Analyzing Agile Development – from Waterfall Style to Scrumban

Marian STOICA, Bogdan GHILIC-MICU, Marinela MIRCEA, Cristian USCATU
Bucharest University of Economics, Romania
marians@ase.ro, ghilic@ase.ro, mmircea@ase.ro, cristiu@ase.ro

The spectacular evolution of information and communications technology in the last two decades and the growing obvious chase of utopian maximization of profit by economic organizations have made an impact on various fields, including that of methodological instruments regarding the lifecycle of software development. We will speak about the necessity of organizational ability to adapt to continuously changing market conditions, which leads undoubtedly to acquiring functional flexibility and, in the end, of business agility. Also, the agility of a business (not only in software development) is supported by the manifestation of agility on all three architectural levels: business, informational, technological. In this study we aim to identify the elements that impact on agility in developing software products, in a gradual approach, from the traditional waterfall model towards approaches like Scrumban. Additionally, we will understand the social dimension of using agile methodologies in software development projects and the main barriers in adopting such methodologies.
***Keywords**: Agile Development, Software Development Life Cycle, Project Management Tools, Incremental Model, Waterfall, Scrum, Kanban, Scrumban.*

1 Introduction

Ever since 1970, when Winston W. Royce created the first forma description of the waterfall model, software development has known a multitude of various approaches. Most of them were built upon the model presented by Royce, bringing new specific characteristics. Significant development for the proposed model are described in [1] from the perspective of main characteristics, software development stages, advantages, drawbacks and utilization recommendations. Beside this description, which may be considered a guide by project managers, another important element is the incremental model for software development, the foundation of the philosophy with the same name. Combining specific elements of the waterfall model with stages and attributes of the prototype model, the incremental development philosophy is the base for what is known today as agile software development (Figure 1).

From a time perspective, agile development started in February 2001, when representatives of 17 software development organization met in Utah, USA, to discuss new and lightweight methods and methodologies to develop projects. The meeting yielded the famous *Manifesto for Agile Software Development*, which includes the 12 development principles. [2] According to the manifest, any project that observes these development principles falls under the category of agile projects. In other words, the representatives of the 17 organizations have identified easier ways to develop software projects and to help others do the same, including these aspects in the 12 principles of Agile Manifesto. Analysts claim [3] that the 12 principles of the Agile Manifesto constitute a "dramatic contrast" from the traditional guide and de facto standard of project management PMBOK – Project Manager's Body of Knowledge.

In a general way, we may say that agile development is a reaction to developers' needs when faced to ever more varied requests from clients. The economic environment is growing in flexibility, providing varied business opportunities, which requires organizations to have the ability to adapt and capitalize on these opportunities. This is only possible as long as the organizations can use agile business architectures, built upon flexible solutions. The agile software development concentrates on the client (beneficiary). Additionally, contrary to

DOI: 10.12948/issn14531305/20.4.2016.01

traditional approach, agile approach does not focus on creating documentation for the product. In turn, this can be a major drawback of the agile paradigm.

