THE CHALLENGE OF IMPLEMENTING SCRUM AGILE METHODOLOGY IN A TRADITIONAL DEVELOPMENT ENVIRONMENT

Renato Takeki Nishijima
Universidade Católica de Brasília
430-0812, Shizuoka-ken Hamamatsu-shi
Hongo-cho 1315-15 Hongo Park ll 102, Japan
renaton@hotmail.com

José Gonçalo dos Santos
Universidade Católica de Brasília
QS 07 Lote 01 EPCT, Águas Claras - CEP: 71966-700
Taguatinga/DF, Brazil
jose.goncalo.santos@gmail.com

ABSTRACT

Agile software development management is gradually gaining more profile and followers among technology companies.

The current article presents a study on agile methodologies in contrast to traditional methodologies as well as the motivation to implement agile software development management techniques in small projects by providing an overview of the challenges, objectives, reasons, advantages and disadvantages compared to traditional software development management.

General Terms

Software Development Methodology

Indexing terms

Agile methodologies; traditional methodologies; RUP; software development management; scrum.

Academic Discipline And Sub-Disciplines

Provide examples of relevant academic disciplines for this journal: E.g., History; Education; Sociology; Psychology; Cultural Studies;

SUBJECT CLASSIFICATION

Software Engineering

COVERAGE

This paper refers to the use of agile methodology for software development worldwide, focusing on the advantages of using such methodology.

TYPE (METHOD/APPROACH)

The methodology used to develop this study was Literary Analysis.

1. INTRODUCTION

Relatively new, agile methodologies have been gaining more and more followers among large companies like Microsoft, Google, EA (Electronic Arts), Yahoo, Siemens, Nokia and others.

The Agile methodologies have been used as an alternative to traditional methodologies (considered heavy), in a world more and more globalized and competitive, which require faster market responses and ensure on-time deliveries also adding value and quality to their products and / or services.

Scrum, as well Extreme Programming (XP) and others methodologies, are classified as agile software development. According to Teles [17], this new concept of methodology has emerged in America in the 90s, in an attempt to create better systems in less time, and produced more economically than usual. These objectives are achieved through a set of values, principles and practices that differ primarily in the traditional way to develop software.

Based on this scenario, this article aims to expose and understand the applicability of Agile Methodology, specifically Scrum in a traditional development environment of IT.

1.1 ENVIRONMENT DESCRIPTION

“The market for software development is extremely inefficient and ineffective” [9].

In an increasingly competitive world, whose changes happen in a frequent and faster pace, the adoption of agile management methods becomes more and more necessary to achieve greater agility and flexibility when creating quality software in a short period time, complying with the constraints of time, cost, scope and quality.

The agile philosophy is still a relatively new concept and still the scene of many discussions.

1.2 MOTIVATION

The main motivation of the present article is to demonstrate the benefits and challenges of implementing agile methods in IT companies whose market increasingly competitive requires fast delivery of high quality systems that answer the requests of change with flexibility.

In a survey conducted by the Cutter Consortium1 in 2001[6], with more than 200 professionals from different companies in the United States, Europe, India and Australia on agile and traditional development methods reached three interesting conclusions:

There was an increase in the number of companies that used some method of Agile Software Development when compared to a similar survey conducted in the preceding year;

Development projects carried out by the agile approach performed better in terms of compliance with the requirements of the business, customer satisfaction and quality;

The Agile received better scores in the category "moral of professionals", considered this result, at the time, astonishing because only 12% of respondents were analysts or programmers.

1.3 PROBLEMS DIAGNOSED

The problems identified in implementing the Scrum agile methodology are:

The main one is the cultural resistance to change within the organization, naturally rejected by withdrawing those involved from their comfort zone.

Clients must be fully committed to the project, having necessary knowledge and available to answer questions when needed.

1.4 GOALS

The purpose of this article is to provide an overview of the difficulties and obstacles, as well as the advantages of implementing agile methods in a traditional development environment.

