

Agile Project Management Principles

Jan Juricek

Abstract—This paper shows main agile principles. Agile approach is compared against standardized, eg. waterfall oriented methodologies (also referenced as a product-based) with group of activities based on iterative and incremental development. The aim of this paper is to introduce the reader of basics agile principles and to define the pragmatic differences between the heavyweight waterfall methods, while agile is a global reaction to traditional approaches of delivering the solution or delivering the products by a standard flow: requirement – plan – delivery – acceptance. The agile principles are researched and aligned with the product-base, a waterfall model and standard project management methodologies such as PMBOK© and Prince2©.

Index Terms—Project management, agile, project management methodologies.

I. INTRODUCTION

Agile Methods are a reaction to traditional ways of developing software and acknowledge the need for an alternative to documentation driven, heavyweight software development processes [1]-[3]. Traditional ways are designated as rigorous.

In the implementation of traditional methods, work begins with the elicitation and documentation of a “complete” set of requirements, followed by architectural and high-level design, development, and inspection. In the practices, [4] practitioners found these initial development steps frustrating and, perhaps, impossible. In a modern, rapidly changing environment, the requirements (or their priorities) are changing.

Changes are almost un-manageable and customers have become increasingly unable to definitively state their needs up front while, at the same time, expecting more from their software or valuable outputs. [5].

Agile actually responds to these challenges. The agile principles are collection of different techniques that share the same values and basic principles based on experience from the change itself and from the approach that can be introduced as a power of the flow.

Regarding this, agile project management [6] has proven to be a useful tool for today's knowledge worker and the project managers in the new economy, which is characterized by more complex and uncertain project situations.

II. HISTORY SURVEY–AGILE MANIFESTO

Back in February 2001 in Utah, Manifesto for Agile

Manuscript received July 3, 2013; revised October 1, 2013.

J. Juricek is with the Department of the System Analyses, University of Economics in Prague, Czech Republic (e-mail: jan.juricek@gmail.com).

Software Development was published to define the lightweight development method approach. This approach and the manifest itself, was primarily addressed to software development area [1].

First, align the standard guidelines for project management with the agile meanings. In this case, process-based methodology will be referenced as a clear representative of the invariant processes and knowledge areas. These representatives will be addressed further as a heavyweight practices.

So-called lightweight agile software development methods evolved in the mid-1990s as a reaction against heavyweight waterfall methods, which were characterized by their critics as a heavily regulated, regimented, micromanaged, waterfall model of development. Proponents of lightweight agile methods contend that they are a return to development practices from early in the history of software development [2].

For the further agile principles overview, the main and complex differentiation between the heavyweight waterfall methods and the agile methods are depicted:

TABLE I: AGILE METHODS

Heavyweight waterfall methods	Lightweight Agile Methods
Clear, described, invariant processes and tools	Individuals and interaction
Approved comprehensive documentation	Working SW
Contract negotiation, long time spend to baseline a contract	Customer collaboration
Following a plan	Responds to changes

Comparison of these principles showed in the Table I, are actually the basic part of the agile manifesto [7]. To further transformation of the entire manifesto to the agile principles, agile methods are comprised and extended, as showed in the Table II.

TABLE II: AGILE MEANING CONTEXTS

Agile Methods	Manifesto Meaning Context	SW Development Context
Individuals and interaction	Self-organization and motivation	Pair-programming, extreme programming
Working SW	Working solution, visible output	Working deployable software
Customer collaboration	Continuous customer and stakeholder involvement	Continuous SW requirements collection
Responds to changes	Quick responses to change	Continuous development

Summarized, the meaning of the entire manifesto [7] stands on real life facts in what makes project to be

potentially successful¹:

- 1) Self-organization and personal individual motivation are important, as well as interaction and co-operation
- 2) Working solution, visible output
- 3) Customer Collaboration
- 4) Responding to changes

Meaning of these points is a conclusion, that the agile approach is a working solution, through the management strategy context (vision, goals, funding), a release context (estimation, back-log), an iteration and daily plans (review, retrospective, standups, component acceptance) and through the continuous integration and collaboration. Agility is bringing values in adaptability, transparency, unity and simplicity as well as visibility in time-spend, overall testing and project status. These meanings should be transformed into the agile principles.

III. AGILE PRINCIPLES

Put simply, agile principles are a different way of managing IT development teams and projects. In this paper, a differentiation and significance from the product-based and strict processes project management methodologies will be accented. PMBOK and Prince2 guidelines, as cross-referenced project management methodologies are selected.

A. Active User Involvement

Active user involvement is basically an agile imperative. (Please see Table III as a comparison between the agile principle and other project management methodologies). It's not always possible to have users directly involved in development projects, particularly if the agile development project is to build a product where the real end users will be external customers or consumers. In this event it is imperative to have a senior and experienced user representative involved.

