

THE ADAPTIVE NATURE OF MANAGING SOFTWARE INNOVATION

Mihai Liviu Despa¹

ABSTRACT

The focus of this article is pointed at adaptive management in the context of innovative software projects. Software development is presented through the filter of innovation. The aspects that differentiate software innovation from any other kind of innovation are highlighted. Adaptive management is addressed from a general point of view. The circumstances that require adaptive management are emphasized. Methods of implementing adaptive management in innovation oriented software projects are submitted and practical examples resulting from personal experience are presented. Innovative software development projects are depicted as a suitable environment for recusant management approaches. The theories, information and facts enunciated throughout the article are summarised. A coherent picture of managing the development of innovative software is assembled.

Keywords: adaptive management, innovation, software development, project management

1. SOFTWARE INNOVATION CHARACTERISTICS

Innovation is the implementation of an idea or an invention which leads to improving and perfecting a product, a method, a theory or a service with the sole purpose of accomplishing, at a higher standard, the objectives they were originally designed for. Innovation may be the subject of an entire project or may occur in a certain fraction of a project.

In software development projects innovation occurs with an increased frequency. Software innovation is not fundamentally different from other types of innovations but there are certain aspects that set it apart.

Scarce resources. You only need a PC and an internet connection in order to start developing software. Keep in mind that the internet connection is not a mandatory resource but will improve the odds of achieving innovative results. Innovation in the software development industry is not bound to sophisticated research laboratories or the prerogative of international scientist teams. A good idea, implemented correctly can be validated by online communities and go on to become a world acknowledged innovation [6].

Subjective views. Because it doesn't have a shape, a colour, or a sent software innovation is very often a matter of perspective. What is to some a useful application or a valuable functionality can prove to others completely useless. For example PR departments regard social media applications as being innovative software products that will shape the future of the internet. They consider social media valuable because it supports interactive dialogue and facilitates communication between individuals. Marketing departments consider social media a bound to fail undertaking because is not economically viable. Shoppers referred

¹ PhD Candidate, Academy of Economic Studies, Bucharest, Romania, mihai.despa@gmd.ro

from social networks such as Facebook, Twitter, LinkedIn and YouTube generated 0.34 % of all online sales on Black Friday, a decrease of more than 35 % from 2011 [1].

User contribution. Users play an important and sometimes direct role in software innovation. User feedback is fundamental in any industry but with software development is very easy to use it at its full potential [5]. Demo versions of an application are often released in order to gather feedback from users. Feedback is collected from blogs, forums, social networks and sometimes with the help of the applications itself by implementing “Contact Us” like functionalities. Open source software encourages users to actively take part in the process of innovation implementation [2]. Software innovation can also be validated with the use of online user communities.

Releases, updates, patches. Software products are evolving entities and they continue to change during their entire existence thus providing the perfect environment for innovation to take place. Some applications are designed from the start to be launched progressively, module by module or in successive individual releases. This gives the developer opportunity windows in which improvements and upgrades can be implemented based on the user’s feedback. Web applications are very easy to update were by changing the source code on the server the changes propagates instantly to all the users. This way new features can easily be implemented or malfunctioning features can be corrected by applying patches.

Ideas. Software companies often collect large numbers of ideas originating from company management, developers, customers and user networks [2]. The software industry can also borrow ideas from all other sectors because of its adaptive capabilities. Basically every human activity can, on a certain degree, be moved or reflected online. This statement translates into the fact that most of the ideas that have been implemented in other fields can be adapted for the online environment. This balanced mix of sources for new ideas insures a solid breeding ground for innovation.

Continuous process. In software development innovation can occur at any stage of the project: analysis, planning, development, testing or deployment. The main reason responsible for this situation is the fact that customers often change their minds about requirements and they provide unclear specifications leaving the developer room to improvise and forcing him to adapt. Innovation opportunities are not confined to the project implementation stages mentioned above. There are even software development methodologies that do not follow the above mentioned stages. Innovation can occur in a project even after deployment. Most applications are easily updateable and innovative features can be added effortlessly.

2. IDENTIFYING ADAPTIVE MANAGEMENT CIRCUMSTANCES

Adaptive management is an aggregate of decision making techniques specially designed to cope with uncertainty. Its aim is to reduce uncertainty, boost the learning process and always deliver a functional application by focusing on incremental and iterative practices [3]. Adaptive management is highly effective on the long term; the challenge consists on making it viable on short and medium term. The circumstances that require adaptive management tactics are diverse and require special consideration.