2. THEORETICAL

It will be made a brief presentation of the traditional methods (section 2) represented by the classical model (section 2.1.1) and the model of iterative and incremental development RUP (section 2.1.3), as well as the advantages and disadvantages (section 2.3 and 2.4) of using traditional methods, and also a brief introduction to agile methods (section 3) with emphasis on the Scrum agile method (section 4). Finally, in section 5 we will have a comparative study between traditional and agile development methods.

2.1 TRADITIONAL METHODS OR "HEAVYWEIGHT"

Traditional methodologies known as heavy or oriented planning, try to anticipate all of the system requirements to be done so prior planning of how the system will work. This is represented as rigorous planning documents that will guide the entire development process [12].

Efforts are directed to obtaining project artifacts through complete and accurate reports, documents and diagrams. Much of the traditional processes in use today is based on the model diagrams of the Unified Modeling Language (UML), such as RUP.

In Traditional Methodology, the project indicates a well-defined beginning and end of the project scope. During the development of the Project Charter, assumptions and constraints defined will provide the basis for modeling the entire project planning following the guidelines of PMI (Project Management Institute) that discusses the best practices of project management.

However, this methodology has emerged in an era when software development was based on mainframes and “dumb terminals”[13]. There were no tools to support development as code analyzers and debuggers. The documentation had to be comprehensive and consistent because it costs to make a change was very high.

2.1.1 Classic Model:

According to Roger Pressman [11] , the classic model was the first methodology published and is still widely used in current days.

[1] Cutter Consortium is a global IT advisory firm dedicated to helping organizations forge solutions to the business technology challenges they face.

The Classic model or "Linear Sequential model", establishes a sequence of steps. Each stage has a beginning and an end with a strict documentation set to be followed between one step and another, such that without the documentation, the process does not continue. The classic example of this model is called “Waterfall” (see Figure 1).

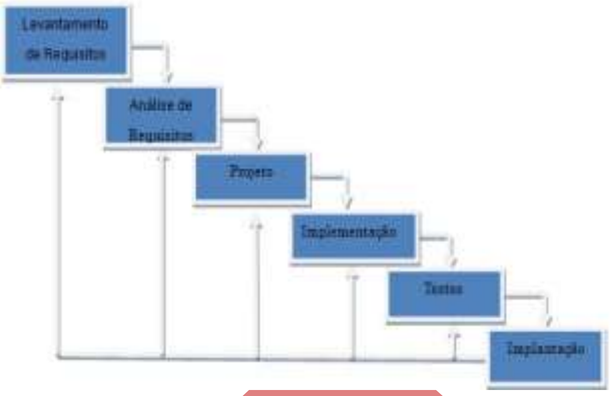

Fig 1: Waterfall Development Model[11]

The spiral model is also a classical model, but unlike the waterfall model, allows return to earlier stages.

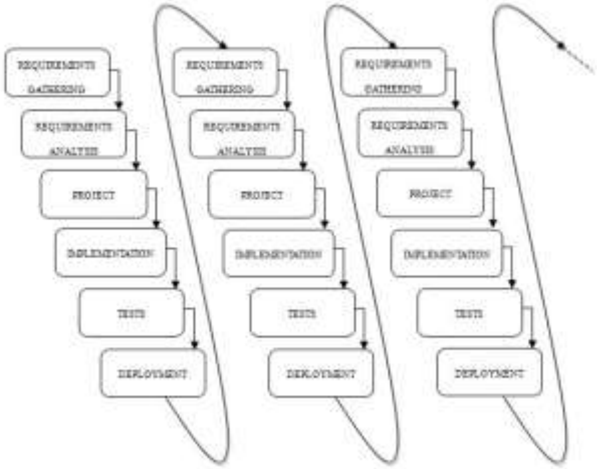

Fig 1: Iterative and Incremental Model [11]

2.1.2 Iterative and Incremental Model

The iterative and incremental model (see Figure 2), replaces the classical waterfall development, making it a bit more dynamic because is divided into a series of time-boxed iterations. Each iteration results in an increment, which implements each of its disciplines.

2.2 RUP

Created by Rational Software Corporation, RUP[2] is a well-defined and structured software engineering process, which clearly specifies who is responsible for each task, how and when they should be performed.

