Optimizing Project Execution

Planning is important – execution of the plan even more

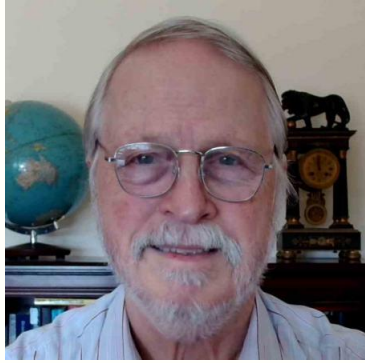
www.malotaux.eu/?id=conferences

www.malotaux.eu/?id=booklets

Niels Malotaux

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

Niels Malotaux



- Independent Engineering and Project Coach
- Expert in helping teams 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
- Project rescue
- Embedded Systems architect (electronics/firmware)
- Project types
  electronic products, firmware, software, space, road, rail,
  telecom, building automation, industrial control, parking system

Quality On Time
Delivering
the Right Result
at the Right Time

Optimizing project execution

- Did you ever dilute quality to meet a deadline ?
- Did you ever need excuses to explain why your deliveries were late ?

- Most ‘impossible’ deadlines are not impossible
- Still delivering great quality

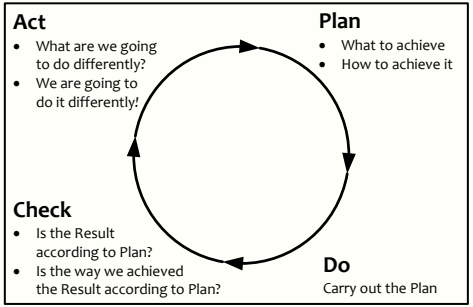
- Sounds incredible ?
- That’s what many thought, until they tried

Some examples

- Electronic product development project (US)
  - *Saved about half a year* (some 20 man-year not wasted) - delivered '*unprecedented quality*'
    Won Prestigious Team Award as part of the company's Technical Excellence recognition program
    (Product manager: https://malotaux.eu/doc.php?id=19 - chapter 4.7.1)
- Space project (NL)
  - *Saved about one year* (some 40 man-year not wasted) - *delivered successfully on time*
- Software project (PL)
  - Deadline in 6 weeks; called it 'mission impossible'
  - With some coaching *delivered successfully in 5 weeks* (proudly saying: "No overtime!")
- Large development project (UK) - abandoned
  - Wasted at least 70 man-year (not following my suggestions)
    Later finalized at huge cost: "satisfying every whim of the customer"
- Accounting software project (DE) - abandoned
  - Project manager admitted: "My professor at university actually said similar things"

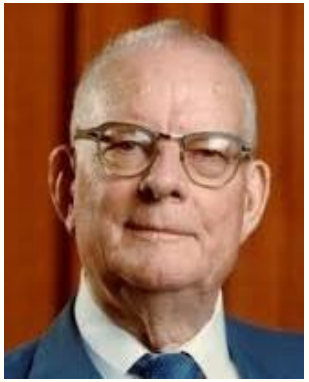
Main secret to be on time, NOT diluting quality

- Half of what we do in our work, later will prove not to have been needed
  - If we see that *after* spending the time, the time is already wasted
  - If we see that *before* we spend the time, we still can decide not to waste the time
- If we save time, we have more time to do the right things right
  still delivering on time
- Doing things wrong, costs about *three times* as much as doing it right the first time
- Quality costs less
- ~~Retrospectives~~ → Prespectives
- We know we're not perfect, that's why we use PDCA

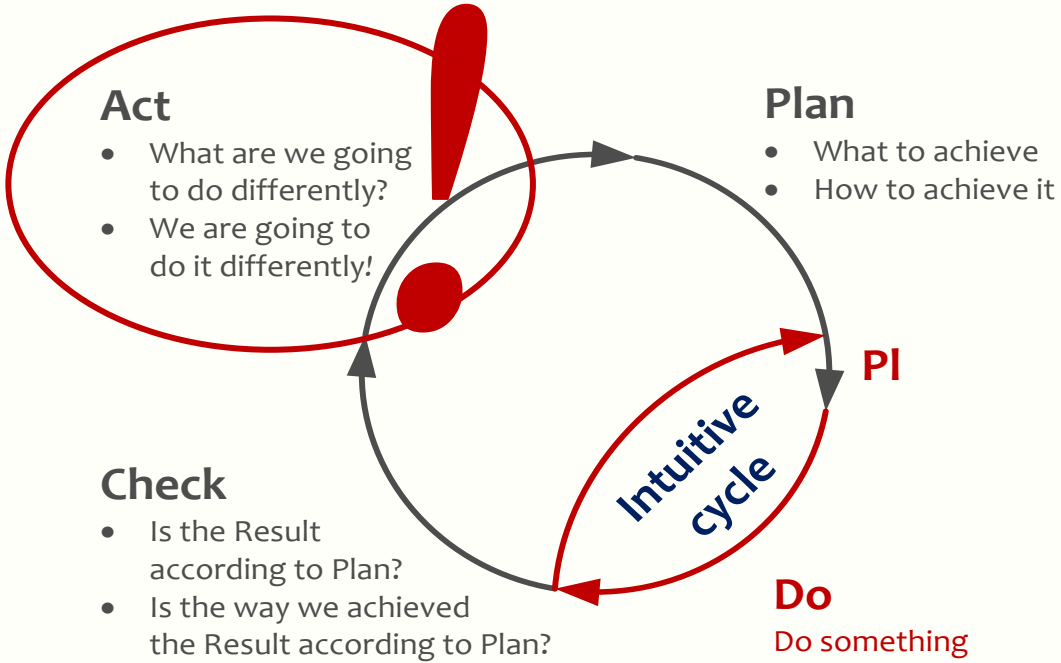


The essential ingredient: the PDCA Cycle

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



Deming



Universal goal

- Delivering the Right Result at the Right Time, wasting as little time as possible (efficiently)