Changing requirements. Software projects are notorious for the fact that clients often, during the development stage, change significant parts of the original requirements. A traditional management method will break down under the pressure of changing requirements. A whole stack of documents representing, use case diagrams, wireframes and design concepts will be useless. The analysis, design, code, test and deployment routines will be toppled with severe consequences and the whole project will have to start all over again. Adaptive management is designed to handle changing requirements and to embrace new challenges [3].

Insufficient specifications. There are countless projects for which the client will not provide sufficient specifications. Either he has not yet decided on the strategy for the entire project; he hasn't got a clear view on the project or simply is reluctant to give up information. To cope with this situation adaptive management is needed. Sometimes just writing code is more important than understanding the business concept. As the project advances the client will make available more and more information and the goal and purpose will become clear.

Choice. If multiple management choices are available for a given state than adaptive management is applicable. Choosing a path, following it while is feasible and then deflect from it and choosing another way is an adaptive method. If only one scenario can be implemented then adaptive management will not generate satisfying results. With only one option available being adaptive will only mean being less productive.

Learning. Adaptive tactics are recommended for project dedicated to learning and deepening certain concepts or skills. There are developers that approach certain projects with the sole purpose of learning new technologies, frameworks or methodologies. For example a software development company can bid for a web project and offer on the side a free complementary Android mobile application. Even if mobile applications for Android OS are not in the company's portfolio the project offers a good learning opportunity.

Uncertainty. Novelty is a distinctive characteristic of software projects. While developing a software project situations that have never been documented by the specialized literature often arise. In this type of circumstances you operate exclusively on assumptions and a wrong decision is likely to spring up from time to time. Adaptive management allows you to contain the damage produced by a wrong decision and get back on the right track.

Programming team. Only a strong and homogeneous programming team can work effectively under an adaptive management system. An adaptive management system emphasizes the value of self-organizing programming teams. The project manager sets the objectives and defines the tasks and the programmers assume the objectives and choose the tasks they are going to perform. Every programmer has a double responsibility: to the project manager, for letting him decide what tasks to perform, and to his fellow programmers, for trusting him and sharing the same common objective.

3. IMPLEMENTING ADAPTIVE MANAGEMENT

Adaptive management is required in software development because of its ability to overcome changing requirements and for its capability to maintain the project's balance

regardless of disturbances. References and theories regarding adaptive management are widely available so in order to avoid redundancy only methods validated by the present paper's author are going to be enunciated.

Few documents. Specification documents, design concepts, wireframes, use case diagrams, user role maps and hierarchies are all valuable management tools but often tend to shift the focus from the development process to the monitor and control area. Managers very often get entangled in a dense weave of paperwork and bureaucracy, losing focus on the actual development of the application. Innovation on paper is just literature. The client is not looking for outstanding management performances; it only requires that the developer delivers an application that fulfils certain requirements.

Human interaction. As much as he can, a manager should personally approach members of his team regarding project related issues or tasks. Interaction is a conducive environment for innovation. Bug reporting and task tracking software are very useful in the software development industry but a personal approach may sometimes offer important feedback on the overall progress of the project. The project progress may look great on paper but reality is sometimes far from management forecasts. When developers welcome bugs or new task with apparently unjustified aversion the project may very well be in a lot of trouble. Avoid sending emails when the recipient is only a few desks apart. Encourage live presentation instead of written reports but also store a written copy.

Working software. Plan and structure the development process so that after approximately two weeks of programming a working module or the backbone of the application can be presented to the client. Working software is one of the most valuable assets a developer should focus on because it keeps clients satisfied and the programming team optimistic. It is also a measurable evaluation method for the progress being made on the project. Comprehensive documentation, brilliant design, tutorials, training, or great contractual terms are futile for the client without working software. Concentrate on making the application work at the expense of best management practises or established marketing guidances.

Embracing failure. In order to achieve innovative results you have to be prepared to fail over and over again. Innovation comes from exploring uncharted territories and many paths lead to failure. The downside of failure is that it doesn't just come with disappointment but also causes financial losses, resignations and worsening customers' relations. Therefore embracing failure is not enough you also have to be prepared to handle the consequences. Never build your entire business on just one client no matter how important, influential or creditworthy. Make sure that at any time you have several recruitment sources available. Use recruitment companies, job portals and keep in touch with university's campuses. Handing job flyers in a campus can offer you amazing recruitment opportunities if you target computer sciences dorms.

Accepting risks. Innovation is always closely related to risk. Risk is often followed by resounding success. A talented manager will only take calculated risks and never start a project or a task without a contingency plan. Always ask "What if?" and if the answer is satisfying don't look back. Every innovative software development project has to benefit from a dedicated risk budget. Allocate financial resources to treat, control and limit risks.

