



**The 7th Balkan Conference on Operational
Research
“BACOR 05”
Constanta, May 2005, Romania**

**AGILE SOFTWARE PROJECT MANAGEMENT
ETHODOLOGIES – PROSPECTS OF THE GREEK
IT MARKET**

VAGELIS MONOCHRISTOU
DR. MARO VLACHOPOULOU
DR. VASSILIKI MANTHOU

University of Macedonia, Department of Applied Informatics, Thessaloniki, Greece

Abstract

Software Project Management and IT Project Management in general, deals with a variety of problems and difficulties resulting in time and budget overrun. During the last decades a number of methodologies have been introduced, starting from the waterfall model up to the today's more flexible, referred to as Agile Methods, aiming at the efficient and effective IT project management. This paper aims to describe shortly the evolutionary steps and the reasons beyond the appearance of these methods, as well to identify the preconditions and the limitations for their acceptance, intending to examine the possibilities for the adoption of the Agile Methods in the Greek IT market. Despite of the factors that influence the adoption of Agile methodologies in Greece, the extent of usage and familiarity with these methods (by presenting the results of a relevant survey conducted by the University of Macedonia), is also examined.

Keywords: *IT Project Management, Software Project Management Methodologies, Agile Methods*

1. INTRODUCTION

Software Project Management and IT Project Management in general, deals with a variety of problems and difficulties resulting in time and budget overrun. Standish Group conducted and finally published in 1995 the results of an extensive research amongst 365 US companies (http://www.pm2go.com/sample_research/chaos_1994_1.php) which compose a

representative sample (in terms of company size and business sector) depicting in an extremely interesting manner, the today's picture. More specifically, and according to the research:

- Only 9%-16% of IT projects are completed on time and on budget.
- Over 50% of IT projects cost two times their original estimates.
- Companies that conducted IT projects in 1995 experienced a nearly 30% cancellation before they ever get completed, with an estimated total cost of \$81 billion.
- Most of the completed IT projects include only half of the originally-proposed features and functions, whilst the total implementation time overruns 200 to 300% of the one primarily estimated.

According to another Standish Group research published in 2000, only 1 of 4 of USA's IT projects are completed on time and on budget, including all of the originally-proposed features and functions.

During the last years, a number of new, more flexible software development Methodologies, referred to as *Agile Methods* have been introduced in well-developed countries, (concerning the area of IT), and especially in the USA. According to the "Agilists", these methods can propose a solution to the above-mentioned problems, derived from the usage of traditional, bureaucratic methodologies.

This paper aims to briefly describe the evolutionary steps and the reasons beyond the appearance of the agile methods, as well as to identify the preconditions and the limitations for their acceptance. Finally, numerous research papers and publications are being overhaul intending to examine the possibilities for the adoption of these methods in the Greek market.

2. EVOLUTIONARY STEPS

In the early years of IT, software development was based on the absolute absence of a formal process. The role of programmer was dominant, and the final result was mainly reliant on the programmer's capabilities. The final user's contribution was minimal concerning only the general description of the problem that the software would solve. Project management contained only the phases of analysis and coding [14]. As IT projects became more complex, the necessity for methodologies that would give directions and aid their management was revealed. Table 1, depicts the basic methodologies according to their date of appearance.

Period	Date of Appearance	Name
1 st Period	Early years of IT	Absolute absence of a formal process
2 nd Period	Beginning of '70s	Waterfall Model [21]
3 rd Period	Middle of '80s	<ul style="list-style-type: none"> ▪ Rapid Prototyping Model [7] ▪ Spiral Model [2]
4 th Period	Beginning of '90s	<ul style="list-style-type: none"> ▪ Incremental model [28] ▪ Evolutionary lifecycle model [10] ▪ R.A.D – Rapid Application Development Method [15] – <i>it is an improvement of the previous one</i>
5 th Period	End of '90s	<ul style="list-style-type: none"> ▪ Extreme Programming (XP) [1] – <i>its appearance considered as</i>

