Oxford
23 April 2009

Niels Malotaux

Controlling Project Risk *by Design*

N R Malotaux - Consultancy
The Netherlands
+31-30-2288868
niels@malotaux.nl
www.malotaux.nl

Niels Malotaux

Controlling Project Risk by Design

Niels Malotaux

Niels Malotaux is an independent Project Coach specializing in optimizing project performance. He has over 35 years experience in designing electronic hardware and software systems, at Delft University, in the Dutch Army, at Philips Electronics and 20 years leading his own systems design company. Since 1998 he devotes his expertise to helping projects to deliver Quality On Time: delivering what the customer needs, when he needs it, to enable customer success. To this effect, Niels developed an approach for effectively teaching Evolutionary Project Management (Evo) Methods, Requirements Engineering, and Review and Inspection techniques. Since 2001, he taught and coached over 100 projects in 25+ organizations in the Netherlands, Belgium, China, Germany, India, Ireland, Israel, Japan, Romania, South Africa and the US, which led to a wealth of experience in which approaches work better and which work less in the practice of real projects.

Niels puts development teams on the Quality On Time track and coaches them to stay there and deliver their quality software or systems on time, without overtime, without the need for excuses. Practical methods are developed, used, taught and continually optimized for:

- Evolutionary Project Management (Evo)
- Requirements Engineering and Management
- Reviews and Inspections.

Within a few weeks of turning a development project into an Evo project, the team has control and can tell the customer when the required features will all be done, or which features will be done at a certain date. Niels enjoys greatly the moments of enlightenment experienced by his clients when they find out that they can do it, that they are really in control, for the first time in their lives.

N R Malotaux
Consultancy

Niels Malotaux
project coach

Result Management

Bongerdlaan 53
3723 VB Bilthoven
The Netherlands
tel +31-30-228 88 68
fax +31-30-228 88 69
mob +31-6-5575 3604
niels@malotaux.nl
www.malotaux.nl

Controlling Project Risk *by Design*

Niels Malotaux

N R Malotaux Consultancy

+31-30-228 88 68 niels@malotaux.nl www.malotaux.nl

1

Niels Malotaux

Project Coach

- Evolutionary Project Management (Evo)
- Requirements Engineering
- Reviews and Inspections

Result Management

- Researching problems in projects
- Finding ways to fundamentally overcoming these problems
- Ploughing back into projects
- Tuning of the results (because theory isn't practice)

2

Risk Definition

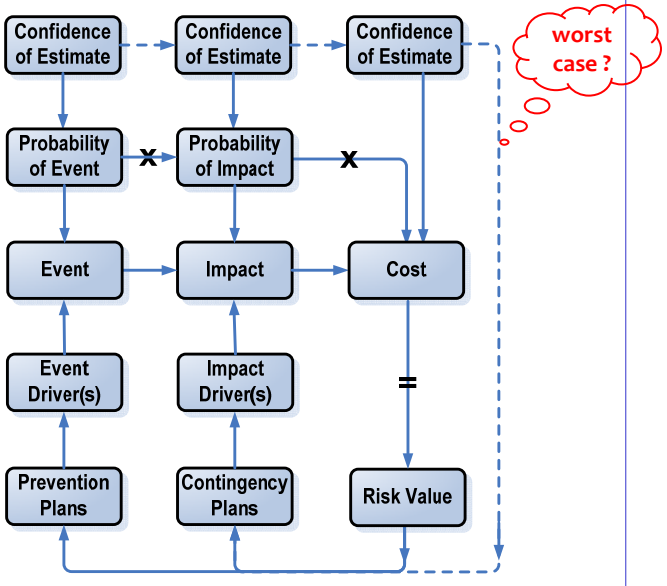
An uncertain event or condition that,
if it occurs,
has a negative effect
on a project's objectives

(PMBOK)