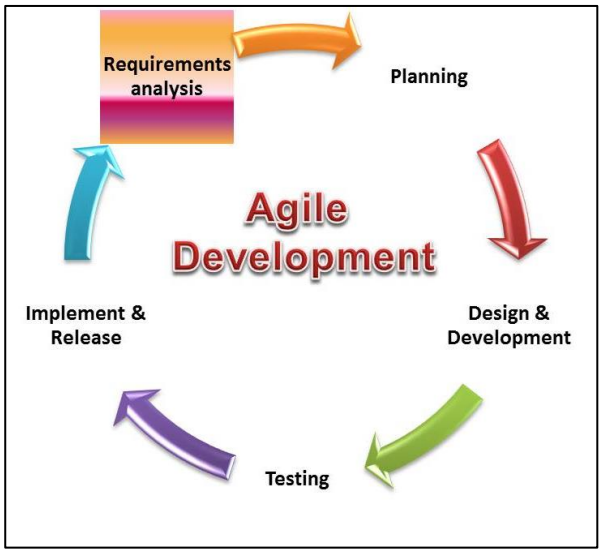

Fig. 1. The Agile Development Cycle

On the other hand, an organization or business agility is a key element in gaining strategic advantages. Therefore, the existence of an agile architecture on organization level may lead to decrease in development time for new processes and increase in flexibility of existing processes. Additionally, the impact of business agility may be measured by the decrease in response time to clients' requests, increase of new client numbers, reduction of costs in adaptation to new economic scenarios and, finally, increase in organization income.

The increasingly visible orientation of software developers towards agile development philosophy does not mean the traditional approach will be completely abandoned. There still are software projects of high complexity, with significant usage targets (at least regional or national level), which do not fit in any way to agile approaches (for example the project regarding national health cards in Romania). Additionally, considering the latest paradigms used on the software development market, agile philosophy is frequently associated with Cloud Computing, in order to highlight once more the flexible character of this approach on economic organization level. [4] Currently, all circumstances allow us to talk about Agile Development as a phenomenon with rapid growth from the microeconomic climate that initiated it towards the macroeconomic level. This claim is supported by the precedent of project oriented development that changed the perceptions or modern organization management. [5] Today, management through projects is a good work practice that supports, with good results, the main functions of the enterprise elaborated by classic management scholars.

2 Agile Development – from Philosophy to Methodology

When talking about the life cycle of software development (or system development in other acceptations) – SDLC, we certainly mean an environment that describes the activities performed during each stage of the development process (Figure 2). Additionally, we mean a detailed plan that describes the way development, maintenance and replacement of the software will take place (as software development process), while observing specific international standards (ISO-IEC 12207).

When detailing the specific activities performed during SDLC, we notice that they require the use of diverse processes and methodologies, selected according to the project goals. In other words, any software development activity calls on the experience of the development team members that will select the best solution matching the project goals. This again demonstrates that there is no single best method for an optimal solution.

DOI: 10.12948/issn14531305/20.4.2016.01

Fig. 2. The Software (System) Development Life Cycle

The literature [3], [6] presents agile development from both philosophical and methodological perspective. Until now it is still unclear where philosophy ends and methodology applies or when methodology is applied in the spirit of philosophy. Only certainty is that when talking about SDLC, the most important development models, in chronological order, are:

a) Waterfall model;
b) V model (verification and validation);
c) Incremental model;
d) RAD model RAD (Rapid Application Development);
e) Agile model;
f) Iterative model;
g) Spiral model.

The incremental model is unanimously considered the precursor of any current agile methodology. Still, in the fifth position in previous list, there is a so called agile model (Figure 3), which is in fact a superior version of the incremental model, with three main attributes:

- Software is developed in fact, incremental cycles;
- Each version is tested to ensure product quality;
- It is used for applications that must be completed in a critical time frame.

