

A Comparative Study between Agile Methods of Software Development

Humberto Rocha de Almeida Neto^{1,2,3}, Edviges Mariza Campos de Magalhães⁴,
Alexandre J. H. de O. Luna¹, Hermano Perrelli de Moura¹, Bruno N. Figueiredo²,
Felipe A. A. Ramalho², Felipe D. Alves², Thiago R. Santos²

¹ Centro de Informática – Universidade Federal de Pernambuco - UFPE
Av. Professor Luís Freire, s/n. Cidade Universitária - CEP 50740-540
Recife – PE – Brasil.

² Ciências da Computação – Centro Universitário de João Pessoa - UNIPÊ
Rod BR 230 - Km 22, Água Fria - CEP 58053-000
João Pessoa – PB – Brasil.

³ Serviço Federal de Processamento de Dados - SERPRO
Avenida Parnamirim, 295 - Parnamirim - CEP 52060-901
Recife – PE – Brasil.

⁴ Empresa de Tecnologia e Informações da Previdência Social - DATAPREV
Av. Getúlio Vargas, 47 - 8º Andar - Centro - CEP 58013-240
João Pessoa – PB – Brasil.

{hran, ajhol, hermano}@cin.ufpe.br, edviges.magalhaes@dataprev.gov.br,
{brunonf15, felipeunipe, felipefda812, thiagorochagti}@gmail.com

Abstract. *Agile methods are being more and more used to develop systems to meet requirements such as agility, efficiency and better response to changes. On the other hand, we now have a considerable range of options that, despite being based on the principles of the agile manifesto, may offer distinct features and applicability. In this article we will present an extension of comparative studies between Agile Methods with the aim of spreading such methods and to assist organizations in adopting agile practices.*

1. Introduction

To obtain better results, companies of Information Technology (IT) have adopted methods of software development increasingly flexible and conducive to frequent changes. Such methods, called agile, are considered a response to the more bureaucratic and methods that have proven ineffective and unproductive in the current context of the need for quick responses to market (Ferreira, 2006).

On the other hand, we have now the main motivation for this research, a typical problem currently faced by software factories: "What are the agile methods best suited to my software development projects?"

Given this context, a huge range of agile methods has been offered as a solution, however, organizations may present some difficulty in choosing the method that best suits the characteristics of their software projects.

A considerable range of options which, despite being based on the same principles of the Agile manifesto (Beck, 2001) may offer different features and applications. XP (Jeffries, 2001) and the Scrum (Schwaber et. al, 2011) are a good example of this. While the first focuses on the activities of Engineering, the second operates strongly on issues related to Project Management.

This paper has as main objective to present an extension of a comparative study of agile methods, through the definition of clear criteria and objectives to support the Software Factories who aspire to adopt agile practices in their daily lives as well as researchers who want to know a little about these agile methods.

2. Methodology for the development of this research

For such studies Abrahamsson (2002) and Luna (2010) were identified during the development of this research and served as the basis for initial construction of the proposed extension of the comparative study.

This research was built based on some methodological criteria. As regards the procedural methods, this work uses the comparative method. This method seeks to explain similarities and dissimilarities from observations of two seasons, or two events (Medeiros 1997).

In the case of this work, was an extension of a comparative study based on the works of Abrahamsson (2002) and Luna et. al (2010) by adding four agile concepts, namely: Kanban Development, OpenUP - Open Unified Process, AgileUP - Agile Unified Process and Agile Modeling.

The methodology used in the preparation of this survey has been developed based on bibliographic and descriptive research.

According to (Marconi & Lakatos, 2009) the bibliographical research is developed from material already prepared. It is developed on the basis of materials that are already available, such as books and scientific articles giving the author a greater knowledge and a better approach to the topic to be researched.

And by the end the descriptive research "that aims to completely describe certain phenomenon", Lakatos and Marconi, (2001, p. 188).

