

Optimizing Project Execution

Planning is important - execution of the plan even more

www.malotaux.eu/conferences

www.malotaux.eu/booklets

Niels Malotaux

Cobb's Paradox (1995):

We know why projects fail

We know how to prevent their failure

So why do they still fail ?

+31-6 5575 3604

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

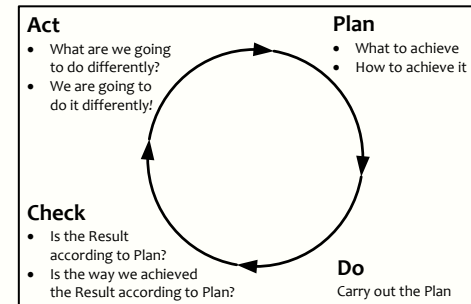
Quality On Time
Delivering
the Right Result
at the Right Time

Optimizing project execution

- Did you ever meet a deadline by diluting quality ?
- 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

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

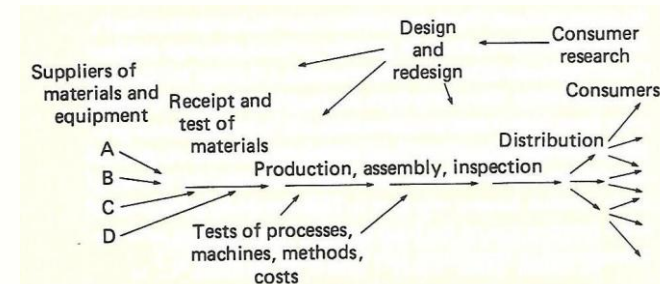
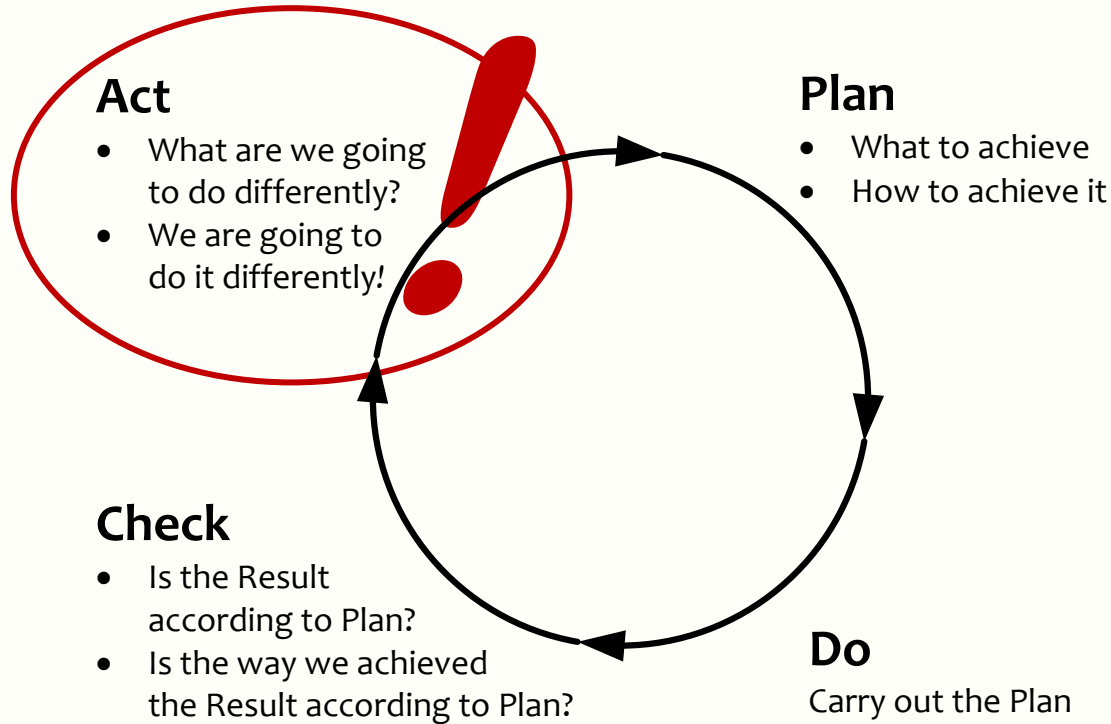


The essential ingredient: the PDCA Cycle

(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

Requirements with Planguage

ref Tom Gilb

quantifying the goal

SMART

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

Specific
Measurable

Benchmarks (Playing Field):

