Evolutionary (Evo) Principles

Niels Malotaux

It's not a method

Just a bunch of add-ins to what you are already doing

Perhaps some alternatives …

niels@malotaux.nl

www.malotaux.nl/conferences

Niels Malotaux

- Independent Project and Organizational Coach
- Expert in helping optimizing performance

- Helping projects and organizations very quickly to become
  - More effective – doing the right things better
  - More efficient – doing the right things better in less time
  - Predictable – delivering as predicted
- Getting projects on track

Result Management

Simple questions

- What do you have to have achieved by the end of next week ?
- Will you succeed ?
- How do you know ?
- What do you have to do to achieve it ?
- How much time does it take ?
- How much time do you have to do it ?
- Does it fit ?
- If not, what would you do ?

Ultimate Goal of a What We Do

Quality on Time

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

Providing the customer with

- what he needs
- at the time he needs it
- to be satisfied
- to be more successful than he was 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

Preflection, foresight, prevention

Do we really 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*

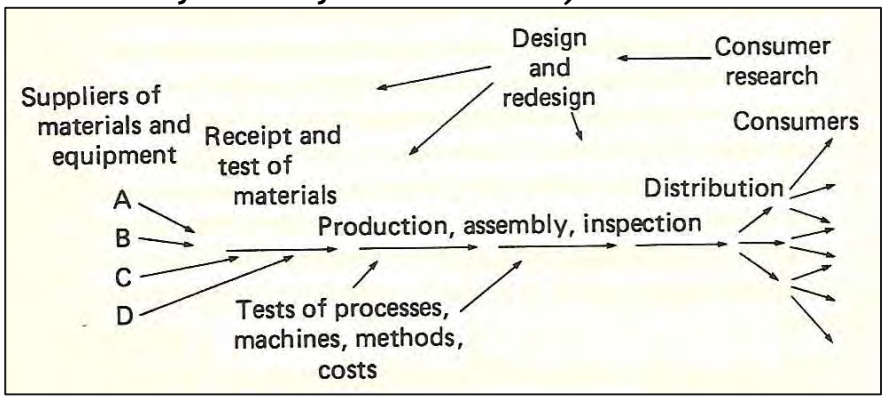
- Hindsight is easy, but reactive
- Foresight is less easy, but proactive
- Reflection is for hindsight and learning
- Preflection is for foresight and prevention

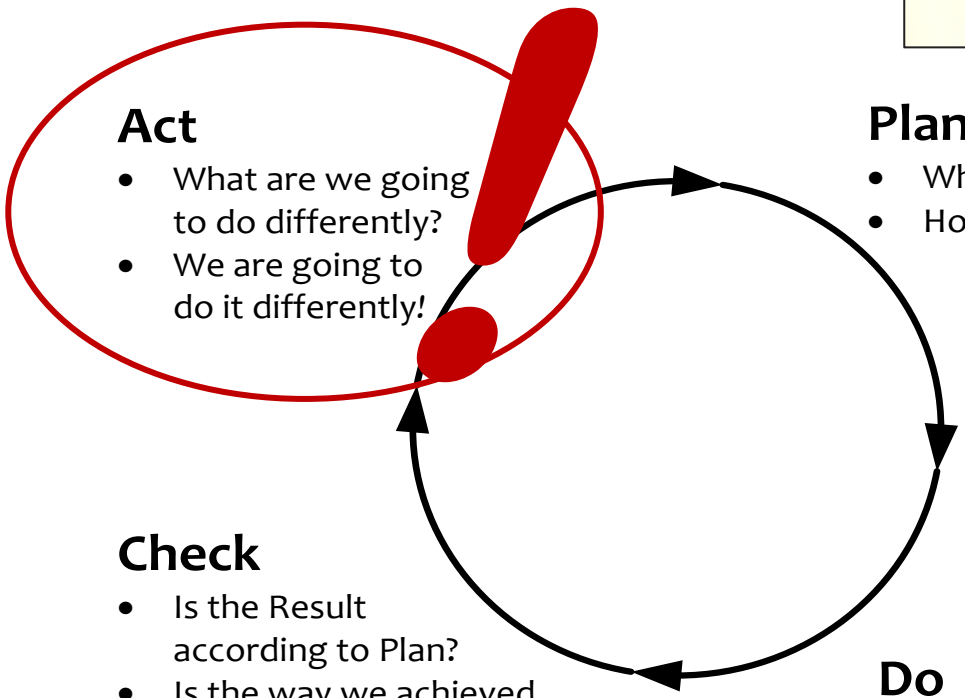
Only with *prevention* we can save precious time

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

The essential ingredient: the PDCA Cycle

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





Act

- What are we going to do differently?
- We are going to do it differently!

Plan

- What to achieve
- How to achieve it

Check

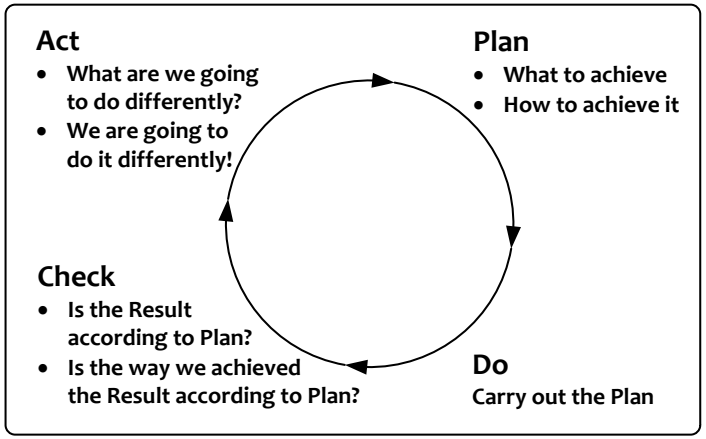
- Is the Result according to Plan?
- Is the way we achieved the Result according to Plan?