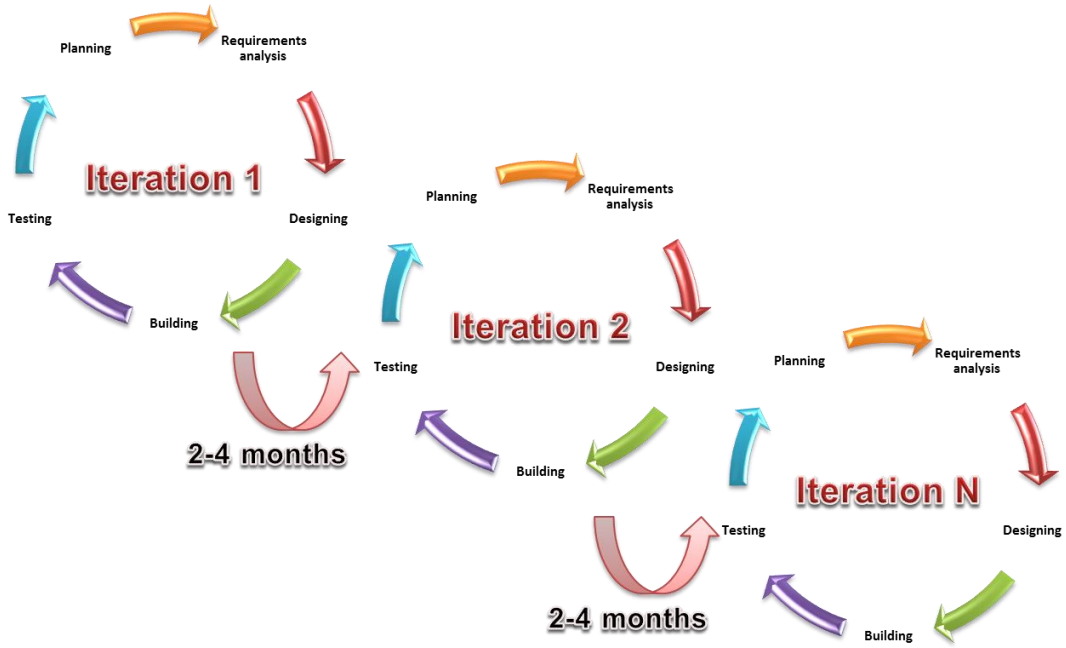

Fig. 3. The Agile Model Diagram

The existence of agile model in SDLC description and its application in software development lead to the idea of the existence of an agile methodology. In order to clear this terminology controversy, researchers suggest approaching the problem on three levels: philosophies, methodologies and tools (Figure 4). [6] On the other hand, in the experience of

DOI: 10.12948/issn14531305/20.4.2016.01

practitioners (including the authors), this three level pyramid transforms in a four level pyramid, specific methods and techniques replacing the tools, while the base of the pyramid is represented by commonly used applications (Figure 5).

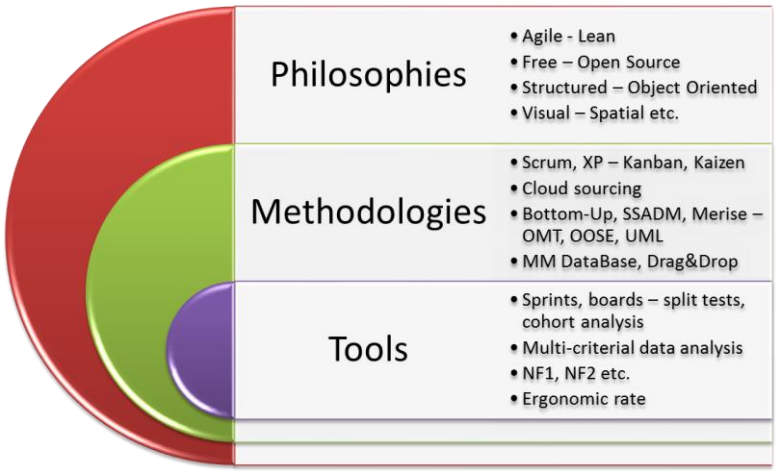

Fig. 4. Three levels pyramid of software development

From this perspective, traditional software development methodologies are today reviewed through philosophies like Agile vs. Lean, Free vs. Open Source, Structured vs. Object Oriented, Visual vs. Spatial etc., knowing significant transformations on upper levels for practical application. They are finalized into development instruments / methodologies like SCRUM (58% of the market), SCRUM/XP Hybrid, Scrumban, Kanban, DSDM, XP etc. Each of these "modern" methodologies, beyond its purely technical characteristics, is built and managed through specific instruments in the generic category of Project Management (MS Project, Sprintometer, VersionOne, Google Docs, Bugzila, IBM Rational etc.).