Past: 2 min [minimum, 2025], 8 min [average, 2025], 83 min [max, 2025]

Current: < 4 min [competitor y, May 2026] ← <who said this?>, <Survey Dec 2025>

Record: 57 sec [competitor x]

Wish: < 2 min [2027Q3, new system available] ← CEO, 19 Jan 2026, <document ...>

Attainable

Requirements:

Time

Traceable

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

Realizable

Impact Estimation principle

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

Possible solutions to achieve it

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

At what 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	<u>Benefits</u> Cost	<u>Benefits</u> Cost	<u>Benefits</u> Cost	

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

Note: If we use Ai to generate the numbers, we miss the deep thinking and debating !

Quality

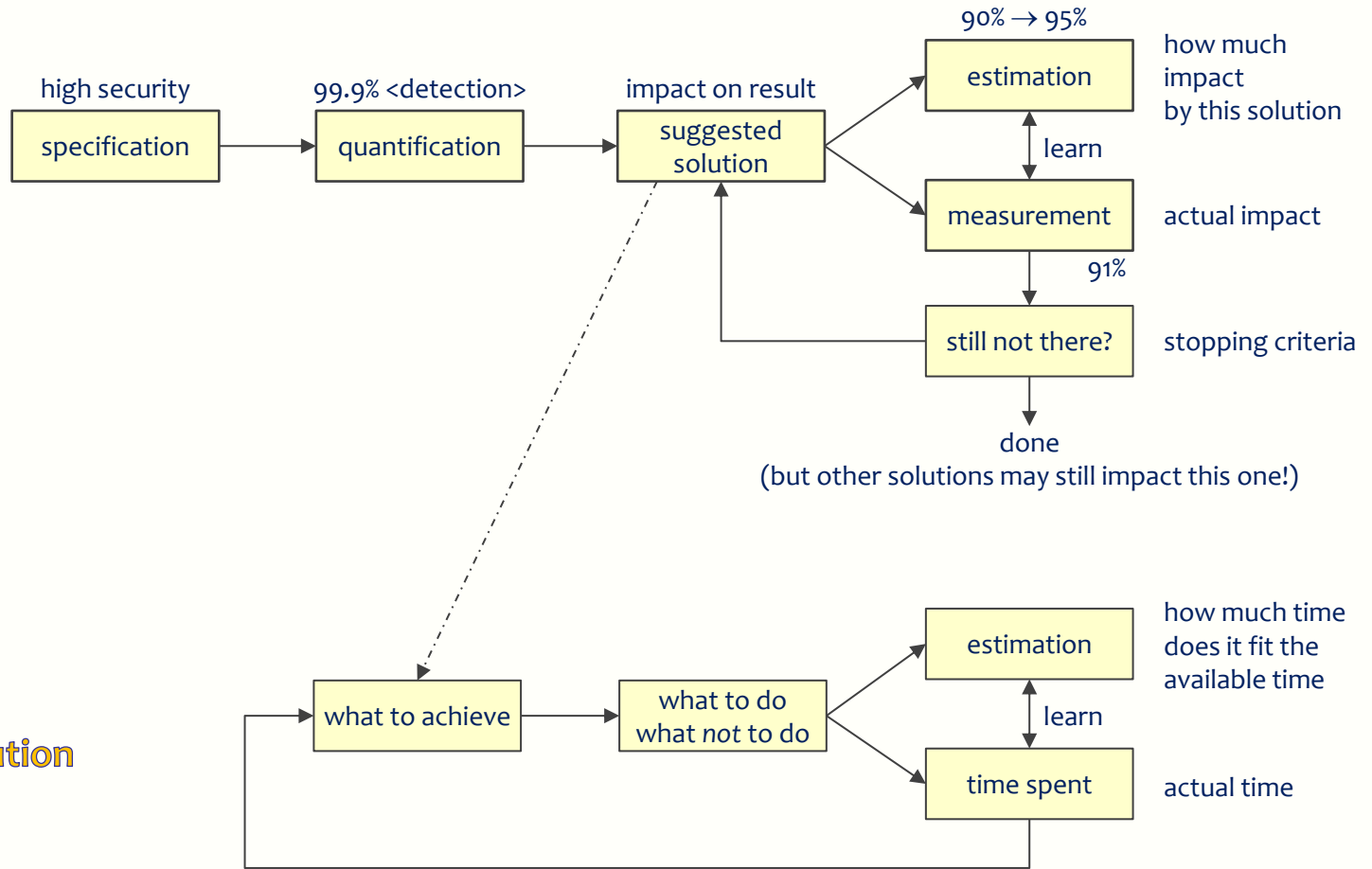
organising the

- what
- why
- for whom
- how much

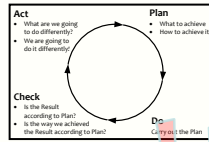
on Time

organising the execution

- how
- when



- **Plan-Do-Check-Act**
 - The powerful ingredient for success



- **Business Case**

Why

- 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

What
How much
Are we done

How

Check and learn
as early as possible

Zero
Defects
Attitude

- **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

Effectiveness
of what we do

- **TimeLine**

- Getting and keeping control of Time: Predicting the future
- Feeding program/portfolio/resource management

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

Evolutionary Project Management elements (Evo)

Tom Gilb

Evo Project Execution

Niels

Weekly TaskCycle plan

- What can we learn from last week (reflecting)
- What should we prevent the coming week (preflecting)
- 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 !