Do

Carry out the Plan

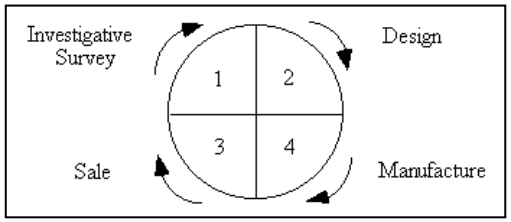
Deming

Evo



- Evo (short for Evolutionary...) uses PDCA consistently
- Applying the PDCA-cycle
  actively, deliberately, rapidly and frequently,
  for *Product*, *Project* and *Process*, based on ROI and highest value
- Quantifying, estimating, measuring, learning
- Combining Planning, Requirements- and Risk-Management into *Result Management*
- We know we are not perfect, but the customer shouldn't find out
- Evo is about delivering Real Stuff to Real Stakeholders
  doing Real Things
  "Nothing beats the Real Thing"
- Projects seriously applying Evo, routinely conclude successfully on time, or earlier

Real Stuff to Real Stakeholders doing Real Things

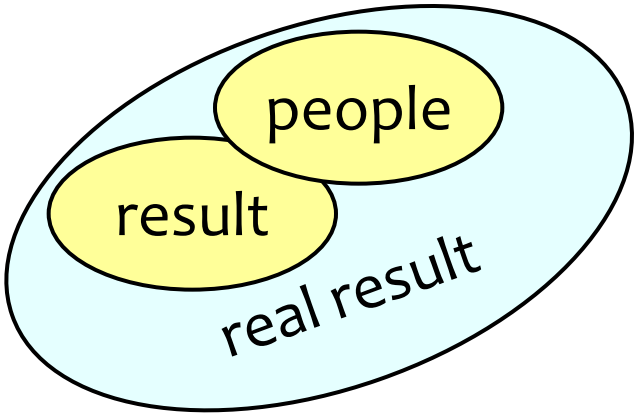


Did you ever do a 'Demo' ?

- Give the delivery to the stakeholders
- Zip your mouth
- Keep your hands handcuffed on your back
- and o-b-s-e-r-v-e what happens
- Seeing what the stakeholders actually do provides real feedback
- Then we can 'talk business' with the stakeholders

- Is this what you do ?

Stakeholders are (not only) people



- Every project has some 30±20 Stakeholders
- Stakeholders have a stake in the project
- The concerns of Stakeholders are often contradictory
  - Apart from the Customer *they don't pay*
  - So they *have no reason to compromise !*
- Some Stakeholders are victims of the project
  They have no reason for the project to succeed, on the contrary
- Project risks, happening in almost every project
- No excuse to fail !

Victims can be a big Risk

What are the Requirements for a Project ?

- Requirements are what the Stakeholders require

but for a project ...

- Requirements are the set of stakeholder needs that the project is *planning to satisfy*

- The set of Stakeholders doesn't change much
- Do you have a checklist of possible Stakeholders ?

No Stakeholder?

- No Stakeholder: no requirements
- No requirements: nothing to do
- No requirements: nothing to test
- If you find a requirement without a Stakeholder:
  - Either the requirement isn't a requirement
  - Or, you haven't determined the Stakeholder yet
- If you don't know the Stakeholder:
  - Who's going to pay you for your work?
  - How do you know that you are doing the right thing?
  - When are you ready?

The Simplest and Best Agile Method - 'Evo'

Tom Gilb

1. Gather from all the key stakeholders the top few (5 to 10) most critical goals that the project needs to deliver
   Give each goal a reference name (a tag)
2. For each goal, define a scale of measure and a 'final' goal level
   For example: Reliable: Scale: Mean Time Before Failure, Goal: 1 month
3. Define up to 4 budgets for your most limited resources
   For example, time, people, money, equipment
4. Write up these plans for the goals and budgets
   Try to ensure this is kept to only one page
5. Negotiate with the key stakeholders to formally agree the goals and budgets
6. Plan to deliver some benefit
   that is, progress towards the goals in weekly (or shorter) increments (Evo steps)
7. Implement in Evo steps
   Report to sponsors after each Evo step (weekly, or shorter) with your best available estimates or measures, for each performance goal and each resource budget.
   On a single page, summarize the progress to date towards achieving the goals and the costs incurred
8. When all Goals are reached: 'Claim success and move on'
   Free remaining resources for more profitable ventures

"Deceptively simple"

Requirements / Goals using Planguage

Tom Gilb

SMART

Specific

Measurable

Attainable

Realizable

Time

Traceable

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

Benchmarks (Playing Field):

Past: 2 min [minimum, 2016], 8 min [average, 2016], 83 min [max, 2014]

Current: < 4 min [competitor y, Jan 2018] ← <who said this?>, <Survey April 2018>

Record: 57 sec [competitor x, Jan 2016]