In order to determine an estimated risk budget a dedicated process must be conducted. To the extent of possible all the risks associated with the project must be identified. For each identified risk a probability of occurrence is associated. Also for each identified risk an impact level has to be determined. The impact represents the effect the materialization of the risk will have on the project. In order to determine an exhaustive risk budget just sum-up the costs of neutralizing each risk.

$$EB_R = \sum_{m=1}^n P_m \bullet I_m \quad (1)$$

EB_R - extended budget risk

P_m - probability that the 'm' risk occurs

I_m - impact the 'm' risk has on occurrence

In order not to oversize the project budget only the ten most relevant risks will be addressed for handling. The actual risk budget formula is:

$$B_R = \sum_{m=1}^{10} P_m \bullet I_m \quad (2)$$

B_R - actual budget risk

P_m - probability that the 'm' risk occurs

I_m - impact the 'm' risk has on occurrence

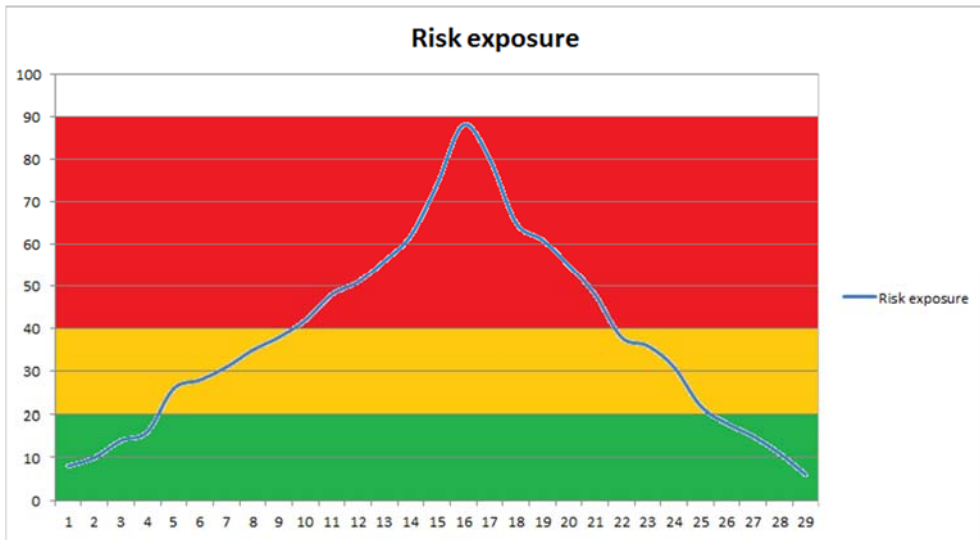


Figure 1

In Figure 1 are listed 29 risks identified for a particular project the author has worked on. Every risk is measured depending on probability of occurrence and impact. Probability is

measured on a scale from 1 to 10; 1 being the lowest probability. Impact is also measured on a scale from 1 to 10; 1 being the lowest impact. By combining probability of occurrence with impact (PxI) every risk is assigned a scoring on a scale from 1 to 100. Risks that have scores greater than 40 are considered critical risks. Risks that have scores in the 20 - 40 interval are considered moderate risks. Risks that have scores under 20 are considered low risks. For this particular project only the top 6 highest ranking risks were addressed. The top risk for this particular project was scored 88 and consisted in a senior developer leaving the team. The risk was that high because the developer already announced his resignation. Because a proper risk budget existed the developer got a salary increase and remained with the team.

Absorbing information. New data and information is introduced to a project manager on a regular basis. Assimilate and use the information in order to bring additional value to the project and do so in compliance with the client's directions. Decision makers must adapt to new information and change the course of the project accordingly even if the new direction doesn't coincide with the initial strategy they laid out. Being inconsistent is not a bad management tactic if it is based on new information or changing circumstances [4].

Reduce uncertainty. Rely on adaptive management to gradually reduce uncertainty by implementing a less stringent management system. The programming team will respond more efficiently to management requests. The client's actions will be met by a proper reaction from the developer. Every successful project is just a benchmark and offers guidelines for future projects; it doesn't create rules or imposes courses of action.

Data and information. A project manager should collect as many data about the project as possible in order to take informed decisions. Collecting data is the process of gathering information in an organized manner and sorting it according to relevance. Data and information are the main assets of a project manager. Based on the data a project manager can make decisions, define strategies, right deviation or measure progress. Project Manager collects data from the following sources: project team, end-users, project owner, similar projects and project's environment. In Figure 2 the data collection process is depicted.