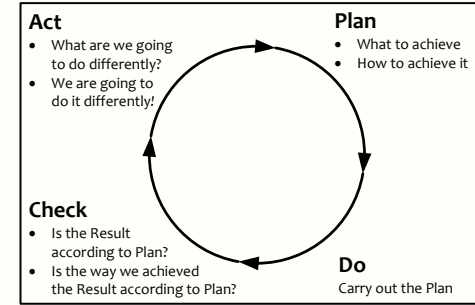
2/3 is default start value - works well with development work



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	

Reflecting

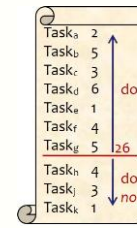
- Was all planned work really done ?
- If a Task was not completed, we learn:
 - Time spent but needed more time ? → effort estimation problem
 - What did I think then, what do I know now → learn (Check and Act)
 - Time not spent? → time management problem
 - Too much distraction
 - Too much time spent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



Immediate
consumption
of metrics

Task _a	2	↑ do
Task _b	5	
Task _c	3	
Task _d	6	
Task _e	1	
Task _f	4	
Task _g	5	
Task _h	4	↓ do not
Task _j	3	
Task _k	1	

Preflecting with 'innocent' questions



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

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 some work ?)
 - **Learning**
 - **Helping**
 - **Socializing**

*Modulation
costs less than
Preparation*

Why is this important ?

- TaskCycle Planning is **not** just planning the work for the coming week
- It exposes issues immediately
- Half of what people do in their work later proves to have been unnecessary
- During the TaskCycle planning we can very efficiently see
 - What our colleagues think they're going to do
 - Make sure we're all going to work on the most important things
 - Not on unnecessary things
 - In line with the architecture and design
 - Leading most efficiently to the goal of the delivery
 - Everyone knows exactly what's going to happen, what not, and why

Update: 19 Feb 2024
When selected, rows with 'ok' will show pink rather than white, indicating that the row cannot be changed anymore when 'ok'.

Last update: 02 Mar 2024 16:09:27 (CET)