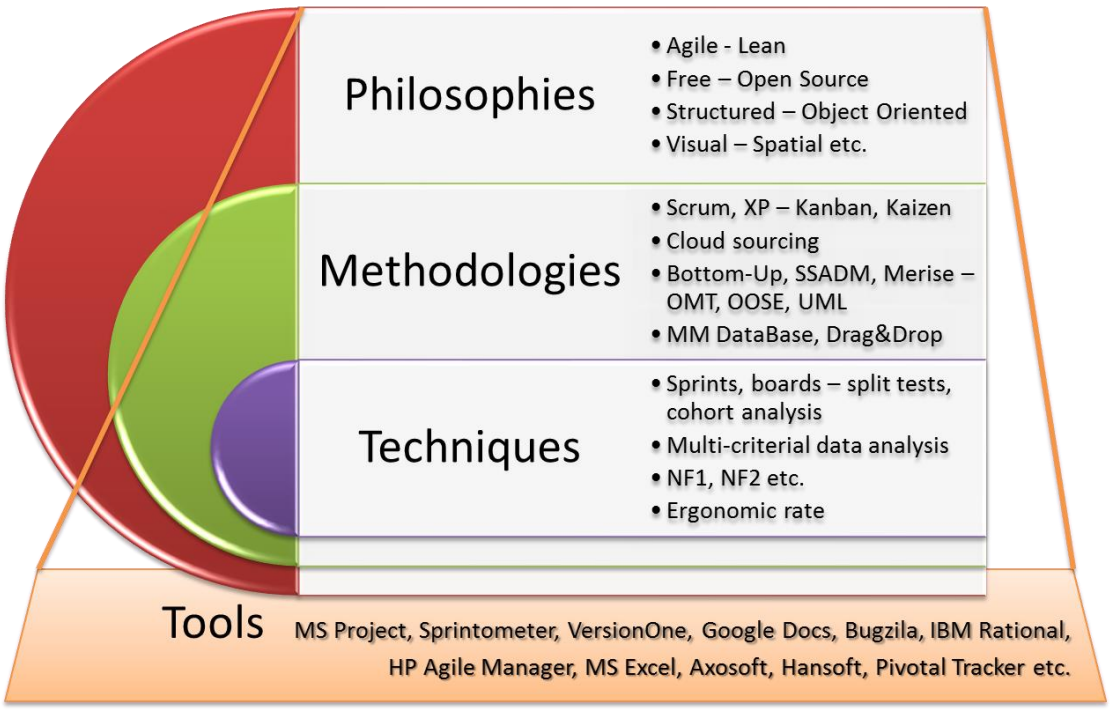

Fig. 5. Four levels pyramid of software development

DOI: 10.12948/issn14531305/20.4.2016.01

Beyond the philosophy vs. methodology discussion, Agile consolidates its software development market position, even if, as one of the SCRUM methodology parents, Jeff Sutherland pointed out in [7], Agile is only a subset of the LEAN philosophy, which, in turn, is a subset of Systems Thinking practices and principles. Invoking the systemic character in agile development provides the characteristic flexibility of this methodology. Traversing the development stages with their inter-connections as well as connections with the environment (represented by the client) constitutes the main advantage over the traditional sequential methodologies or models.

3 From Waterfall to Scrumban, through Scrum and Kanban

After seeing the main SDLC models published over time, in order to fully achieve the goals of this study, we must also review the most popular methodologies used to implement the agile philosophy [6]: Extreme Programming (XP), Feature-Driven Development (FDD), Adaptive System Development (ASD), Dynamic Systems Development Method (DSDM), Lean Software Development (LSD), Kanban, Crystal Clear, Scrum. Also, there are a series of practices and frameworks used by software developers to provide agility to their products. Some of them are: Agile Modelling (AM), Rational Unified Process (RUP), Test-Driven Development (TDD), Scaled Agile Framework (SAF), Rapid Application Development (RAD), Empirical Control Method (ECM).