- 0% probability is not a risk
- 100% probability is an issue or a problem

3

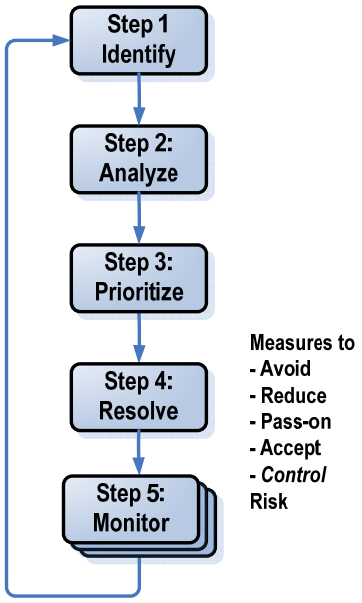
Risk Model



$$V_R = P_e * P_i * C$$

4

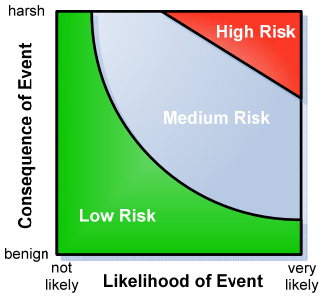
Risk Management

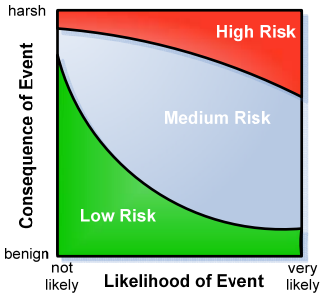


5

Prioritize Risk?

ref. INCOSE SE Handbook

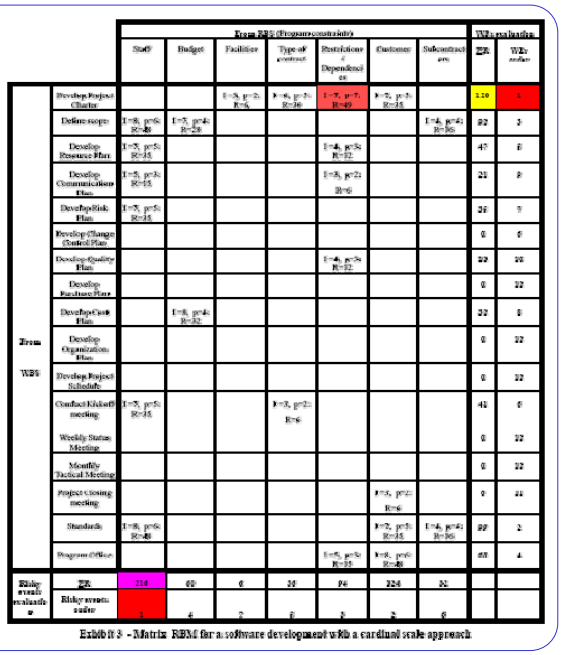




Risk Priority = Likelihood x Consequence ??

6

Mathematical Risk Management can be risky



From WBS	From RBS (Program constraints)							WP's evaluation	
		Staff	Budget	Facilities	Type of contract	Restrictions / Dependencies	Customer	Subcontractors	ER	WPs number
	Develop Project Charter			I=3, p=3; R=9	I=6, p=3; R=18	I=7, p=7; R=49	I=7, p=3; R=21		110	1
	Define scope	I=8, p=6; R=48	I=7, p=4; R=28					I=4, p=4; R=16	92	3
	Develop Resource Plan	I=3, p=5; R=15				I=4, p=8; R=32			47	6
	Develop Communication Plan	I=3, p=5; R=15				I=3, p=2; R=6			21	9
	Develop Risk Plan	I=3, p=5; R=15							15	7
	Develop Change Control Plan								0	0
	Develop Quality Plan					I=4, p=8; R=32			32	16
	Develop Procurement Plan								0	22
	Develop Cost Plan		I=8, p=4; R=32						32	8
	Develop Organization Plan								0	22
	Develop Project Schedule								0	22
	Conduct Kickoff meeting	I=3, p=5; R=15			I=3, p=2; R=6				41	9
	Weekly Status Meeting								0	22
	Monthly Tactical Meeting								0	22
	Project closing meeting						I=3, p=2; R=6		9	22
	Standards	I=8, p=6; R=48					I=7, p=3; R=21	I=4, p=4; R=16	89	2
	Program Office					I=5, p=5; R=25	I=8, p=6; R=48		98	4
Risky events evaluation	ER	214	60	0	30	96	224	32		
	Risky events number	1	0	0	0	2	2	0		

