

EXTREME PROGRAMMING – AGILE METHOD USED IN PROJECT MANAGEMENT

Cruceru Anca

Romanian- American University, Faculty of Management- Marketing, 1B Expozitiei Blvd, Bucharest, cruceruanca@yahoo.com, 0723508894

Fotache Liviu Christian

Economical Studies Academy of Bucharest, 132 Brestei Street, Craiova, liviuchristian@yahoo.com, 0728297207

The main characteristic of the XXI-st century is change, which affects the development of organizations' activities. In this context, the organizations have to adapt their administration methods and systems to the new changes. An efficient administration method can also be project management, which, shall be held by the Agile Project – a project management system. From the multitude of Agile methods, the most important and most frequently used is Extreme Programming. Abiding by the principles and values that represent the basis of this method, the management of one project shall be more efficient in obtaining the desired results.

Keywords : Changing, Project Management, Agile Methods, Extreme Programming (XP)

JEL classification: D3, M15, O3

The organization of the XXI-st century carries out its activity in a dynamic environment, where change is a central element. In the second half of the XX-th century there appeared a “special” way of management, project management. Project management is an essential instrument for all organizations and can be successfully used in this environment in continuous change. The purpose of project management is to prevent as many dangers and problems as possible that could appear and to plan, organize and control the activities, so as the project to be finalized taking in consideration all the existing risks. Along with the appearance of the new changes, in the exterior environment of organizations and also in the internal environment of organizations, new discoveries in the project management are achieved. The most “spectacular” discoveries are in the informatics field, by creating new project management systems to ease the project management activities and to obtain the desired results – Primavera Project, **Agile Project Management** etc.

1. About Agile...

In the 90s, people became interested in developing software methodologies for the new business environment. As a result, the Agile methods were established, methods which showed the best practices resulted from experience of hi-tech developers and project managers. The purpose was to sustain the work of these successful developers.

In order to contribute at shaping the foundation principles of the Agile software development practices, and also to improve, diversify and increase the efficiency of this software, The Agile Alliance was created (www.agilealliance.org)

The Agile methodologies have in common values and principles.

The Agile Alliance created a manifesto with 4 main enunciations and 12 principles that sustain the manifesto. The manifest is the following:

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- individuals and interactions over processes and tools;
- working software over comprehensive documentation;
- customer collaboration over contract negotiation;
- responding to change over following a plan.”[6]

The 12 principles that sustain the manifesto are:

- even late in development are expected and welcomed changing requirements;
- the most important priority is to satisfy the customer needs through early and continuous delivery of valuable software;
- deliver working software varies frequently from a couple of weeks to a couple of months;
- the projects must be build with motivated individuals; you must support and trust them.
- the conversation face-to-face is the most efficient and effective method to give and receive information in a development team;
- daily, in a project, business people and developers must work together;
- to obtain progress you must create working software;
- the agility is enhanced by continuous attention to technical excellence and good design;
- the sustainable development is promoted by agile processes;
- the most essential element is simplicity -the art of maximizing the amount of work not done;
- at some established intervals, the team must reflect on how to become more effective and must adjust its behavior accordingly.
- from the self-organizing teams result the best architectures, requirements, and designs.

The Agile software development practices upgrade the correlation between the business value and the software development. The main values of the Agile development are severed towards productivity and efficiency. A company that adopts an Agile development methodology invests itself in front of the competition even before initiating the competition.

The major Agile methodologies are:

- Dynamic Systems Development Method (DSDM)
- Scrum
- Crystal Methods
- Lean Development (LD)
- Feature Driven Development (FDD)
- Extreme Programming (XP).

2. About Extreme Programming (XP)

The most important methodology is Extreme Programming, which was developed eight years ago, by Kent Beck, Ward Cunningham, and Ron Jeffries. The methodology delivers the software “your customer needs when it is needed”[8]. XP represents a discipline of software development, which has as based values: community, simplicity, feedback and courage. XP emphasizes team work, and also, authorize your developers to answer to changing customer requirements. Managers, customers, and developers (part of a team) pursue to deliver a quality software. Through XP we can experience the view of the cost of change and its emphasis on technical exquisite through refactoring and test-first development. XP tenders a system of dynamic practices. Actually, XP “works by bringing the whole team together in the presence of simple practices, with enough feedback to enable the team to check where they are and to tune the practices to their unique situation”.[2]

The principles that embody the values are:

- open, honest communication;
- quality work;
- rapid feedback at all levels;
- assume simplicity;
- embrace change;
- play to win;
- concrete experiments;
- small initial investment;
- incremental change;
- accepted responsibility;
- honest measurement;

- travel light;
- teach learning;
- local adaptation.

3. When and How can we use Extreme Programming?

Extreme Programming can be used to solve problems in areas that need change. Also, it has been set up to address the problems of project risk. XP has been set up for small groups of programmers. The requirements are an extended development team, testability and productivity. XP provides a set of daily practices that, used together with simple rules, have been substantiated to efficiently obtain high quality software.

These practices are:

Whole team- In an XP team: everyone is involved all the time, the team members communicate with one another by talking, everyone is kept fully informed, and everyone works together. The team includes a business representative (the “customer”) and some programmers, a coach who keeps the team on track, a manager who allocates resources and removes external impediments, and other specialists.

