What makes an IT Project successful?

In one of my earlier posts I tried to describe what kind of attributes a leader should have to be successful, and in this post I would like to go a big further and try to define what should make an IT project a success.

Wikipedia says that

In project management a project consists of a temporary endeavor undertaken to create a unique product, service or result.

By reading the above definition, there are two ingredients that define a project:

1. A project is a **temporary action**, therefore every project has a start and an end.
2. It has an **output**: product, service or a result

**A perfect project**

Let's start with some statistics:
This chart data is based upon The Standish Group report. The Standish Group is a Massachusetts-based consultancy responsible for publishing the CHAOS reports since 1994. The reports are based on studies of IT projects and track the success, challenged and failure outcomes of each project.

It's not hard to see that in average, since 1994 ~27% of all the project is successful. In other words, there is a higher chance that your project is not going to be successful than that it would. Which is scary.

**Project Management**

Usually, a project is run by a project manager and the typical project manager has lots of responsibility and very little authority, which implies that the person that runs the project in cannot fully control it. A project plan in this circumstances can but not necessarily does represent the reality.

Under these conditions, to run and to achieve a successful outcome can be really challenging.

**So, what would be the common project “attributes” that would make project “a success”**?

There is a list of goals that every project would need to meet in order to be a success:

1. Meet the business requirements
2. Deliver and maintain on schedule
3. Delivered and maintained within budget, and
4. Deliver the expected business value and return on investment.

Needless to say that one has to work really hard in order to achieve all the objectives. In order to achieve the points of the above list, there are some best-practices* should be followed and that could help the positive outcome of the project.

1. Clear and clearly articulated goals
2. Comprehensive, long-term, planning, when possible
3. Early definition of deliverable quality criteria
4. Active executive support with a shared vision throughout the project’s life
5. Carefully planned implementation
6. Concise, consistent, complete, and unambiguous business and technical requirements
7. Realistic estimates and schedules
8. Early risk analysis and ongoing risk management
9. Planning for business process change management
10. Proactive issue resolution
11. Stakeholder involvement throughout the life cycle
12. Defined and consistently executed change management to minimize scope increases
13. A skilled Project Manager experienced in the execution of project management best practices
14. Standard software infrastructure
15. Execution of a formal system development methodology (such as the State’s System Development Life Cycle)
16. A competent team
17. Commitment to success

The above points are taken from the Maryland’s Department of Technology and could be downloaded by following this link.

**Agile Project Management**

An alternative to the “classical” project management is, now not so new, agile project management, that I really believe is the “right thing” if adopted correctly. I am mentioning the Agile Project Management as I believe that is more suitable for a modern and a dynamic way of managing projects, at least when it comes to the IT industry, where I’ve got most of my experience.

In one sentence, if that’s even possible, what is the definition of the agile project management:
Agile project management is a value-driven approach that allows Project Managers to deliver high-priority, high-quality work—and look like rock stars to their stakeholders. It’s nothing like the plodding, costly and error-prone approach to project management, which has delivered inconsistent results for years...

Agile Project Management reduces complexity by breaking down the many months long cycle of building requirements for the whole project, building the entire product and then testing to find hundreds of product flaws. Instead, small usable segments of the software product are specified, developed and tested in manageable, two to four week cycles.

– Source: http://www.versionone.com/agile-project-management/

So, instead of having long running projects, with an unknown outcome, let’s have smaller chunks of perfectly managed iterations that would lead to a higher degree of success.

Unfortunately, I couldn’t find any reference on internet that would judge what is the degree of success of the agile projects alone, but eventually this will be published by some research companies. Please let me know if you find some!

**Final thoughts**

In this post I’ve mentioned the principal points that would make a project successful, together with a list of best-practices that, if followed correctly, could lead to a project success. Unfortunately, historically and statistically there are higher chances that the project fails and hopefully with new techniques (agile!??) this could be avoided, or to some degree improved. What do you think about it?

**Join the agile-code today!**

Email Address

Subscribe

**About the author:**

My name is Zoran Maksimovic. I’m a passionate programmer and interested in everything about Software Development, Object-Oriented Design and Software Architecture. Feel free to contact me or to know more about me in the about section.

http://www.agile-code.com/blog/what-makes-an-it-project-successful/
What makes an IT Project successful?