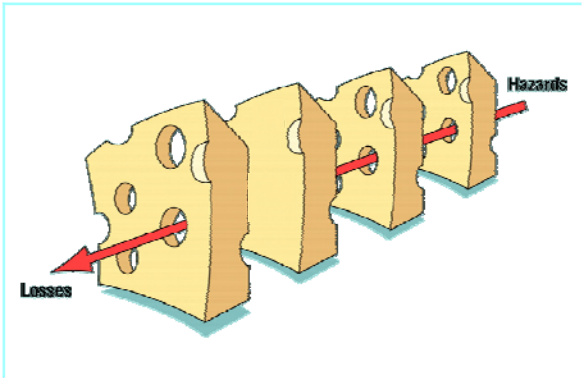
Exhibit 3 - Matrix RBM for a software development with a cardinal scale approach

ref
Carlo Rafele,
David Hillson,
Sabrina Grimaldi

7

Swiss Cheese model

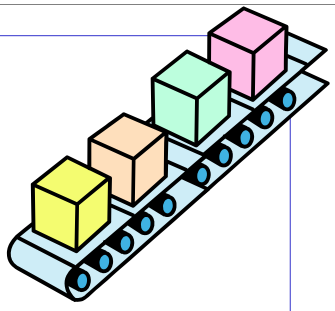
ref James Reason



Can we add some cheese from Holland?

8

Controlling Risk *by design*



- Every (software) project is unique
  (otherwise it's production)

however

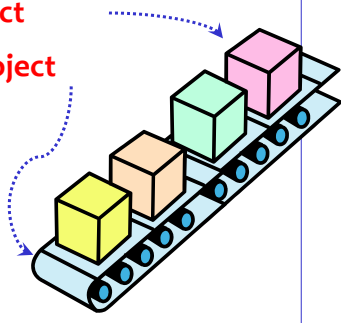
- A lot is always the same:
  - Every project is done by people
  - No project is very much unique
  - There are many similarities (with *known* risks)
  - So, a lot is predictable
  - We know Requirements will change (don't know *which*)
  - Engineers control risks *by design* (= *engineering*)

9

Known risks are hardly risks

- Most of the real risks are in the product
- Most of the known risks are in the project

$$V_{Risk} = P_{event} * P_{impact} * C \qquad P_{event} = 1 \qquad P_{impact} \rightarrow 0$$



- We don't only design the product,
- We should also *design the project*
- If we control 80% of the risks by design
- We have more time to handle the 20% *real* risks

10

Universal Project Goal

- 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

11

Defect and Risk

If a Defect is:

a cause of a problem experienced by a stakeholder of the system, ultimately by the customer

- Not satisfying the Goal is a defect
- Being late is a defect
- Being over budget is a defect

Then a Risk is:

an event that may cause a defect

12

Murphy's Law

- Whatever can go wrong, will go wrong
- Should we accept fate?

Murphy's Law for Engineers:

- Whatever can go wrong, will go wrong …

Therefore:

- We should actively check all possibilities that can go wrong and *make sure that they cannot happen*