The RUP is based on three basic elements: Use cases that guide the whole process of development, architecture-centric and iteration, incorporating the best practices of software development through guidelines and skeletons that help programmers to focus on the project. These best practices are:

- Iterative development;
- Requirement Management;
- Component based architecture;
- Use of visual modeling software;
- Continuous quality improvement software;
- Change management control software.

[2] The Rational Unified Process (RUP) is a software process product, originally developed by Rational Software, which was acquired by IBM in February 2003.

The main goal of RUP is to meet the needs of users ensuring production of high quality software that meets a timeline within a predictable budget.

The RUP has five main items:

- Roles
- Activities
- Artifacts
- Workflows
- Disciplines

According to Per Kroll [10], RUP organizes the software development into four steps:

- Home (Inception);
- Development (Elaboration);
- Construction (Construction);
- Transition (Transition).

They deal with issues about planning, requirements gathering, analysis, implementation, testing and deployment of software. Each phase has a crucial role to achieve the objective, distributed among various professionals such as System Analyst, Designer, Test Designer, among others.

Within each phase of the RUP, the tasks are categorized into nine disciplines listed below:

1st. Business Modeling

Business Modeling ensures that the goals and expectations of all stakeholders of the project are aligned with the organization's goals.

2nd. Requirements

The purpose of Requirements is: To define the limits of the system and, according to the requirements of this new system, create use cases that will be the basis for estimating the costs and development efforts. The project stakeholders must understand and accept everything that the system should do.

3rd. Analysis and Design

The purpose of Analysis and Design is: To turn requirements into a design of the system to be built. Produce technical specifications that will be followed in the implementation of each use case of the system.

4th. Implementation

The purpose of Implementation is: To transforming the logical and physical models created in Analysis and Design in usable source code and test the developed components as units.

5th. Tests

The purposes of testing are: To find, document and address the flaws in the quality of the software produced. These defects arise from a comparison between what was produced with what was specified in the requirements and logical and physical models of the product;

6th. Deployment

Deployment ensures that the software produced is available to your end users.

7th. Configuration and Change Management

Provides Change control and maintain the integrity of each of the artifacts produced in the project. Each of these artifacts must be identified, assessed and have all configuration and maintenance defined;

8th. Project Management

The purpose of Project Management is: To balance competing objectives, manage risk, monitor the project and treat rules that deliver a product that will meet the expectations of customers and end users.

9th. Environment

The main objective is to set up the environment to run the development process and its activities, providing the tools and processes necessary to ensure that all project activities can be adequately performed by each role.

As shown in Figure 3, the RUP has two dimensions:

- The horizontal axis represents time and shows aspects of the life cycle of the process as it develops. It represents the dynamic aspect of the process.

- The vertical axis represents the disciplines, which logically group the process content. It represents the static aspect of the process.

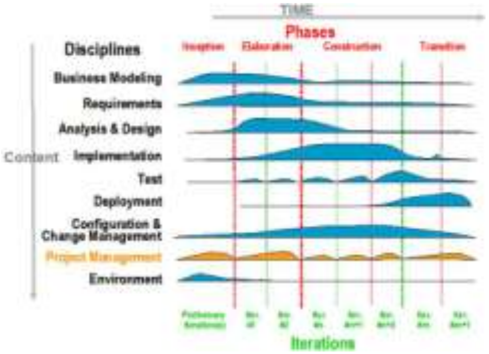

Fig 3: RUP Life Cycle (IBM.com)

2.3 PMBOK

The PMBOK ® (Project Management Body of Knowledge) is a guide approved by ANSI (American National Standard) and was developed by the PMI (Project Management Institute) that discusses the best practices of project management. The PMBOK ® addresses all the vital areas of good planning and guides project managers to achieve their objectives leading on time, with quality and budget with the least possible contingencies.

The main objective of PMBOK ® is to visualize the project management as a set of linked and integrated processes. Processes are a series of actions that cause results through 42 structured processes into 5 groups and nine knowledge areas. The five process groups of the PMBOK ® are: Initiating, Planning, Executing, Monitoring and Controlling, and Closing.