	<p style="text-align: center;"><i>the starting point of the Agile Methods</i></p> <ul style="list-style-type: none"> ▪ SCRUM [23] ▪ Crystal Family [4] ▪ Feature Driven Development [17] ▪ Dynamic Systems Development Method [26] ▪ Adaptive Software Development [11] ▪ The Rational Unified Process (RUP) [13] – <i>There is a confliction about whether RUP can be considered as Agile or not.</i> ▪ Open Source Software Development [16]
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Table 2.1 – Evolutionary Steps in the appearance of IT project management methodologies

In the beginning of 1970s, the *waterfall model* was introduced by W.W. Royce (1970). The name of the model - which has to be noted that was not given by the author - indicated the way that the certain model faced an IT project. The project was implemented through a series of phases that were performed sequentially with milestones, documents and reviews at the end of each phase. As soon as one phase was ended, it was impossible to return to a previous one. Emphasis was given on preparing a detailed technical specification plan, up-front in development. These two preconditions constituted the strongest point of criticism against this model, especially in the current days where there's great uncertainty surrounding both the requirements of the client and the available technological solutions.

As a result the need for developing models that would have the ability to respond towards the strongly appeared uncertainty arised. Thus, from the middle of 1980s, several methods started to appear (3rd Period – Table 2.1), that tried to provide alternative ways for overcoming the limitations of the *waterfall* model. Typical examples of such methods were the *rapid-prototyping model* [7] and the *Spiral model* [2]. Both models suggested the use of the software prototype (in fact a draft, non-functional “plan” of the final product, attempting to replace the written requirements of the *waterfall* model), which was shown to the customer at the early stages of development, aiming at “collecting” the necessary feedback, that would help with the designing of the system. On the contrary, the spiral model differentiated by emphasizing on project's risk assessment. Conclusively, two key elements were the main characteristics of the above models. Prototypes developed were not functional parts of the final system but only used for demonstrational purposes, whereas during the designing and implementation phase of the system, the *waterfall* model was used. In fact, even though these models composed an improved, “evolutionary” version of the waterfall model, they significantly depended on it, including as well the majority of its disadvantages.

On later stages (4th Period – Table 2.1), several methods emerged that tried to move one step further. The main characteristic of these methods was the iterative implementation of functional sub-systems of the final system throughout the development of the system. One of the fundamental objectives of these methods was the reduction of failure risks associated with the system, by gradually delivering parts of the designed functionality to the customer. Decriers of the above methods supported that although customers actively participated throughout the implementation of the system and their posed opinions were taken into account, by the end of one sub-system development, it was impossible to revise it. Furthermore, any plans for significant changes to any of the future - designed sub-systems, based on customer's remarks, were not feasible.

In this point, it is worth mentioning the *CMM - Capability Maturity Model for Software* [18], [19]. Although this model did not represent a methodology, it became broadly acceptable in the IT industry, due to its detailed description of the principles and the practises that companies had to follow in order to improve its software development processes. The model was organized in 5 levels, each of them determining the maturity extent of the company's capability towards software development, while providing measurement and evaluation criteria. Companies that know the implementation criteria of the next level in advance are directed to constant improvement.

Lately, and especially during the last 5 years, new and more flexible methods appeared. According to their supporters, Agile Software Development Methods can improve even more the existing ones, based on a quite different approach that includes: iterative development (small versions of the software with rapid development cycles), customer collaboration (customer in fact participates in all phases of software implementation), adaptability (last minute changes is allowed). Despite of their differences, Agile methodologies (5th Stage – Table 2.1) present a lot of similarities especially in the way that they deal with the management of an IT project. Their basic principles are described in Agile Software Development Manifesto (www.agilemanifesto.org), which was published during 2001 by a team of IT experts:

- People and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over punctual appliance of a plan

More specifically, agile methodologies consider people (members of the development team) as the most important element. The development team's main objective is to iteratively deliver operational software (on a weekly basis, monthly basis, depending on the size of the project and the method used etc.). Developers should be exhorted to produce simple software code in order to reduce documentation. On the other hand, although the requirement for contracts is recognized, it is proposed to be considered as the mean of creating a confidence relationship with customer. Finally, development team (which includes both developers and the customer) must be allowed to modify the initial designing when necessary. In fact, agile methods propose the use of managerial and technical processes that continuously adjust to changes derived from experiences gained during development, changes in software requirements, and changes in the development environment.

