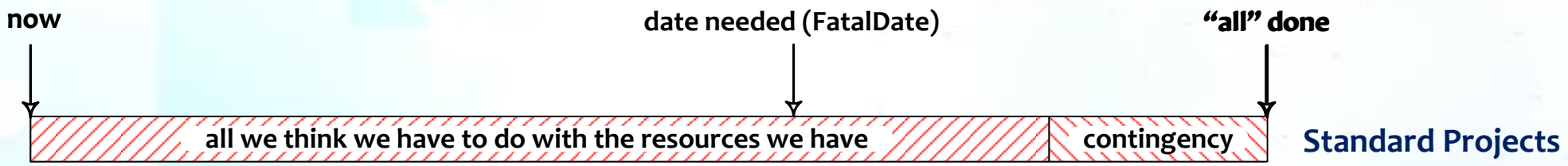
- Some suggestions …
- Result: well reviewed, great meeting, everyone satisfied

Biweekly DeliveryCycle

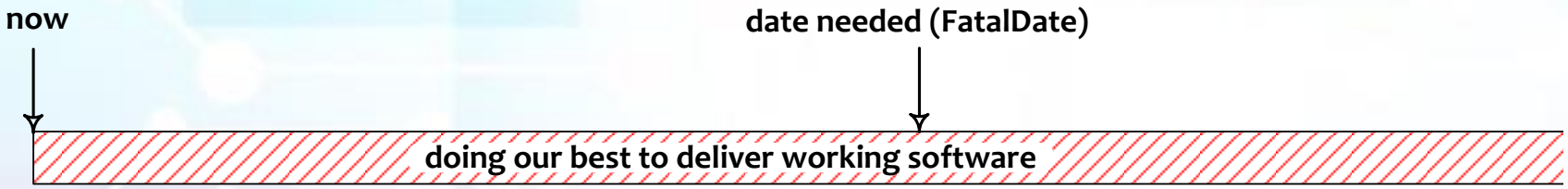
- Are we *delivering* the right things, in the right order, to the right level of detail for now
- Optimizing requirements and checking assumptions
  - Better assume our assumptions may be incorrect
  - Suppliers: We better assume that their assumptions may be incorrect
  - What will generate optimum feedback
  - We deliver to eagerly waiting stakeholders
  - Delivering ‘juicy bits’, if we have to make them eagerly waiting
- Not more than 2 weeks

TimeLine

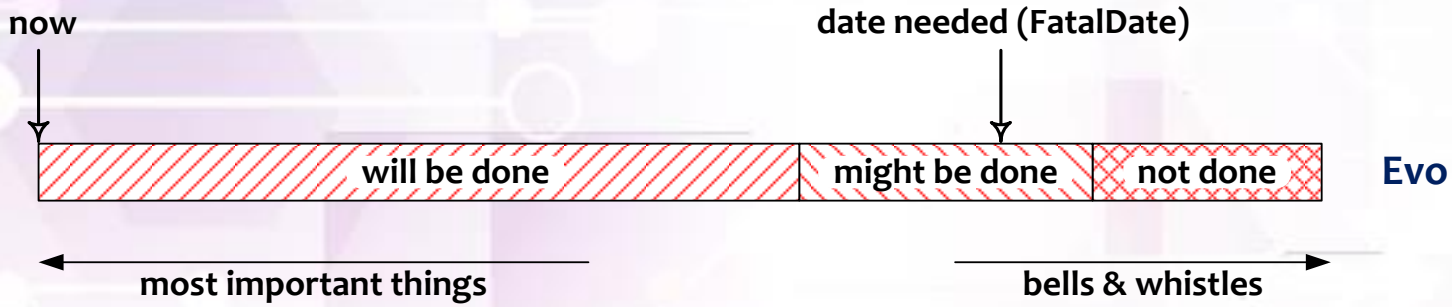
How do we know that we do, and get, what is needed, when it's needed ?