Wish: < 2 min [2020Q3, new system available] ← CEO, 19 Jan 2018, <document ...>

Requirements:

Tolerable: < 10 min [99%, Q4] ← SLA

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

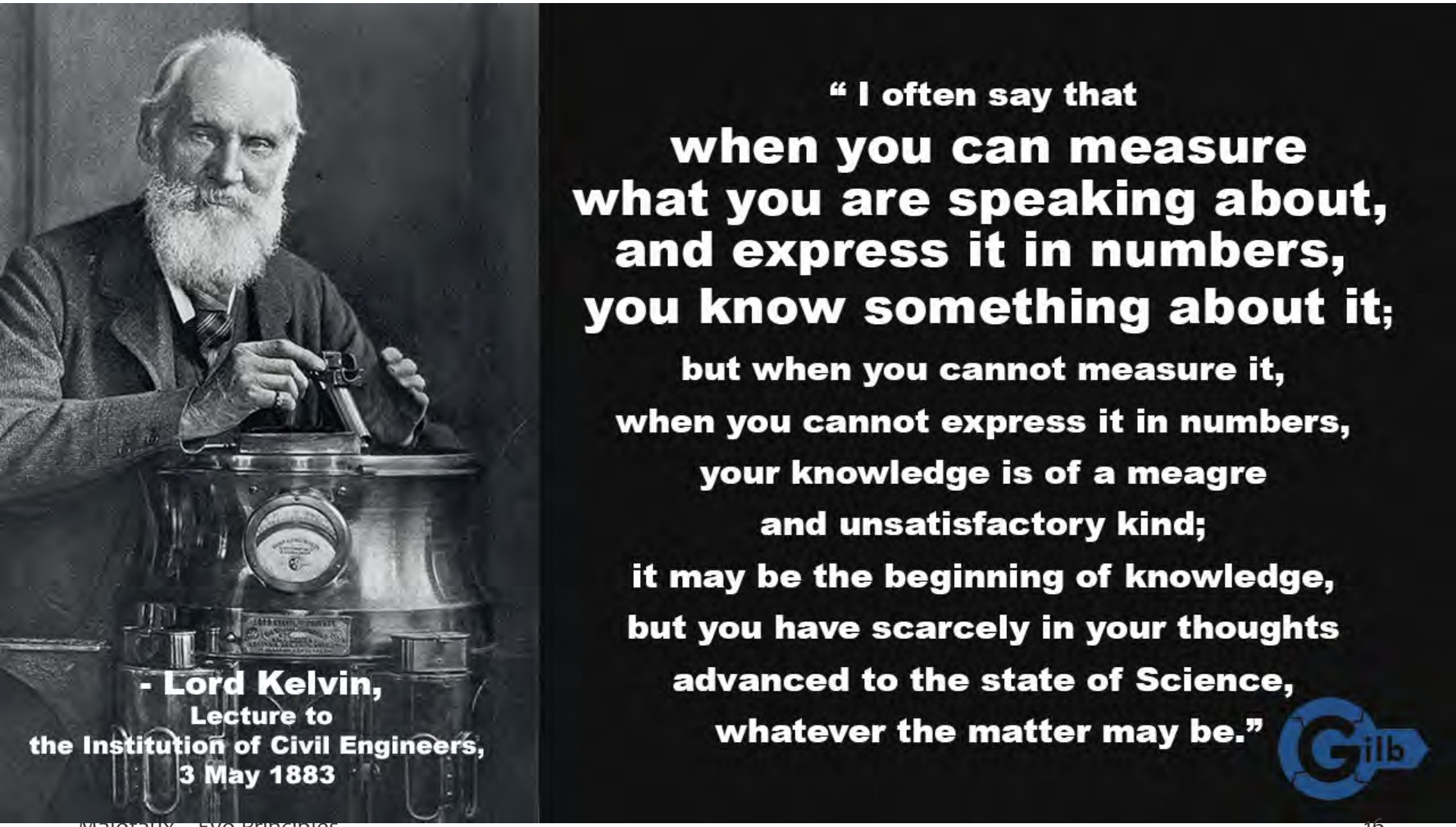
Goal: < 15 min [99%, Q2], < 10 min [99%, Q3], < 5 min [99%, Q4] ← marketing

Tom Gilb quote

Tom Gilb

- 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

Estimating, measuring, learning (implies quantification !)



It’s not about the functions

- Banks bank for thousands of years
- What do they do ?
- How can they handle their business ?

Improving on *existing* qualities

- Measured values !

	V8.5	V9.0	
• Usability.Productivity:			
• Time to set up a typical specified report	65	20	min
• Time to generate a survey	120	0.25	min
• Time to grant access to report, distribute logins to end-users	80	5	min
	265	25.25	min
• Usability.Intuitiveness:			
• Time for medium experienced programmer to find out how to do ...	15	5	min
• Capacity.RuntimeConcurrency			
• Max number of concurrent users, click-rate 20 sec, response time < 0.5 sec	250	6000	users

after FIRM / Gilb 2005

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

Zero Defects Attitude

Check as early as possible

Right Result

Quality On Time

Right Time

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 ?

How about your requirements ?