3. AGILE METHODS PRINCIPLES – CONDITIONS OF ADOPTING

The research team that wrote the *Agile Software Development Manifesto* additionally described a number of principles that had to be taken into account so as to be considered that an agile “environment” was adopted [5], [6], [12], [27]:

- Our highest priority is to satisfy the customer through early and constant delivery of operational software modules
- Customers and development team must work together on a daily basis throughout the project

- Any changes on the initially planned requirements should be accepted even in late stages of implementation
- Frequently hand outs of operational software modules should be indispensable
- The operational degree of the software hand outs is the primary measure of progress
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done
- The best architectures, requirements, and designs emerge from self-organizing teams
- The most efficient and effective method of conveying information to and within a development team is through face-to-face conversations
- Continuous attention to technical excellence enhances agility
- Simplicity is pivotal.
- Project development teams should evaluate their effectiveness at regular intervals and adjust their behaviour accordingly.

It is more than apparent, that the above described principles define the prerequisites that have to be efficiently addressed in order for an Agile Methodology to be adopted. Specifically:

- Customer should be located in proximity to the development team, always available when needed.
- Members of the development team must be located in the vicinity of each other.
- Documentation does not constitute panacea.
- User requirements will emerge during the development phase of the software.
- Development team must have the necessary expertise to manage and adopt all the changes which will arise.
- Evaluation of software artefacts can be restricted to frequent informal reviews and tests.
- Reusability of software won't be the primary goal.
- Cost of changes should not dramatically be increased over time.
- Software will be developed in increments.
- There is no need to get prepared for changes towards the initial designing, since any change can be effectively handled by modifying the source code.

4. LIMITATIONS ON ADOPTING AGILE METHODOLOGIES

The above prerequisites respectively define the limitations on adopting the *Agile methodologies* [27], [4]:

NEED FOR SPECIALISED HUMAN RESOURCES

During the project, customer expert or experts should be available at all times and ready to devote 100% of their time; unfortunately, and mainly in the Greek reality, this significantly increases the cost of the project as well as creating acceptance difficulties from the majority of administration boards. Additionally, it generates an extra need for the presence of managers supporting the customer's side by contributing with concrete proposal and solutions in potential organizational obstacles.

DIRECT COMMUNICATION

Companies operate using decentralized structures and with distributed development teams, (in various places or countries) face communication obstacles. The need for continuous and face-to-face communication between the members of the development team is sometimes difficult leading to the need for documentation reversing thereby one of the main conditions of Agile methodologies. For Greek IT companies, it appears that this cannot be considered as an important problem, since the majority of IT companies do not fall to the above category, because of their limited size and their structure. Nevertheless, there are some difficulties, since the development team has to be located in the same place. Significant problems arise also in the case of subcontracting, which is a very common situation in the Greek IT market. As a conclusion, it is almost impossible to consider that an *Agile* method is followed when part of the project has been assigned to a subcontractor.

QUALITY ASSURANCE

Questions arise related to the quality assurance, as it is described from international standards (ISO, TickIt, etc.), and followed by the majority of companies worldwide. Does the strict conformance with company procedures and the complete documentation ensure quality? Aren't there cases where the delivered software was consistent to all the requirements of a Quality System, but did not satisfy customer's requirements? Isn't it likely that quality finally is the extent of customer's real requirements satisfaction, as Agile Methodologies supporters claim? [8]. The answer is not easy and cannot be given with a single paragraph analysis. The fact is that one of the main principles of adopting *Agile* methodologies comes in sharp contrast with the modern aspects for documentation, required by quality systems. In addition, it should be taken into account that companies have invested a significant amount of time and money in the development and maintenance of such systems, thus these investments cannot be ignored.