In spite of this density of agile development methodologies, waterfall (as sequential, linear development model) remains the most popular SDLC. This happens because its main advantages (ease of use, detailed definition of requirements and importance of documentation) easily overcome the drawbacks (lack of flexibility and partial deliverables). Agile and Lean philosophies were built upon the bases of traditional waterfall philosophy and are represented by Scrum and Kanban methodologies. For a better understanding, we propose a comparative approach of the two new philosophies, through seven differentiating/common aspects (Table 1).

Table 1. Agile vs. Lean philosophy (adaptation from [6])

	Agile	Lean
Purpose	Rapid execution of tasks, easy adaptation to changes	Intelligent development by elimination of elements that are useless for the client
Finality	Flexibility of development process	Sustainability of development process
How it started	Initially designed for software development, then extended to marketing and currently applied in various other domains	Initially designed for traditional factory manufacturing processes, then extended to all industries
Action way	Product backlog – sprint backlog – iteration (sprints) – potentially shippable result	Build-measure-learn
Demonstration of progress	Definition of "done"	Validated learning
Specific methodologies	Scrum, XP, FDD, DSDM, Crystal Methods etc.	Kanban, Kaizen etc.
Specific instruments	Sprints, boards, Scrum Master, acceptance tests, user story mapping etc.	Hypotheses, split tests, customer interviews, funnel and cohort analysis, Customer Success Manager etc.

DOI: 10.12948/issn14531305/20.4.2016.01

As mentioned above and seen in Table 1, Scrum and Kanban are the representative methodologies for the two discussed philosophies. We already know Scrum as a "popular" methodology, based on the 12 principles of the Agile Manifesto of 2001 (for details, see [2], pages 73-74). In this line, the latest report regarding agile development published by VersionOne in 2016 [8] indicates a market share of 58% for Scrum and only 5% for Kanban (while Scrumban approaches 7%!).

Implementation of Scrum in an organization involves establishing small sized teams tasked with simple jobs for short terms, up to two weeks. All aspects of Scrum implementation are highlighted by Jeff Sutherland (a guru of Scrum) in [9]. It is worth noting that Scrum is not a process or a technique for building products, software or of another nature. Scrum is rather a framework that involves and employs various processes and techniques to build something. Built on the empirical process control, Scrum is based on three main pillars: transparency, inspection/verification and adaptation.

On the other hand, when speaking of Kanban, we must say from beginning that this methodology has a more pregnant technical, practical, industrial character. It was inspired by Toyota Production System (TPS) and Lean Manufacturing (or Lean Production) – concepts developed by the Japanese industrial philosophy "Muda" (which means avoiding waste by removing anything not useful – elimination of waste). Kanban is a less structured methodology than Scrum; its principles can be applied to any ongoing process (even a Scrum process). It is a visual framework (Japanese meaning for "visual sign") used to implement agile projects, showing what, when and how much to manufacture.

In a more practical approach, Scrum is useful when organization activity requires a "reboot" and Kanban is recommended when the ongoing activity must be improved. In the Kanban approach, there are four fundamental principles upon which the methodology is built [6]:

a) Improving communication and collaboration through monitoring ongoing tasks;
b) Limitation of ongoing activities in order to avoid propagation of initiated and unfinished tasks;
c) Measuring and optimization of work flows, anticipation of future problems;
d) Pursuing continuous improvement as result of activities.

In 2009, Henrik Kniberg [10] performed a comparative analysis of the two methodologies, highlighting common aspects and main differences (Table 2).