Are they

- Unambiguous (to the intended readership)
- Clear to test
- Quality requirements expressed quantitatively
- No design (solutions) in the requirements (goals)

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

Zero Defects Attitude

Check as early as possible

Right Result

Quality On Time

Right Time

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

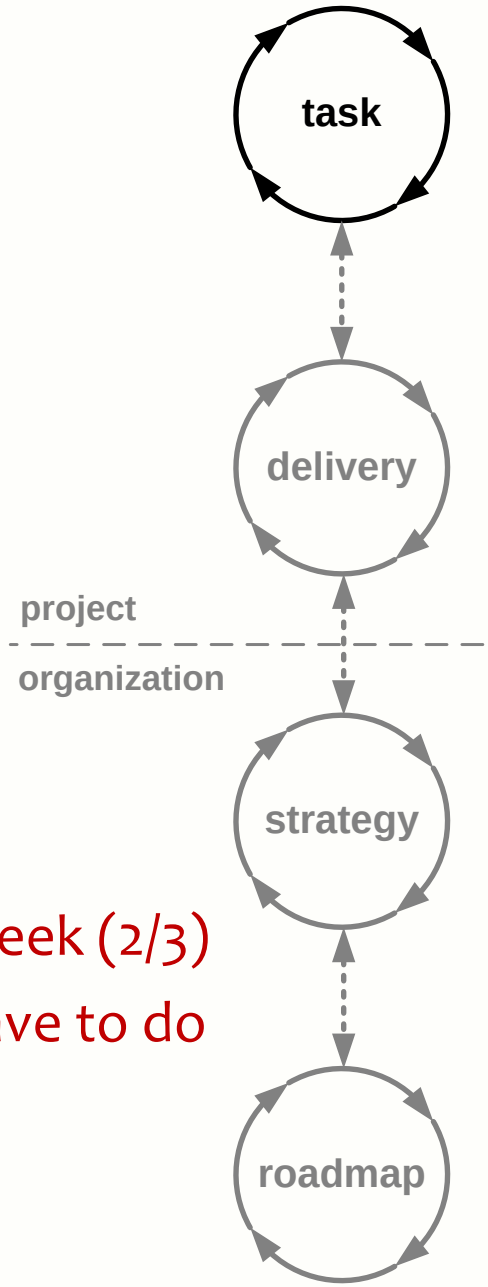
What will happen and what will we do about it ?

Evolutionary Planning
prevention is better than cure

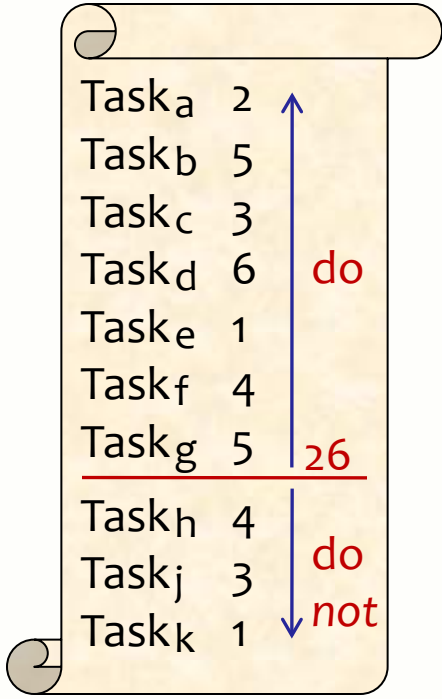
- Weekly TaskCycle
  - Short term planning
  - Optimizing estimation
  - Promising what we can achieve
  - Living up to our promises
- Bi-weekly DeliveryCycle
  - Optimizing the requirements and checking the assumptions
  - Soliciting feedback by delivering Real Results to *eagerly waiting* Stakeholders
- TimeLine
  - Getting and keeping control of Time: Predicting the future
  - Feeding program/portfolio/resource management

Evo Planning: 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



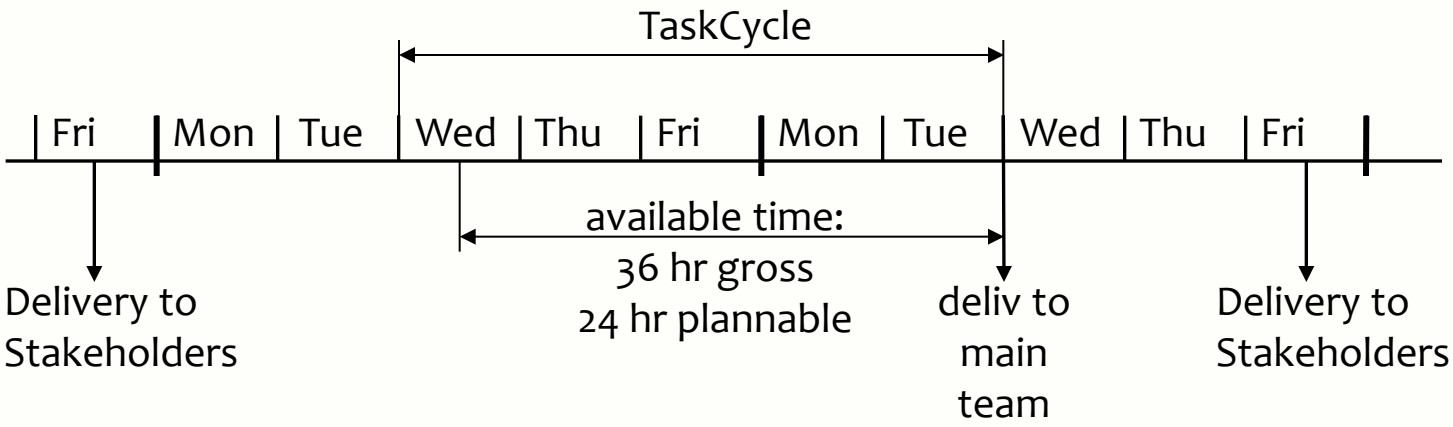
Every week we plan



- 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
- *Write it down ! Our fuzzy mind isn't good enough !*