Tasks

DeliveryCycle: Del 1
memo for Del 1

note for Del 1

TaskCycle: Cyc 2
memo for Cyc 2

note for Cyc 2

me niels
project/team TextCan
today 2024-03-02 00:09

add
delete

filterOff
prev
next

cycle: 1
plan: 1

hours of: niels
in cycle: Cyc 2

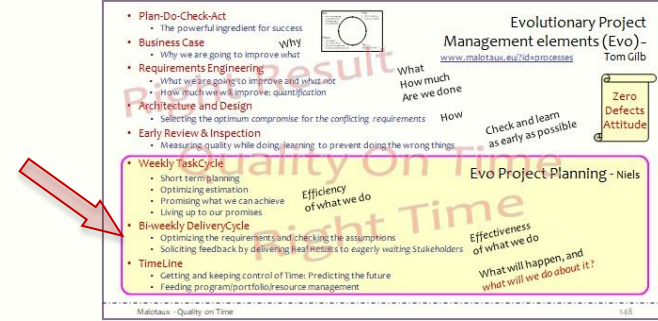
total: 26
done: 0
to do: 26

edit other:
☐ names
☐ task-cycles
☐ close

print
help

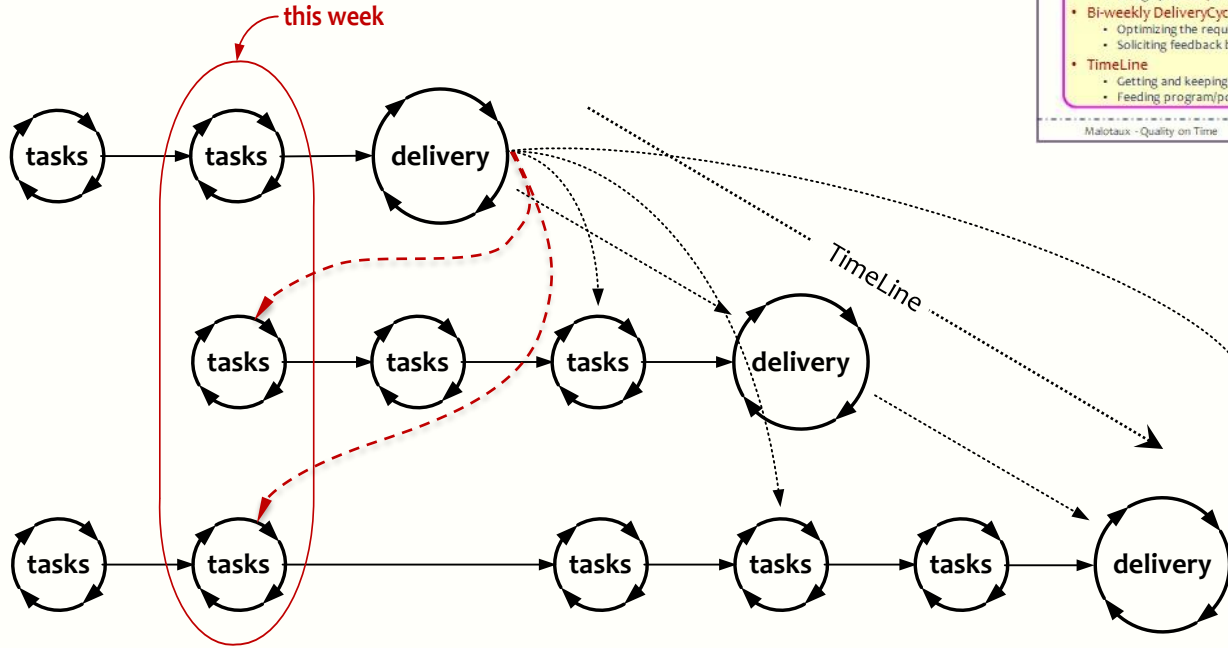
# id	# delCyc	delDue	# taskCyc	cycDue	# name	# hrs	ok	description
9	Del 1	2024-03-16	Cyc 2	2024-03-16	niels	3		task description 9
8	Del 1	2024-03-16	Cyc 2	2024-03-16	niels	4		task description 8
7	Del 1	2024-03-16	Cyc 2	2024-03-16	niels	3		task description 7
6	Del 1	2024-03-16	Cyc 2	2024-03-16	niels	5		task description 6
5	Del 1	2024-03-16	Cyc 2	2024-03-16	niels	8		task description 5
4	Del 1	2024-03-16	Cyc 2	2024-03-16	niels	3		task description 4
3	Del 1	2024-03-16	Cyc 1	2024-03-09	niels	6	ok	task description 3
2	Del 1	2024-03-16	Cyc 1	2024-03-09	niels	3	ok	task description 2
1	Del 1	2024-03-16	Cyc 1	2024-03-09	niels	1	ok	task description 1

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

Tasks feed Deliveries



Evolutionary Project Management elements (Evo)- Tom Gilb

www.malotau.eu/?id=processes

Zero Defects Attitude

Check and learn as early as possible

What How much Are we doing

Why

How

Efficiency of what we do

Effectiveness of what we do

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

TimeLine

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

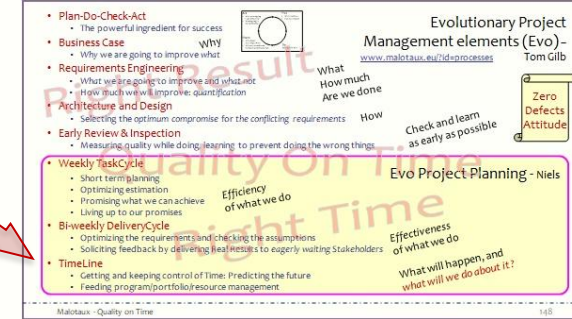
- Getting and keeping control of Time: Predicting the future
- Feeding program/portfolio/resource management