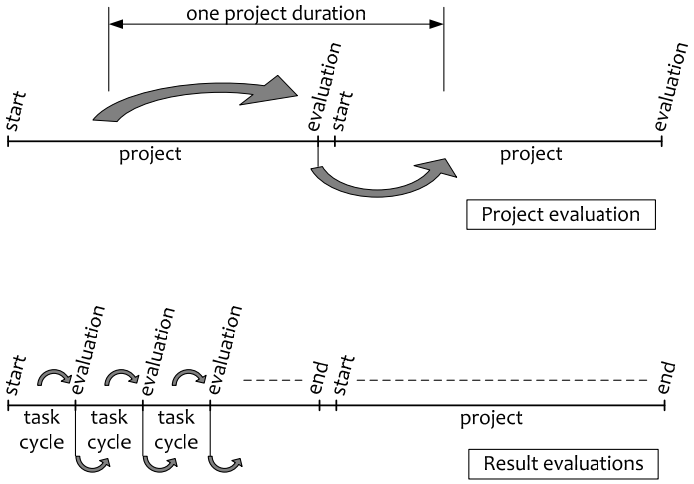
13

**If your previous project was late,
your current project will probably be late**

**If we don't learn from history,
we are doomed to repeat it**

14

Project evaluations



15

Preflection, foresight, prevention

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

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

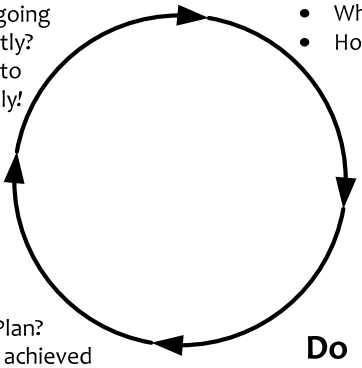
Only with *prevention* we can save precious time

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

16

The essential ingredient: the PDCA cycle

(Deming cycle)



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

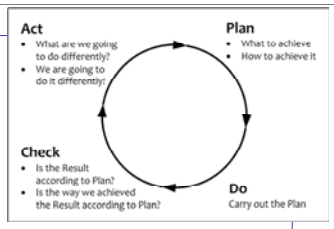
Plan
- What to achieve
- How to achieve it

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

Do
Carry out the Plan

17

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
- Combining Planning, Requirements- and Risk-Management into *Result Management*
- We know we are not perfect, but the customer should never find out
- Evo is about delivering Real Stuff to Real Stakeholders doing Real Things

  "Nothing beats the Real Thing"
- Controlling risk *by design*
- Projects seriously applying Evo, routinely conclude successfully on time, or earlier

18

Evolutionary Project Management (Evo)

- Plan-Do-Check-Act
  - The powerful ingredient for success
- Business Case
  - *Why* we are going to improve *what*
- Requirements Engineering
  - *What* we are going to improve *and what not*
  - *How much* we will improve: quantification
- Architecture and Design
  - Selecting the optimum compromise for the conflicting requirements
- Early Review & Inspection
  - Measuring quality while doing, learning to prevent doing the wrong things

Zero Defects Attitude

Right product

Evo Project Planning

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

Right time

19

Project Risks

wrong	wrong
wrong	right
right	wrong
right	right

- Delivering the wrong things at the wrong time (being late!)
- Doing the wrong things for too long
- Promising more than we can do
- Not living up to our promises due to unrealistic optimism
- Not being able to predict what we can have done when
- Making too many mistakes because of fatigue or sloppiness
- Others causing us to fail
- Interrupts

20

Project Risks

(*Boehm, 1991)

- Personnel shortfalls*
- Unrealistic schedules and budgets*
- Management impeding the workers
- Developing the wrong functions or properties*
- Developing the wrong interface*
- Developing user interfaces ignoring real human behaviour
- Gold plating, doing more than necessary*
- Changing requirements*
- Real-time performance shortfalls* (no issue with real *engineers*)

21

Personnel Shortfalls

Boehm 1991

- There are a certain number of people in the organization
- If we don't get the people we think we need, they are working on more profitable activities
- Using TimeLine, we inform management about the consequences

- This is not risk - it's choice

22

Unrealistic schedules and budgets

Boehm 1991

- How can we speak about realistic schedules if the requirements will change anyway?
- If the time/cost budgets are insufficient to get a profit, we shouldn't start or continue
- If management insist on unrealistic schedules (*Check*), they may need education (*Act*), or their aim is to fail
- People can quickly learn to change from optimistic to realistic estimators and thus live up to their promises
- We continuously update the TimeLine to predict what we will *surely* achieve, what *not* and what we *may* achieve
  - Using "Earned Value" to calibrate the "Value Still to Earn"