FLEXIBLE USER'S REQUIREMENTS

The aspect that user's requirements will emerge during the development stages comes in contrary to the modern practice, where user's requirements represent in fact the subject of signed contracts where concrete and pre-agreed time schedules and budgets are appointed. These contracts are usually the tool of solving any disputes or misunderstandings between the parties. A more flexible contract, as "Agilists" propose frightens the upper management.

KNOWLEDGE AND EXPERTISE OF PROJECT TEAM

Difficulties related to the project team members arise, since Agile Methods require the presence of experts with significant experience not only in technical background but also in the managerial one. Additionally, project team must include expert (s) with special knowledge on the particular methods.

COST ISSUES

The "Agilists" belief that reusing part or whole of the final product (software) is not the primary goal comes in sharp contrary with the modern business theories of cost

compression through the automation of production. It is actually true that, especially in IT industry, the development of software is extremely costly and sometimes results in losses for companies. Modern management requires software that can be reused or sold many times, covering likewise the initial costs or even the losses.

CONTROL MECHANISMS

A lot of difficulties arise when the project team is large. Effective control, communication and collaboration require documentation and control mechanisms.

SECURITY ISSUES

Big obstacles are confronted when the project is crucial and security is the primary factor (e.g software applications for the army).

5. INTERNATIONAL REALITY

Despite the several difficulties and limitations, along with the scepticism that exists concerning their advantages, Agile methods are broadly being used by companies that develop software. A few data from research supporting all those mentioned are distinctively cited:

- The *Software Development Times* Magazine (July 2003 Issue) published the research results of an *Evans Data Corporation* [9] research, according to which a rate of 16% of North America companies totally use Agile methodologies in their projects.
- A Cutter Consortium research [3] conducted among 200 Managing Directors / IT Managers, that constitutes a respective sample from the point of the geographical allocation of the companies' type and size, recorded that during 2001, 21% of the participants used Agile methodologies to more than 50% of their projects. Additionally, in the year 2002, 34%, and in 2003 almost *half of the participants* expected that more than 50% of their projects would be conducted using Agile methodologies.
- The research of Giga Information Group [25] in 2002 anticipated that in a period of the following 18 months the 2/3 of IT companies in USA would mostly use Agile methods in their projects.

Furthermore, results of other researches ([22], [20], [24]), that are of great interest, concerning those companies that already use Agile methodologies, are presented below:

- Companies that use Agile Methods are usually "young" in years of operation (<10 years) with a small number of employees (< 100). It is worth mentioning though that a significant number of bigger companies gradually adopt (usually in pilot projects) Agile methodologies.
- The "trigger" for having applied Agile methods is the unsatisfactory rate of acceptable products / projects, and the timetable and budget overruns.
- Most of the project teams are small in size. Teams with 5-10 members comprise the majority, while teams with less of 5 members follow. There are a small number of companies that are using such methodologies for teams with more than 40 members.

- In all projects there is at least one (1) full-time on-site customer participating in all phases of the project.
- Applications developed are mainly web and client - server applications.
- Java is the dominant programming language used.

Companies using Agile methods stated that:

- Productivity was increased in 93% of them.
- Quality of the final product was increased in 88% of them.
- Total business satisfaction from their use was high in 83% of them.
- Total cost of production was decreased in 49% of them.

It is notable that almost all companies using Agile methods keenly state that they will continue to use them. Moreover, companies participating in these researches, regardless of using such methodologies or not, strongly believe that during the next years the majority of their projects will be implemented using Agile Methods. These answers have more value, if we take into account that the questionnaires have been filled by IT experts, who surely have the sense of IT market, despite of their final acceptance of the Agile Methods. Generally, the projects that used Agile methodologies were rated as successful by the participating companies, having better schedule, better cost performance and better quality [20], [24]). Finally, according to the same data, the main “barrier” was management opposition, whereas the risks related to Agile Methods categorised according to their percentage were: problems with on-site customer, opposition against the method, lack of special tools, not enough training on the method, unskilled developers [22].