The nine knowledge areas of project management PMBOK ® are: Integration, Scope, Time, Cost, Quality, Human Resources, Communications, Risk and Procurement.

It's worth remembering, the PMBOK ® is not a management methodology, but rather a guide that comprises knowledge of a set of best practices.

2.4 ADVANTAGES

The traditional management methodology is still the most used. But this situation is changing with the introduction of agile methodology that is a gaining a huge number of followers and supporters all over the world.

One of the major advantages of the traditional method is to obtain data on historical facts and repetition, improving the ability of the process through standardization, measurement and control of the project.

2.5 DISADVANTAGES

The Classic model was the only model until the mid-1990s. Since then, several studies questioning the effectiveness of the model began to emerge.

It was noticed that the traditional models hid a number of serious problems, such as:

• Problem for delivery within the scope of considering all the features requested (Standish Group, 1995).Based on 8330 projects, only 16.2% of these projects were delivered on time and cost stated initially. About 31% were canceled, and 52.7% were delivered with only 61% of the functionality provided without respecting deadlines and cost an average of 159% higher than forecast;

• High occurrence of errors, since the traditional methodology hampers the implementation of changes during the development process;

• A famous article by Fred Brooks, "No Silver Bullet: Essence and Accidents of Software Engineering"[4], demonstrated that the idea of fully specify the software before its beginning is totally impossible;

• Should be applied only in situations where the system requirements are stable and foreseeable future requirements;

•The traditional method of development (CSS) does not allow changes;

•The RUP is characterized by the large amount of documents generated that sometimes slows down the development of the project;

•Agile processes have emerged as an alternative to traditional processes, because they were best suited for environments where requirements change is frequent.

3. AGILE "LIGHTWEIGHT" DEVELOPMENT METHODOLOGIES

3.1 HISTORY

lthough the base of Agile Methods have been IIDD[3], acronym for iterative and incremental design and development, method adopted by engineers over 75 years ago and that, in mid-1950s was applied in software development. This method preached the advantages of "avoiding the discouragement of management" and "increase customer satisfaction.".

In 1976, Tom Gilb, in his book Software Metrics, argued that "the evolutionary development resulted in the delivery of superior software, launching a movement that was intended agile and lightweight iterations, able to provide rapid results and visible business benefits with frequency."

In the 1990s, the modern Agile began to take shape when in 2001, Kent Beck and 16 other noted software developers, writers and consultants met to discuss the best ways in the process of software development, creating thereafter the Agile Alliance and the establishment of Agile Manifesto[4].

Through this work, it stated that:

- Individuals and interactions over processes and tools;
- Working software over comprehensive documentation;
- Customer collaboration over contract negotiation;
- Responding to change over following a plan.

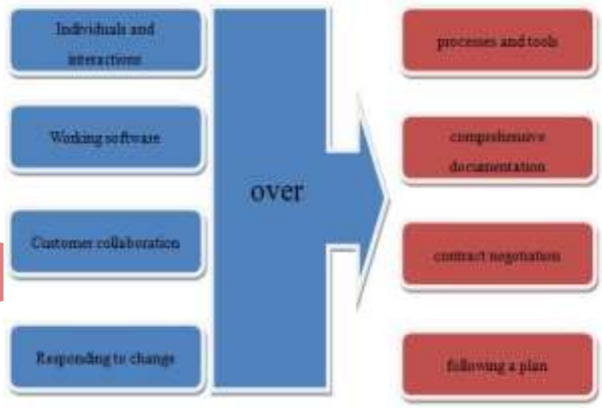

Fig 4: Agile Manifesto (UCB)

The Agile Alliance defines 12 principles[5] for those who want to achieve agility:

1- Our highest priority is to satisfy the customer from the beginning through continuous delivery of valuable software;

2- Changes in requirements are welcome, even late in development. Agile processes harness change for the customer´s competitive advantage;

3- Delivery of working software frequently, every two weeks to two months, preferably in the shortest time;

4- The business personnel and developers must work together daily throughout the project;