Quality on Time

- Providing the customer with
  - what they need
  - at the time they need it
  - to be satisfied
  - to be more successful than 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

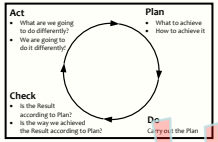
Do we know *our* goal ?

- Do we know what we are supposed to *achieve* the coming weeks ?
- Do we know what we are supposed to *achieve* the coming week ?
- Do we know what each of us is supposed to *do* to contribute to successfully achieving:
  - what should be achieved
  - in optimum order

Evolutionary Project Management elements (Evo)

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 *Execution*

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 ?

Requirements with Planguage

ref Tom Gilb

quantifying the goal

SMART

Specific

Measurable

Definition:

RQ27: Speed of Luggage Handling at Airport

Scale: Time between <arrival of airplane> and first luggage on belt

Meter: <measure arrival of airplane>, <measure arrival of first luggage on belt>, calculate difference

Attainable

Benchmarks (Playing Field):

Past: 2 min [minimum, 2024], 8 min [average, 2024], 83 min [max, 2024]

Current: < 4 min [competitor y, May 2024] ← <who said this?>, <Survey Dec 2024>

Record: 57 sec [competitor x]

Wish: < 2 min [2026Q3, new system available] ← CEO, 19 Jan 2025, <document ...>

Realizable

Time

Traceable

Requirements:

