Analysis of Return on Investment in Different Types of Agile Software Development Project Teams

Goran MILANOV[1], Angelina NJEGUŠ[2]
[1]Banca Intesa a.d, Belgrade, Serbia
[2]Singidunum University, Belgrade, Serbia
gmilanov@live.com, anjegus@singidunum.ac.rs

This exploratory study of IT project teams in Serbia investigates how the choice of agile methods in different development project teams affects the return-on-investment (ROI). In this paper different types of software project teams are analyzed in order to examine and identify the business-value of using agile methods. In various software development project teams, the ROI of agile methods is yet to be fully explored, while the ROI of traditional methods is well-understood. Since ROI is important indicator of the projects success, in this paper we examine the factors that influence the ROI both from software solution customer point of view, and different agile project teams.

Keywords: *Agile Methodology, Scrum, Return on Investment, Project Teams, Software Development*

1 Introduction

The literature dealing with agile methodologies often states the faster return on investment (ROI) as an advantage over the traditional methodologies. As confirmation of this claim, it is stated that the return on investment becomes greater than zero immediately after the first sprint [1]. The ideal return on investment would be a delivered product which, after the first Scrum sprint possesses functionalities that can start repaying the investment [2] [3]. However, the product that results from the first sprint is very difficult to have a sufficient set of functionalities so that the client could immediately have a return on investment. For example, it is difficult for a bank to have the return on investment after the first sprint in which the functionality of client data entry was implemented but not the functionality which enables the bank to provide consumer loans to customers. Therefore, the key issue is how to determine the critical mass of functionalities [4] which defines a set of required functionalities whose implementation starts the return on investment for the client.

The critical mass of functionalities can be defined by the client only; that is, only the software solution purchaser who defines required functionalities, features and quality of the specific software solution can estimate which minimum set of functionalities and features of the software solution in what moment on the market can lead to the beginning of the return on investment. This setting is possible in systems:

- which can be introduced in phases
- in which a certain set of functionalities is launched on the market in order to retain the interest of the market for a specific product or set of services.

On the other hand, many systems due to their functional demands or the marketing strategy cannot be introduced in phases but have to be fully implemented before production. A recent systematic literature review conducted by Rico [5] identified the ROI of agile and traditional methods, in so far a very few empirical studies have focused on a different types of project teams. In the focus of any software development project are project teams therefore examination of the main impacts on ROI on different types of development teams are examined in this paper.

2 Literature Review

2.1 Agile Methodologies

Agile development software methods are often referred to as “lightweight” approaches to IT project management, as they are in direct contrast to the traditional long-term, plan-driven, document-heavy, bureaucratic approach to managing software development [6]. According to the Agile Manifesto [7] agile is based on a set of principles that focus on customer value, iterative and incremental delivery, intense collaboration, small integrated teams, self-organization and small and continuous improvements [8]. Agile methodologies have been influenced largely by Lean Production techniques, which were introduced in the Toyota Production System (TPS) by Taiichi-

Ohno[9]. The impact of lean thinking is only beginning to be felt in the software industry today. Lean software is based on a set of proven economic and mathematical principles that describe the flow of product information through larger business value chain, and include five elements such as [1]:

- Sustainably delivering value fast
- Respect for people
- Continuous improvement
- Management support
- Product development flow

There are many popular implementations of agile methodology, such as Scrum, XP (eXtreme Programming), DSDM (Dynamic Systems Development Model), Crystal, OUP (Open Unified Process), FDD (Feature Driven Development), MSF (Microsoft Solution Framework) for Agile, Agile project management framework (APM), Lean development, Rapid Application Development (RAD) etc. (Figure 1).

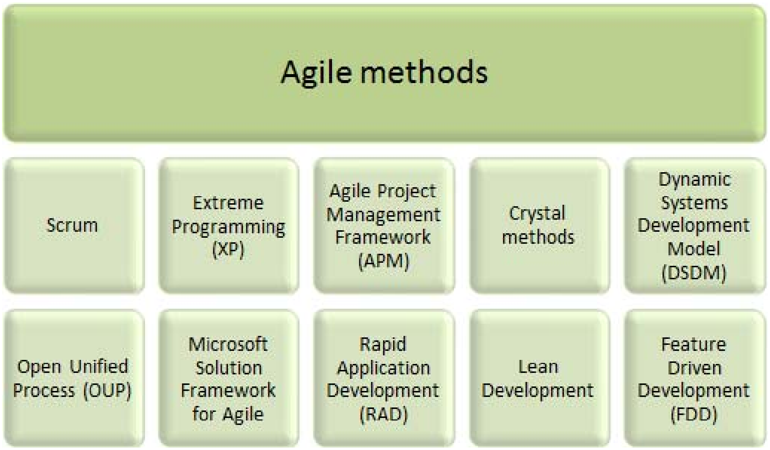
Agile methods
Scrum
Extreme Programming (XP)
Agile Project Management Framework (APM)
Crystal methods
Dynamic Systems Development Model (DSDM)
Open Unified Process (OUP)
Microsoft Solution Framework for Agile
Rapid Application Development (RAD)
Lean Development
Feature Driven Development (FDD)

Fig. 1. Agile methods

In recent years, research in the area of agile methods has grown due to the increase in the number of software project teams that use an agile approach. A major impact on project success is often related to agile methodologies. However, in spite of the growing popularity of agile management the rate at which software projects are failing is still alarming. According to Scott W. Ambler [10], the success rate for agile projects is 83% for small teams, 70% for medium-sized teams, and 55% for large teams (more than twenty-five people). Based on these facts it is clear that team size has a direct incidence on the success rate of the project. Since return on investment is an important indicator of the project success, in this paper, the key factors that influence the ROI in different project teams are examined.