Malotau - Quality on Time

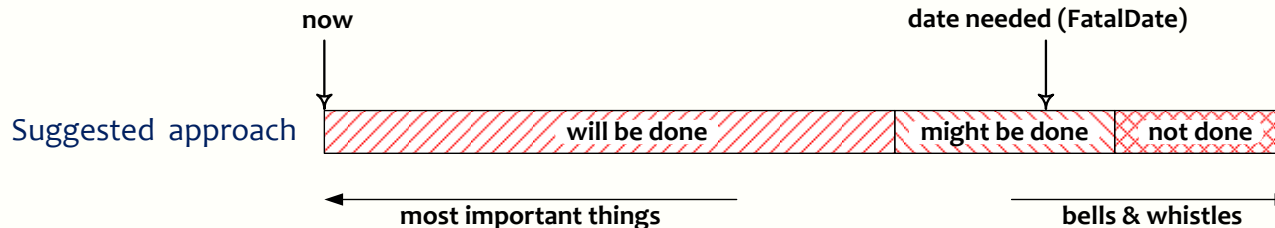
148

TimeLine

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

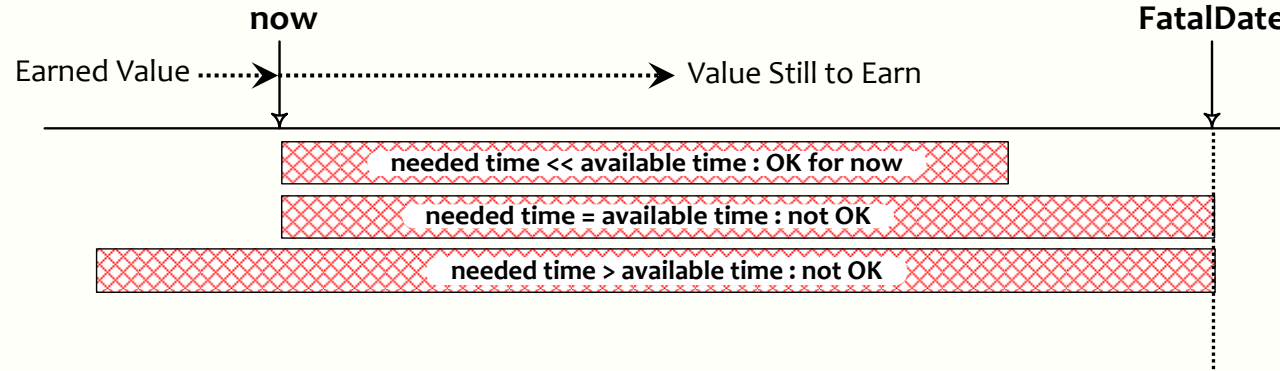


Agility
 Quickly and appropriately
 responding to whatever is,
 and whatever happens
 Can be used anytime, anywhere



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

At the time they need it



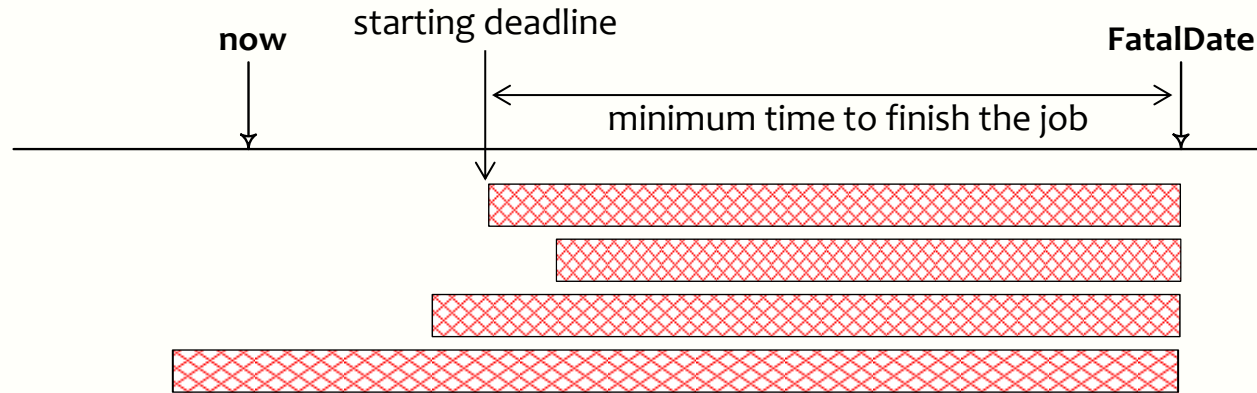
- Value Still to Earn
- versus
- Time Still Available