5- Build projects around motivated individuals. Give them the environment and support they need and trust that they will do the job;

6- The most efficient and effective method to bring information into the development team is face-to-face conversation;

7- Working software is the primary measure of progress;

8- Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain a constant pace indefinitely;

9- Continuous attention to technical excellence and good design enhances agility;

11- The best architectures, requirements and designs emerge from self-organized teams;

12 - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

[3] IIDD- *Iterative and Incremental Design and Development*

[4] http://agilemanifesto.org/

[5] http://agilemanifesto.org/principles.html

gile methodologies bring concepts that are not applied in the traditional methodology as higher customer participation in the process, extremely short iteration and the emphasis on automated testing.

Agile development provides important benefits, but it should be noted that it is not applicable to all kinds of projects, products, people and situations, nor is contrary to traditional practices of software development.

3.2 AGILE PROCESSES

One of the best known methods in this category are XP (Extreme Programming) or XPM (Extreme Project Management) and SCRUM, which will be focused in this article. We can also quote many others as: DSDM (Dynamic System Development Method), Crystal, Feature Driven Development (FDD), Lean, Test-driven development (TDD), etc…

4. SCRUM[6]

Scrum is an agile software development method that was conceived by Jeff Sutherland in the early 90 s and in recent years featured collaborations of Ken Schwaber and Mike Beedle, aligned with the Agile Manifesto.

The SCRUM follows an iterative and incremental philosophy to optimize predictability and risk control. His principles guide the development activities within a process that incorporates the framework of activities: requirements, analysis, design, development and delivery, worth remembering that Scrum is not synonymous with agile management, but one of many management frameworks mentioned above.

4.1 MAIN ROLES OF SCRUM

- **Product Owner**- Is the one who represents the stakeholders;
- **Scrum Master**- is a facilitative team leader who ensures that the team adheres to its chosen process and removes blocking issues;
- **ScrumTeam** – Is the project team designated to analysis, implementation, design, testing etc... .

4.2 ARTIFACTS

- **Product Backlog** - It's basically a list of requirements (stories) specifying the priorities of the Product owner in a language easily understood by the customer.
- **Sprint Backlog** - Artifacts that represent activities conducted within the predefined time.
- **Release Burndown** - Document that measures the Product backlog remaining.
- **Sprint Burndown-** This is a document that measures the Sprint Backlog items remaining over time in a Sprint.

4.3 SCRUM PROCESS CYCLE

Each of activities takes place within the framework of a standard process and time called Sprint. The amount of Sprints needed to complete each activity varies with the size and design complexity.

A SCRUM project can be started even if you have only a superficial view, which will be clarified as the project evolves.

4.4 TIME-BOXES

Time-Boxes are short-term events. It is a simple technique applied to software development. In Scrum, the technique applies to meetings and Sprints making them more objective. Its components are:

- Sprint planning meeting;
- Release planning meeting;
- Sprint;
- Daily Scrum;
- Release planning meeting;
- Sprint review meeting;
- Sprint Retrospective.

4.4.1 Release planning

The team discusses the plans and goals of the release, its general characteristic features and risks, improving understanding and communication.

4.4.2 Sprint Planning

[6]http://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum%20Guide%20-%20Portuguese%20BR.pdf

It is held a planning meeting (Sprint planning) between the team (Scrum Team) and the customer (Product Owner). It elaborates a list (which can add new items throughout the project) of all the features and requirements expected of the software, known as Product Backlog.

At the beginning of each Sprint, a list of items that are part of Product Backlog Sprint Backlog is known as selected according to priorities.

4.4.3 Sprint

It represents one cycle (iteration) which may last for two to four weeks. Generally the four-week cycle is adopted in order to obtain better overall visibility.

4.4.4 Sprint Review

The team holds an informal monthly meeting lasting an average of four hours and can be adapted if Sprint is shorter.

The team analyzes the changes made in increments committed to Product Backlog, reviewing features and discussing difficulties and successes during the Sprint, to decide what to do next.

“Sprint Review provides valuable input to subsequent Sprint Planning Meetings.” [16].