2.2Agile Method "Scrum"

Scrum is an agile, lightweight framework for managing and controlling software development in rapidly changing and distributed environments [11]. Scrum is especially useful for project management and teamwork because it provides effective communication in the form of boundary objects (artifacts) and spanners (roles), it provides effective social integration by building up project team, and it provides needed control and coordination mechanisms [12]. Scrum has three major components:

- **Scrum roles -** simple team structure that involves roles such as: Product Owner, Scrum Master, and Team Members. Product owner identify and prioritize system functionality in form of a Product backlog. Scrum master is responsible for tracking project status, defines the sprint duration, runs the daily meetings, helps team productivity, resolve current problems etc. The Scrum team is self-organizing, and usually a cross-functional team that consists of five to ten people who work full time on the project. The team members are responsible for building and testing system quality.
- **Scrum artifacts -**the main artifacts are: User stories, Product Backlog, Sprint Backlog, and Burndown chart. User story is specifications of requirements that consists of a title and a brief, usually narrative, description of the desired functionality. Features and user stories express functional requirements. The product backlog consists of features, nonfunctional requirements, bug fixes, and all that needs to be done in order to deliver a system with quality,

on time and within specified resources. The sprint backlog is the set of priority stories planned for implementation in an iteration (called Sprint). Sprint usually lasts two or four weeks. There is only one backlog for the team. At the start of the sprint, the team breaks down each story into a set of tasks, and when a task is finished it is registered on a burn down chart.

- **Scrum process -**major activities are: the sprint planning meeting, the kickoff, the sprint, the daily Scrum, and the sprint review meeting. The Scrum process starts with a sprint planning meeting where the targeted functionalities are broken down and estimated. In the first part of Scrum meeting, the team defines the product backlog (the list of the user requirements), and determines the sprint goal (the formal outcome from particular sprint). In the second part of the meeting, the focus is on creating the sprint backlog. On the kickoff meeting the team defines the high-level backlog and the major project goals. Every day the Scrum team meets in a daily short stand-up Scrum meeting where they share what they did, plan to do or are doing, and also share eventually problems or new ideas. The sprint review meeting is more informal and is held at the end of each sprint where developed functionalities are presented to the product owner.

The Scrum process as defined in the theory [2] cannot support the processes that occur in practice as a result of changed requirements (functional and non-functional). To answer real process requests [13], Scrum framework has to be modified to support the necessary processes and activities that follow the process and implementation of the required changes. The modified Scrum process that involves activities for successful requirements change management during project is shown in Figure 2.

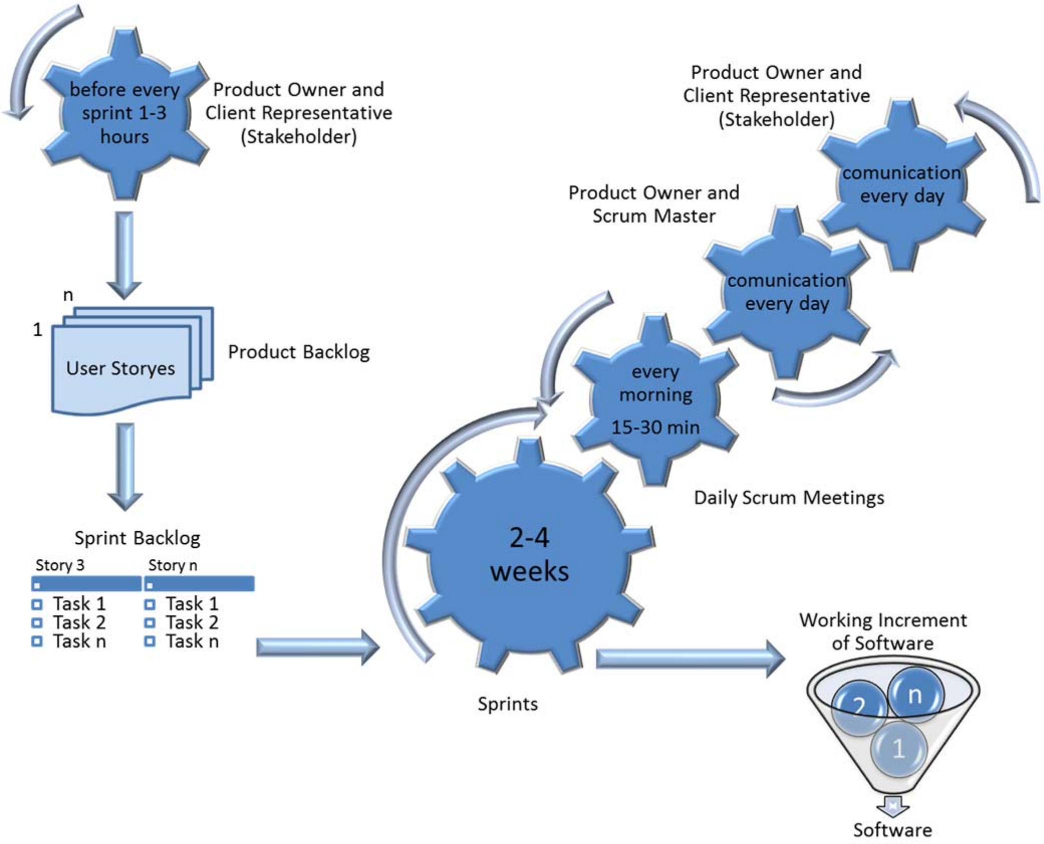

Fig. 2. The modified Scrum process [13]