6. THE GREEK REALITY - PROSPECTS OF THE GREEK MARKET

It can not be said that Greek IT market follows the international trend on using Agile methods. In fact, Greece is considerably falling short. According to an on-line survey conducted by the University of Macedonia, in which 174 IT Experts participated:

- The rate of satisfactory knowledge (answer “thorough knowledge” and “satisfactory knowledge”) on the most important among these methodologies hardly exceeds 13% in average
- Concerning the four fundamental values as they are described in the Agile Manifesto, it is particularly interesting the fact that in average half of the participants (rate of 56.6%) agree (answered “totally agree” and “fairly agree”).
- Concerning the basic principles of these methodologies, in 9 out of 10 of participants answers, the agreement rate (answered ‘totally agree” and “fairly agree”), fluctuated between 60% and 90%.
- In the question: *“If you could choose the software development management method to use, would you choose one that fulfils the above principles?”* only 1.7% of the participants answered that would not use any of the Agile methodologies.
- As far as the adoption possibilities of these methodologies by Greek experts is concerned, the participants do not seem to share the same optimism with the Americans, as only 30% of them consider this as possible in near future.

Even though the results confirmed our initial estimation for the low degree of popularity of the Agile methodologies to Greek IT experts, nevertheless we were positively surprised, since it is proved that although in theory the particular

methodologies are described as “revolutionary” their principles have great response and degree of acceptance.

On the other hand, Greek IT business environment can be considered as quite interesting. Factors that influence the adoption of Agile methodologies in Greece are presented in Table 6.1.

Factors that support the adoption of Agile methodologies	Factors that create difficulties on the adoption of Agile methodologies
The majority of Greek IT companies are small in size, whereas project teams with more than 20 members are not typical.	Subcontracting in IT Projects is a frequent phenomenon.
Numerous small sized companies which are more flexible in the necessary managerial and operational changes.	Lack of experienced managers that will undertake the responsibility for the necessary managerial and operational changes.
The majority of implementing IT projects are not large or critical.	A significant part of implementing IT projects concerns the public sector, something that creates difficulties on accepting such methodologies.
Significant number of experienced IT specialists.	Lack of expertise on the subject.
Significant number of educational institutions that can undertake the necessary academic research	Lack of financing from the companies’ side and lack of connection between research and Greek IT market.

Table 6.1 Factors that influence the adoption of Agile methodologies in Greece

7. CONCLUSIONS

Despite the many positive and innovative aspects of the Agile Methods, project managers globally are still trusting and using traditional methodologies and primary the waterfall model, arguing about the effectiveness of using Agile Methods. The main reason behind this scepticism is the fact that in Agile “way” of software development there are not milestones which can be used as a comparison mean between the estimated and the actual. In addition, the principle that Agile Methods propose require a change on the “companies’ culture”, where special attention should be given to the customer and project team. Finally, academic research on the subject is minimal.

In practice of course, traditional methodologies are not used in their whole extent, mainly because of their complexity and the bureaucracy that require. The need for continuous control and documentation, the strict contracts and time schedules, limit the flexibility that IT projects must have during their development.

The important point that someone has to understand in relation to the philosophy of Agile Methods is that Agilists do not consider procedures, tools, detailed documentation and contracts as worthless. They just believe that the Agile Manifesto Principles [12] are more important and contribute more to the final product, and customer’s satisfaction.

It should be also mentioned that Agile Methods can not be effectively used on every type of project. Their limitations show the types of project for which they are more appropriate. Projects where security is the primary factor, or projects that are large and very complex with large project teams and involvement of many companies (subcontractors), create an environment inappropriate for these methodologies. By deciding which agile method covers their needs, and using it primarily in small and less important projects, a company can evaluate the effectiveness of the method. This way the

company will have the necessary data to decide whether it should adopt the method in whole or not. In any case, the honest participation and commitment of all involved is necessary and imperative.

Although Agile methodologies do not constitute by definition a panacea, we strongly believe that they can considerably help companies, especially the Greek ones, in specific types of projects, to become more competitive, produce better results on time with high quality, and in competitive cost. In a following stage, a number of more specific researches investigate Greek IT business environment and opportunities, related to the acceptance of Agile Methods.