The Product Owner verifies which Product Backlog items were completed in the Sprint, and discusses with the development team, what will be the new priorities. If there is no impediment, a new Product Backlog is generated starting a new Sprint.

4.4.5 Sprint Retrospective

The team and Scrum Master meet to discuss what went well and what to improve in the next sprint. The product owner does not attend this meeting.

“At the end of the Retrospective, the Scrum Team should have identified improvement measures to be implemented in the next Sprint” [16].

Each of these sprints completed and delivered, will be implemented toward to the full product.

4.5 DAILY SCRUM

On each day of a sprint, the team holds daily standup meetings time-boxed to 15 minutes in same place and same time to follow the progress of development making use of Burndown Chart for monitoring tasks. Its members should answer questions like:

- What difficulties and obstacles are in your way?
- What was done since the previous daily Scrum meeting?
- What will be done before the next meeting?

Daily Scrums improve communications, identify and remove impediments to development, highlight and promote quick decision-making, and improve the Development Team’s level of project knowledge**.**

The Scrum Master must lead the meeting and ensure that all the members of team are present, keeping the brevity of the interventions and strengthen the rules.

“The Daily Scrum Meeting is the continuous inspection and adaptation mechanism of Scrum” [16].

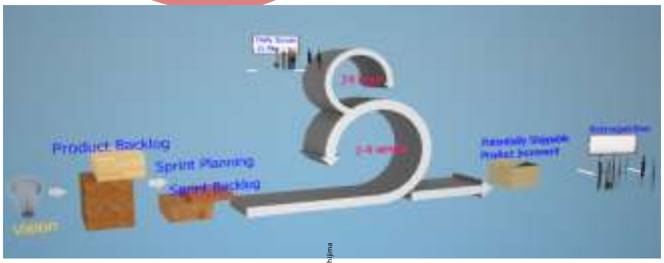

Fig 5: Scrum Cycle[11]

Renato [illegible]ki Nishijima

5. AGILE VS. TRADITIONAL

"***Most*** *of the* ***agile*** *practices are* ***nothing new***" [6]

One of the most frequent questions is: Which is the best, traditional or agile methodology?

It's a hard question to answer because it depends on many factors such as company culture, size of project life cycle, etc.

According to Soares [15]:

"Heavy methods should be applied only in situations where the software requirements are stable and future requirements are predictable. These conditions are difficult to achieve, since the requirements for development of software are changeable. Among the factors responsible for changes in requirements are dynamic organizations, changes in laws and changes requested by stakeholders, which usually has difficulties in defining the scope of future software".

T**hat's where the advanta**ge comes from the use of Agile Methods. The option of using agile methodologies provides a means to evaluate and respond to frequent changing of requirements or external factors experienced during the development of software, making it more flexible.

From the individual analysis of the traditional Methodology RUP and the Agile Scrum Methodology, both discussed in this article, it will be made a brief introduction of the comparison between these two methodologies.

5.1 SCRUM VS. RUP

IBM Rational Unified Process, or RUP, uses the spiral lifecycle (iterative) and is widely used in large projects in software development, but nothing prevents it from being used on smaller projects although not recommended. The RUP is heavy, with emphasis on documentation and large teams with well-defined roles, but slow to respond to requirements changes, a fact that is very commonplace in the software development environment.

Scrum is a methodology based on Agile Manifesto, where more emphasis is given to a working software over the comprehensive documentation, and that satisfies the requirements and features specified by the customer in the shortest time possible.

Such as RUP, Scrum is a method based on iterations, but with more freedom to set the size of these iterations, called Sprints in Scrum defined by a small team formed generally by 5-9 members working in only 3 roles: Product Owner, Scrum Master and Team (team Scrum).

The Scrum Guide says that "Scrum is focused on the development of complex products." Can be adapted to various situations and also complement traditional methodologies such as RUP, MPS.br[7], PMS, and others.

We cannot say which is best. The choice of methodology will depend on many factors, such as team size, project size, if the customer can follow the project closely, among other.

6. DEPLOYMENT

The three main critical success factors for an agile project are: culture, people and communication.

