

Asset management for dummies

Ype Wijnia

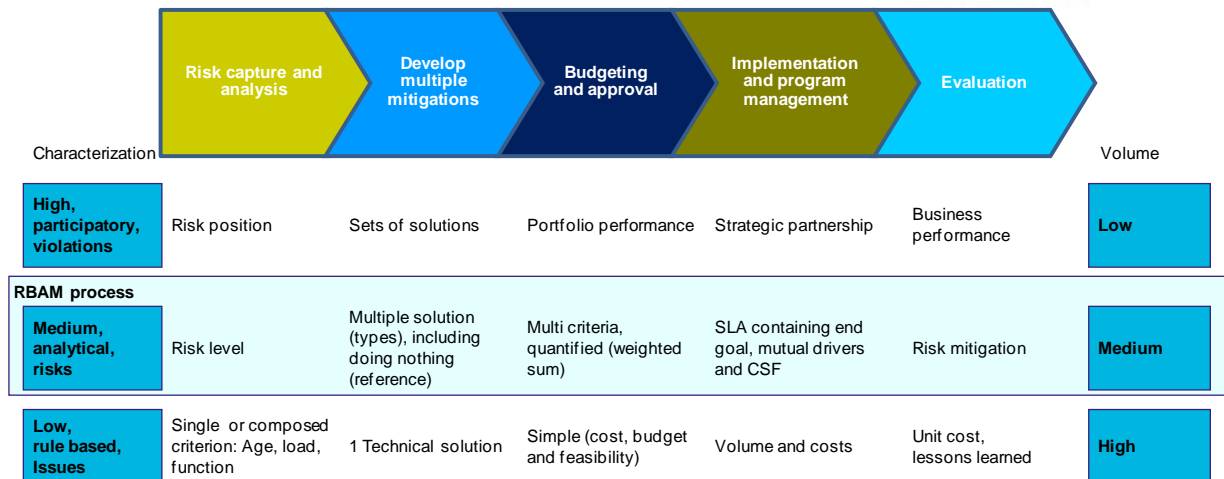
14 September 2012

Everybody with a grain of brain knows that there are simple and complex problems. If the computer does not work because the plug is not connected, it is a relatively simple problem. You plug the computer in and it works again. The great thing about simple problems is that many people are capable of solving them. As a matter of speaking, one could teach a chimpanzee to do the trick, given that enough bananas are available. It becomes more difficult if the computer shows a blue screen once in a while (for the Apple users amongst us: a long time ago, computers were pimped typewriters and the screen only showed letters. A blue screen looks similar). A blue screen of death indicates a fatal error that cannot be recovered, sometimes with the location in the memory where the error occurred. For most users that is utterly incomprehensible. Finding the precise cause of the error is practically impossible. In the tens of years of evolution of both chip and operating system, new developments were placed on top of the existing generation, with all historic errors still embedded deeply, and the potential of conflicts between old and new code. To find the cause of the crash one would have to review millions of lines of code with the potential of introducing new problems with edits of the code.

Fortunately computers have a simple solution for this kind of unexpected problems: you restart the computer and the odds are the same problem will not ever occur again. This is simply because the fatal error was the result of an exotic series of actions. The simple procedure can be performed by any user. It also happens to be the first line of the script used at helpdesks. If it does not help, a number of standard configurations will be checked, based on characteristics of the error. This may even be done from a distance. If the helpdesk is not able to solve the problem, they often advise a new installation, or even a new computer. No helpdesk employee will ever explore the depths of the code, that will be done by the companies that issues the programs. And they do, as they bring out patches and updates (just repairs of faults) at a high rate. Once every so many years they bring out a completely new version, with added functionality, but also with the potential for many new errors.

In dealing with computer problems, it is easy to recognize layers. The first level is that of the individual computer that does not do what it is supposed to do. Such a local problem is a bottleneck. Using simple rules (the helpdesk follows a script) a series of solutions is tested, with the replacement of the computer as the ultimate answer. Given the costs of a computer and the costs of just identifying the error this can be the best solution indeed. If a certain type of problem occurs more often, the software developer will start working on a more structural solution. This involves some serious debugging of the code or adding extra controls, though the software in general stays within the functional specs of the original release. The objective of those developments is to prevent the problem from happening again in future. The bottleneck is kind of generalized into a risk, which is a problem that might occur in future. The final level is that of the release change. The functional specifications of the software may be reviewed, performance requirements may be increased, new functionalities may be added or old functionalities may be removed.

A similar layering of problems can be recognized within asset management. We often use the schedule below as a starting point.



For simple bottlenecks like failures or assets in a bad condition you send out a team with a set of simple instructions that will arrive by means of a decision tree at one technical solution. That will be budgeted in standardized unit costs, and the implementation will be driven based on those standardized costs. If a new kind of problem occurs, a new recipe has to be written, which requires an analytical approach. In its core this is the risk based asset management process, with risk analysis, formulating multiple alternatives (also alternative in the approach, like technical, legal, commercial), different approaches to the implementation (different contracts or SLAs). On the top layer it really becomes complicated, because one has to decide how the available resources will be allocated over all risks that together drive the performance of the whole. If that performance is below standard, we say it is a violation (the agreement is not followed), though it should be stressed that it is not necessarily a violation in the legal meaning. To deal with those violations you will have to negotiate with the stakeholders, as one of the options to meet the target is to adjust the target to what has been achieved. Working together with the stakeholders is key in dealing with violations.

Following this logic, bottlenecks are simple problems. Yet, in practice one often sees smart people being deployed on bottlenecks, though it has to be said that they are complicated. As some bottlenecks can require very expensive solutions, it definitely pays out to think about them instead of blindly following rules. However, people working on bottlenecks often do not do anything else anymore. Bottlenecks show up anytime anyplace, begging for attention. This leaves no time for improving the rules for simple bottlenecks. As it does not matter much per bottleneck, no one will question this. But as there are much more simple than complicated problems, it does not require a great leap in thinking to realize that there is more to be gained in simple problems than in complicated ones. It may even bring enough benefits to allow for the incidental too expensive solution for a complicated problem. Phrased differently, it brings more value to be 90% correct in 100% of the problems, than 100% correct in 10% of the problems and the rest just some moderate performance.

This is what we call asset management for dummies. Keep it simple and care for rules that are easy to apply. This allows the whole organization to participate and much more value is created than when a bunch of Einsteins reviews a number of projects. Those smart employees should be used to guarantee the simple rules provide the best answer for the organization as a whole. And if that requires apples instead of bananas, that seems a reasonable price to pay.