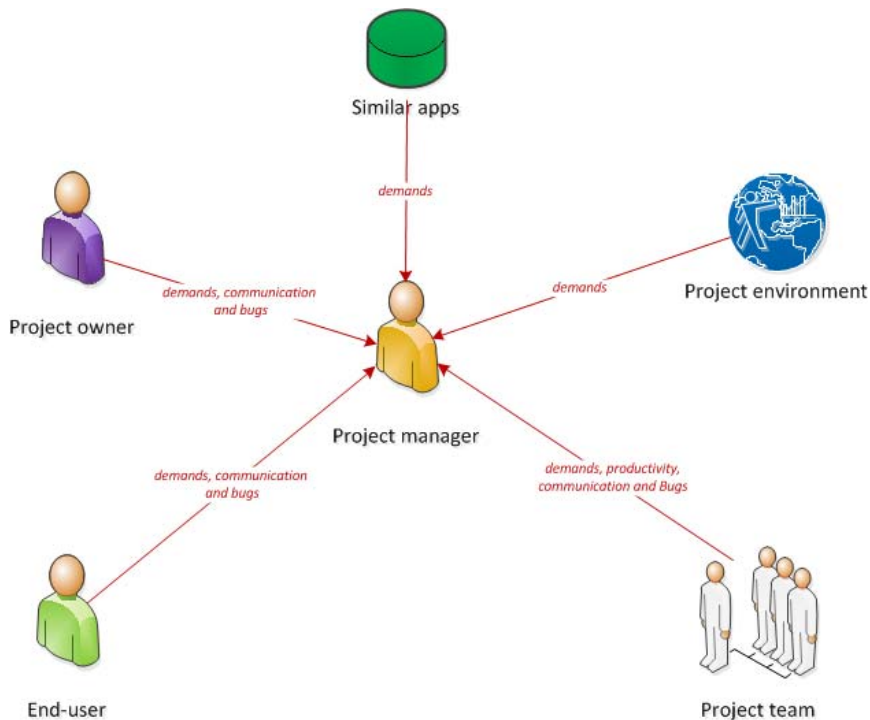


Fig. 2 - Collecting data for IT project management

Specific IT project management data:

- **demands** - represents all the specifications, instructions, requirements submitted by the parties involved in the project. The project manager has to take into account the project's team members requirements, the project owner's specifications and the end-users instructions.
- **legal framework** - represent all the laws and regulations under which the project must follow, respect and comply with. The legislative framework creates opportunities and constraints for a project manager. The project manager is required to know and optimally use facilities of the existing legal framework.
- **productivity** – is the measure of the project team's efficiency and / or the efficiency of a team member. Productivity is calculated on the basis of actual progress and allows for predictions and simulations related to the future development of the project.
- **communication** - it refers to the effectiveness with which the project stockholders communicate. Communication is a very important element in a project and is targeting both the project team and the project owner, suppliers, collaborators or end users
- **bugs** - malfunctions or components that are not compliant with the specifications submitted by the project owner. Bugs are important data for the project manager because they provide an overview of the project's quality.

- **standards** - represents all the characteristics that the application has to meet. Standards are set by the project owner, the common industry practice or similar apps already on the market

4. CONCLUSIONS

Adaptive management is not a guaranteed recipe for implementing innovative software development projects. You can use adaptive software development for the most common software projects with great success. If aiming for innovation concentrate on the basics of software development. Use few specifications and design documents. Try to communicate directly with your development team even for the most insignificant issues. Send emails only if you believe that the information needs to be reordered for future references. Use reporting for monitoring the programmers' activity, encourage them to present them live but also store a written copy for evaluations and assessments. Concentrate on delivering working software as early in the project as possible. Do not be afraid of failure but welcome it as prepared as possible. Budget risk into your financial forecasts. Be open to new information and act on it.

5. REFERENCES

- [1] IBM Digital Analytics Benchmark, *Black Friday Report 2012*
- [2] M. PIKKARAINEN, W. CODENIE, N. BOUCART, J.A.H. ALVARO, *The Art of Software Innovation: Eight Practice Areas to Inspire your Business*, 2011, ISBN 978-3-642-21048-8, Springer-Verlag, Berlin Heidelberg
- [3] C. MURPHY, *Adaptive project management using Scrum*, Methods & Tools, 2004, Volume 12, Number 4, pg 10, ISSN 1023-4918
- [4] L. VIRINE, *Adaptive Project Management*, 2008, PM World Today
- [5] E. HIPPEL , *Democratizing Innovation*, 2006, Massachusetts Institute of Technology, Cambridge, pg. 216, ISBN 9-780262-002745
- [6] M. GHENEA, *Antreprenoriat: drumul de la idei către oportunități și succes în afaceri*, 2011, Editura Universul Juridic, pg. 329, ISBN 978-973-127-516-1