The process of cultural change is the main obstacle in the adoption of Scrum within organizations. In face of this situation, [7] as well as Ambler [2], argue for a gradual transition from classical processes for agile development, making the transition smoother.

The senior management must be aware of the principle that the deployment of agile methods in the company, may affect beyond the development team, customers, managers and even other departments and areas with no apparent link with the project. They should establish a general consensus on the new form of work thus avoiding, wear and future problems. In this context, it is imperative that the Human Resources department be involved from the beginning of the deployment, acquiring knowledge of the new way of working to providing support, solving doubts and easing the anxieties of those involved in the transition process of change [7].

In the adoption of Scrum, first and foremost there needs to be awareness that the decentralization of control is inevitable, that is, an organizational environment where it is no longer necessary figure of superiors. Naturally this kind of change is rejected by the people involved by taking them from their comfort zone. The acceptance of these changes generally happens because there is the reward of performing the work more enjoyable, creative in a way that makes sense, then generating enthusiasm and satisfaction according to "The Enterprise and Scrum" by Ken Schwaber.

Once overcome the barrier of rejection, through the process of understanding, acceptance and recognition of the benefits, the theoretical concept of the Scrum framework and the introduction of its artifacts, roles and ceremonies are presented to the team.

Traditional Work teams give place to self-organized teams with autonomy. In Scrum, what prevails is the mutual cooperation of the members of the team. The Scrum Master and the Product Owner are set to their well-defined roles. Leadership is distributed among all members.

The Scrum Master has a fundamental role as a disseminator of this new culture, organizing and leading the team with no influence of upper management in an environment of mutual trust, working mainly as a mediator of the team.

7. CONCLUSION

The Agile Scrum methodology can be seen as a refinement of iterative methodologies with emphasis on the interaction between project members and Stakeholders[8].

[7] MPS.Br (Brazilian Improvement Software Process or *Melhoria de Processos do Software Brasileiro* in Portuguese) is at same time a program for software development improvement in Brazil (MPS.BR) as well a process (MPS Model).

Even the PMI (Project Management Institute) recognized the need to provide alternatives to suit the present scenario, more dynamic, where changes are more frequent, with the Rolling Wave Planning, or "planning in waves," which is based on progressive planning project, where the work to be done in the short term is more detailed, and progresses to next step, or wave, as the project proceeds and later details become clearer. So, it has agile technical features defined in the PMBOK ® Guide, and looks a lot like agile iterative development.

Thus, it was noticed that Scrum can be used in addition to the traditional methods. It has the distinction of being a good communication tool, which provides a much more collaborative and pleasant to work with.

It strengthens the ties of relationship with the customer, because they become part of the team through the figure of the Product Owner. Generates greater motivation of programmers, reduces bugs, and changes in requirements can be performed at any time, providing greater flexibility for development.

The implementation of Scrum can bring great benefits, but its success will depend on the conduction of cultural change strategy within the organization and the commitment of the entire team.

REFERENCES

[1] AGUIAR, Luiz. Introduction to Scrum. 2008. Available at: <http://www.gonow.com.br/blog/introducao-ao-scrum> Accessed on October 1, 2012.

[2] AMBLER, S.W. When does(n´t) agile modeling make sense. Apr, 2002c. Available at: < http://www.agilemodeling.com/essays/whendoesAmWork.htm>. Acessed October 14,2012

[3] BASTOS, Felipe. Scrum Projects Simple and Complex. 2012. Available at: <http://blog.felipebastosweb.com.br/2012/scrum-em-projetos-simples-e-complexos/> Accessed October 14, 2012.

[4] BROOKS, Frederick. No Silver Bullet: Essence and Accidents of Software Engineering. 1987. Available at: <http://www.cs.nott.ac.uk/~cah/G51ISS/Documents/NoSilverBullet.html>.

[5] CISNEIROS, Hugo. Agile Development Model. 2009. Available at: <http://www.devin.com.br/modelo-scrum/> Accessed October 14, 2012.

[6] COCKBURN A. and HIGHSMITH J. (2001a; 2001b). Agile Software Development: The People Factor. IEEE Software 34(11):131–133.