2.3 Return on Investment of Agile Methods

There are many types of major metrics, models, and measures for measure the performance of agile software development projects. Some popular techniques are payback analysis, return on investment, net present value etc. However, ROI metrics are usually used to evaluate the economic value of investment in information technology

(IT) [5]. ROI is a percentage rate that measures the relationship between the amounts the business gets back from an investment and the amount invested [14]. However, ROI is a common way to measure the business value of agile methods for developing new software products [15], which can be presented as:

$$\text{ROI} = \frac{Benefits - Costs}{Costs} \cdot 100\%$$

where:

- *Benefits* – total amount of money gained from agile methods (includes economic benefit from using new system)
- *Costs* – total amount of money spent on agile methods (includes training, coaching, automated tools, etc.)

The ROI of agile methods when compared to the traditional methods is significant, because of the lower costs, fewer defects, and lower total life cycle costs [15]. Agile development focuses on cross-functional teams, rapid iterations with continuous customer input, on building working software that people can get hands on quickly, versus spending a lot of time writing specifications up front [16]. The most popular agile method is Scrum which is used by many large and small companies including Microsoft, Google, Motorola, SAP, Cisco, etc.

Scrum structures software development in cycles of work called Sprints which last no more than one month. At the beginning of each Sprint, a cross-functional team selects customer requirements from a prioritized list of features written in the product backlog. When it comes to Scrum theory, return on investment (ROI) is usually calculated by the formula [16]:

$$\text{ROI} = \textit{Business value} \,/\, \textit{Effort}$$

where *Business value* and *Effort* are expressed in points given to the Product backlog items. The Scrum role of Product Owner is responsible not only for monitoring the return on investment but also for its management by making the Product Owner responsible for the prioritization of the Product backlog items. By calculating the ROI for the Product backlog items, a prioritization of the Product backlog items can be done and the ROI can be maximized immediately in the first sprints. Yet it requires a great simplification and is possible only in the ideal case i.e. theory.

This calculation of the ROI is essential for the Scrum team, but it is not necessarily important to the software purchaser, especially to management. The management will always calculate the ROI as the ratio of invested money and return on investment. The ROI is not affected only by the methodology [17] [18] [19] but also by the required functionalities, the technologies used in the development, the platform on which the front-end applications is executed, marketing, contracting, etc.

3 Proposed Methodology

The following chapter focuses on the methodology used to identify and analyze ROI within different project teams. This study has been made with the development teams where authors of this paper were project managers or part of project teams. The ROI is analyzed both from perspective of different types of development teams and customers who purchase software solutions.

The most teams that develop software solutions can be divided to:

1. teams which develop products for their own performance on the market
2. teams that develop products at the request of another company (contracts)
3. teams that develop products for internal use in companies where they work (in house)

For each type of teams Scrum agile method is used. From customer perspective, ROI is analyzed in two types of projects:

1. Implementation of the ERP solution
2. Implementation of the e-commerce solution

Within these two types of projects, traditional methods are used, such as ASAP methodology for implementation of SAP ERP solution and Microsoft Solution Framework (MSF) for implementation of e-commerce solution.

During and after projects implementation, data that had been collected were costs and benefits (income) of the developed projects. Considered costs were costs associated with developing the system, and operation and maintenance costs.

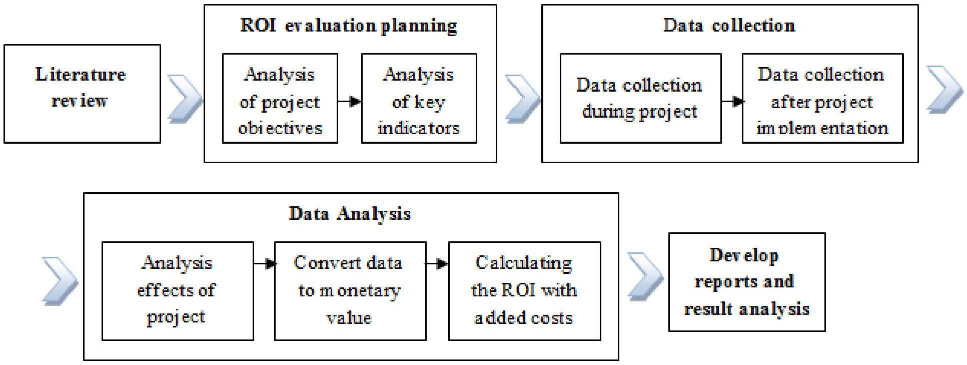
Literature review
ROI evaluation planning
Analysis of project objectives
Analysis of key indicators
Data collection
Data collection during project
Data collection after project implementation
Data Analysis
Analysis effects of project
Convert data to monetary value
Calculating the ROI with added costs
Develop reports and result analysis

Fig. 3. Proposed research methodology

Considered benefits of the project were both tangible and intangible benefits. They are represented as total amount of money gained from used methodology (derived from operation of new system). The proposed research methodology is represented in Figure 3.

