RESEARCH PAPER | Computer Science | Volume : 5 | Issue : 9 | September 2015 | ISSN - 2249-555X



Comparative Study on Software Development Methodologies

KEYWORDS: software development, project management, development methodology

Dr. Kishor H Atkotiya

Head, Dept. of Computer Science, J.H. Bhalodia Women's College, Rajkot

ABSTRACT *This paper focuses on the current state of knowledge in the field of software development methodologies. It aims to set the stage for the formalization of a software development methodology dedicated to innovation orientated IT projects. The paper starts by depicting specific characteristics in software development project management. Managing software development projects involves techniques and skills that are proprietary to the IT industry. Also the software development project manager handles challenges and risks that are predominantly encountered in business and research areas that involve state of the art technology. Conventional software development stages are defined and briefly described. Development stages are the building blocks of any software development methodology so it is important to properly research this aspect. Current software development methodologies are presented. Development stages are defined for every showcased methodology. For each methodology a graphic representation is illustrated in order to better individualize its structure. Software development methodologies are compared by highlighting strengths and weaknesses from the stakeholder's point of view. Conclusions are formulated and a research direction aimed at formalizing a software development methodology dedicated to innovation orientated IT projects is enunciated.*

Introduction

The management process of a software development projects follows the basic rules of project management but also includes particular features. A software development project manager has to deal with challenges and setbacks that are proprietary to the IT industry. Also in software development there are benefits and strong points that help ease the burden of management.

Software development projects are notorious for frequently changing initial planning and specifications. In order to identify the main reasons for changing specification during the development stage of a software product debates were started on LinkedIn project management groups. Debates were initiated on 11 LinkedIn groups starting the discussion with the following introduction: Software development projects are notorious for frequently changing initial planning and specifications. What do you believe are the main reasons for changing specifications during the development stage? Project managers responded on 3 out of the 11 groups. The 3 groups are: PMO – project management office, Agile Project Management Group and International Society for Professional Innovation Management. Based on the information collected from the LinkedIn discussions and on the author's own experience as software development project manager specifications change due to the fact that:

- The project owner identifies new business opportunities and decides to integrate them into the software being developed;
- Due to the technical nature of software development projects there is a lack of shared understanding of excepted outcomes;
- original planning was based on specifications that were misinterpreted by the project
- Manager or poorly illustrated by the project owner
- project team is unable to implement planned functionalities due to lack of expertise or
- technological limitations;
- the context in which the software is going to be used changes thus generating the need for the software to change;
- new technology or software product is launched on the market.

Changing specifications has a negative impact on the project management process as it reduces predictability and exercises pressure on the budget and deadlines.

The software development field is characterized by high dynamics of technology and standards. Programming languages evolve, new frameworks arise and fall with astonishing speed, user interfaces become more and more diverse as software is required to work on a larger array of devices. PHP server-side scripting language registered 16 releases on new and improved versions in 2014 alone [1]. The software development community widely accepted Phalcon and Laravel, as two of the most powerful PHP frameworks. Both were released in 2012. The project manager has to keep up with the latest trends in software development in order to meet the project owner's requirements and in order coordinate effectively the project team.

Software development projects involve project teams that are made up of highly skilled and highly trained individuals with predominantly technical backgrounds. Highly skilled and highly trained individuals will require significant compensation for their work and that will translate into high man-hour or man-day rates. So the project manager is under considerable pressure to

provide accurate time estimates in terms of required man-days as every inconsistency will generate significant additional costs. Also highly skilled individuals don't integrate well in a team as they have a tendency of being arrogant and selfabsorbed. The project manager should be able to exploit their ego in the best interest of the project and mitigate disputes and opinion clashes that occur between team members.

Software development projects are often implemented by teams that have members distributed all over the globe. Building software does not require formal face-toface com-

Volume : 5 | Issue : 9 | September 2015 | ISSN - 2249-555X

munication. Task assignment and task tracking is done by using online management tools like Pivotal Tracker, Basecamp or Producteev. Code version control is ensured by using versioning tools like SVN or GIT. File sharing is accomplished by using tools like Dropbox, Google Drive or Box. Online meetings can take place using Skype video conferences.

Current software development methodologies

A software development methodology is a set of rules and guidelines that are used in the process of researching, planning, designing, developing, testing, setup and maintaining a software product. The methodology also includes core values that are upheld by the project team and tools used in the planning, development and implementation process. This paper

reviews 20 of the most popular software development methodologies and highlights their core characteristics. The analysis includes specifying the scale of the project the methodology is suited for, the stage project owner feedback is delivered and a graphic representation of the methodology. For coherence reasons stages defined in the second section of the article are also used in depicting the graphic representations of the methodologies.