TABLE III: USER INVOLVEMENT COMPARISON

PMBOK and Prince2 Project Methodology accents	Agile Principle
Role of the Senior user is part of the steering board (projectboard) or project committee	Role of the Product owner, who is accountable for maximize product's value and team's work
Active user involvement is slightly accented	Active users are part of the development team itself
Assure the user involvement primarily through the reporting process	Increasing visibility to the project through the integrated team of the customer and the supplier
Visibility of the project is high in the initiation phase, low in the build phase and high again in the acceptance stage or closing phase	Overall project visibility is stable

According to the research at this field, e.g. By Center for Applied Software Engineering at the Free University of

¹ According to the definition of a successful project is one whose output was achieved on time and within defined quality and within the approved budget

Bozen [8], or in Turning Chaos into Success [9] user involvement is essential agile approach. Understanding what customers really want requires their constant involvement in the project. And agile assume changes in the customer requirements within the project life cycle.

B. Empowered Team to Manage from down –to–up

By Kelly Watters, the project team must be empowered to make decisions in order to ensure that it is their responsibility to deliver the product and that they have complete ownership. Any interface with the project team is disruptive and reduces their motivation to deliver [7].

User involvement comparison is showed in Table IV.

TABLE IV: USER INVOLVEMENT COMPARISON

PMBOK and Prince2 Project Methodology accents	Agile Principle
Manage by exception approach – project manager has a strict budget and all important changes must be reviewed by board. It takes time and additional effort.	Product Ownersgip is established on the project management level
Interference with the board reduces motivation by the team level of management	Requirements are priorities and evaluate in the team level of project management

It is an agile principle, that the team must establish and clarify the requirements together, prioritise them together, agree to the tasks required to deliver them together and estimate the effort involved together.

C. Flowing Requirements

In agile development, requirements evolve, but timescales are fixed. Recognizing that customers require maximum flexibility and ability to rapidly adjust their solution capacity, and also wants to have fixed budget and fixed scope, is going opposite against each other. Please see the requirements dealing comparison in Table V.

TABLE V: REQUIREMENTS DEALING COMPARISON

PMBOK and Prince2 Project Methodology accents	Agile Principle
Change Management is established as a rigors process and change is typically exceptional	Changes are usual part of the project.
Fixed budget, fixed scope, following plan	Requirements evolve, budget burn-chart
No or low requirement prioritization, there is only Business Requirement Statement baseline	Requirements prioritization in each iteration, planning at each iteration start
Traditional or plan-driven approach with known requirements and comprehensive cost–benefitanalysis and established development plan	Periodic teamwork with focus on 'value-up' through quick delivery of working software, unknown requirements
Project Requirements aligned with project plan must be approved by project board	Agile Development teams capture high level requirements in workshops, working together in a highly collaborative way so that all team members understand the requirements as well as each other.

By research of requirements prioritization strategies [10] which define also Agile plus (Agile approach with

consideration of cost in addition to value) and Hybrid strategy (A combination of PB and Agile plus with variable iterationsizes), the most relevant factor for requirements prioritization is dynamism, defined as the percentage of requirements change per month – the more dynamics project and the more question in requirements, the better usage of the agile approach in the overall project success will be.

Agile works on the premise that requirements emerge and evolve, and that however much analysis and design team do, this will always be the case because team cannot really know for sure what they want until they see and use the output.

Summarized, agile requirements are ideally visual and should be barely sufficient, i.e. the absolute minimum required to enable development and testing to proceed with reasonable efficiency [8].

D. Quick, Small, Incremental Releases and Iteration

This principle logically continues on requirements evolving. In product-based projects, the (simplified) lifecycle is Analyze, Develop, Test—first gathering all known requirements for the whole product/output, then developing all elements of the software, then testing that the entire product is fit for release. In agile software development, the cycle is Analyze, Develop, Test; Analyze, Develop, Test; doing each step for each feature, one feature at a time, as showed in Table VI..

TABLE VI: A RELEASE AND AN ITERATION COMPARISON

PMBOK and Prince2 Project Methodology accents	Agile Principle
Initiate – plan – manage product delivery – close the project	Iteration in several days, check the requirements, attest the function and review with the customer
Traditional release	Incremental release

By [4], [8] the advantages of this iterative approach are:

- 1) Reduced risk: clear visibility of what's completed to date throughout a project
- 2) Increased value: delivering some benefits early
- 3) More flexibility/agility: can choose to change direction or adapt the next iterations based on actually seeing and using the software
- 4) Better cost management: if, like all-too-many projects, you run over budget, some value can still be realized.