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

Designing a Delivery



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

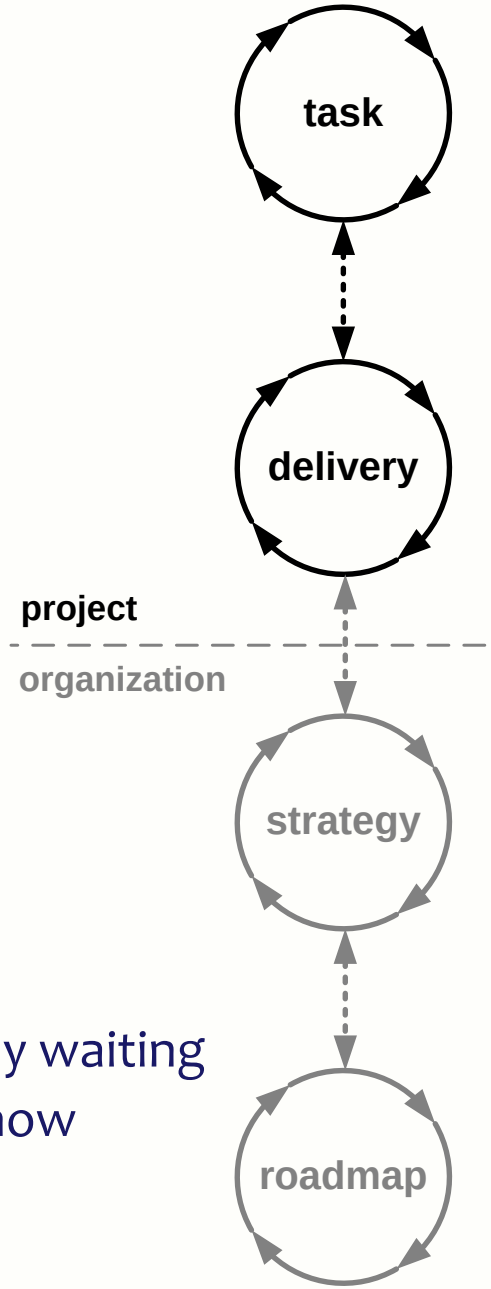
Gregory	
Draft design	6
Finish design	6
Work for deliv	3
-	1
-	2
-	2
-	3
-	5
-	6
XMLa	4
XMLb	4
Total	42

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

Jerome	
XMLa	3
XMLb	3
...	

DeliveryCycle

- Are we *delivering* the right things,
  in the right order
  to the right level of detail for now
- Optimizing requirements
  and checking assumptions
  1. What will generate the optimum feedback
  2. We deliver to *eagerly waiting* stakeholders
  3. If they're not eagerly waiting, but should,
     we deliver them *juicy bits*, to make them eagerly waiting
     - What will make Stakeholders more productive now
- Not more than 2 weeks