Table 2. Scrum vs. Kanban - differences (adaptation from [10])

Scrum	Kanban
Team involved in a specific iteration	Optional involvement
Uses speed (velocity) as a measure for improving processes	Uses deadline / lead time as a measure for improving processes
Prescribed estimations	Optional estimations
One sprint backlog belongs to a team	Kanban-board may be shared
Involves using at least three roles (Product Owner / Scrum Master / Scrum Team)	Does not use roles
The Scrum-board is reset between sprints	The Kanban-board does not change, it is persistent
For each sprint, priorities are established on the sprint backlog	Establishing priorities is optional

DOI: 10.12948/issn14531305/20.4.2016.01

As mentioned in title, in the last two decades the software project development approaches have evolved from waterfall model to agile ones, like Scrum and Kanban. The similarities between the two agile methodologies on one hand (they use transparency to improve processes – see Kaizen model, oriented on anticipated delivery of the product, based on self-organizing teams, breaking tasks in subtasks etc.) and differentiating aspects on the other hand have been lately just as many motives to try a combined approach of software development processes. This is how Scrumban is born – one of the newest hybrid approaches in software development that calls on technical and methodological "compromises" between the parent methodologies (Figure 6).

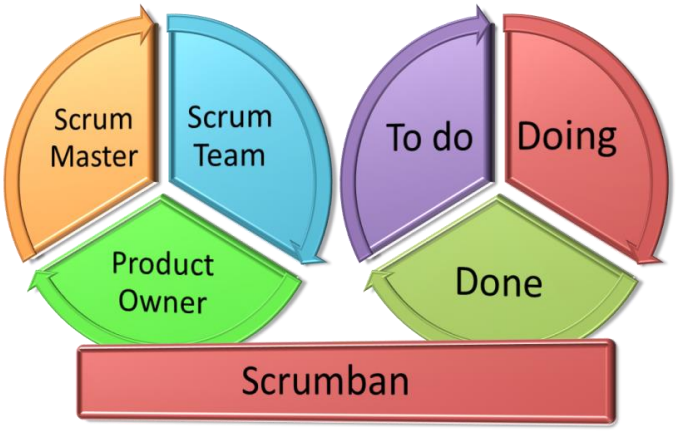

Fig. 6. Scrum + Kanban ≈ Scrumban

In Scrumban, development teams may adapt to production requirements and interests of the stakeholders, without being burdened by the project methodology. Scrumban inherits from Kanban the concept of elimination of elements that might lead to unwanted results, thus avoiding unnecessary processes. Also, Scrumban may optimize the teams' effort in order to achieve the quality standards assumed. [11] In other words, Scrumban ensures a slow transition from Scrum to Kanban.

According to the latest studies regarding the impact of agile methodologies on software development market [8[[12], Scrumban, as hybrid methodology, has gained one percent (from 6% in 2015 to 7% in 2016), but remains behind other two hybrid methodologies, that keep their position in surveys: Scrum-XP Hybrid (10%) and Custom Hybrid (multiple methodologies – 8%).

4 Agile Development and Social Relationships

Beyond philosophy and development methodologies, Agile is also a way to stimulate team work, which give it attributes of a socializing instrument. Looking 10-15 years ago, we would not have thought socializing would take place in the workspace. Work context socializing was (and still is in many cases) identified with team-building meetings, located as far as possible from the actual work place, preferably in isolated spaces (technologically, ecologically, demographically). Today, one of the most efficient methods to keep happy especially the younger generation is socialization at work place. Human resource specialists claim that employees will work more efficiently if there is a formula that combines their lives with the type of work performed. One of the first solutions towards this goal was online communication [13].

This is supported by the same HR specialists, noting that a large part of the current work force is made up of "generation X and Y". [13] Still, the complexity of socializing phenomenon goes to unexpected dimensions which are apparently disjoint: technology vs. psychology or abiding by contract (contractualism) vs. management.

DOI: 10.12948/issn14531305/20.4.2016.01

Without exaggerating too much, it seems that today the circulatory and locomotive systems of our daily lives have been "replaced" by technology. If you are not "connected" through technology, you don't "exist". In other words, technology has become an important part of socialization and Agile means technology too, beyond other aspects.

Psychology is the guiding science for the individual during his life, as a sort of GSP of human existence. The fundamental psychology concepts are often transferred to various domains, like systems theory and behavioral analysis. Knowing the behavior of a system provides remarkable opportunities to anticipate its reactions to various perturbing stimuli.