BIBLIOGRAPHY

- [1] Beck, K. (1999) *Extreme programming explained: Embrace change*. Boston, Addison-Wesley;
- [2] Bohem, B. (1998) "A Spiral Model of Software Development and Enhancement". *IEEE Computer*, vol.21, 61-72;
- [3] Charette R. (2001), "The Decision is in: Agile versus Heavy Methodologies", Cutter Consortium, e-Project Management Advisory Service, Executive Update, vol. 2 No.19;
- [4] Cockburn, A. (2002b) "Learning from Agile Software Development – Part One", *Crosstalk*, Oct.2002, vol.15 (10), 10–14;
- [5] Cockburn, A. (2002c) "Learning from Agile Software Development – Part Two", *Crosstalk*, Nov.2002, vol.15 (11), 9–12;
- [6] Connell, J.L., and Shafer, L. (1989) *Structured Rapid Prototyping: An Evolutionary Approach to Software Development*. Yourdon Press, Englewood Cliffs, NJ;
- [7] Crispin, L. (2001) "Is quality Negotiable?" available at www.xpuniverse.com/2001/xpuPapers.htm (accessed on June 2003);
- [8] Evans Data Corporation (2003) "North American Development Survey, vol. 1, - Process Methodology Primarily Used" available at www.evansdata.com/n2/surveys/northamerican/2003/na_2003_1_xmp3.shtml (accessed on July 2003);
- [9] Gilb, T (1988). *Principles of Software Engineering Management* Addison-Wesley, NY;
- [10] Highsmith, J. A. (2000) *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. New York, NY, Dorset House Publishing;
- [11] Highsmith, J. A. (2002) *Agile Software Development Ecosystems*, Boston, Addison-Wesley;
- [12] Kruchten, P. (1996) A Rational Development Process. *Crosstalk* 9(7), 11–16;
- [13] McCormack, A., et al. (2002) "Flexible Models of Software Development: Must we trade off efficiency & quality", Submitted to *IEEE Software*, Jan 2002;
- [14] Martin, J. (1991) *Rapid Application Development*. Macmillan, New York;
- [15] O'Reilly, T. (1999) "Lessons from Open Source Software Development". *Communications of the ACM*, vol. 42, no. 4), 32–37;

- [16] Palmer, S. R. and Felsing, J. M. (2002) *A Practical Guide to Feature-Driven Development*. Upper Saddle River, NJ, Prentice-Hall;
- [17] Paulk, M.C., et al. (1993a) *Capability Maturity Model for Software, Version 1.1* (CMU/SEI-93-TR-24, ADA 263403), Pittsburgh, PA: Software Engineering Institute, February 1993;
- [18] Paulk, M.C., et al. (1993b) *Key Practices of the Capability Maturity Model, Version 1.1* (CMU/SEI-93-TR-25, ADA 263432). Pittsburgh, PA: Software Engineering Institute, February 1993;
- [19] Reifer, D. J. (2002) "How Good Are Agile Methods?" *IEEE Software*, 14-16, July/August 2002;
- [20] Royce, W.W. (1970) "Managing the Development of Large Software Systems". *Proceedings, IEEE Wescon*, August 1970;
- [21] Rumpe, B., Schröder, A. (2001) "Quantitative Survey on Extreme Programming Projects". *Technical Report*. Munich University of Technology;
- [22] Schwaber, K. and Beedle, M. (2002) *Agile Software Development with Scrum*. Upper Saddle River, NJ, Prentice-Hall;
- [23] Shine Technologies Pty Ltd (2003) "Agile Methodologies Survey Results" *available at www.shinetechnologies.com/agile_survey_results.jsp* (accessed on June 2003);
- [24] Stapleton, J. (1997) *Dynamic systems development method – The method in practice*. Addison Wesley;
- [25] Sliwa, C. (2002), "Users Warm Up to Agile Programming", *ComputerWorld*, 36(12): 8, 2002;
- [26] Turk, D., France, R., Rumpe, B. (2002) "Agile Software Processes: Principles, Assumptions and Limitations." *Technical Report*. Colorado State University;
- [27] Wong, C. (1984) "A Successful Software Development". *IEEE Transactions on Software Engineering*, vol. SE-10, no.3, 717-727.