The Planning Game- We have to take into consideration two aspects:

- Release Planning: the customer articulates the required items, the programmers evaluate the difficulty, and together they establish a preliminary plan (initial order of development, set out what can be accomplished with the available people/money/time).
 - Iteration Planning: the required items are broken down into tasks and estimated in more detail, and allocated to programmers.
- Customer Tests-** these tests address two problems in software development – “how do the programmers know when they're finished?”, and “how does the team know that everything that was working last iteration is still working this iteration?”. For each problem, the XP customer defines one or more automated tests that will show that the feature is working.

Small Releases:

- XP teams try to enhance the system, by adding daily, small features or parts of a feature;
- XP teams liberate tested software that can be displayed to a production environment at the end of every iteration;
- XP teams release to the final end-users as frequently as possible (every iteration for some teams).

Simple Design- XP teams take into consideration the simplest thing that could possibly work, implement it, hoping that the changes that occur are easier to withstand. They try to keep things as simple as possible so as to be easy to understand and to eliminate quarrel.

Pair Programming- The production work in an XP team is achieved by two programmers sitting side-by-side, working on the same machine. Pair programming means converging on the task at hand and diffusing knowledge around the team, especially if you change the pairs regulated.

Test-Driven Development- These tests are designed, implemented and owned by the programmers and they are distinct from the customer tests. These tests are helpful by constraining programmers to center on the “what”, the interface itself, before the “how”, the implementation and to realize that there are tests for every face of the system.

Design Improvement- Is facilitated by simple design, test-driven development, and pair programming, all these reducing the cost of change (design evolution process called “refactoring”).

Continuous Integration- In XP teams it is important to understand that everyone's code works together. XP teams constraint integration as often as they can.

Collective Code Ownership- In an XP team anyone can change anything, anywhere, because XP projects are team activities. This will improve code quality because each piece of code is looked at by many persons.

Coding Standard- The code must be produced by the same person who has to be capable and competent for this purpose.

Sustainable Pace- The XP teams try every day practices that make software development more sustainable and predictable.

Metaphor- In an XP team, the same vocabulary and language has to be used, therefore most time they speak through metaphor- through the mention that the software system is the same as some other system and that they're already colloquial with. This metaphor is useful as it increases the communication compression and also, it can function as an architectural vision.

Other Useful Practices- XP is a discipline that evolves and there exist other practices that are based on the XP principles and that add value for most projects:

Stand-up Meetings- for a good timing, the XP teams have a brief, daily meeting, where the following questions are answered:

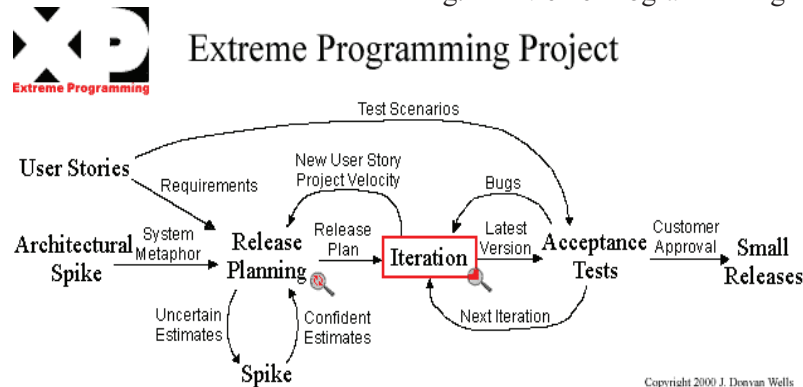
- “What have they done since the last meeting?”
- “What do they plan to do before the next meeting?”
- “Do they have any obstacles?”[5]

These meetings are called stand-up meetings because everyone stands up.

Retrospectives- At the end of every iteration, an XP team should have a retrospective, which analyzed what went wrong and what should be improved in the following iteration.

A better way to show how the practices interact, with who they interact, with the purpose to form a development methodology, is to represent an XP Map:

Fig. 1 Extreme Programming Map



Source: www.extremeprogramming.org

As a **conclusion**, we can say that XP is complex, but also agile, in order to adapt to the changes of the XXI-st century. XP's practices are helping in managing the augmentation of the team and its interface to the customer. Actually, in an XP process, the team has the possibility to grow, change and adapt to the opportunities and threats from outside background, that affect the business needs. Also, XP can transform the whole delivery organization, not only the augmentation of the team. We can state that XP is not “the best of the Agile methodologies, just the best known”. [5]

Bibliography:

1. E. Burdus, Gh. Caprărescu, A. Androniceanu, M. Miles, “ Managementul schimbării organizaționale”, Ediția a II-a, Ed. Economică, București
2. Ganesh Sambasivam, “Extreme Programming”, <http://www.agilealliance.org/system/article/file/1376/file.pdf>
3. Glenn B. Alleman, “Chapter X: Agile Project Management Methods for IT Projects”, 2002 (“The Story of Managing Projects: A Global, Cross- Disciplinary Collection of Perspectives”, Dr. E. G. Carayannis and Dr. Y. H. Kwak, editors, Greenwood Press / Quorum Books, 2002)

4. James K. McCollum, Cristian Silviu Bănaciu, 'Management de proiect. O abordare practica », Ed. Universitară, București, 2005
5. Steve Hayes (Khatovar Technology), Martin Andrews (Object Consulting), "An Introduction to Agile Methods", <http://www.khatovartech.com>, <http://www.objectconsulting.com.au>
6. "Studiu de caz. Dezvoltarea Agile. Abordarea Endava"- Copyright Endava, 2007
7. www.agilealliance.org
8. www.extremeprogramming.org