3. Selecting agile methods with a view to this study

Although currently there is a considerable range of agile methods available on the market, only some of these were selected to be covered in this study. Research results presented on the use of agile methods were instrumental in the selection of input methods that would make the results of this research. The following will show the results of searches of the state of use of agile methods in Brazil and worldwide.

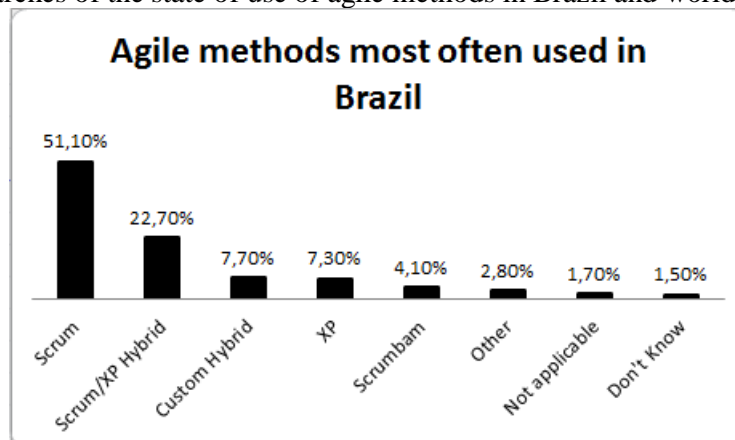


Figure 1. Agile Methods State in Brazil. Source: adapted from (AgilCoop, 2011).

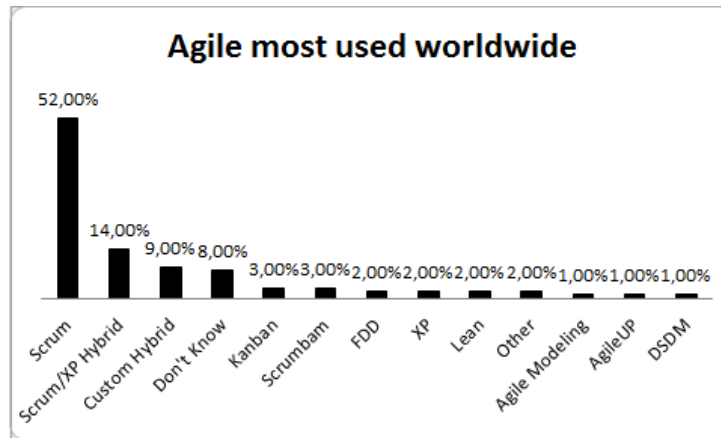


Figure 2. Agile methods used worldwide. Source: adapted from (Versione, 2011)

We observe that in Brazil and in the world the three most used agile methods are Scrum hybrid of XP with Scrum, agile custom hybrid.

Comparing the figures, it is observed that there is a difference in the adoption of agile methods, because besides the aforementioned, the Kanban Development, Agile Modeling and AgileUP were also presented as agile in ascension worldwide, and thus were also included in this work. Apart from these agile methods were treated as part of the scope of this work the OpenUP, who despite not being highlighted in Figure 2 presents strong basement in AgileUP. In surveys we did not identify many scholarly works that speak about these agile methods which can hinder a possible adoption of these agile methods.

The following will be presented a brief overview about the four models that are the focus of expansion of comparative analysis proposed in this work. The proposal is for more people and software developers learn more about this alternative methodological and may in future adopt them in your projects.

3.1. Agile Modeling

According to Ambler (2003), Agile Modeling (AM) is a practice based method for effective modeling software. This method does not follow a prescriptive process, i.e. does not define detailed procedures such as the creation of a given model type. Instead, AM is a collection of values, principles and practices that can be applied by software professionals in their daily lives.

It should be noted, however, that the AM is not in essence simply an agile development method such as XP and Scrum, for example, but an Agile modeling technique. Thus we can say that AM aims to build and maintain effective systems models and efficient and, therefore, can be used in the context of agile methods, thereby increasing the efficiency of modeling and documentation of the project

The principles of AM include: (1) software is the main goal: software that works; (2) enable their next effort is a secondary goal: to always think in the next functionalities; (3) travel light: fewer documents during the project - select documents to be maintained during the development process; (4) assume simplicity; (5) accept change; (6) implement incremental changes; (7) modeling with a purpose: to meet the reality, to improve communication; (8) build multiple models;(9) work with quality; (10) get quick feedback.