Standard Projects



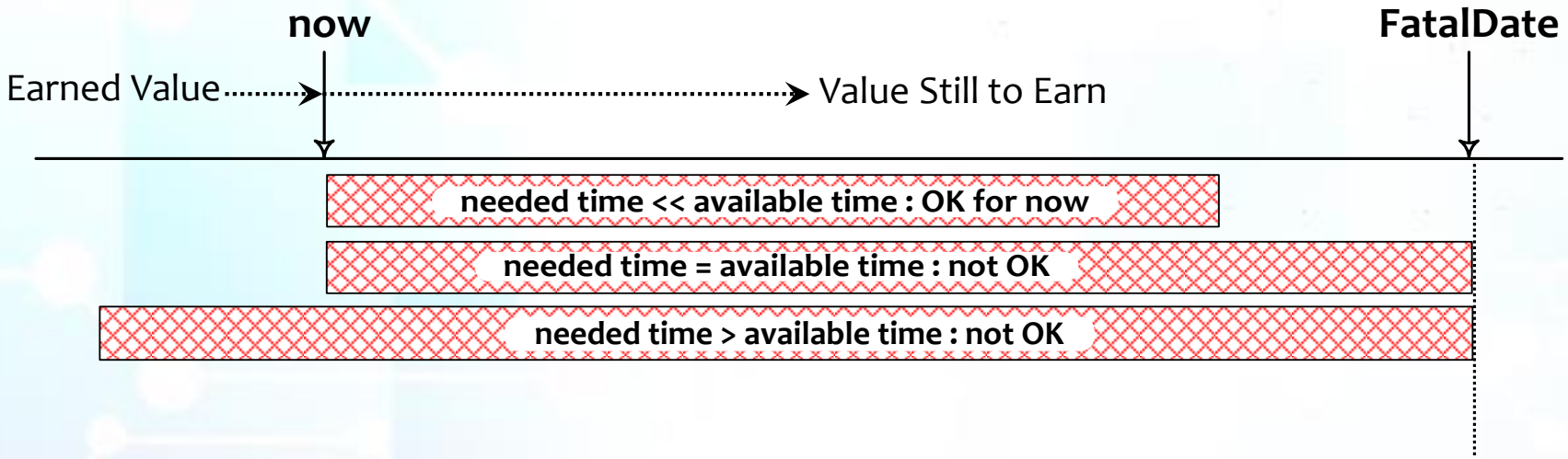
Agile



Evo

Better 80% 100% done,
than 100% 80% done
Let it be the most important 80%

What do we do if we see we won't make it on time ?



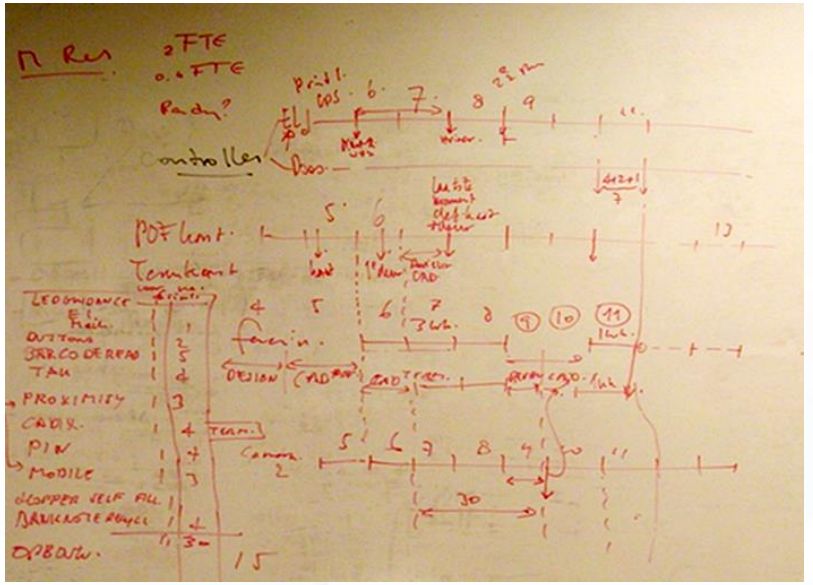
- Value Still to Earn

versus

- Time Still Available

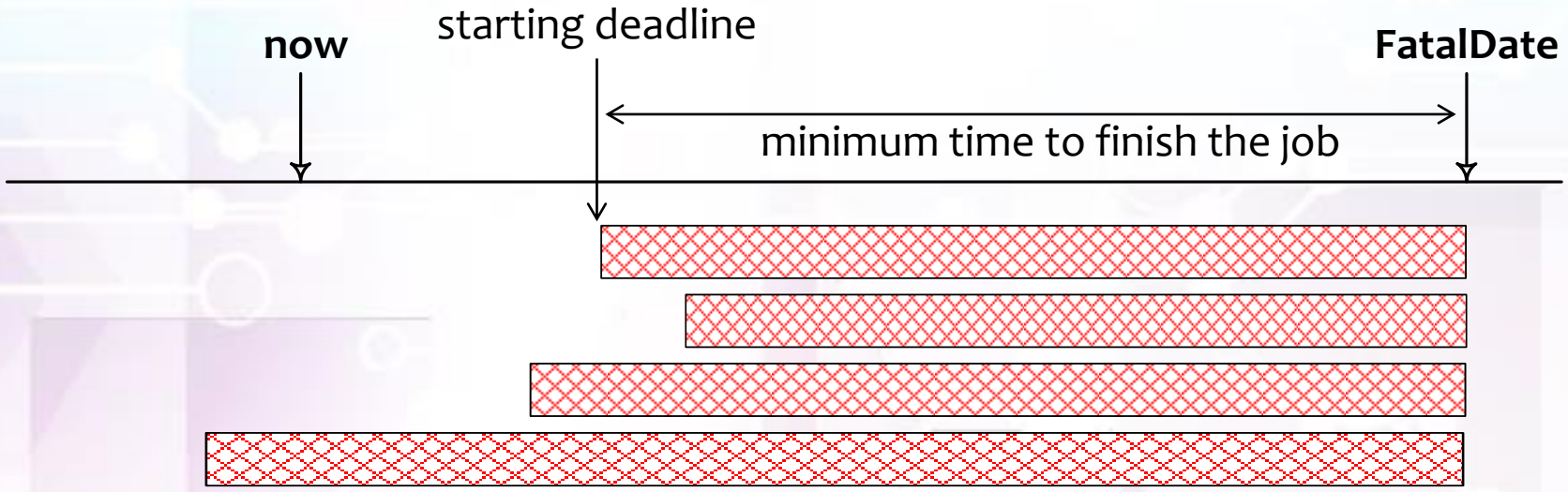
If the match is over, we cannot score a goal