Tolerable: < 10 min [99%, Q4] ← SLA

Tolerable: < 15 min [100%, Q4, Heathrow T4] ← SLA

Goal: < 15 min [99%, Q2], < 10 min [99%, Q3], < 5 min [99%, Q4] ← marketing

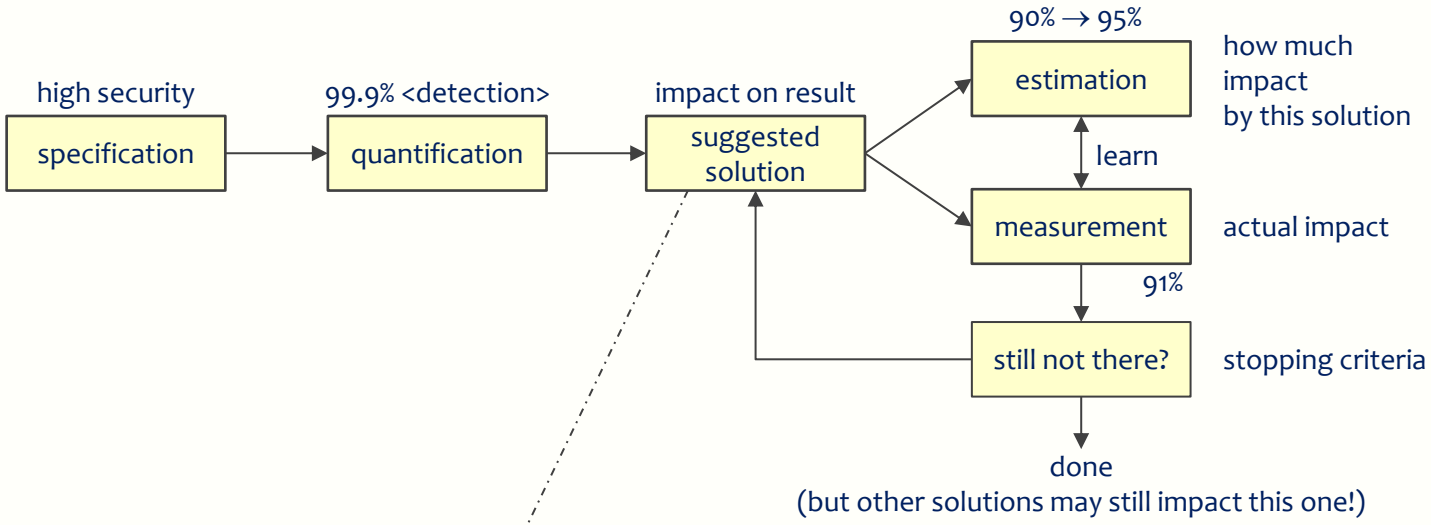
Tom Gilb quote

- 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

Quality

organising the

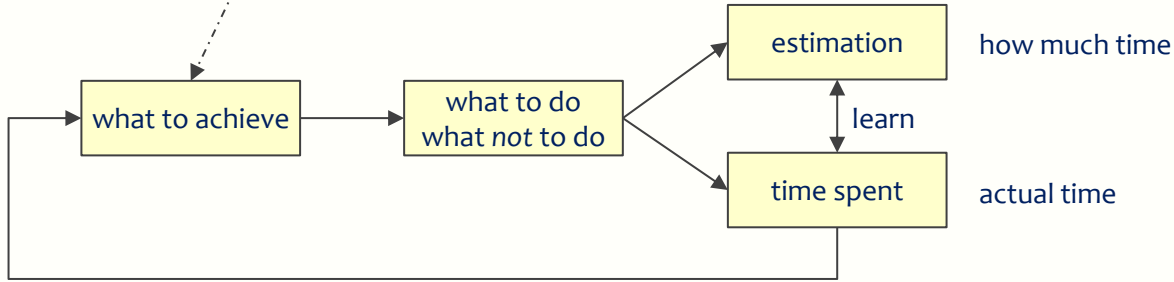
- what
- why
- for whom
- how much



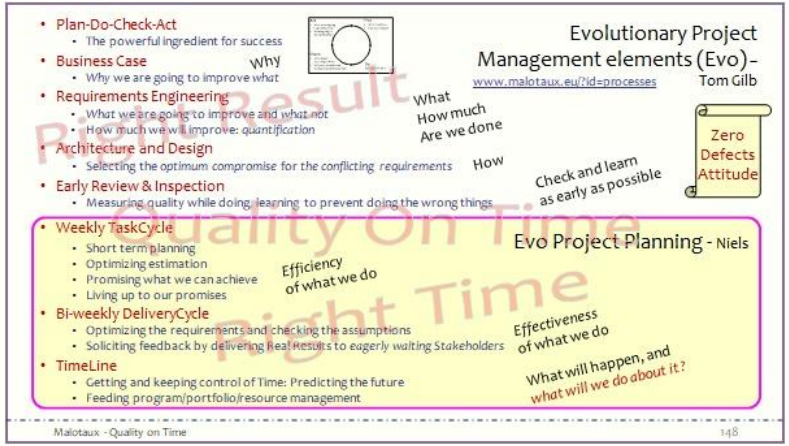
on Time

organising the

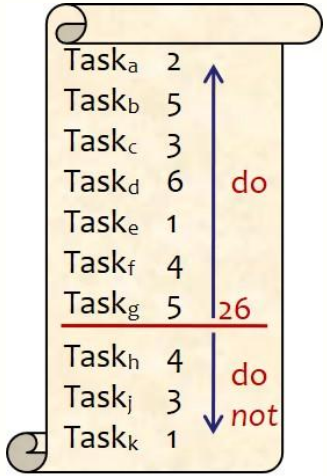
- how
- when



Weekly TaskCycle plan

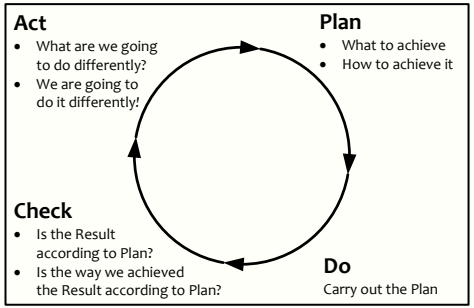


- What are we supposed to achieve
- How much time do we have available
- 2/3 of gross available time is net plannable time
- What is most important to do
  in order to achieve what we're supposed to achieve
- Estimate net effort needed to do these things
- Which most important things fit the net plannable time
  (default 2/3 of gross available time, 26 hr per week at 40hr work-week)
- What can, and what are we going to do
- What are we *not* going to do
- *Writing it down ! Our fuzzy mind isn't good enough !*



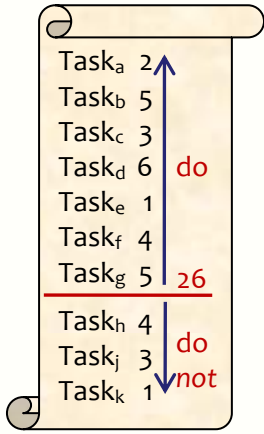
2/3 is default start value
this value works well with development work

Every week: reflecting and preflecting



- Was all planned work really done ?
- If a Task was not completed, we learn:
  - Time spent but needed more time ? → effort estimation problem
    - What did I think then, what do I know now → learn (Check and Act)
  - Time not spent? → time management problem
    - Too much distraction
    - Too much time spent on other (poorly-estimated) Tasks
    - Too much time spent on other things
- Close unfinished Tasks after having dealt with the consequences
  - Learn from it
  - Define remaining work - can put on the Candidate Task List
  - Declare all Tasks finished after having taken the consequences
    (we cannot work in history)
- Continue with planning the Tasks for the next week

Immediate consumption of metrics



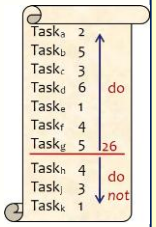
cycle	who	task description	estim	real	done	issues
3	John	*Net time available: 26*				
		aaaaaaaaa	3	3	yes	
		bbbbbbbb [Paul]	1			
		ccccccccc	5	13	yes	
		ddddddddd	2			
		eeeeeeee	3	2		
		fffffffffff	2	1		
		~~ggggggggg~~	6	7	yes	
		hhhhhhhh	4			
			26	26		
4	John	*Net time available: 26*				
		jjjjjjjjjjjjj	3			for team x
		~~kkkkkkkkk~~				for team x
		mmmmm	5			for team x
		~~nnnnnnnn~~				for team x
		~~ppppppppp~~				for team y
		qqqqqqqq	12			for team y
		rrrrrrrrrrrr	6			for team y
		~~sssssssssss~~				for team y
		~~ttttttttttt~~				for team y
			26			

TaskCycle Analysis
(reflecting)

learning

TaskCycle Planning
(preflecting)

'Innocent' questions for preflection



People come in with their week plan,
come out with a modified plan,
more going to work on the right things,
less spending time on unnecessary things.
Immediate savings.
From day one.

