THE ADVANTAGES OF AGILE METHODOLOGIES APPLIED IN THE ICT DEVELOPMENT PROJECTS

Vangel Fustik

Faculty of Electrical Engineering and IT, UKIM, Skopje
e-mail: vfustic@feit.ukim.edu.mk
R. Macedonia

Abstract: The paper presents an overview of the agile methodologies considering important characteristics in software development projects such as process, project team structure, documentation, practices, software types and tools. The advantages are discussed and presented on real project. A survey that was realized recently point out the usage of particular methodologies that are most efficient and cost effective in modern and global project environment. Among the agile, Scrum is mostly accepted methodology with specific characteristics in comparison to other agile methodologies. The results from the survey are discussed and useful conclusions could be obtained in order to understand the attitude in creating the business of the IT companies.

Key words: Agile methodologies, Software development projects, Agile project management.

1. INTRODUCTION

In modern terminology Agile project management is defined as an iterative approach to managing software development projects that focuses on flexible and continuous releases incorporating customer feedback in every different activity in project. In accordance with the Toyota lean production concept (1940), software development teams have embraced agile methodologies to reduce costs and increase transparency while quickly addressing their customers' ever-changing needs. Agile approach helps software teams collaborate better and innovate faster than ever before. In agile approaches, leadership encourages teamwork, accountability, and close communication in project development. Business stakeholders and developers must work together to align the product with customer needs and company goals.

For clarification, traditional agile project management can be divided in two categories: Scrum and Kanban. While scrum is focused on fixed-length project iterations, kanban is focused on continuous releases. Upon completion, the team immediately moves on to the next.

Considering that the project success or failure depends upon the selection of appropriate type of software development methodology, some experience could be helpful. For example, for the projects where the scope of works is well known, or for very small and inexpensive projects - waterfall model is the best. For mission critical projects spiral model is best, for unskilled client prototype model is best, and recently object orient model was popular because it can interact with real items. However, it does not mean prototype model cannot be used for other projects, it simply means they are good for that type of projects along with other type of projects. Whatever model we choose for developing software projects there are mandatory stages [1], as follows:

- Requirement gathering - experts discover whether the proposed software project is feasible or not under technical, economic conditions.
- Requirement analysis - professionals gather information about proposed project. How much resource, manpower needed for this software project is determined in this stage.
- Designing - In this stage object oriented models are used to design the software, means how software work. It is very useful for good coding.
- Coding - In this phase programming languages are used to develop or create software. It uses many different languages and software to develop or complete the projects.
- Testing - It is the phase in which some testing methods or standards are used to test the developed software using black box testing unit testing integration testing, etc.
- Implementation /Deployment - after various types of testing, if software pass all testing conditions then it is implemented or deployed in client system.
- Maintenance - After implementation organization or company make contract to serve their service of certain time period as maintenance such as adding some component or module or feature, change in platform.

Since, Agile software development methodologies are in fast development and usage, this research is oriented towards their project application and among them especially Scrum methodology is emphasized.

2. AGILE SOFTWARE DEVELOPMENT METHODOLOGIES

Fast development of agile methodologies relates to the time before 2000 and their expansion was indicated in 2001 by the group of consultants and experts working in software development projects. They created the Agile manifesto [2] which recommends values and principles common for all agile methodologies. As the most important values the following are listed:

- individuals and interactions over processes and tools,
- working software over comprehensive documentation,
- customer collaboration over contract negotiation,
- responding to change over following a plan;

In the dynamic global market if applying these values, it is easier to respond to requirements and conditions. There are few important items that need to be discussed in the context of the new agile methodological approach: process, project team structure, documentation, practices, software types and tools. IT companies work in a new environment that is global and economically driven. That means that the project teams could be elsewhere and communicate to provide documentation and project delivery on different. In a survey, exposed latter in the paper some of the findings are discussed. Agile approach is focused on flexibility and it is not linear and deterministic. In general, it tends to develop software as soon as possible, with iterations, and doing refinement and reprioritization in every iteration. Agile methods that appeared first were: eXtreme Programming, Crystal methods, Adaptive Software Development (ASD), Scrum and Dynamic Systems Development Method (DSDM). Later evolved: Feature Driven Development (FDD), Lean Development, Open Source Software Development and others. Statistically, four of them are the most frequently used: XP, Scrum, DSDM and ASD.

3. THE MAIN CHARACTERISTICS OF AGILE METHODOLOGIES

3.1. Scrum methodology

The main activity in Scrum project management [3] is the Sprint, a time boxed iteration that usually lasts between 1-4 weeks, with the most common sprint length being 2 weeks.