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



Plan-Do-Check-Act

- The powerful ingredient for success

Business Case

- Why we are going to improve what

Requirements Engineering

- What we're 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
- Selecting feedback by delivering real results to eagerly waiting Stakeholders

Time/Lime

- Getting and keeping control of Time: Predicting the future
- Feeding program/portfolio resource management

Evolutionary Project Management elements (Tom Gilb)

www.magnet4u.eu/gilbprocesses

Zero Defects Attitude

Evolutionary Project Planning - Niels

What How much Are we done

Check and learn as early as possible

Efficiency of what we do

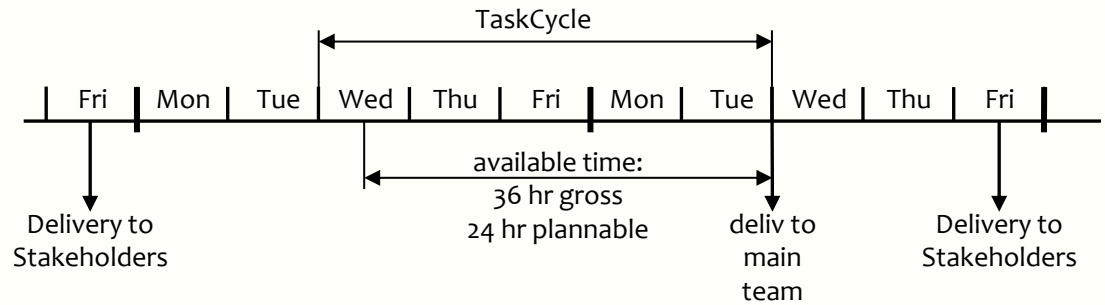
Effectiveness of what we do

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



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

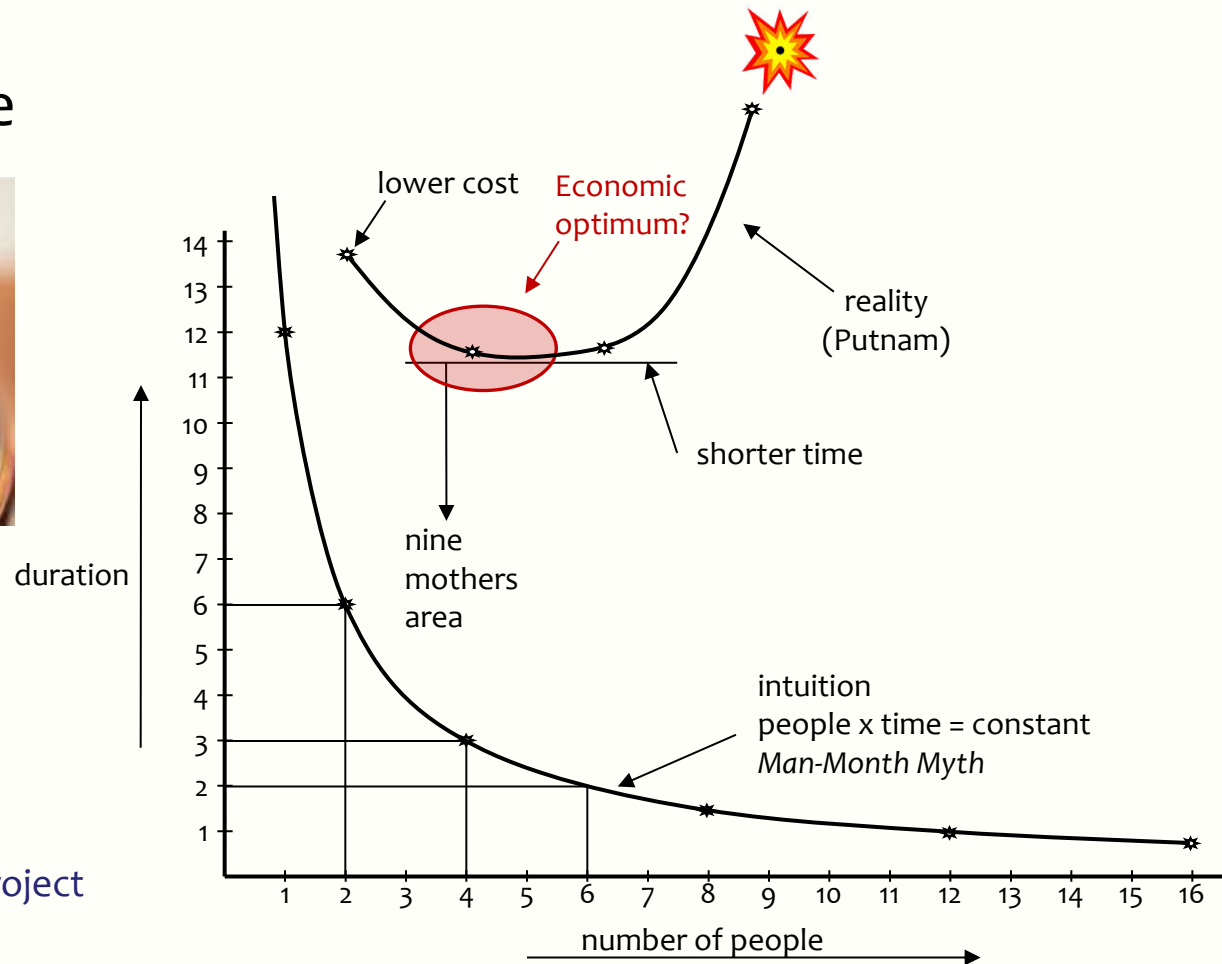
1. Hoping for the best (fatalistic)
2. Going for it (macho)
3. Working Overtime (fooling ourselves and our boss)
4. Moving the deadline
 - Parkinson's Law
 - Work expands to fill the time for its completion (we spend the time given)
 - Student Syndrome
 - Starting as late as possible, only when the pressure of the FatalDate is really felt