- Really ?
- Should we ?
- Why would we do that ?
  (never use 'you')
- Who's waiting for that ?
- What do they need ?
- How much do they need ?
- When do they need it ?
- Is it really necessary ?
- Is it really necessary now ?
- How do we know ?
- Why ?
- What happened ?
- What could we do ?
- Will we be on time ?
- What makes us think that ?
- What's different this time ?
- Is this the best way to do it ?
- Does it fit the available time ?
- *If there is any problem:*
  What-are-we-going-to-do-about-it ?

Never challenge an estimate !
If we don't understand the estimate,
only say:
- What are you planning to do ?
- Now estimate again …

If they insist:
- Let them find out themselves

Client statement

- People felt having not enough time to do things properly
- But having quantified it makes the problem clearer, and *more confronting*

- Team members plan their work weekly, and share this in the weekly team meeting
  - People realize how little time they had between all the scheduled meetings
  - Everyone knows what the others are going to do and why
  - We can help each other when we are in danger of forgetting something …
  - As a stepping stone to get longer periods of time under control
  - Problems come to the surface more clearly so that we can do something about them sooner

Ref Project Manager

Making best use of limited available time

- After the work, the time is already spent
- Before the work, we still 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 change history, only improve the future

Optimizing planning efficiency

(efficiency = effectiveness + time)

- 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)
  - Retrospection (what can we learn from what happened ?)
  - Prespection (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

Modulation costs less than *Generation*

Biweekly DeliveryCycle

- Are we *delivering* the right things, in the right order, to the right level of detail for now
  (TaskCycle: Are we *doing* the right things, in the right order, to the right level of detail for now)
- Optimizing requirements and checking assumptions
  - Better assume *our*, *and their*, assumptions may be incorrect
  - Better assume that suppliers' assumptions may be incorrect (use Active Synchronization)
  - 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

Active Synchronization

Somewhere around us, there is the bad world

If we are waiting for a result outside our control, there are three possible cases:

1. We're sure they'll deliver the Right Result at the Right Time
2. We're not sure
3. We're sure they'll not deliver the Right Result at the Right Time

- If we are not sure (case 2), better assume case 3
- From suppliers already using these techniques we can expect case 1
- We act like case 1 (now we know how to do that !)

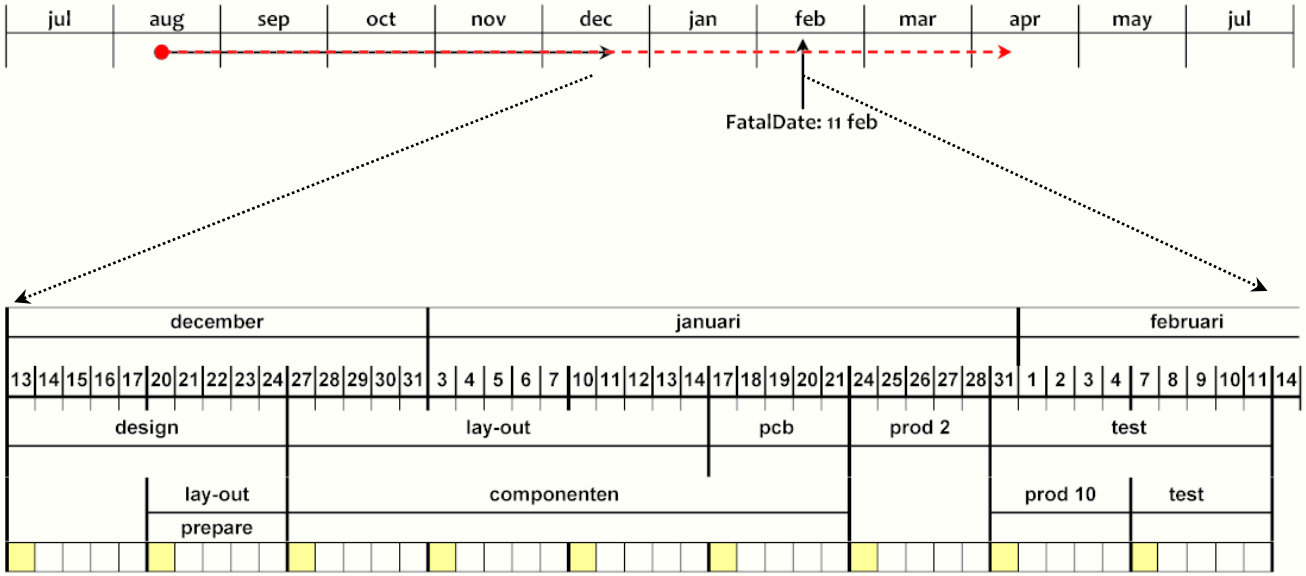
In cases 2 and 3: Actively Synchronize: Go there !