- Sprint Planning Meeting
- Daily scrum or daily standup
- Sprint Review
- Sprint Retrospective

Agile Project Management is about embracing change, even late in the development stage. It□™s about delivering the features with the greatest business value first, and having the real-time information to tightly manage cost, time and scope. Agile Project Management [4] reduces complexity by breaking down the many-months-long cycle of building requirements for the whole project, building the entire product and then testing to find hundreds of product flaws. Instead small, usable segments of the software product are specified, developed and tested in manageable, two- to four-week cycles. In projects Scrum methodology is a proven and widely adopted method for achieving software agility. With short sprints, this iterative cycle can be repeated until satisfactory work items have been completed, the budget is critical and/or a deadline is coming shortly. It is in contrasts [4] to the more traditional waterfall style approach that fixes the project scope upfront, requiring the extensive creation of requirements, analysis and design documentation before development can get started. It is appropriate only for routine projects where Scope of works is well known, and such project already being realized. In other projects, with unclear scope of works (specifications) delays and budget overruns are common, and the failure to prioritize the feature set often results in low quality of project outcomes that are overloaded with features that the client/user does not actually require.

Shifting from traditional project management approaches to Scrum project management requires an adjustment in terms of the activities that are carried out, the artifacts that are created and the roles within the project team:

- The Product Owner handles setting project goals, handling the tradeoff of schedule versus scope, adapting to changing project requirements and setting priorities for product features.
- The ScrumMaster guides the team to prioritize their tasks and removes impediments to handling their tasks. Agile project management with scrum is an entirely new world!

- The Team Members directly handle most of the task assignment, daily detail management, progress reporting and quality control for the product.

Project teams have to adopt different rules and apply different practices. These arguments lead to the necessity for a new team organization. For example, it can be the case when people work in pairs and use the same computer to solve difficult issues. Many teams start out using spreadsheets to manage the product backlog and task boards to see and change the state of tasks during the current sprint, often with a whiteboard and sticky notes. This approach tends to work well for small, co-located teams. However, as the backlog increases and remote members require project visibility many organizations implement a more sophisticated tool to centrally manage projects and enable cross-team collaboration.

3.2. Lean Software Development

It is an iterative agile methodology originally developed by the principles and practices of the Lean Enterprise movement, and the practices of companies like Toyota. Lean Software Development focuses the team on delivering Value to the customer, and on the efficiency of the "Value Stream," the mechanisms that deliver that Value. The main principles of Lean methodology include:

- Eliminating Waste
- Amplifying Learning
- Deciding as Late as Possible
- Delivering as Fast as Possible
- Empowering the Team
- Building Integrity In
- Seeing the Whole

Lean methodology eliminates waste through such practices as selecting only the truly valuable features for a system, prioritizing those selected, and delivering them in small batches. It emphasizes the speed and efficiency of development workflow, and relies on rapid and reliable feedback between programmers and customers. Lean uses the idea of work product being "pulled" via customer request.

3.3. The Kanban Method

Usually this method is used by organizations to manage the creation of products with an emphasis on continual delivery while not overburdening the development team. Like Scrum, Kanban is a process designed to help teams work together more effectively.

Kanban is based on 3 basic principles:

- Visualize what you do today –workflow
- Limit the amount of work in progress –WIP
- Enhance flow: when something is finished, the next highest thing from the backlog is started.

Kanban promotes continuous collaboration and encourages active, ongoing learning and improving by defining the best possible team workflow.

3.4. XP

This methodology was originally described by K. Beck, and has emerged as one of the most popular and controversial agile methodologies. XP is a disciplined approach to delivering high-quality software quickly and continuously. It promotes high customer involvement, rapid feedback loops, continuous testing, continuous planning, and close teamwork to deliver working software at very frequent intervals, typically every 1-3 weeks. The original XP recipe is based on four simple values: simplicity, communication, feedback, and courage and with additionally twelve supporting practices: Simple Design, Test-Driven Development, Coding Standards, etc.

3.5. Crystal methodology

The Crystal methodology is one of the most adaptable approaches to software development. Crystal is actually comprised of a family of agile methodologies such as Crystal Clear, Crystal Yellow, Crystal Orange and others, whose unique characteristics are driven by several factors such as team size, system criticality, and project priorities. This Crystal family addresses the realization that each project may require a slightly tailored set of policies, practices, and processes in order to meet the project' unique characteristics. Like other agile process methodologies, Crystal promotes early, frequent delivery of working software, high user involvement, adaptability, and the removal of bureaucracy or distractions.