3.2. AgileUP - Agile Unified Process

The Agile Unified Process (AgileUP) is a simplified version of the Rational Unified Process (RUP), designed by Scott Ambler. The agile principles that incorporates AgileUP traditional RUP does not have. As well as the OpenUP, AgileUP demand balancing agility and risk control (Ambler, 2005).

AgileUP practices are based on agile techniques such as Test-Driven Development (TDD), Agile Model Driven Development (AMDD), Agile Modeling and refactoring of code (Ambler, 2005). In addition, AgileUP philosophy part of principles that the team knows what they are doing, simplicity, agility and focus on high-value activities (Ambler, 2005). AgileUP teams typically seek to deliver development versions always at the end of each iteration as recommended following the development lifecycle iterative and incremental. As for your process, AgileUP provides a simpler model than the one presented by RUP.

According (Ambler, 2005), AgileUP introduces the concept of stages being divided into: (1) Inception: The goal is to identify the initial scope of the project, a potential architecture for your system, and to obtain initial project funding and stakeholder acceptance; (2) Elaboration: The goal is to prove the architecture of the system; (3) Construction: The goal is to build working software on a regular, incremental basis which meets the highest-priority needs of your project stakeholders; (4) Transition: The goal is to validate and deploy your system into your production environment.

In this context, AgileUP introduces the concept of disciplines, which should be carried out systematically and phase vary dependent on the demand of team effort. The disciplines and their goals are: (a) Model: understand the business of the organization;(b) Implementation: transform the model into executable code and perform a basic level of testing;(c) Test: make an objective assessment to ensure quality;(d) Deployment: plan for the delivery of the system and execute the plan to make the system available to end users;(e) Configuration Management: manage access artifacts of your project;(f) Project Management: delegate the project activities;(g) Environment: support the rest of the effort.

3.3 Kanban Development

Kanban is a Japanese word and literally means "card" or "signaling". It is a concept related to the use of cards (post-it and others) to indicate the progress of the production flows in manufacturing companies. (Agency, 2010)

Pull Production principles, "just in time", total quality and continuous improvement also inspired the software industry and gave rise to the approach of Lean Software Development. Lean Software Development provides a set of principles on the application of a set of techniques from industry and applied in software development. (Agency, 2010)

Kanban is the Lean framework aiming at adaptation to both development and operations, is used to implement the concept of Pull Production, where the actual output of finished products at the end of the Assembly line, said the pace of introduction of raw material in the system. This avoids accumulation of unfinished products along the line leading to decreased quantity of WIP-Work in Progress. With fewer intermediate products, we have an overload in the system and can then adapt better and faster to the context of changes in customer demand. (Anderson, 2010)

The 5 core properties of a Kanban implementation second (Anderson, 2010) are: limiting work in progress, showing the workflow, measure and optimize the flow, make explicit policies of progress, manage quantitatively.

The Kanban uses a visual flag that serves as a signaling tool, leaving explicit the value stream through the process in progress and also facilitates the visualization of activities that are impacting on the success of the project causing all eyes turn to this activity to be resolved and the project continue walking.

3.4 OpenUP - Open Unified Process

The OpenUP is a lightweight unified process implementing iterative and incremental approaches in a structured life cycle. In this way this method seeks to address an agile and pragmatic philosophy focusing on the collaborative nature of software development (OpenUP, 2010).

The four principles of OpenUP are: (1) balance the competing priorities to maximize the benefit to stakeholders; (2) work together to align the interests and share the understanding; (3) focus on architecture as soon as possible thereby reducing the risk and arranging development; and (4) evolve continuously to get feedback and promote improvements. Furthermore, this method provides well-defined roles, they are: (2) Interested, (2) Analyst; (3) Architect; (4) Developer; (5) the testator; and (6) project manager. Each of these papers has responsibilities that relate to the artifacts that must be created, modified, or used under its responsibility.