Now we are already much more efficient

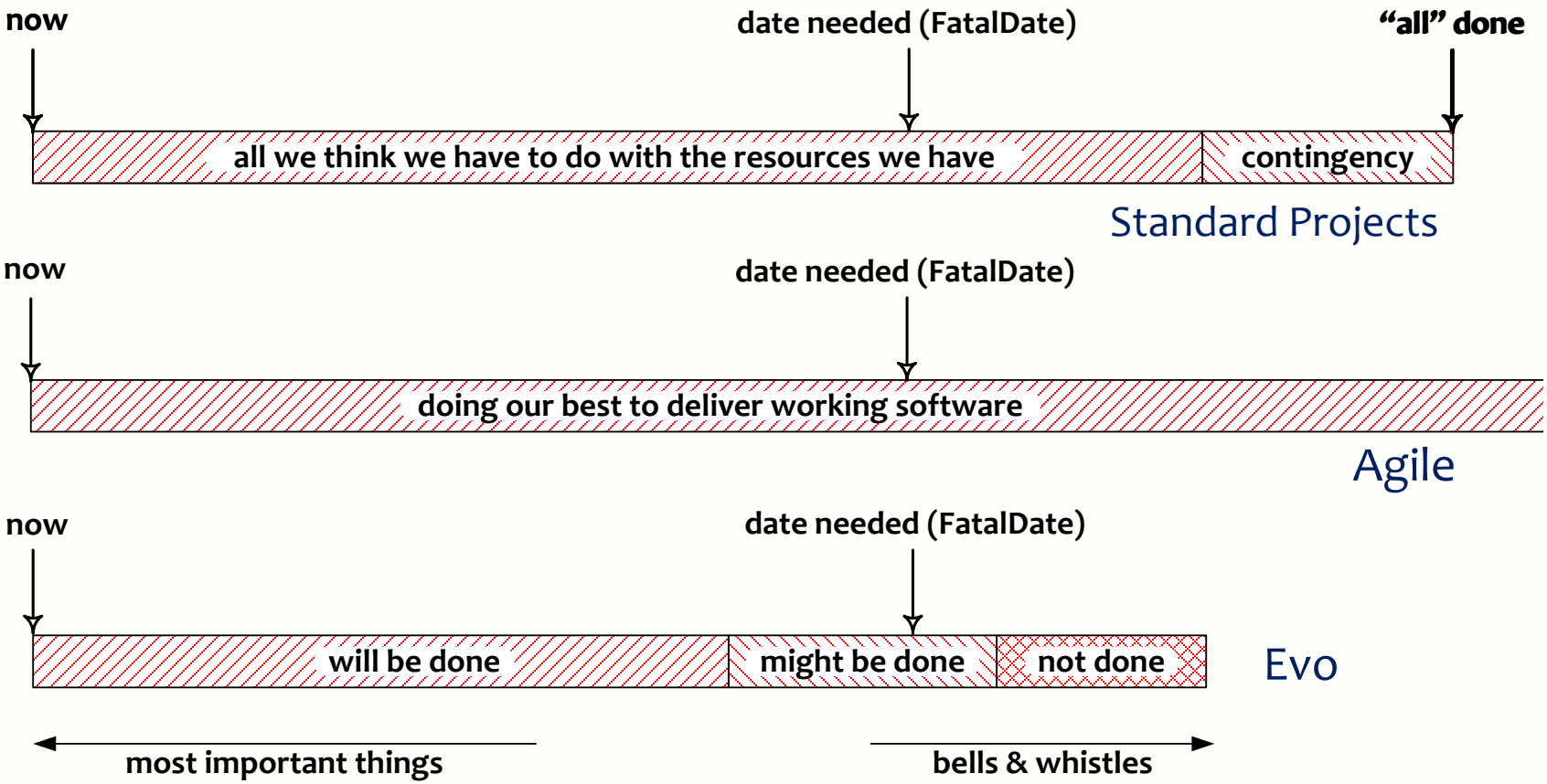
- Organizing the work in very short cycles
- Making sure we are doing the right things
- Doing the right things right
- Continuously optimizing (what not to do)
- So, we already work more efficiently

but ...

- How do we make sure the whole project is done on time ?

TimeLine

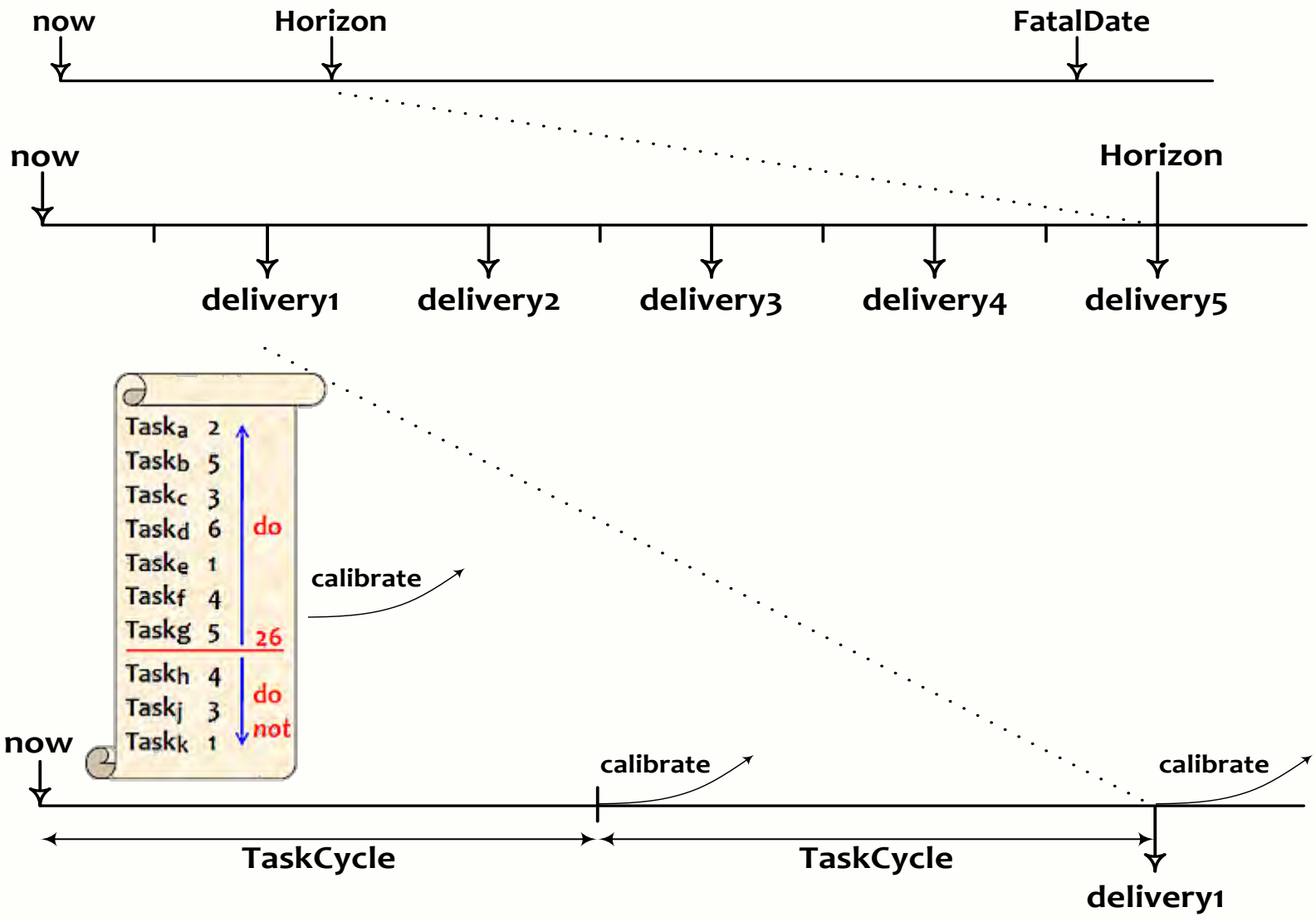
What the customer wants, he cannot afford