3.6. Dynamic Systems Development Method (DSDM)

Since1994, this method grew out of the need to provide an industry standard project delivery framework for what was referred to as Rapid Application Development (RAD) at the time. DSDM methodology has evolved and matured to provide a comprehensive foundation for planning, managing, executing, and scaling agile process and iterative software development projects. DSDM is based on nine key principles that primarily revolve around business needs/value, active user involvement, empowered teams, frequent delivery, integrated testing, and stakeholder collaboration. DSDM specifically calls out "fitness for business purpose" as the primary criteria for delivery and acceptance of a system, focusing on the useful 80% of the system that can be deployed in 20% of the time. Requirements are base-lined at a high level early in the project. Rework is built into the process, and all development changes must be reversible. Requirements are planned and delivered in short, fixed-length time-boxes, also referred to as iterations, and requirements for DSDM projects are prioritized using following rules: **M -** Must have requirements; **S -** Should have if at all possible; **C -** Could have but not critical; **W -** Won 't have this time, but potentially later

3.7. Feature Driven Development (FDD)

FDD is a model-driven, short-iteration process. It begins with establishing an overall model shape. Then it continues with a series of two-week "design by feature, build by feature" iterations. The features are small, "useful in the eyes of the client" results. FDD designs the rest of the development process around feature delivery using the following eight practices:

- Domain Object Modeling
- Developing by Feature
- Component/Class Ownership
- Feature Teams

- Inspections
- Configuration Management
- Regular Builds
- Visibility of progress and results

4. COMPARISON OF THE RELEVANT FEATURES FOR SOFTWARE PROJECT DEVELOPMENT

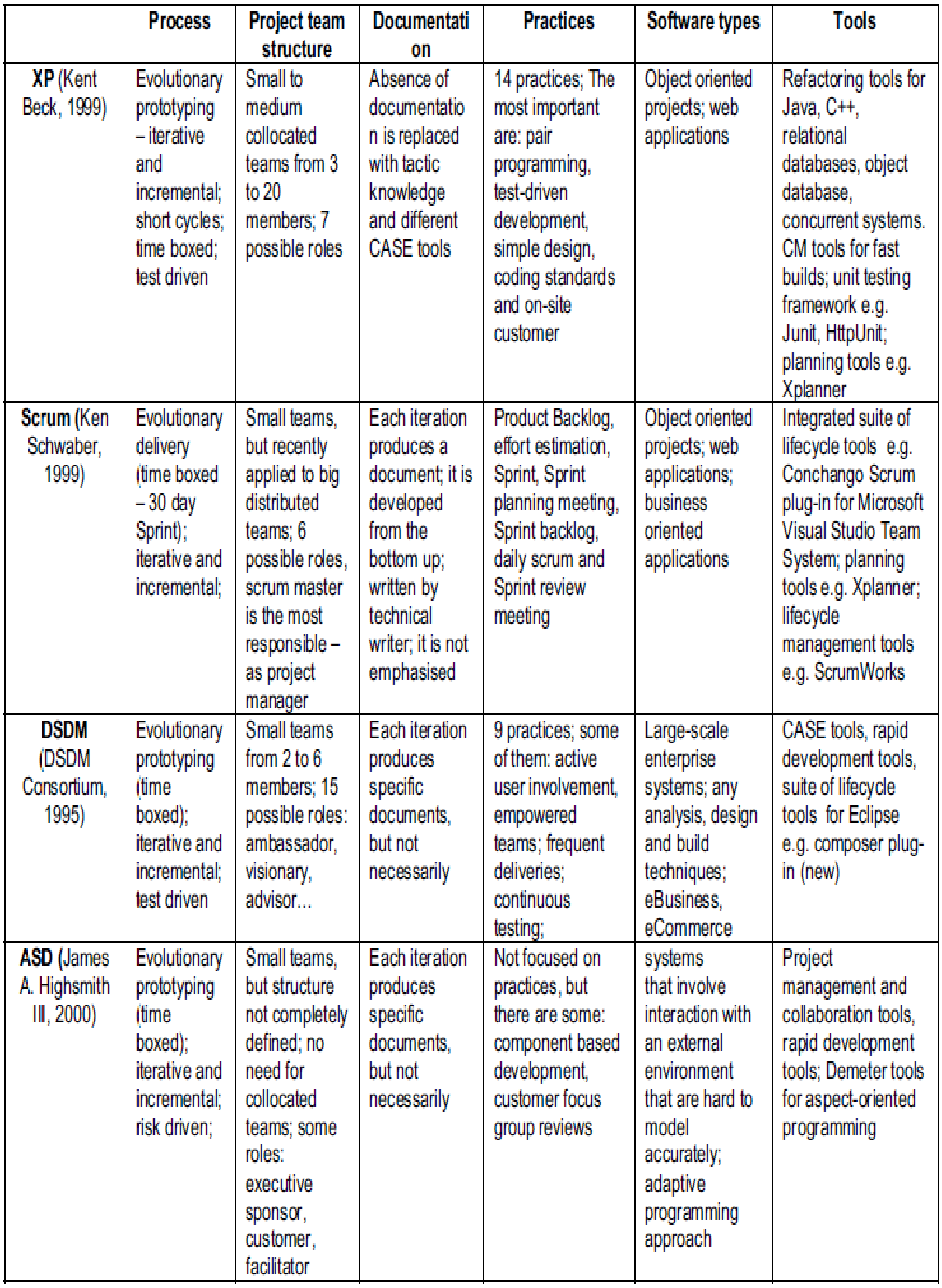
There are many successful efforts, as it is done in [5, 6], to point-out some software development project characteristics and to make comparison among software development methodologies. In Table 1 some of the characteristics are presented, adapted from [5]. In such project characteristics the application of Risk management should be treated up to type of contracts depending of the type of the software development project. The qualitative characteristics for positive and negative impact are listed. However, to evaluate real impact quantification should be made and that way appropriate decisions could be made.