The OpenUP structure the project lifecycle into four phases: (1) design; (2) preparation; (3) and (4) Transition. In each of the four phases of the life cycle of a project there is a landmark that determines formally at the end of each phase. (OpenUP, 2010). In addition, the project is divided into iterations planned and timed intervals, which usually is measured in weeks. These iterations direct the team at predictable and incremental delivery of value to stakeholders. The iteration plan defines what must be delivered and as a result there is a demonstrable construction or deliverable.

The focus of OpenUP is intended for small teams working together and in the same location. Among the team members are: stakeholders, developers, architects, the project manager and the testers. (OpenUP, 2010).

4. Results: Extending comparative analyses about agile methods

The work of Abrahamsson (2002) and Luna et. al (2010) were the basis for the extension of comparative analysis in this research proposal. From the criteria and methods developed in the scope of each of these works, it became possible to build a comparative analysis between which, in a second moment, added four agile methods (Kanban, Agile modeling, AgileUP and OpenUP).

Table 1. Comparative analysis between the agile methodologies, expanding the study by Luna et. al (2010), with addition of methods: Kanban, Agile Modeling, AgileUP and OpenUP.

| Methods | Key points | Main features | Limitations/Flaws |
|---------|---|---|--|
| XP | Customer-driven development. | Refactoring improves system performance and is responsible for the changes. | Little attention in management practice. |
| SCRUM | Small, sortable, self-development cycle of up to 15 days. | Product overview well defined and repeatable. | Lack of integration testing and silent in relation to aspects of implementation. |

| | | | |
|----------------|--|---|--|
| XPM | Complements the lack of managerial approach of XP. Recommends joint application. | Facing e-Projects. The results lead the planning and changes are encouraged. | The practices are very subjective, you need high degree of maturity of the project manager to put them into practice. |
| APM | Believes strongly in adoption of principles that explore the understanding of human behavior. | Considers that the collective behavior is characterized by a superposition of order, self-organization and a collective intelligence that is greater than the sum of its parts. | Requires a lot of experience of the Manager in leading people to extract the best result of the method. Not recommended for little mature teams. |
| YP | Simplified process that rests on practices of XP, RUP and Agile Modeling. | For use in academic or commercial projects of small and medium-sized businesses. | Recommended for small-scope projects, which can be completed within four months. |
| FDD | Formed by five processes and short iterations. | Simple method development for features and object modeling. | Focus only on the project and implementation. |
| CRYSTAL | Several methods with different characteristics. | Ability to select the most suitable method to the project. | Difficulty in the use of estimates. |
| DDSM | Use of the RAD, team with autonomy to make decisions. | Uses the prototyping and have various roles (responsible) for performing the same activity in the method. | Only the team members have access to the procedures of this method. |
| ADS | Focuses on the Adaptive cycle, collaborative and iterative development. | From the philosophy of adaptive systems. | Are based more on the concepts and culture than in agile practices. |
| ADS | Focuses on the Adaptive cycle, collaborative and iterative development. | From the philosophy of adaptive systems. | Are based more on the concepts and culture than in agile practices. |
| KANBAN | Focuses the work on progress. | Limited work thus avoiding accumulation of tasks and so can adapt better and faster to changes in customer demand. | No worries about the estimates. |
| AGILE MODELING | Facilitates the understanding of those involved in the project in the quickest way possible and practical. | Used to increase the efficiency of modeling and documentation. | Is a supplement to existing methods, it is not a complete methodology. |
| OPENUP | Discusses an agile and pragmatic philosophy that focuses on the collaborative nature of software development. | Focus on the significant reduction of risks making mandatory regular meetings. | Recommended for local teams. |
| AGILEUP | His philosophy the following principles that the team knows what they are doing, simplicity, agility and focus on high-value activities. | Simplified version of RUP and agile principles which incorporates demand balancing agility and risk control. | There is some slowness of the first version in relation to each other, increasing the risk of change. |

In table 1 presents a comparative study of agile methods, based on the study by Luna et al. (2010) and supplemented by this work, pointing out the key points, the main features and flaws between the methods presented here.