1. Showing up increases our priority
2. We can resolve issues which otherwise would delay delivery
3. If they are really late, we'll know much earlier

Deadlines ('fatal dates')

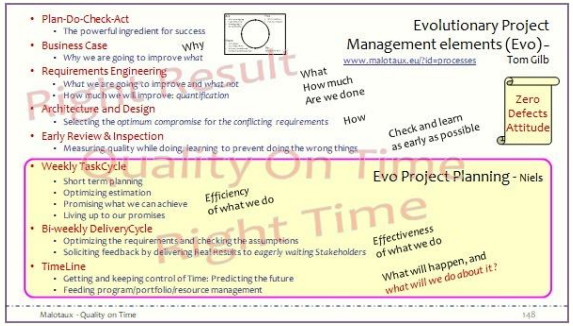
- Make us more efficient
  - Parkinson's Law: "People spend the time available"
- Real deadlines
  - Else *we don't need it anymore*
  - Next year it will be obsolete (mobile phone chip)
  - Competition will have grabbed the market (VOIP: Voice over IP)
  - We'll have to wait another two years (parking system)
  - We'll have to wait another 30 years (planetary constellation)
- Fake deadlines (manager's dream)

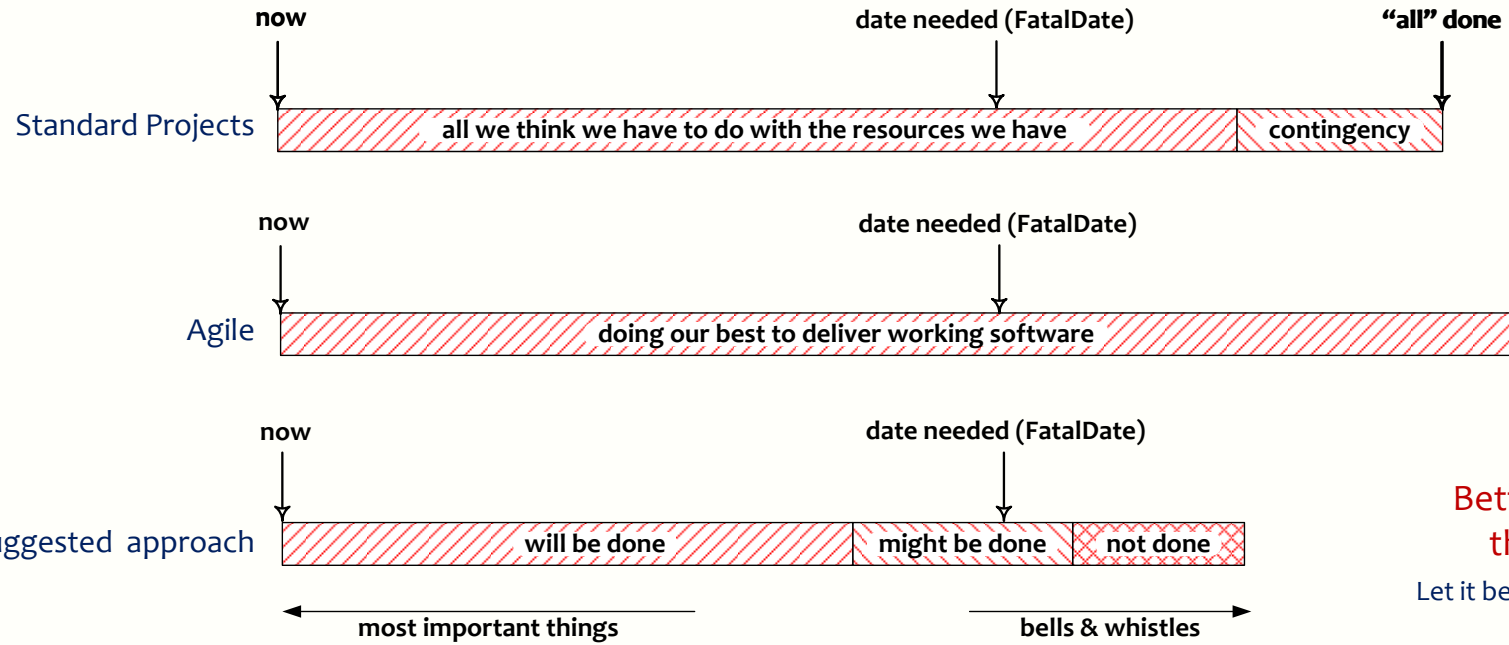
Hard deadline



TimeLine

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

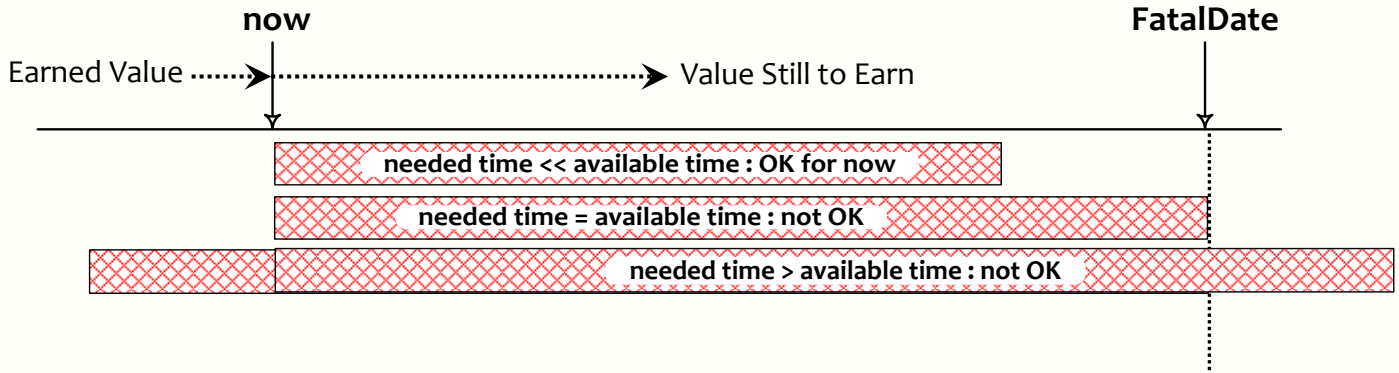




Better 80% 100% done,
than 100% 80% done

Let it be the most important 80%

At the time they need it



- Value Still to Earn

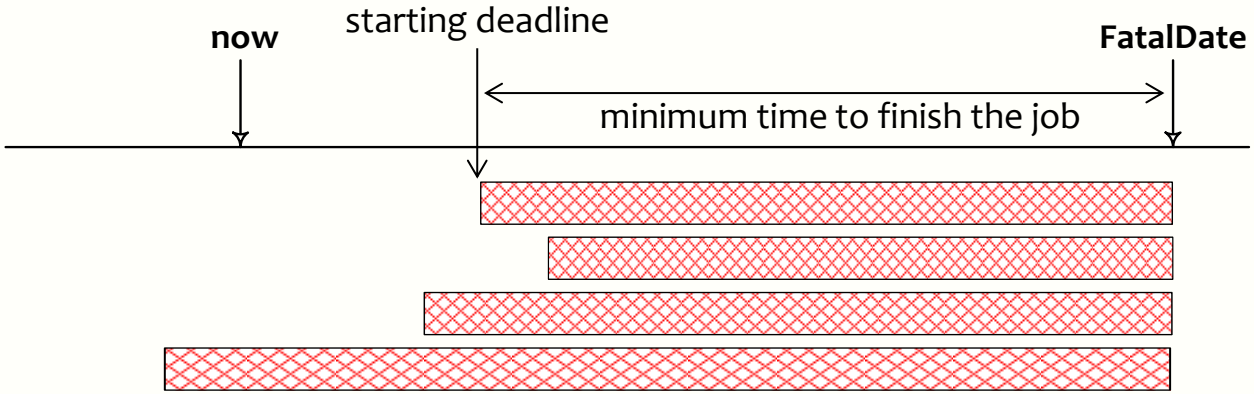
versus

- Time Still Available

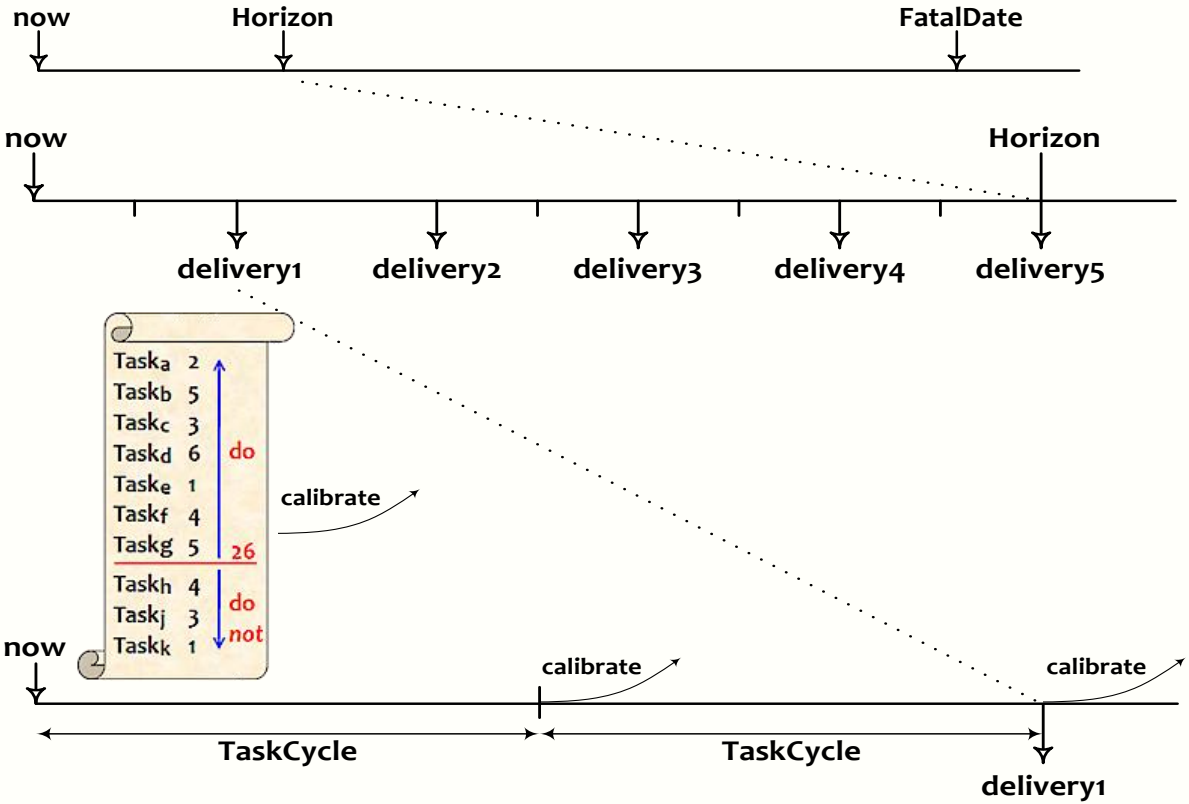
If the match is over, you cannot score a goal