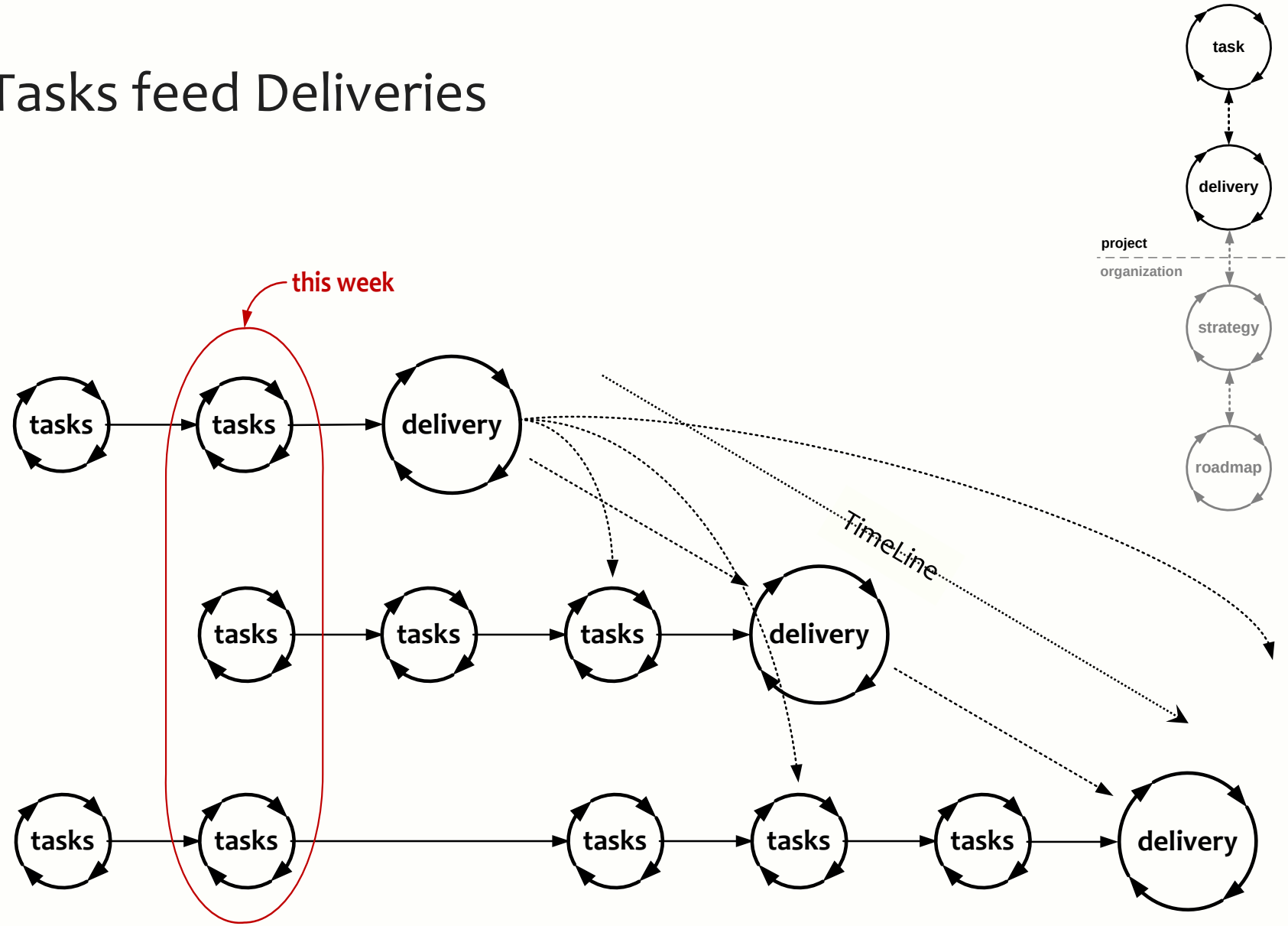
- Better 80% 100% done, than 100% 80% done
- Let it be the most important 80%