The individual is a biological system with a profound cybernetic character. In the context of team work, specific to Agile methodology, knowing the individual behavior of the employee becomes a critical aspect. Speaking of Agile and the inter-human relations we must also approach the managerial side. The contract theory proposed by J.J. Rousseau in 17th century (with later completions, addendums and feminist adaptations) becomes (again) interesting in agile context.

Beyond the individual work contract (or convention), a social contract on the level of the agile development team gives it a larger representativeness, greater trust. The advantages of the social contract are not directed against the management. They must be perceived as a mean of helping the team self-stimulate, evolve and achieve the maximum performance.

On the other hand, management as science, philosophy or paradigm must not give ground. AS proof, the decision makers' orientation toward modern management models places the employee (or the team, in the case of agile development) in the center of the organization goals (see the anthropocentric paradigm in contemporary management). Thus, creating the optimal work conditions for the employee becomes a goal, once achieved it indirectly generates the economic result and plus value for the business.

Unfortunately, in modern organization management, the main factor blocking large adoption of agile manner in business development is the organizational culture itself. From a managerial perspective, beside information, the organizational culture is a neo factor the impacts more and more the development of businesses and achieving economic goals. The latest goals in agile development [8] [14] show an obvious negative impact regarding the role of organizational culture, both failed agile projects (46%) and general agile adoption as current work method (55%).

Regarding the reasons of failure in adoption of agile projects, the next two positions are taken by the lack of experience in using agile methods (41%) and lack of managerial support (38%).

The main barrier blocking the adoption of Agile is the ability to change the organizational culture, followed by the organization resistance to change (42%) and existence of a more rigid framework in place (waterfall type) – 40%.

5 Conclusions

There are a few aspects that must be mentioned in the conclusion of our study. On one hand agile methodologies tend to occupy the largest part of the software development market. Even more, the agile way of managing projects is translated more and more to other sectors, at least through an agile model of project management. On the other hand, regardless of the philosophy used to approach the issues, sooner or later most businesses must reorient towards an agile model, considering the permanently changing (economic, social, political, cultural etc.) conditions in the surrounding environment. Certainly there is no "absolute best" agile development methodology, each project bringing its own goals and requirements.

Still, emerging from the sphere of process control and industrial production, Scrum and Kanban methodologies have generated the Scrumban hybrid, which may be a "compromise" solution for the supporters and adversaries of the parent methodologies.

DOI: 10.12948/issn14531305/20.4.2016.01

Last but not least, the agile development has a powerful social aspect, stimulating human interaction at the workplace and team spirit. In order to consolidate this status, development teams may confidently call on solutions similar to social contracts. Still, organizational inertia of managerial or functional order is a significant barrier blocking the adoption of Agile on a large scale.

References

[1] M. Mircea, *Arhitecturi informaționale agile în mediul colaborativ de afaceri*, ASE Bucharest, pp. 1-154, 2015, ISBN 978-606-34-0025-4

[2] M. Stoica, M. Mircea, B. Ghilic-Micu, *Software Development: Agile vs. Traditional*, Informatica Economică, vol. 17 no. 4/2013, pp. 64-76, pg. 73-74, DOI: http://dx.doi.org/10.12948/issn14531305/17.4.2013.06

[3] *What's the Difference? Agile vs Scrum vs Waterfall vs Kanban*, https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban, 24 may 2016

[4] B. Ghilic-Micu, M. Stoica, C. Uscatu, *Cloud Computing and Agile Organization Development*, Informatica Economică, vol. 18 no. 4/2014, pp. 5-12, pg. 9-11

[5] R. Gareis, *Happy Projects! Managementul proiectelor și programelor*, 3rd edition, Ed. ASE Bucharest 2010, ISBN 978-606-505-314-4

[6] *How to choose between Agile and Lean, Scrum and Kanban — which methodology is the best?*, https://realtimeboard.com/blog/choose-between-agile-lean-scrum-kanban/#.WCbR79KLR1t, 8 November 2016