Table 1. Software project development characteristics [5]

Project characteristic	***Positive impact***	***Negative impact***
Scope of works (specifications) - frequently changing	*-to finish the project in complex projects when unknown risks occur*	*-exceeding project budget* *-postponed deadlines* *-stress and discontent for the project team*
Fast innovations in technology and standards	*- new possibilities in design* *- new opportunities in coding*	*-software could become obsolete for the planned time for use* *-more time for research of the project team*
Skilled workforce	*-innovations in project implementation*	*-high cost paid to human recourses*
Project team(s) distributed globally	*- complementary skills* *-possibilities for low cost for human resources*	*- difficulties in communication, monitoring and project control* *-integrating new code is more challenging*
Only one agile methodology used	*-high skilled project team in particular methodology* *-appropriate devoted project roles and responsibilities*	*- un-appropriate in some software development projects* *- un-adequate project roles*

In Table 2 some relevant features of the software development projects are compared for different software development methodologies. Such researches are very useful, but in nowadays fast changing technology and rapid increase of new complex requirements for software development projects, we should be prepared for dynamic changes in comparison results.

Table 2. Comparison of software development methodologies with relevant features [6]



	Process	**Project team structure**	**Documentation**	**Practices**	**Software types**	**Tools**
XP (Kent Beck, 1999)	Evolutionary prototyping – iterative and incremental; short cycles; time boxed; test driven	Small to medium collocated teams from 3 to 20 members; 7 possible roles	Absence of documentation is replaced with tactic knowledge and different CASE tools	14 practices; The most important are: pair programming, test-driven development, simple design, coding standards and on-site customer	Object oriented projects; web applications	Refactoring tools for Java, C++, relational databases, object database, concurrent systems. CM tools for fast builds; unit testing framework e.g. Junit, HttpUnit; planning tools e.g. Xplanner
Scrum (Ken Schwaber, 1999)	Evolutionary delivery (time boxed – 30 day Sprint); iterative and incremental;	Small teams, but recently applied to big distributed teams; 6 possible roles, scrum master is the most responsible – as project manager	Each iteration produces a document; it is developed from the bottom up; written by technical writer; it is not emphasised	Product Backlog, effort estimation, Sprint, Sprint planning meeting, Sprint backlog, daily scrum and Sprint review meeting	Object oriented projects; web applications; business oriented applications	Integrated suite of lifecycle tools e.g. Conchango Scrum plug-in for Microsoft Visual Studio Team System; planning tools e.g. Xplanner; lifecycle management tools e.g. ScrumWorks
DSDM (DSDM Consortium, 1995)	Evolutionary prototyping (time boxed); iterative and incremental; test driven	Small teams from 2 to 6 members; 15 possible roles: ambassador, visionary, advisor…	Each iteration produces specific documents, but not necessarily	9 practices; some of them: active user involvement, empowered teams; frequent deliveries; continuous testing;	Large-scale enterprise systems; any analysis, design and build techniques; eBusiness, eCommerce	CASE tools, rapid development tools, suite of lifecycle tools for Eclipse e.g. composer plug-in (new)
ASD (James A. Highsmith III, 2000)	Evolutionary prototyping (time boxed); iterative and incremental; risk driven;	Small teams, but structure not completely defined; no need for collocated teams; some roles: executive sponsor, customer, facilitator	Each iteration produces specific documents, but not necessarily	Not focused on practices, but there are some: component based development, customer focus group reviews	systems that involve interaction with an external environment that are hard to model accurately; adaptive programming approach	Project management and collaboration tools, rapid development tools; Demeter tools for aspect-oriented programming