Even more important:
Starting Deadlines



Starting deadline

- Last day we can start to deliver by the end deadline
- Every day we start later, we will end later

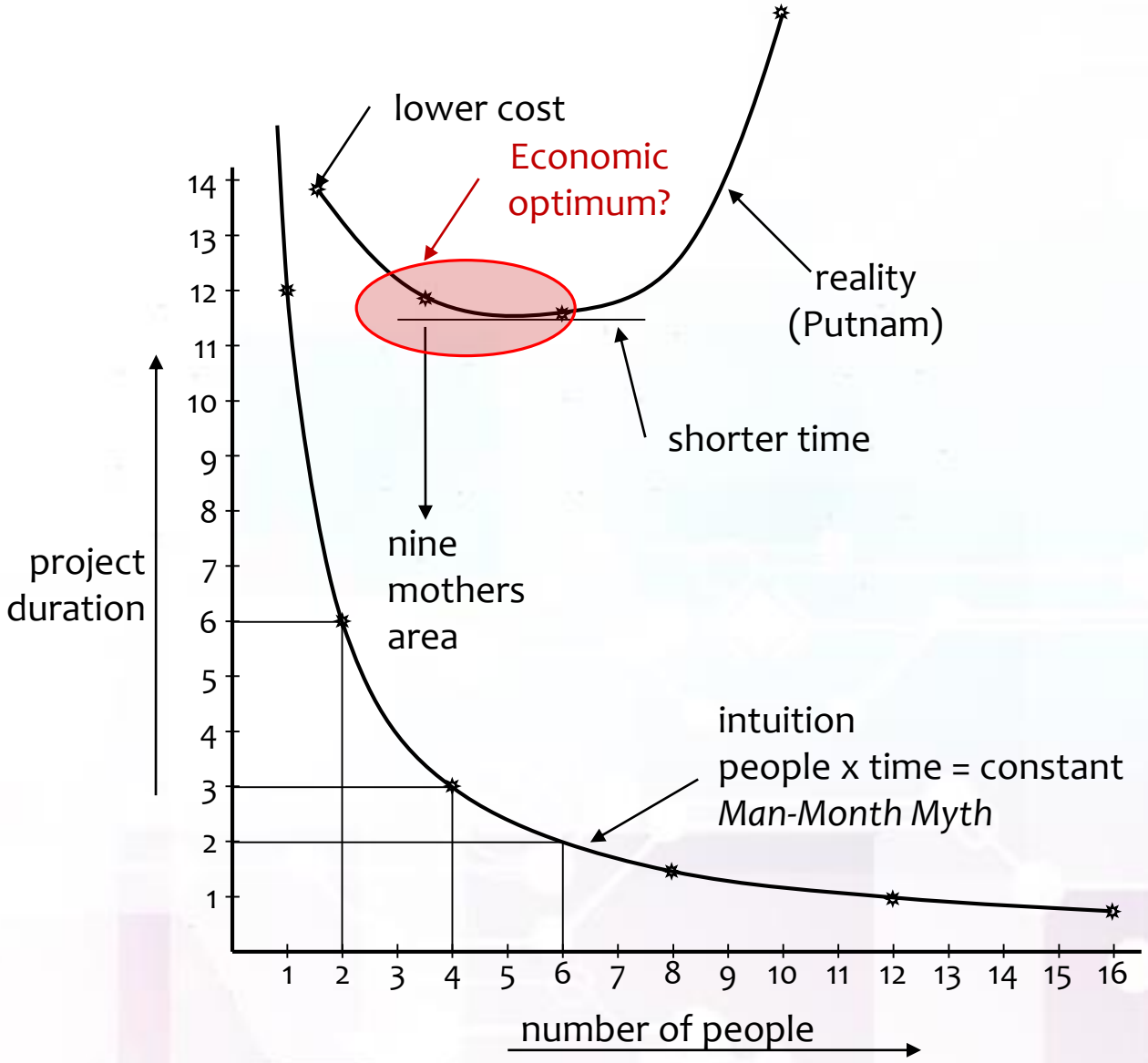


Deceptive options

- Hoping for the best (fatalistic)
- Going for it (macho)
- Working overtime (fooling ourselves and our boss)
- 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 into the wrong direction

Adding people ?