4 Analysis and Results of Return on Investment in Different Project Teams

Projects for development of software products differ in many ways, not only in the functionalities that the product should possess, in the technologies used for development, or project management methodologies but also in the manner of obtaining and initiating development projects, the model of financing, the strategies for market performance, marketing, etc.

4.1Teams Which Develop Products for Their Own Performance on the Market

Teams that develop products for wider market, they develop products based on the company management vision or based on the idea of an employee or on the market research. These teams usually develop business solutions such as ERP, CRM, BI systems or systems for some specific needs of certain branches of activity, but also for small and medium systems with different purposes (Web sites, services, etc.). These projects take longer to finish since they are usually massive. All investments in product development are on the side of the company that works on the software development. Business decisions about product development should be based on the market research and projections of market trends. After its appearance on the market, the product may return on investment and make some profit. Due to the changes on the market or because of financial reasons, the company management may decide, at some stage, that the project is unprofitable and may stop it. Figure 4 represents the summary amount of investment and return on a monthly basis on the project of a web based software system which includes a web application for a mobile client and a web administration application used for billing certain services. Scrum was used for project management.

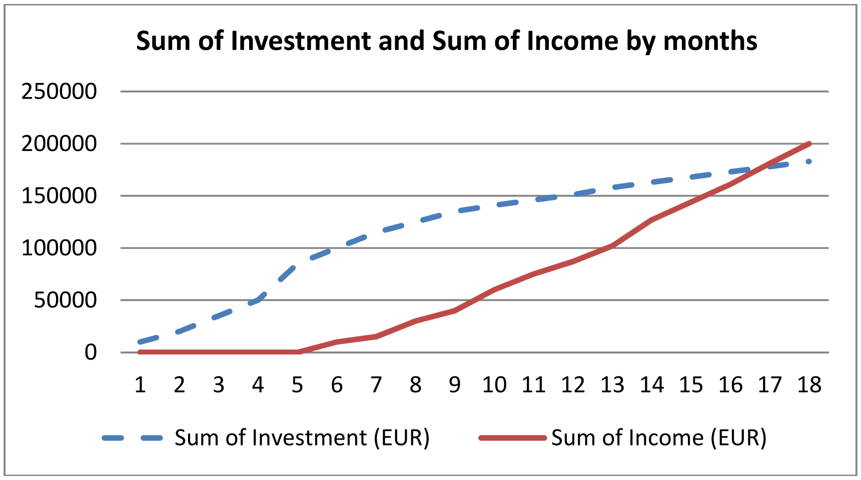

Fig. 4. Sum of investment and sum of income by months of the web based project

All sprints were completed on time and at the end of each sprint the set of anticipated functionalities was delivered. The company that developed the software made a business decision to invest in the development of the aforementioned software system and offer it via the web to different business systems. The company that developed the software system, hosted the solution, too; that is, they invested in and provided the server hardware, server licenses, links, support and maintenance. In the fifth month since the beginning of the project, the server hardware and server licenses were purchased and the necessary links to the Internet were leased and set up. In the sixth month, the first production version was implemented and the first contract was signed with an external system user.

Figure 5represents the percentage of the return on investment per month. It can be seen that the first return of funds occurred in the sixth month since the beginning and that for the total return on investment of 100%, 17 months were needed.

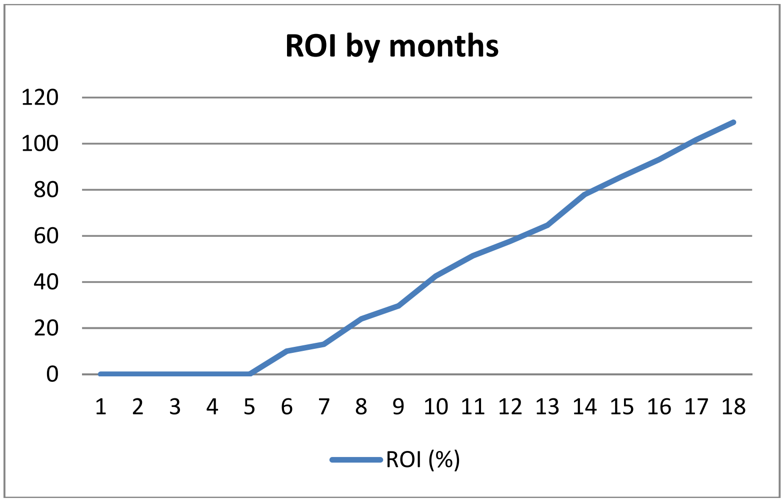

Fig. 5. ROI by months for web based project

The factors that have an impact on the return on investment are the following:

- the product needed to have all the required functionalities so that it could be put into production environment i.e. so that the exploitation of the product could start. Thus the product could not be put into production before the end of the last sprint;
- the sale depended on the set of functionalities that could be presented to potential buyers. Because of that, a larger part of the functionalities (over 80%) had to be finished before presenting the product to the buyers;
- the sale depended on references that the use of the specific product have had because there have already been similar products on the market. Therefore, it was important for the product to offer new functionalities and to show excellent results for the first users;
- the sale was not too dependent on the traditional marketing activities because the product was developed for a specific industry sector and was not intended for a widespread use. Therefore, marketing expenses did not burden the project additionally and the return on investment did not depend on the traditional marketing activities.

The project was successful and fulfilled the expectations of management at this particular market. The choice of Scrum framework for managing this development project did not have any crucial influence on the ROI and the success of the project.