23

Continuing stream of Requirements changes

Boehm 1991

- Requirements do change because
  - We learn
  - They learn
  - The circumstances change
- If we would deliver according to obsoleted requirements, we don't create customer success
- We *know* that requirements will change, so we have to find out quickly which will change:
- *Provoking* requirements change as quickly as possible

24

Real time performance shortfalls

Boehm 1991

- This is why we have Performance Requirements
- We use engineering practices to make sure the system *will* work according to the requirements
- Real time predictability is easy, if you know how to do it
- If you don't know how to do it, it's not a Risk, it's an Issue (if it can go wrong, it will - Murphy)

25

Managers ignorance

- The product has to generate income
- If management impede the workers to produce the product in the most optimal way ...
- Management usually is not stupid
- But if we don't supply them the right facts ...

- The boss *may* mess up the Result, *if* he's the owner of the company
- All the others have the option to leave

26

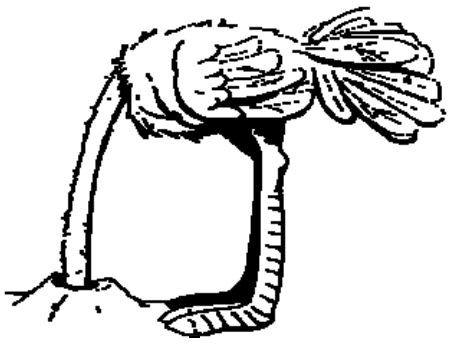
Worst risk

The worst risk is that we forgot an important issue

- It's within our control, but we didn't see it before it happened
- It's beyond our control, but we saw it too late and/or we didn't react appropriately
- The trick is to be ahead of any problem, before it occurs
- Don't ostrich: actively take your head out of the sand!
- If anybody complains, we're too late

If we control 80% of the risks *by design*, we have a lot more time to address the remaining 20%

27



The problems in projects are not the real problem,
the real problem is that we don't do something about it

28

Links

- www.gilb.com
  Tom Gilb's website: Evo guru
- www.malotaux.nl
  Niels' activities: Evo evangelist
- www.malotaux.nl/nrm/Evo
  Evo pages
- www.malotaux.nl/nrm/Insp
  Inspection pages
- www.malotaux.nl/Booklets
  1 Evolutionary Project Management Methods (2001)
  2 How Quality is Assured by Evolutionary Methods (2004)
  3 Optimizing the Contribution of Testing to Project Success (2005)
  3A Optimizing Quality Assurance for Better Results (2005)
  4 Controlling Project Risk by Design (2006)
  5 TimeLine: How to Get and Keep Control over Longer Periods of Time (2007)
  6 Human Behavior in Projects (2008)
  7 How to Achieve the Most Important Requirement (2009)
- www.malotaux.nl/nrm/Evo/ETAF.htm
  Download the Evo Task Administrator (ETA) tool
  (expects MSAccess 2000~2003)