E. Complete First, then Move to the Next

Features developed within iteration, should be 100% complete by the end of the Sprint [5]. All requirements are moving to the backlog² and must be developed or completed by its priorities. By this principle, the one iteration could not merge into the other one. In product-based methodologies, phases can overlap.

F. Test Early and Often

As its definition [8], in agile approach, testing is integrated throughout the lifecycle; testing the output continuously throughout its development. Agile development does not

have a separate test phase as such.

Product-based methodologies accent the integration testing as a final proof of shippable solution. In this case, partial tests seem to be inadequate. However, regarding the Agile SW testing in a large-scale project research [11], the project team cut by an order of magnitude the time required to fix defects, defect longevity, and defect-management overhead. Even on such a large-scale project, the team achieved full regression testing at each iteration and developer testing. It also resolved all defects over a significant time period that included both personnel changes and team growth. This research and its data show that agile testing works.

G. Collaboration between All Stakeholders

Agile culture always stresses on cultivation (understanding the need and possibility of the change), competence (motivation, individualities, continues attention to technical excellence), control and customer collaboration.

In this context, agile principle accents that business people and developers (team members) must work together daily through the project; the best architectures and clear functional requirements and its design emerge from self-organizing teams; and the most efficient and effective method of capturing information is face-to-face conversation. The collaboration comparison is showed in Table VII.

TABLE VII: COLLABORATION COMPARISON

PMBOK and Prince2 Project Methodology accents	Agile Principle
Heavy-weight documentation with both the process plans and rich content	Keeping requirements and documentation lightweight, and acknowledging that change is a normal and acceptable reality in software development
Multi-level management and strict communication and reporting activities	Reducing project roles
Defined stakeholder management	N/A

Agile principles of collaboration include keeping requirements and documentation lightweight, and acknowledging that change is a normal and acceptable reality in software development [1], [2], [8]. This makes close collaboration particularly important to clarify requirements just-in-time and to keep all team members (including the product owner) 'on the same page' throughout the development.

IV. CONCLUSION

Agile is bringing solution and values to the various projects, characterized by a large degree of ignorance in the total requirements in a changing environment. In comparison with traditional methods, agile gives clear instructions, clearly functioning in the real project practice, leading to increased visibility, adaptability, and business value and decreasing of the project risks.

Agile management is certainly subject to further

²Product Backlog: Prioritized requirements list with estimates to turn them into completed functionality

examination and testing for specific projects at specific delivery. Without a doubt, agile management cannot be simply adjusted to all kinds of IT projects without deeper consideration and research based on exact data coming from particular projects.

REFERENCES

- [1] Wikipedia. Agile Software Development. [Online]. Available: http://en.wikipedia.org/wiki/Agile_software_development
- [2] G. M. Weinberg, "Iterative and Incremental Development: A Brief History," *Computer*, vol. 36, ISSN: 0018-9162, 2003.
- [3] K. Beck, A. Cockburn, R. Jeffries, and J. Highsmith. Agile manifesto. [Online]. Available: <http://www.agilemanifesto.org>, 2001, 12-4-2002.
- [4] J. Highsmith, *Agile Software Development Ecosystems*, Boston, MA: Addison-Wesley, 2002.
- [5] D. J. Fernandez, "Agile Project Management – Aligism versus Traditional approaches," *The Journal of Computer Information Systems*, 2008, ISSN: 088-74417.
- [6] B. Kent *et.al.*, "Manifesto for Agile Software Development," *Agile Alliance*, 2010.
- [7] K. Watters. All about agile. [Online]. Available: <http://www.allaboutagile.com/agile-principle-2-agile-development-team-must-be-empowered/>

- [8] C. Martina and S. Alberto, "Project Management in Plan-Based and Agile Companies," *IEEE Computer Society*, 2005, ISSN: 07407459.
- [9] J. Johnson, "Turning Chaos into Success," *Software Magazine*, Dec. 1999, pp. 30.
- [10] D. Port and B. Tung, "Simulating mixed agile and plan-based requirements prioritization strategies, proof of concept and practical implications," *European Journal of Information Systems*, 2009, pp. 317-331, ISSN: 0960085X.
- [11] D. Talby, O. Hazzan, Y. Dubinsky, and A. Keren, "Agile SW Testing in a Large-scale project," *IEEE Software*, Published by the IEEE Computer Society, 0740-7459/06, 2006.



Jan Juricek was born in Prague, Czech Republic in 1980. Mr. Juricek graduated from the University of Economics in Prague, Faculty of informatics and statistics in major specialization of information management.

Besides his PhD studying, he is currently working as a head of PMO in the business sector and is active in professional project management practices for 8 years.

Mr. Juricek is the holder of the international certification for project management practices – ©Prince2 Practitioner, is lecturing techniques of project management and is a member of professional associations for project management in Czech Republic.