4.2Teams that Develop a Product Based on a Contract with an External Company

A large number of contracts for development of software products are signed after a tender where the ordering party (Client) defines what kind of product wants to purchase. Based on the documents that the client has created and which are usually not clear enough, the potential supplier of the desired software gives an offer defining the delivery time and price. Only after signing the contracts, a detail analysis, documenting of what is to be done (user scenarios) and the development itself start. Common characteristic for all these development projects is that the initial analysis is very short and it is based on insufficient information. Also, the development time and price is decided by the sellers who, by their best offer for the client, try to get the project, get the commission and engage their development team. In many cases the development cost exceeds the contract price. The reasons for this can be the following:

- the client constantly expands the functionality requirements because the tender documentation wasn't clear and complete enough;
- the necessary resources and development time were underestimated because of insufficient information;
- the seller gave the lowest possible price in order to get the contract and the sales commission and the blame for the failure of the project was transferred to the development team;
- the original offer did not include maintenance cost and the contracts concerning maintenance were not signed although in most cases these development teams maintain concrete solutions and provide customer support.

The Figure 6 represents the sum of investment and the sum of income by months for a project obtained in the tender. Software development project consists of an application for tablet PC in vehicles which communicates with headquarters via the web, a Pocket PC application for agents on the field, and an application for GPS tracking of the client's vehicles. The first installment of the total amount was paid to the development company right after the signing of the contract and the beginning of the project. Scrum was used for project management. All sprints were finished on time and at the end of each sprint a set of functionalities was delivered to the client. Before every sprint there were some changes and the additional requests from the committee formed by the client and consisting of the client's employees and an external oversight body. Each change and additional requirement had to be negotiated because the requests in the tender documentation were not clearly defined and detailed so every new requirement made the project more and more expensive. However, the firm that developed the project delivered the first version of the project and collected the second installment from the client. Then, some senior managers of the client who had seen the product for the first time started defining additional requests claiming that the tender documentation implied that by default. Instead of paying the last installment one month after the delivery of the first version as the contract had defined it, and under the pretext that the product was not complete without the additional changes and improvements, the last agreed upon installment was paid in the fourteenth month from the beginning of the project. In Manifesto for Agile Software Development [7], its authors say that they appreciate more "the cooperation with clients than contractual arrangements". The point is that a company which develops and sells software solutions has to get paid adequately for its work in order not only to return on investment but also to make profit.

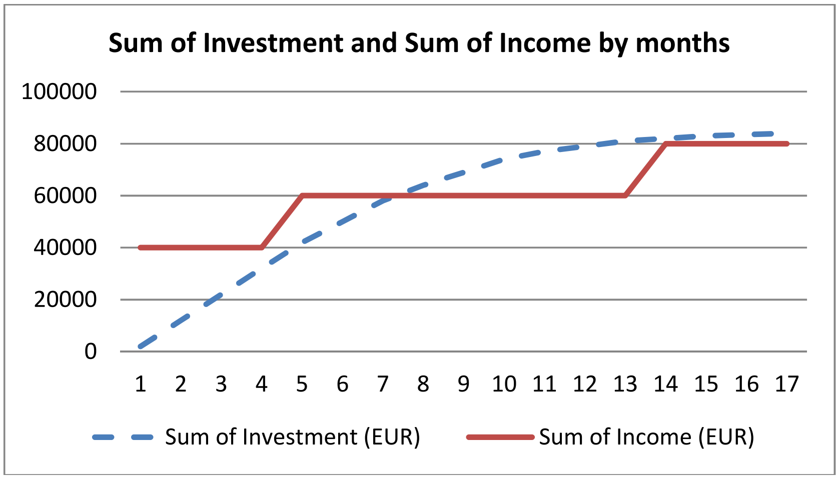

Fig. 6. Sum of investment and sum of income by months for tender based project

In the Figure 7, which represents the percentage of ROI, it can be seen that the ROI did not reach 100% even after the last installment. Not only did not the company that worked on the development make profit but they were also at a loss. In many cases in practice, the ROI in the projects acquired in tenders does not exceed 80%.

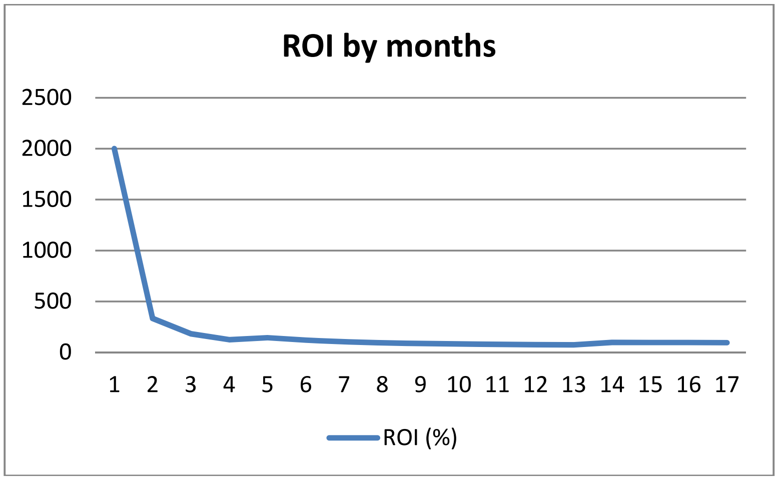

Fig. 7. ROI by months for tender based project

The factors affecting the return on investment are as follows:

- the product had to have all the required functionalities so that it could be placed in the production environment i.e. that it could begin to be used. Thus the product could not be placed in production before the end of the last sprint;
- obtaining the contract in the tender depended mostly on the lowest possible price of the product that the bidder defined;
- the sale did not depend on investment in any traditional marketing activities.