29

Controlling Project Risk *by Design*

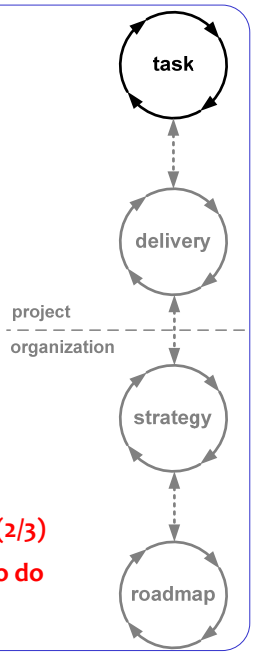
Niels Malotaux

N R Malotaux Consultancy

+31-30-228 88 68 niels@malotaux.nl www.malotaux.nl

30

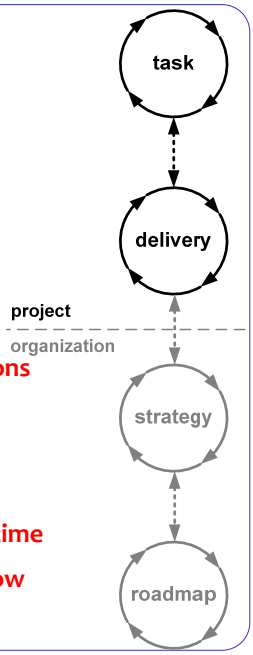
Cycles in Evo: Weekly TaskCycle



- 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

31

Cycles in Evo: DeliveryCycle

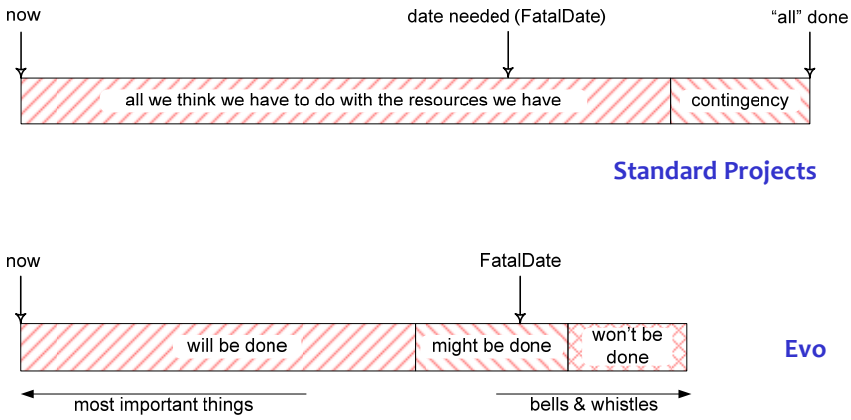


- Are we *delivering*
  the *right things*,
  in the *right order*
  to the right *level of detail for now*
- Optimizing requirements and checking assumptions
- What will generate the optimum feedback
- We deliver only to *eagerly waiting* stakeholders
- Delivering the juiciest, most important stakeholder values that can be made in the least time
- What will make Stakeholders more productive now
- Not more than 2 weeks

32

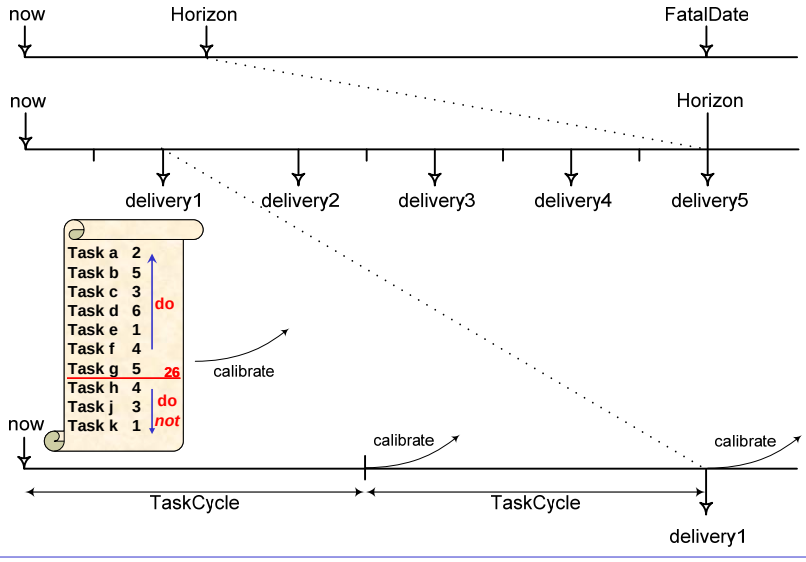
TimeLine

What the customer wants, he cannot afford



33

Result to Tasks and back



34

Controlling Project Risk *by Design*

Niels Malotaux

N R Malotaux Consultancy

+31-30-228 88 68 niels@malotaux.nl www.malotaux.nl

35