If it easily fits …

Result to Tasks and back

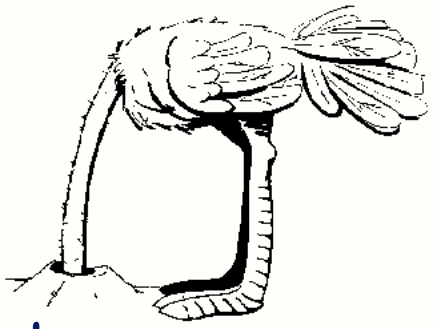


Tasks feed Deliveries



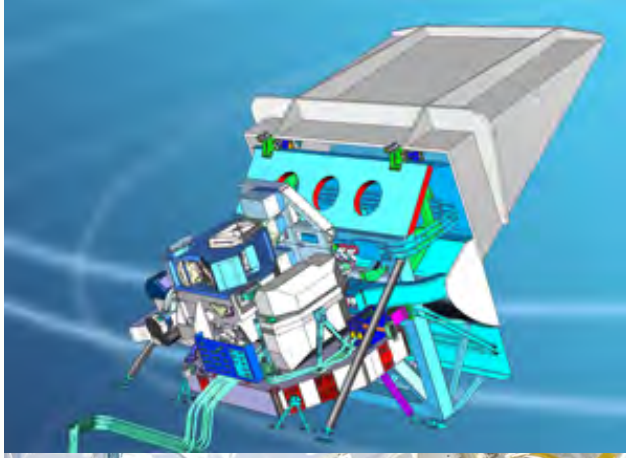
TimeLine

- The TimeLine technique doesn't solve our problems
- It helps to expose the real status early and continuously
- Instead of accepting the undesired outcome,
  we *do something about it*
- The earlier we know, the more we can do about it
- We start saving time from the very beginning
- We can save a lot of time in any project,
  while producing a better outcome



If, and only if, we are serious about time !

Earth Observation Satellite



- Very experienced Systems Engineers
- Using quantified requirements routinely
- 8 year pure waterfall project (imposed by ESA)
- 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
- Recently: delivered 1 day early (instead of expected 1 year late)
- Savings: at least 40 man-year (about €6M)
- How did they do that ?

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

Zero Defects Attitude

Check as early as possible

Right Result

Quality On Time

Right Time

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

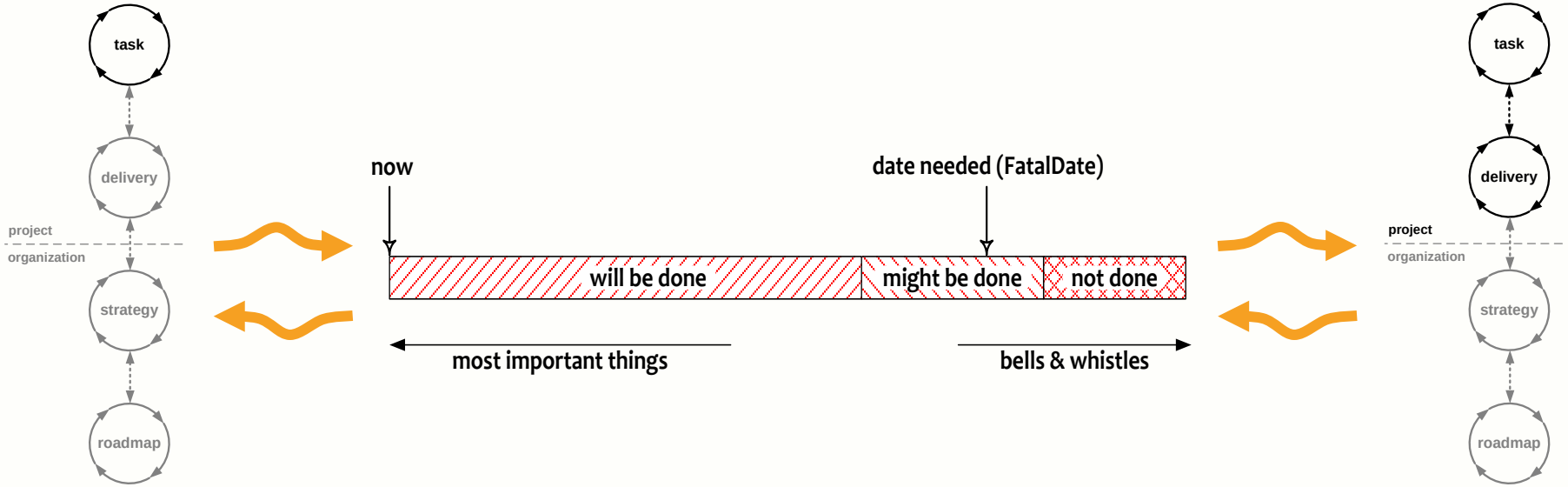
What will happen and what will we do about it ?

Evolutionary start pattern

- Evo day
  - Explanation of the Evo approach
  - Organizing the work of the coming week
  - Goal: at the end of the day, people of the team know what they are going to work on and why
- Weekly Evo day
  - Execution of the 3-step procedure

Evolutionary introduction pattern

1. Introducing *Tasks*
   How to organize the work → Short term view
2. Introducing *TimeLine*
   The *design* of the project → Longer term view
3. Introducing *Deliveries*
   Focusing on Results → Connecting long and short



What could we discuss about ?

- Evolutionary – Evo – Principles
- Cases
  - Introducing Evo immediately saves time while delivering better results
- Estimation exercise
  - Are we optimistic or realistic estimators
- Human Behaviour
  - Understanding human behaviour is the start to doing something with it
- How to move towards Zero Defects
  - Prevention is better than cure
  - Quality costs less
- Help ! We have a QA Problem !
  - TimeLine case
- How Systems Engineers learnt to meet all deadlines
  - Saving one year with > 40 people