The project was not successful and did not meet the expectations of the project implementers. On the other hand the client did get a quality product at a very low price. The scheduled project realization time limit was not broken, that is from the point of view of the client the project was successful. The choice of Scrum as framework for managing this particular development project did not have any crucial influence on the ROI and the project success.

The contracts for development of software projects are not obtained in tenders only. They can be obtained in direct negotiations with a client and these projects are more favorable for a software development company.

4.3 Teams which Develop Products for Use in the Company Where They Work (in House)

After the initial development that should secure the minimum of functionalities of a specific software product which cover the required minimum of business processes, all further activities are developing of new functionalities or changing the existing functionalities. The initial software development is sometimes avoided by purchasing the source code of an existing software solution from a software development company. The most requests for new functionalities or demands are not significant when it is compared with required resources, time and cost. In most cases, development teams also provide user support and maintenance of specific solutions.

The work of these teams is almost impossible to organize in accordance with agile or traditional methodologies because of large number of demands, small size of these demands and very dynamic development. For example, in these teams, daily Scrum meetings do not make any sense because:

- teams are very small (2-3 developers usually). In most cases only one developer works on realization of one request;
- many demands that should be realized require few man-hour, not few man-day;
- due to the heterogeneity of requests, it is not possible to consolidate larger number of small requests into a larger one in order to organize team work.

It is impossible to track the return on investment (ROI) when it comes to working in teams on a large number of requests for changes and improvements that are small in scale. Since these teams are employed in business systems, they are treated as services for the business systems and the salaries of the team members are treated only as an expense and the return on investment is not considered because the small size of the requests prevents that. On the other hand, when it comes to great number of changes and improvements, it is possible to organize outsource teams but in most cases their engagement is also treated as an expense and the ROI is not tracked.

5 ROI Analysis from the Aspect of a Software Product Customer

ROI is not important only for companies that develop software products but also for companies that purchase software products for their own use or their performance on the market via the Internet.

5.1 ROI in Projects of ERP Solution Implementation

Figure 8 represents the summary amount of investment and return by months for the project of acquisition and implementation of an ERP system by one industrial company. Along with the purchase of ERP system licenses, the project anticipated a certain customization (adjustment) of the system, migration of existing data and training of users and administrators. After seven months, from the beginning of the project, the necessary hardware and software licenses were bought. After eight months the system was deployed and after nine months from the beginning of the project the last installment was paid. The system returns an estimated monthly amount.

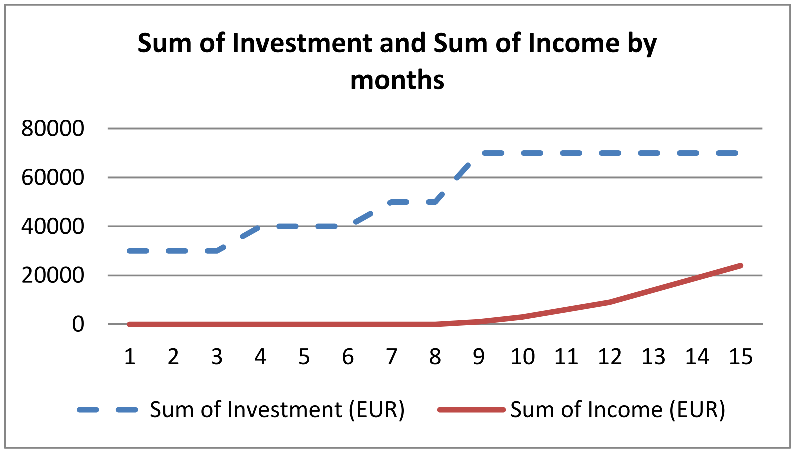

Fig. 8. Sum of investment and sum of income by months for ERP based project

In figure 9, which represents the percentage of ROI, it can be seen that the ROI in these cases is very slow.

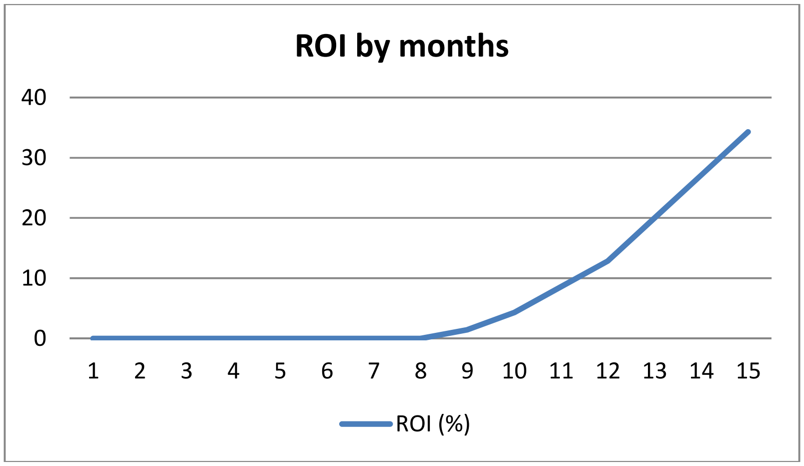

Fig. 9. ROI by months for ERP based project

The factors affecting the return on investment are as follows:

- the product had to have all the required functionalities so that could be deployed to the production environment i.e. that could begin to be used. Therefore, the product could not be deployed before the end of the last sprint;
- the client uses the system in his internal business operations.