To show the table 1 the reader will have specific information about the agile methods most commonly used in Brazil and in the world.

However, for the adoption of an agile approach, organizations must be willing to change their perception in relation to their customers, to re-evaluate how they view their projects and take some risks.

5. Conclusions

Although it is not the solution to all problems, the agile method shows one way to work better long term quality and features such as a better acceptance and adaptation to change. Issues such as these have been shown as important competitive differentiators for software development organizations.

Using A method appropriate to the organizational and project reality, further enhances the production of activities, adapt perfectly to the characteristics and needs. In spite of the agile methods are based on the principles of the Agile manifesto, nevertheless, some of these have peculiarities and its specific features.

This work presented a brief extension of a comparative study between agile methods. It is hoped that this study will support the dissemination of agile practices as well as serve as initial input to support the Software factories that aspire to adopt agile practices in their daily lives and also to researchers who want to know a little more about these methods.

6. Future Works

As future work, are being mapped new comparison criteria and other agile methods with a view to the integration of research. From this comparative study will seek to understand how the best agile practices have been used. In addition this research to serve as a foundation for future studies involving agile methodologies.

References

Abrahamsson, P. (2002). Agile software development methods: Review and Analysis.

AgilCoop. (2011). Resultado da pesquisa: Métodos ágeis no Brasil - estado da prática em times e organizações. Available in: <<http://www.agilcoop.org.br/MetodosAgeisBrasil2011>>. Accessed on 30 May 2012.

Agency, US Environmental Protection (2010). Lean Thinking and Methods. Available in: <<http://www.epa.gov/lean/environment/methods/kanban.htm>> Accessed on 17 abr 2012.

Ambler, S. W. (2005). Agile Unified Process. Available in: <<http://www.ambysoft.com/unifiedprocess/agileUP.html>>. Accessed on 20/04/2012.

Ambler, S. W. (2003). Agile Modeling. Available in: <<http://www.agilemodeling.com>>. Accessed on 20 Apr 2012.

Anderson, David J. (2010). Kanban: Successful Evolutionary Change for Your Technology Business.

Beck, K. et al. (2001). Agile Manifesto. Available in: <<http://www.agilemanifesto.org>>. Accessed on 27 Apr 2012.

Jeffries, R. XP Magazine Contents: What is Extreme Programming? (2001). Available in: <<http://xprogramming.com/xpmag/whatisxp>>. Accessed on 27 Apr 2012.

LUNA, ALEXANDRE J. H. DE O.; COSTA, CLEYVERSON P.; DE MOURA, HERMANO P. (2010). A necessidade de ser Ágil. Engenharia de Software Magazine, No. 37. ISSN 1983-1277. Available in: <<http://www.devmedia.com.br/post-21425-A-necessidade-de-ser-agil-Revista-Engenharia-de-Software-Magazine-37.html>>. Accessed on 20 Apr 2012.

Marconi, M. d., & Lakatos, E. M. (2009). Fundamentos de metodologia científica. São Paulo: Atlas.

OpenUP (2010). Open Unified Process. Available in: <<http://epf.eclipse.org/wikis/openup/>> Accessed on 20 Apr 2012.

Research, Analysis.Net. (2010). State of Agile Development Survey Results. Available in: <http://www.versionone.com/state_of_agile_development_survey/11/> Accessed on 22 Apr 2012.

Santos, A. D. (2003). Agile Modeling - Overview. Available in: <<http://www.ambysoft.com/books/agileModeling.html>> Accessed on: 20 Apr 2012

Schwaber, Ken, e Jeff Sutherland. (2011). The Scrum Guide. Available in: <<http://www.scrum.org/scrumguides/>> Accessed on 27 Apr 2012.