Waterfall is the first methodology

generally acknowledged as being dedicated to software development. Its principals are for the first time described by Winston W. Royce even though the actual term waterfall is not used in the article [24]. It emphasizes meticulous planning and it outputs comprehensive documentation. The Waterfall methodology is linear sequential process where every stage starts only after the previous has been completed. Each stage has its own deliverables. The Waterfall methodology is predictable and values rigorous software planning and architecture.

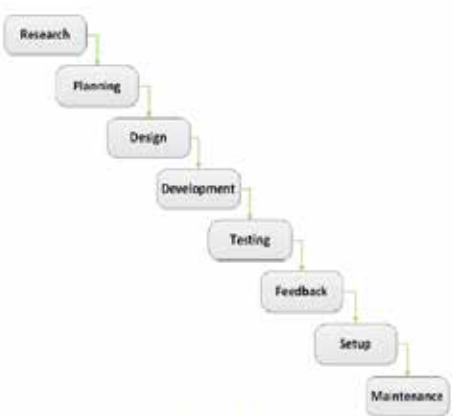


Fig. 1. Waterfall methodology

Prototyping is a methodology that evolved out of the need to better define specifications and it entails building a demo version of the software product that includes the critical functionality. Initial specifications are defined only to provide sufficient information to build a prototype.

The prototype is used to refine The project owner's feedback is received after the prototype is completed. The Prototyping methodology is suitable for large scale projects where is almost impossible to properly define exhaustive requirements before any actual codding is performed. Prototyping methodology is also suitable for unique or innovative projects where no previous examples exist.



Fig. 2. Prototyping methodology

Software development methodologies characteristics

Methodology	Characteristics	Strengths	Weaknesses
Waterfall	- comprehensive documentation - meticulous planning - linear-sequential process - each phase has its own deliverables	- easy to manage - easy to understand for the project owners and team	- working code is delivered late in the project - does not cope well with changing requirements - low tolerance for design and planning errors
Prototyping	- build one or more demo versions of the software product - project owner is actively involved - prototypes are meant to be discarded	- accurate identification of application requirements - early feedback from the project owner - improved user experience - early identification of missing or redundant	- leads to unnecessary increase of the application's complexity - increased programming effort - costs generated by building the prototype

Conclusions

Software development methodologies follow two major philosophies: heavyweight and lightweight. Heavyweight methodologies are suitable for projects where requirements are unlikely to change and the software complexity allows for detailed planning. Heavyweight methodologies are easy to understand and implement. They provide solid documentation and appeal to project owners because they are well structured and showcase tangible deliverables for every stage of the project. With heavyweight methodologies the project manager can easily perform tracking, evaluation and reporting. The project owner is considerably involved only in the research and planning stages. Lightweight methodologies are suitable for projects were specifications are unclear or are likely to change due to project internal or external factors. Lightweight methodologies are based on an incremental approach were software is delivered in

multiple consecutive iterations, all of them being functional versions of the application. Lightweight methodologies provide great flexibility and can easily adapt to change. They promote early delivery of working code, self-organizing teams and adaptive planning. The project owner is heavily involved in all the stages of the project as its input and feedback is critical for the success of lightweight methodologies. When choosing a software development methodology project owner profile, developer's technical expertise, project complexity, budget and deadlines must be taken into account. Often no methodology will fit perfectly the profile of a specific project. Then the best matching methodology should be used or in the case of experienced project teams and project managers a combination of methodologies could be introduced. In the case of innovative software development projects a new methodology is required.

Volume : 5 | Issue : 9 | September 2015 | ISSN - 2249-555X

This topic is a subject for further research in the software development field References

REFERENCE [1] http://php.net/ChangeLog-5.php | [2] J. E. Cooling, T. S. Hughes, "The emergence of rapid prototyping as a realtime software development tool", Proceedings of the Second International Conference on Software Engineering for Real Time Systems, 18-20 Sep. 1989, Cirencester, UK, Publisher: IET, 1989, pg. | 60-64 | [3] C. Larman, V. R. Basili, "Iterative and Incremental Development: A Brief History", Computer, vol. 36, no. 6, pg. 47-56, 2003, doi:10.1109/MC.2003.1204375 | [4] B.W. Boehm, "A spiral model of software development and enhancement", Computer, vol. 21, no. 5, pg. 61-72, 1988, doi: 10.1109/2.59 | [5] K. Beck, "Embracing change with extreme programming", Computer, vol. 32 , no.10, pg. 70 – 77, 1999, doi: 10.1109/2.796139