The project was successful for the client and the project implementer and met the expectations of both. The project implementer used his own methodology in the development of the project but the choice of methodology did not have any crucial influence on the ROI and the project success.

5.2 ROI in the Project of Implementation of E-Commerce Solution

For companies that buy software for their own performance on the market via the Internet, the return on investment depends on factors such as: good market analysis, well-designed application and its contents, well thought-out market strategy, marketing, etc. Figure 10represents the summary amount of investment and the summary amount of return on investment by months on the project of implementation of a site for selling goods which the purchaser of the solution offers on the Internet. For project management, an internal modification of Microsoft Solution Framework (MSF) was used. All iterations were completed on time and after the each iteration a set of functionalities was supplied. Eight months since the beginning of the project, it was set up on production. The system had to have all the required functionalities, i.e. it could not have been set up in phases because the market performance strategy and market positioning relied on several functionalities that the competition, which had already had sites for sale on line, should not have seen before the time.

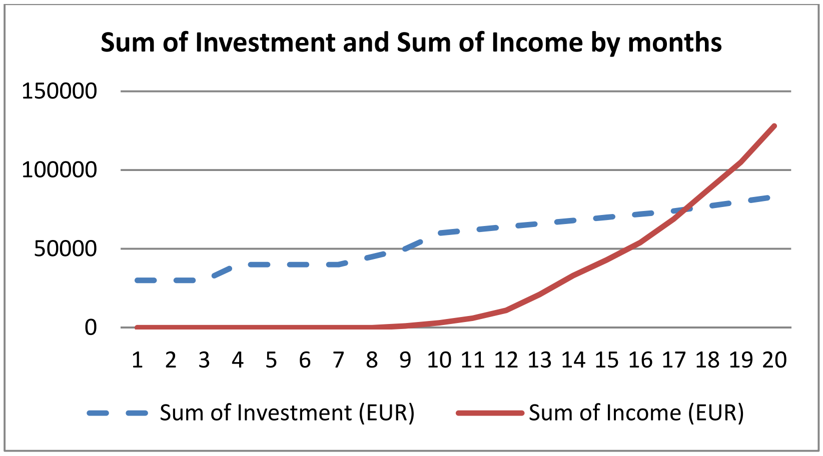

Fig. 10. The sum of investment and the sum of income by months for e-commerce based project

As it can be seen in figure 11, the company in question had the return of ROI higher than 100% only after eighteen months, meaning that the company started earning money on the specific software system.

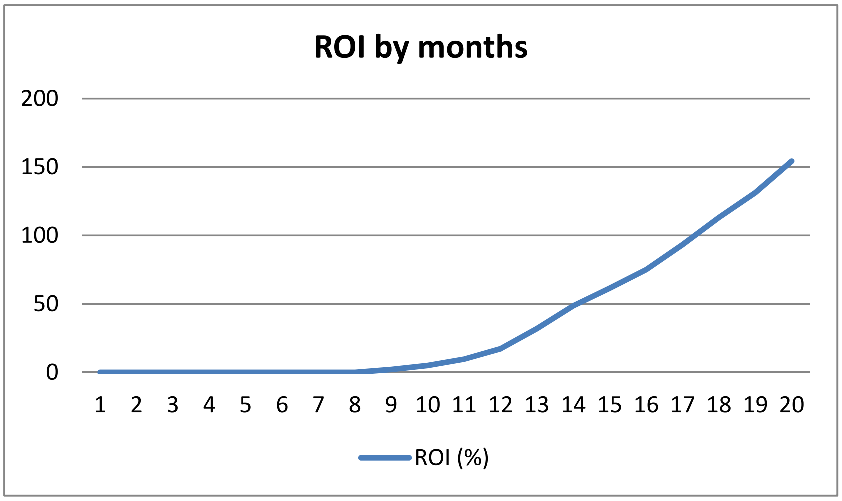

Fig.11. ROI by months for e-commerce based project

The factors affecting the return on investment are as follows:

- the product had to have all the required functionalities so that it could be placed in the production environment i.e. that it could begin to be used. Thus the product could not have been placed in production before the end of the last sprint;
- the success of the project depended on some new functionalities that other systems for the same purposes did not have;
- the success of the project depended on the market analysis where the sale of the assortment of goods that the client offered was analyzed;
- the success of the project depended on the planned strategy in the market;
- the success of the project depended on investing in traditional marketing activities.

The project was successful and met the expectations of the management of both the client and the implementer of the project. The choice of MSF as a management methodology for this specific development project did not have any crucial influence on the return of investment (ROI) and the success of the project.

6 Conclusion

If a development team delivers a software product with required functionalities, on the planned schedule with planned resources and costs, then the specific project has been successful for the development team. This does not mean that the specific project has been successful for the entire company and also, does not mean that the project will be financially viable. The choice of some agile methodology for managing development projects, can bring additional optimization to the development team and reduce development cost [5]. On the other hand, most teams which work together on different projects for many years and use some iterative (traditional) methodology for project management, have already optimized their work on the project by adjusting the methodology that they use. Thus, the choice between Scrum and any other agile methodology is unlikely to make any significant optimization of work in such teams, but finer adjustments. Therefore, the use of agile methodologies for successful teams cannot be a decisive factor where the return on investment is concerned [20].The return on investment in practice depends on many factors such as the product quality, the features, the services and contents it offers, the market analysis, the market projections, the contracting, good market strategy, marketing, etc. Therefore, there are many decisive factors in addition to the optimization of the development team that should be kept in mind, planned, monitored during the project in order the project to be successful for all parties involved in the project.