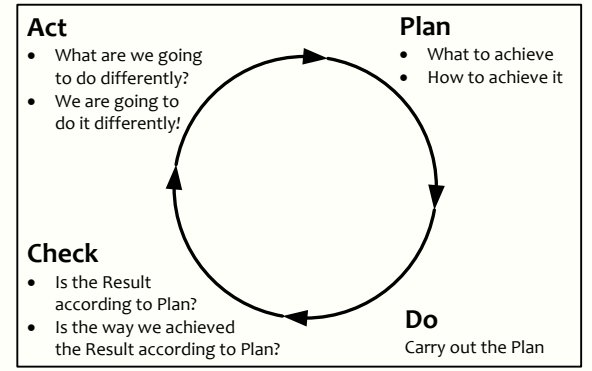
Brooks' Law (1975)

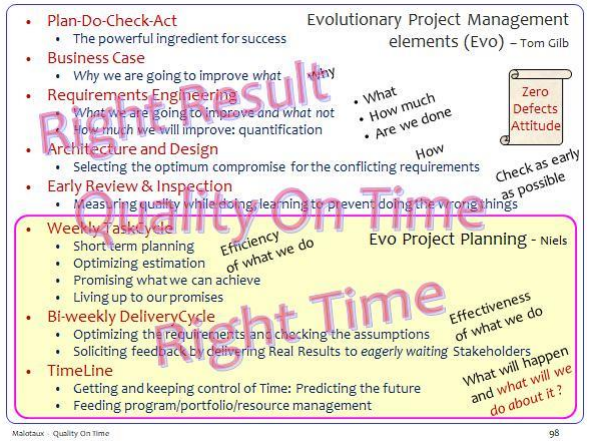
Adding people to a late project makes it later

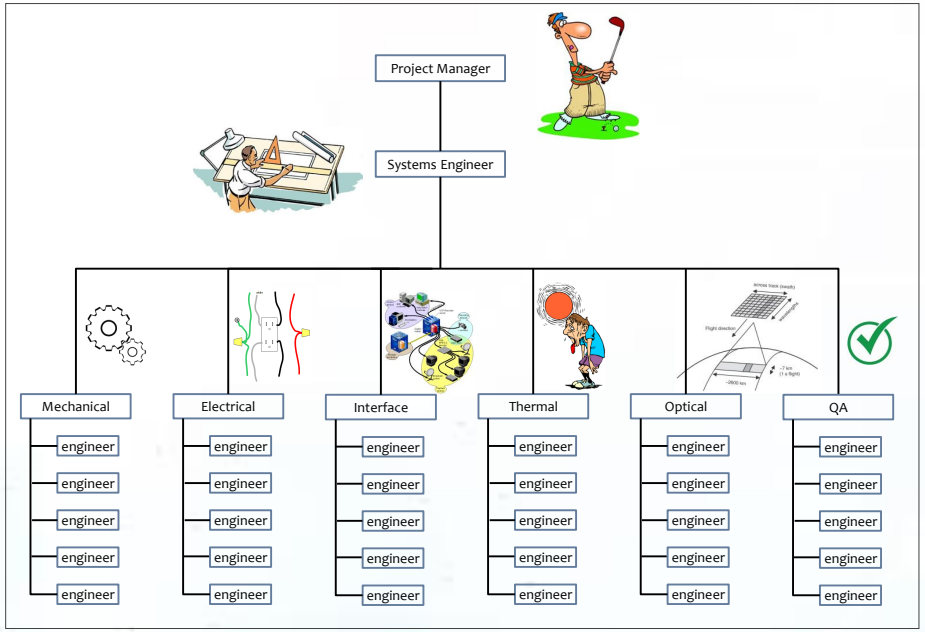
Saving time

Continuous elimination of waste

- 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

Ref Project Systems Engineer

Why did it still take so long before actual launch ?

- The launch was delayed caused by issues you cannot predict even with the Evo approach:
  - 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

Ref Project Systems Engineer

How Space Systems Engineers learnt to meet all deadlines

or

How to be on time

Niels Malotaux

niels@malotaux.eu
www.malotaux.eu/conferences

Would this help *you* to deliver better results in less time ?

Or do you have a better suggestion ?