[7] *What is Agile?*, https://agileinsights.wordpress.com/tag/agile-development/, 26 August 2013

[8] *The 10th annual STATE of AGILE Report*, VersionOne Inc. 2016, http://stateofagile.versionone.com/

[9] J. Sutherland, *The Scrum Guide*, http://scrumguides.org/scrum-guide.html

[10] H. Kniberg, *Kanban vs Scrum. A practical guide*, Deep Lean, Stockholm, May 19, 2009, www.crisp.se/henrik.kniberg/kanban-vs-scrum.pdf

[11] P. Gambill, *Scrumban: A different way to be Agile*, October 14th, 2013, http://www.deloittedigital.com/us/blog/scrumban-a-different-way-to-be-agile

[12] *The 9th annual STATE of AGILE Report*, VersionOne Inc. 2015, http://stateofagile.versionone.com/

[13] *Socializarea la locul de munca fidelizeaza angajatul*, http://www.wall-street.ro/articol/Careers/51004/Socializarea-la-locul-de-munca-fidelizeaza-angajatul.html, October 2008

[14] *The 2015 State of Scrum Report*, https://www.scrumalliance.org/social-networks, 2015

Marian STOICA received his degree on Economic Informatics from the Bucharest University of Economic Studies in 1997 and his doctoral degree in economics in 2002. Since 1998 he is teaching in Bucharest University of Economic Studies, at Economic Informatics and Cybernetics Department. His research activity, started in 1996 and includes many themes, focused on management information systems, computer programming and information society. The main domains of research activity are Information Society, E-Activities, Tele-Working, and Computer Science. The finality of research activity still today is represented by over 80 articles published, 25 books and over 40 scientific papers presented at national and international conferences. Since 1998, he is member of the research teams in over 30 research contracts with Romanian National Education Ministry and project manager in 5 national research projects.

DOI: 10.12948/issn14531305/20.4.2016.01

Bogdan GHILIC-MICU received his degree on Economic Informatics from the Academy of Economic Studies Bucharest in 1984 and his doctoral degree in economics in 1996. Between 1984 and 1990 he worked in Computer Technology Institute from Bucharest as a researcher. Since 1990 he teaches in Bucharest University of Economic Studies, at Economic Informatics and Cybernetics Department. His research activity, started in 1984 includes many themes, like computers programming, software integration and hardware testing. The main domain of his last research activity is the new economy – digital economy in information and knowledge society. Since 1998 he managed over 25 research projects like System methodology of distance learning and permanent education, The change and modernize of the economy and society in Romania, E-Romania – an information society for all, Social and environmental impact of new forms of work and activities in information society.

Marinela MIRCEA received her degree on Economic Informatics from the Bucharest University of Economic Studies in 2003 and his doctoral degree in economics in 2009. Since 2003 she is teaching in Academy of Economic Studies from Bucharest, at Economic Informatics and Cybernetics Department. Her work focuses on the programming, information system, business management and Business Intelligence. She published over 40 articles in journals and magazines in computer science, informatics and business management fields, over 30 papers presented at national and international conferences, symposiums and workshops, she was member over 15 research projects and project manager in 2 national research projects. She is the author and coauthor of 12 books. In February 2009, she finished the doctoral stage, and her PhD thesis has the title Business management in digital economy.

Cristian Răzvan USCATU received his degree on Economic Informatics from the Bucharest University of Economic Studies in 1997 and his doctoral degree in economics in 2007. In the present he is Associated Professor, PhD, currently working with the Bucharest University of Economic Studies, Faculty of Economic Cybernetics, Statistics and Informatics, Department of Economic Informatics and Cybernetics. Competence areas: computer programming, data structures. He is author/co-author of more than 10 books and 40 papers published in national and international journals.

DOI: 10.12948/issn14531305/20.4.2016.01