5. Adding people



Brooks' Law (1975)

Adding people to a late project
makes it later



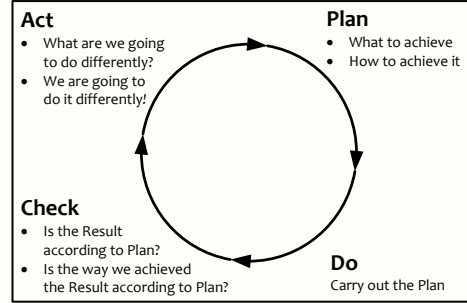


6. Saving time

Continuous
elimination of waste
(www.malotaux.eu/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
 - 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
 - Spending 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, right order
- **TimeBoxing** - much more efficient than FeatureBoxing



(www.malotaux.eu/evo)

(www.malotaux.eu/designlog)

(www.malotaux.eu/projectmanagement)

(www.malotaux.eu/PDCA)

(www.malotaux.eu/timeline)

(www.malotaux.eu/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'

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, 28 hr per week at 40hr workweek)
- 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

Malotau - BCS June 2015 www.malotau.eu/weeklyplanning 13

'Innocent' questions for prefection



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.

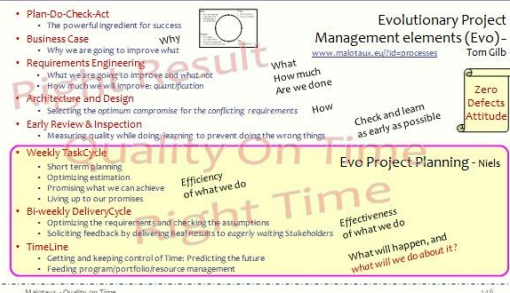
Never challenge an estimate! If we don't understand the estimate, only say:

- What are you planning to do?
- Now estimate again...
- If they insist:
- Let them find out themselves

- 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?

Malotau - BCS June 2015 16

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 we
 - 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 requirement and checking the assumptions
 - Soliciting feedback by delivering results to eagerly waiting Stakeholders
- Timeline
 - Getting and keeping control of Time: Predicting the future
 - Feeding program/portfolio/resource management

What How much Are we doing

Check and learn as early as possible

Evo Project Planning - Nils

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

Zero Defects Attitude

Malotau - Quality on Time 145

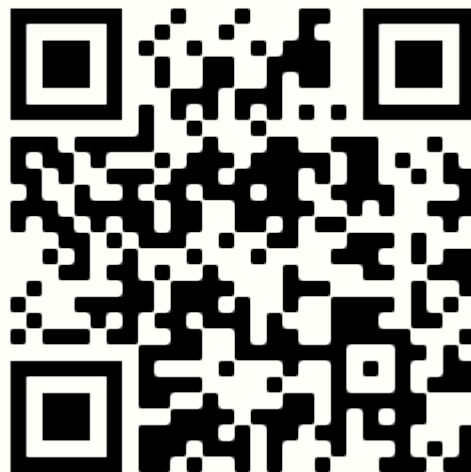
No excuse anymore !