5. THE ADVANTAGES OF THE AGILE PROJECT MANAGEMENT

5.1. The usage of the Agile methodologies in Project management

The managers in ICT companies are frequently faced with a question "When should we use Agile methodology?". Some of the answers could be written according to the survey [7] with particular interviews with experienced managers in R. Macedonia that correspond with the wider experience [9], with the benefits of the Agile methodologies in software project development, such as:

- Unlike the traditional methods (waterfall, V, etc), in agile methods very limited planning is required to get started with the project. Agile assumes that the end users' needs are ever changing in a dynamic business and IT world. Changes can be discussed and features can be newly affected or removed based on feedback. This effectively in most of the examples gives the customer the finished system they want or need.
- There is not a clear specification about Scope of works in term of what the final product should look like and in that case new changes are needed to be implemented. The freedom of the agile approach gives to change is very important. However, clients are in position or being able to change the scope of the project. Then, the new changes can be implemented at very little cost because of the frequency of new increments that are produced. Similar situation could appear when the product is intended for an industry with rapidly changing standards.
- ICT company has skilled developers who are adaptable and able to think independently. To implement a new feature they need to lose only some time of the work (of a few days, or even only hours) to roll back and implement it.
- There is a pressure for rapid production being more important than the quality of the product.
- The business partners in a project (developers and stakeholders) also get more freedom of time and options than if the software was developed in a more rigid sequential way (traditional methods). Having options for Change management provide them the ability to leave important decisions until more or better data or even entire hosting programs are available.

The experience above, documented the clear benefits of the entire project using agile methodologies in order to meet the success criteria.

5.2. The most important benefits of applied Agile methodologies in Project management

1. Incorporating continuous integration and daily testing

In Agile development defining and elaborating the requirements are just in time so that the knowledge of the product features is as relevant as possible. In addition, incorporating continuous integration and daily testing into the development process, allows the development team to address the issues while they're still fresh. The completion of the works as follow: developed, tested, integrated, and documented.

2. Keeping customers involved and engaged throughout the project

The customer is always involved since in some software development projects the progress of development needs high visibility and flexibility to change is very important. This implies customer involvement in the development process and satisfaction of the Need for project initiation and completing the project mission. The project team demonstrate working functionalities to customers in every sprint review.

3. Transparency in daily project activities

There is visibility of each step of the project for both parties with the agile instruments such as Sprint meetings.

4. Possibility for faster Return of Investment

Since the agile development is iterative, the features are delivered incrementally so the benefits are realized early while the product is in development process. That way the product could be functional and 'ready to market' product after few iterations. That is great advantage in fast-moving markets. The project team is focusing on a Business value allowing the customer (client) to determine the priority of features in the most valuable order.

5. Risk reduction

One of the most valuable advantage of the agile methodology is risk reduction with the adaptation to the client's needs and preferences through the development process. Agile team commonly uses user comments and experience with business-focused acceptance criteria to define product features. By focusing features on the needs of real customers, each feature incrementally delivers value, not just an IT component. The new changes that are needed can be implemented at very little cost because of the frequency of new increments that are produced. The ongoing activities in project virtually eliminate the chances of absolute project failure or not to a situation for completely project fail.