Even more important: Starting Deadlines

- To meet Delivery Deadlines, focus on *Starting Deadlines*
- Starting Deadline
  - Last day we can start to deliver by the delivery deadline
  - Every day we start later, we will end later

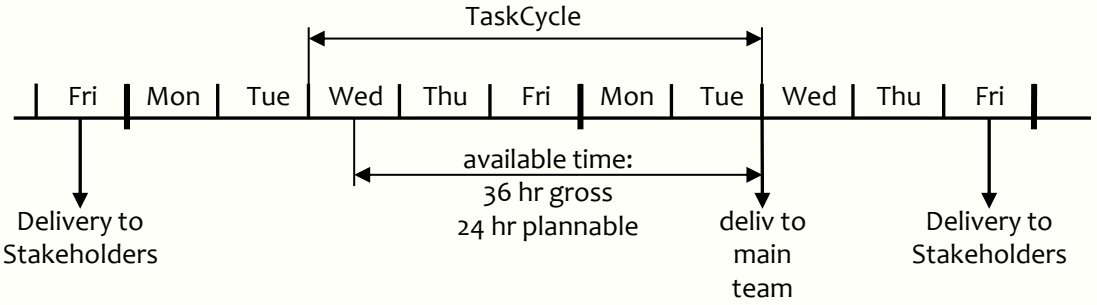


TimeLine: Result to Tasks, and back



Designing a Delivery

not only *designing* the product
also *designing* the way to get there



Serge (ProjLead)	
MbWA	3
Planning nxt wk	3
Work for deliv	4
-	6
-	2
-	1
-	5
Total	24

Gregory	
Draft design	6
Finish design	6
Work for deliv	3
-	1
-	2
-	2
-	3
-	5
-	6
XMLa	4
XMLb	4
Total	42

Gregory (later)	
Draft design	0
Finish design	0
...	

Jerome	
XMLa	3
XMLb	3
...	

What would have happened if we wouldn't have *designed* this delivery?