- 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 ?!

- 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
- 9a 予測可能なプロジェクト - 適切なタイミングで適切な結果を出す方法 - same, in Japanese
- RS Measurable Value with Agile (Ryan Shriver - 2009)
Use of Evo Requirements and Prioritizing principles



Optimizing Project Execution

Planning is important - execution of the plan even more

www.malotaux.eu/?id=conferences

www.malotaux.eu/?id=booklets

Niels Malotaux

+31-6 5575 3604

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

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"

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

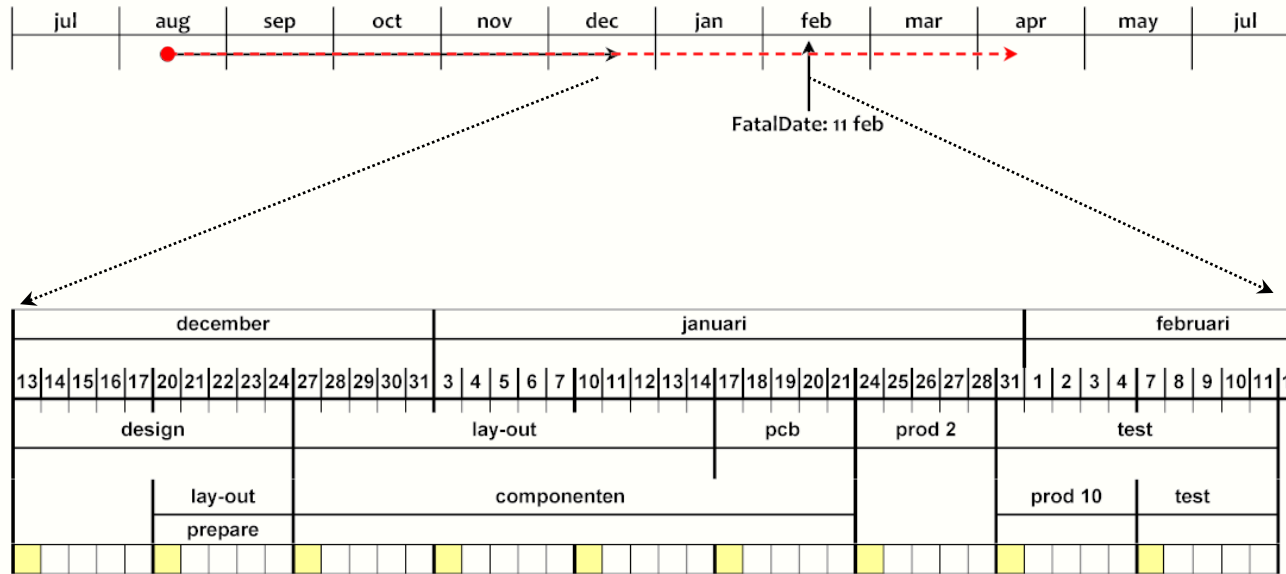
Client statement

- 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

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



cycle	who	task description	estim	real	done	issues		
3	John	Net time available: 26						
		aaaaaaaaa	3	3	yes			
		bbbbbbbbb [Paul]	1					
		ccccccccc	5	13	yes			
		dddddddd	2					
		eeeeeeee	3	2				
		fffffffffff	2	1				
		ggggggggg	6	7	yes			
		hhhhhhhhh	4					
			26	26				
4	John	Net time available: 26						
		jjjjjjjjjjj	3			for team x		
		kkkkkkkkkk				for team x		
		mmmmm	5			for team x		
		nnnnnnnn				for team x		
		oooooooo				for team y		
		qqqqqqqq	12			for team y		
		rrrrrrrrrrr	6			for team y		
		ssssssssss				for team y		
		tttttttttt				for team y		
			26					

TaskCycle Analysis
(reflecting)

learning

TaskCycle Planning
(preflecting)

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