Niels Malotaux
N R Malotaux - Consultancy, the Netherlands, niels@malotaux.nl, +31-30-2288868

Controlling Risk *by design*

If we do nothing, the risk that we won't accomplish something is 100%. In order to accomplish what we want to accomplish, we organize a project, and at the end of the project the risks are to be reduced to an acceptable level. Te level will never be zero, as, for example, a meteorite could strike our result just before delivery of the project result.

Some time ago, I was asked to present a lecture about how to handle risk. My first reaction was "I hardly think about risk in my projects. What's wrong with me?" The second reaction was" "Everything I do in projects *is* about reducing and controlling risk. I only don't call it risk."

As a project coach, I collect, teach and optimize project management methods that make projects routinely succeed on time. The set of methods I call "Evo", which is short for Evolutionary Project Management Methods[1].

Evo combines project-, requirements- and risk-management into *result*-management. As projects are all about risk reduction, Evo provides methods how to control risk *by design*, rather than as a separate process.

The goal of a project

Let's define as the main goal of our efforts in projects: *Providing the customer with what he needs, at the time he needs it, to be satisfied, and to be more successful than he was without it*.

If the customer is not satisfied, he may not want to pay for our efforts. If he is not successful, he *cannot* pay. If he is not more successful than he already was, why should he invest in what we deliver anyway?

Of course we have to add that what we do in a project is *constrained by what the customer can afford and what we mutually beneficially and satisfactorily can deliver in a reasonable period of time*.

The cause of a problem in what we delivered, if experienced by the stakeholders contrary to the goal, is called a defect. If there are no defects, we will have achieved the goal. If not, we failed.

Risk is the chance that a defect may emerge.

Evolutionary Project Management Methods (Evo)

If we appropriately organize projects in very short Plan-Do-Check-Act (PDCA) cycles, constantly selecting only the most important things to work on, we will most quickly learn what the real requirements are and how we can most effectively and efficiently realize these requirements. We spot problems quicker, allowing us more time to do something about them. Preventing defects is usually better than fixing them, although prevention needs a specific attitude that usually doesn't come naturally. We need an attitude that we don't *want* to cause defects and that we will minimize the risks proactively. If we have a clear goal for our project, we can focus on achieving that goal. If management does not set a clear goal (which they should), we should set the goal ourselves.

The essence of Evo is actively, deliberately, rapidly and frequently going through the PDCA cycle, for the product, the project *and* the process, constantly reprioritizing the order of what we do based on Return on Investment (ROI), and highest value first, with a Zero Defect attitude. In my experience as project manager and as project coach, I observed that those projects, who seriously apply the Evo approach, are routinely successful on time, or earlier.

Evo is not only both iterative (using multiple cycles) and incremental (we break the work into small parts), but above all Evo is about learning how to do things better. We systematically and proactively anticipate risks before they occur and work to prevent them. We may not be able to prevent all the problems, but if we prevent most of them, we have a lot more time to cope with the few problems that slip through, before they materialize as a stakeholder problem.

Issues, experience and practical implementation details can be found in references [1] and [2]. Reference [3] describes how the testing process in software can be redesigned to better contribute to the goal. The V&V process in other disciplines can be organized likewise.

[1] The term Evo and Evolutionary as project management approach was coined by Tom Gilb in 1976. It has been used in many projects, including various large military and space projects, since the 60's.

Cycles in Evo

In Evo, we use several learning cycles:

- The *TaskCycle* [4] is used for organizing the work, optimizing estimation, planning and tracking. We constantly check whether we are doing the right things in the right order to the right level of detail. We optimize the work effectiveness and efficiency. TaskCycles never take more than one week.
- The *DeliveryCycle* [5] is used for optimizing the requirements and checking the assumptions. We constantly check whether we are moving to the right product results. DeliveryCycles focus the work organized in TaskCycles. DeliveryCycles normally take not more than two weeks.
- TimeLine [6] is used to keep control over the project duration. We optimize the order of DeliveryCycles in such a way that we approach the product result in the shortest time, with as little rework as possible.

During these cycles we are constantly optimizing:

- The product [7]: how to arrive at the best product (according to the goal).
- The project [8]: how to arrive at this product most effectively and efficiently.
- The process [9]: finding ways to do it even better. Learning from other methods and absorbing those methods that work better, shelving those methods that currently work less.

If we do this well, by definition, there is no better way.

The biggest risk

The biggest risk is the risk that we'll still overlook something:

- It's within our span of control, but we overlooked it
- It's not within our span of control, but we didn't anticipate, or we haven't done enough to avoid the problem to occur (we use *active synchronization* [10] to avoid this)
- The trick is to be ahead of problems, before they occur
- We don't ostrich, we actively take our head out of the sand
- If somebody complains, we're too late
- If the FatalDay is there, excuses or finger pointing is irrelevant
- If we didn't deliver, we failed

Conclusion

We design not only what we will deliver, we also *design* the way we work to succeed in our goal. Evo seeks to optimize this process, by constant learning to doing things better. In this process, risks are not handled separately, but as a natural part of running a project. Therefore Evo is full of small details designed to ensure success. The process itself being evolutionary as well, Evo constantly optimizes the ways to ensure success. Some people fear that this may take a lot of extra time. In practice we see that it saves time: The first project adopting these methods, generally completes successfully in 30% shorter time.

References

[1] N.R. Malotaux: *Evolutionary Project Management Methods*, 2001.
http://www.malotaux.nl/nrm/pdf/MxEvo.pdf
Issues and first experience, having a chance to introduce Evo at a large company.

[2] N.R. Malotaux: *How Quality is Assured by Evolutionary Methods*, 2004.
PNSQC 2004, http://www.malotaux.nl/nrm/pdf/Booklet2.pdf
Practical details how to implement Evo, based on experience in some 25 projects in 9 organizations.

[3] N.R. Malotaux: *Optimizing the Contribution of Testing to Project Success*, 2005.
Draft paper for PNSQC 2005, http://www.malotaux.nl/nrm/pdf/TestingEvo.pdf
How to apply the Evo ideas on the testing process to iterate more quickly towards Zero Defects.

[4]	TaskCycle	ref [2] chapter 5.1	ref [1] chapter 3C
[5]	DeliveryCycle	ref [2] chapter 5.1	ref [1] chapter 3C
[6]	TimeLine	ref [2] chapter 5.5 and 6.8	
[7]	Product	ref [2] chapter 4.2	
[8]	Project	ref [2] chapter 4.3	
[9]	Process	ref [2] chapter 4.4	
[10]	Active Synchronization	ref http://www.malotaux.nl/nrm/Evo/ActiveSynchr.htm	

Additional information can be found at: http://www.malotaux.nl/nrm/Evo