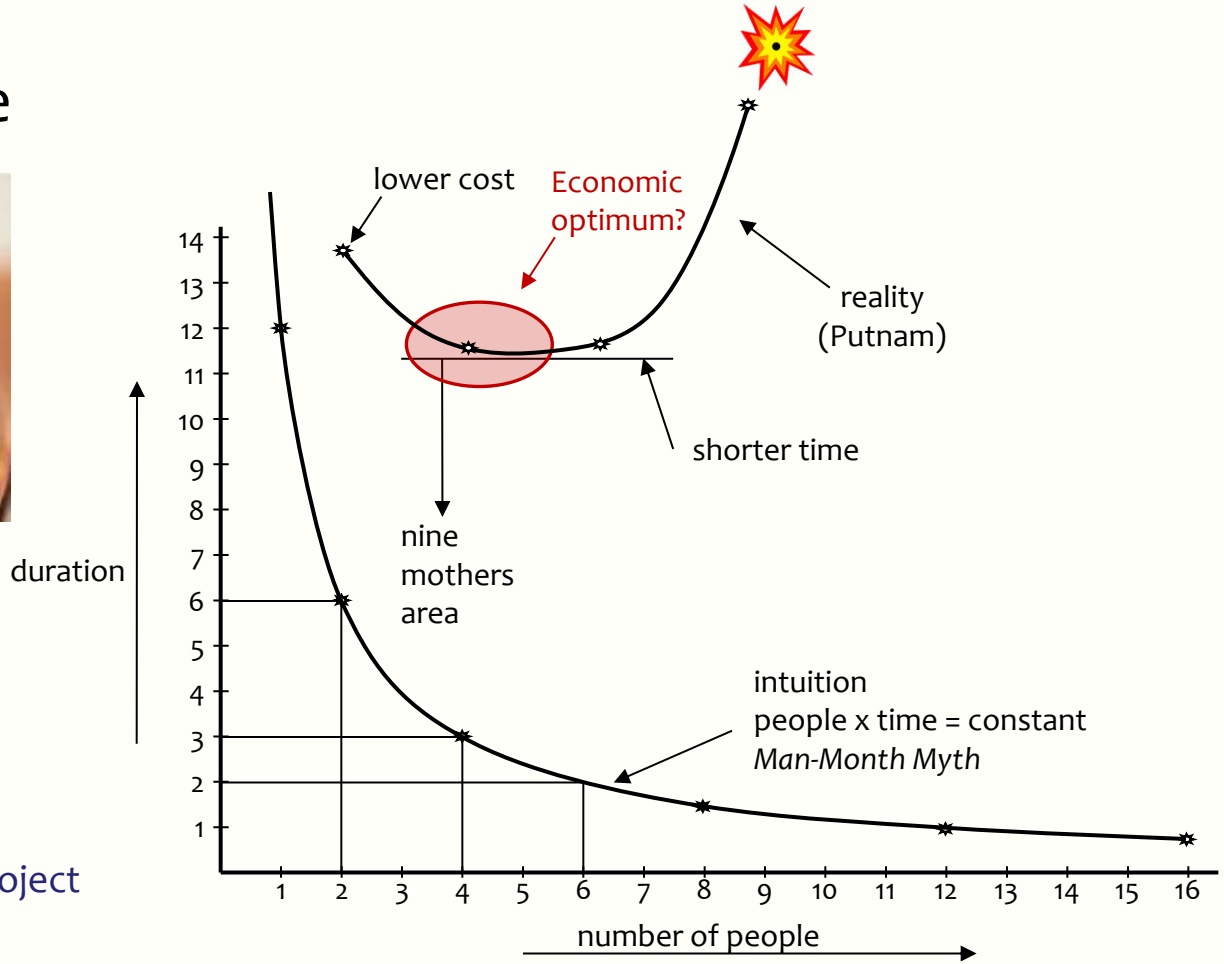
How to be on time

7 options

Deceptive options

1. Hoping for the best (fatalistic)
2. Going for it (macho)
3. Working Overtime (fooling ourselves and our boss)
4. 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 FatalDate is really felt

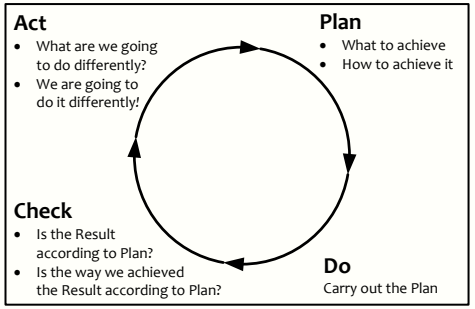
5. Adding people



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

6. Saving time

Continuous elimination of waste (www.malotaux.eu/?id=essenceoflean)



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 (www.malotaux.eu/?id=evo)
  - *Not* doing what later proves to be superfluous
- Efficiency in *how* we do it - doing things differently
  - The product (www.malotaux.eu/?id=designlog)
    - Using proper and most efficient solution, instead of the solution we always used
  - The project (www.malotaux.eu/?id=projectmanagement)
    - Spending less time, instead of immediately doing it the way we always did
  - Continuous improvement and prevention processes (www.malotaux.eu/?id=PDCA)
    - Constantly learning doing things better and overcoming bad tendencies
- Efficiency in *when* we do it - right time, right order (www.malotaux.eu/?id=timeline)
- TimeBoxing - much more efficient than FeatureBoxing (www.malotaux.eu/?id=timeboxing)

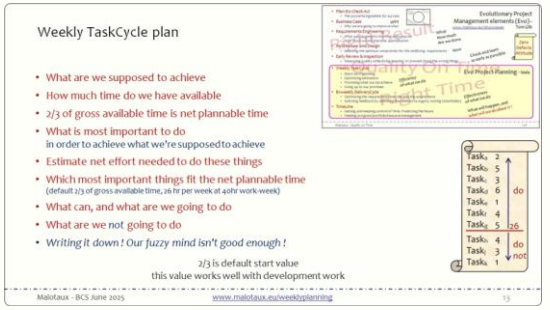
Option 7 ?

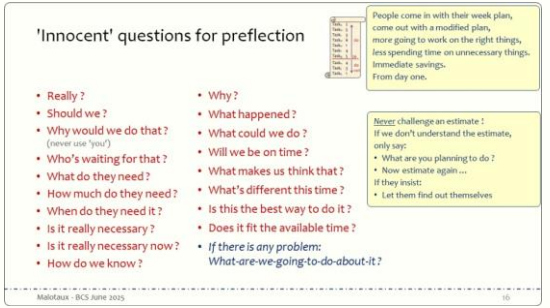
Killing the project

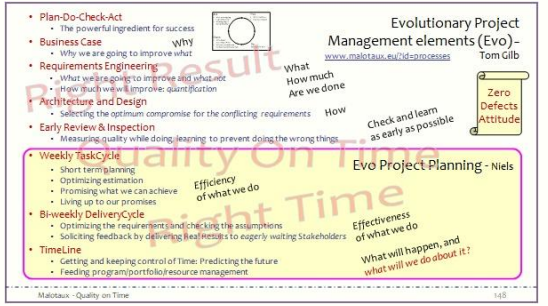
- now, rather than after 9 times the original budget

How would I start

- Don't tell people what to do
  - Montessori: Help me to do it myself
- Observing
  - Weekly planning
  - Gemba - 'walking around'
- Asking 'innocent' questions
  to let them recognize issues,
  which they usually are quite capable to solve
- If that's too tough, provide some 'suggestions'







No excuse anymore !

- Optimizing delivering the Right Results at the Right Time isn't really difficult
- I showed you some examples
- So, there is no excuse anymore
  if you're not sure, just ask !

What are we going to do about it ?!

More

www.malotaux.eu/?id=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/?id=insp Document Inspection pages

Optimizing Project Execution

Planning is important – execution of the plan even more

www.malotaux.eu/?id=conferences

www.malotaux.eu/?id=booklets

Niels Malotaux

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

From 60hr to 26hr, delivering better results

- One of three in a team insisting to work 60 hr the next week
- That probably won't all be done by the end of the week, right ?
- "Yes, but it >has< to be done !"
- Isn't that bad for your health? The others nodded
- After a long discussion, he gave in: planned 26 effort hours for that week
- Few weeks later he took me apart:

  Niels, thank you for your advice
  I was just nuts, getting too tired spending so many hours
  Now, every week I plan 26 net hours,
  getting done *more than ever before*