[7] COHN, M ; FORD, D.. Introducing an agile process to an organization. Available at: <http://www.mountaingoatsoftware.com/articles/introducing-an-agile-process-to-an-organization>. Acessed October 12, 2012.

[8] FERSTE, Mauricio Antonio. Agile Methodologies. 2009. Available at: <http://waltercunha.com/blog/index.php/2009/06/29/metodologias-ageis/> Accessed October 13, 2012.

[9] JOHNSON, J..The Chaos Report. (1995, 2001, 2003). The Standish Group International Inc. West Yarmouth, MA.

[10] KROLL, Per. The spirit of the RUP. 2001. Available at: <http://www.ibm.com/developerworks/rational/library/content/RationalEdge/dec01/TheSpiritoftheRUPDec01.pdf>. Acessed October 12, 2012.

[11] PRESSMAN, Roger S. Software engineering 6th edition. Publisher: MCGRAW-HILL HIGHER EDUCATION.

[12] ROCHA, Thayssa Águila da; OLIVEIRA, Sandro Ronaldo Bezerra; VASCONCELOS, Alexandre Marcos Lins de. Process adaptation for software factories. 2004. Available at: < http://www.simpros.com.br/Apresentacoes_PDF/Artigos/Art_12_Simpros2004.pdf > Acessed October 13, 2012.

[13] ROYCE, Winston W.. Managing the development of large software systems. 1970. Available at: <http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf>. Acessed October 13, 2012.

[14] RSIMOES01. How to build high performance teams. 2011. Available at: <http://scrumex.com.br/blog/?p=675> Accessed October 20, 2012.

[15] SOARES, Michel dos Santos. Comparison of Traditional and Agile Methodologies for Software Development. 2009. Available at:<http://wiki.dcc.ufba.br/pub/Aside/ProjetoBitecIsaac/Met._Ageis.pdf> Accessed on October 5, 2012.

[16] SCHWABER, Ken; SUTHERLAND, Jeff. The Scrum Guide. Available at: <http://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum%20Guide%20-%20Portuguese%20BR.pdf> Accessed on October 1, 2012>.

[17] TELES, Vinícius Manhães. SCRUM. 2008. Available at: <http://improveit.com.br/scrum> Access: Accessed October 5, 2012.

[18] VICENTE, Dinei. What is Scrum? 2008. Available at: <http://www.tuister.com.br/scrum/> Accessed October 13, 2012.

[8] Stakeholders: A person, group or organization that has interest or concern in an organization.

Renato Takeki Nishijima

Renato Takeki Nishijima was born in Santos, Brazil. He has graduated high school Data Processing Technician from Bradesco Foundation in 1991, and since then living in Japan. He has worked at Sony Co. for eight years, acting as quality manager complying with six-sigma doctrine in the Optical Device Development and Manufacturing Department. In 2011 he was awarded with the JLPT(Japanese Language Proficiency Test) Certificate of Proficiency Level N2 and now pursuing Level N1. Presently, he is working at the IT Sector of the Consulate-General of Brazil in Hamamatsu, Shizuoka Prefecture, Japan and is a fifth semester student in analysis and systems development at Catholic University of Brasília.

José Gonçalo dos Santos

José Gonçalo dos Santos was born in Cuiabá-MT, Brazil. He has graduated in Mathematics from the Federal University of Mato Grosso (1994) Technology and IT Management from the Catholic University of Brasilia, postgraduate in computational mathematics from the Federal University of Mato Grosso (1996) and requirements engineering and business process Federal University of Rio Grande do Sul (2012), Masters in Computer Science from the Federal University of Santa Catarina (2001) and Ph.D. in Computer Science from the Federal University of Santa Catarina (2004). Has experience in the area of Computer Science, with emphasis in Computer Systems Architecture, mainly in the following themes: expert systems, artificial intelligence, statistical analysis, data mining, database, ITIL, Object Oriented Analysis, Oriented Programming objects, programming languages (Java, C / C + +, Delphi, PHP), and SQL Development JAVA WEB.