References

[1] D. Leffingwell, *Agile software requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, Pearson Education, Boston, MA, 2011.

[2] K. Schwaber, “Agile Project Management with Scrum”, *Microsoft Press*, 2004.

[3] B. Hunter, M. Fowler, G. Hohpe. (2002, August 12) Agile EAI Methods: Minimizing Risk, Maximizing ROI [Online]. Available: http://www.hohpe.com/Gregor/Work/docs/AgileEAIMethods.pdf.

[4] A. Njeguš, G. Milanov, “Qualitative Comparison of Agile and Iterative Software Development Methodologies”, In *Proceedings of the Nineteenth Telecommunications Forum* (TELFOR), Belgrade, Serbia. 2011.

[5] D.F. Rico (2008, August 12) What is the Return on Investment (ROI) of Agile Methods [Online]. Available: http://www.afei.org/WorkingGroups/ADAPT/Documents/rico08a[1].pdf.

[6] B. Boehm, “Get Ready for Agile Methods, With Care”, *IEEE Computer*, 35(1), pp.64-69, 2002.

[7] K. Beck *et al.* (2001, August 2012). Manifesto for Agile Software Development. *The Agile Manifesto* [Online]. Available: www.agilemanifesto.org.

[8] V. Lalsing, S. Kishnah, S. Pudaruth, “People factors in agile software development and project management”, *International Journal of Software Engineering & Applications* (IJSEA) 3(1), pp. 117-137, 2012.

[9] R. Nair. (2012, August 12). Agile n Scrum [Online]. Available: http://www.agile-community.com/profiles/blogs/agile-n-scrum.

[10] S.W. Ambler. (2010, August 12). IT project success survey. *Ambysoft* [Online]. Available: http://www.ambysoft.com/surveys/success2010.html.

[11] J.O. Coplien. (2011, August 2012). It’s Ordered - Not Prioritized. *Scrum Alliance* [Online]. Available: http://www.scrumalliance.org/articles/367-

its-ordered--not-prioritized, accessed 22 August 2012.

[12] L. Pries-Heje, J. Pries-Heje, „Agile and distributed project management: a case study revealing why Scrum is userful“, *19th European Conference on Information Systems – ICT and Sustainable Service Development*, Finland, 2011.

[13] G. Milanov, A. Njeguš, „Requirements Change Management According To Modified Scrum Process”, In *Proceedings of the 20th Telecommunications Forum* (TELFOR), Belgrade, Serbia. 2012.

[14] J.L. Whitten, L.D. Bentley, K.C. Dittman, *Systems Analysis Design Methods*. McGraw-Hill, NY, 2004.

[15] D.F. Rico, H.H. Sayani, S. Sone, *The Business Value of Agile Software Methods: Maximizing ROI with Just-In-Time Processes and Documentation*. J.Boss Publishing, Fort Lauderdale, FL, USA, 2009.

[16] P. Deemer, G. Benefield, C. Larman, B. Vodde. (2010, August 12) The Scrum Primer [Online]. Available: http://www.goodagile.com/scrumprimer/scrumprimer.pdf.

[17] R. Bar-Nahor. (2005, August 2012). SCRUM: The Product Owner-It is all about ROI. *Amdocs* [Online]. Available: http://www.volaroint.com/file/DC-%20VOLARO-%20Training-Scrum-%20Amdocs-%20Scrum-About%20ROI.pdf.

[18] K. Playfair (2008, August 12) Two Sides of the Same Coin: Risk Management and ROI in Scrum. [Online]. Available: http://www.danube.com/news/Risk_Management_and_ROI_in_Scrum.

[19] R. Sabbagh, M. Garrido. (2009, August 12). Scrum and the Recession. [Online]. Available: http://www.scrumalliance.org/articles/127.

[20] H.F. Cervone, „Understanding agile project management methods using Scrum“, OCLC *Systems & Services*, Emerald, Vol. 27 No. 1, pp. 18-22, 2011.

Goran MILANOV, MSc is the chief of the Development and Acquisition ICT application services at the Department for Information and Communication Technologies in Banca Intesa from Belgrade, Serbia. His research focuses on Database Management Systems, Business Intelligence Systems Development, Software Development Methodologies, and Web Services. He has published a long list of papers on these topics in regional journals and conferences, and in the past six years he has participated as a regular speaker at the annual Microsoft Conference “Sinergija” held in Serbia. Goran Milanov has been employed in many software developing companies and took part in a number of large software developing projects that were developed for different business systems in the country.

Angelina NJEGUŠ, PhD is an Associate Professor of Informatics and Computing at Singidunum University from Belgrade, Serbia. Her research focuses on Information Systems Analysis and Design, Business Intelligence Systems, Data Quality and Master Data Management, Software Development Methodologies, SOA and Cloud Computing. She has published a long list of papers and seven books. Angelina Njeguš has practical experience in a number of projects, where she worked as software project manager or consultant. Currently she teaches Information Systems at the Department of Informatics and Computing, Information Systems in Tourism and Hospitality at the Department of Tourism and Hospitality, and Decision Support Systems on master studies at Singidunum University.