6. A SURVEY FOR PROJECT TEAM WORKFORCE ENVIRONMENT IN IT SOFTWARE DEVELOPMENT BUSINESS

This research is characterized by the use of different data sources from different industry sectors. The survey has been performed by questionnaire, and it was oriented towards the used methodology in software development projects and the impact of the project teams location and some other project characteristics. The questionnaire was hosted electronically on a website and invitations to participate were distributed via e-mail or personal invitation. It was very easy to fill up with the free software tool Google Forms, enabling additional advantage with different workflow depending on the answer provided. The survey includes 48 project teams with 472 members (in the moment of the survey) in R. Macedonia from different types of industry and business areas in the period of one month [7]. The questionnaire (with structural questions) include all relevant issues reflecting the project habits from a view of the used methodology for software development to their business in a new economic and project environment. The most relevant results from the survey are presented on following figures (with the question) as follow:

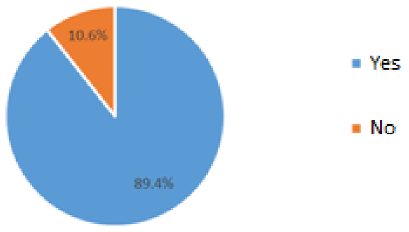


Fig. 1. Do you work in distributed project team?

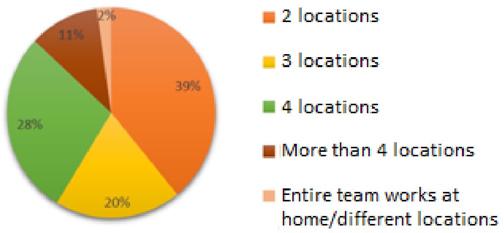


Fig. 2. On how many locations is located project team?

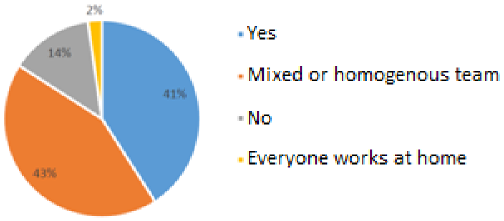


Fig. 3. The presence of Analyst, Designer and Tester on each location

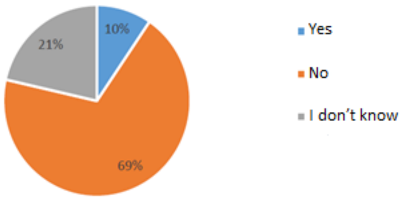


Fig. 4. Is distributed project team more efficient?

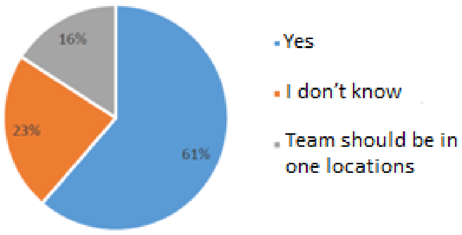


Fig. 5. Do you prefer distributed project team?

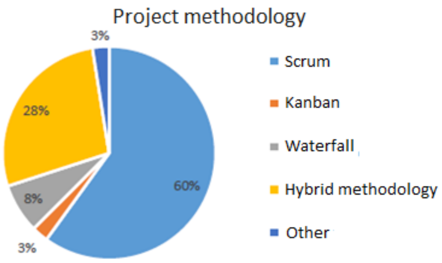


Fig. 6. Which project methodology is used in software development projects?

As we can conclude from the survey [8], the IT companies are mostly running their business in distributed project teams that are located preferably on two locations. One of the specific characteristics is the necessity of presence of the project analyst, designer and tester on each location as the project team members. However, the most interesting conclusion is that the responders are convinced that distributed project team is not more efficient than centered team. It seems more likely that the project team members have difficulties in such working environment implementing and developing software products with agile methodologies in first place Scrum. But, for the IT company' business, worldwide distributed teams are the reality and necessity. That may be the explanation why the responders prefer distributed project teams on Fig. 5. Since agile methodologies are used in over 60% of the responders in questionnaire, preferably Scrum methodology, the IT companies are faced with complex situations working with distributed team that should be overcome and solved respecting project specifics.

7. CONCLUSION

The recent researches in this topic are oriented towards finding the most efficient and appropriate software development model for IT company project success. A comparison of agile methodologies shows some similarities and differences presented in the paper. The differences are in terms of software development techniques and software tools that are mostly used. The advantages are pointed out especially in the new project environment the ICT companies are running their business.

However, modern and innovative software development projects require "writing well" integral and comprehensive documentation to record and patent any original output that might result from the project delivery. In addition, it requires a considerable degree of flexibility since the client (user) specifications are changed very often in the project lifecycle. That is the reason why Scrum methodology is used mostly. From the survey we can also conclude that there is a great percent of the usage of hybrid methodology with combination of two or more methods, adaptive for particular project specifications and requirements. The best matching methodology should be used in the case of experienced project teams and project managers. Otherwise, a combination of methodologies could be introduced. There should be strategic analysis conducted by the company management which and how particular methodology will be used for complex projects. The location of project team and overcoming the difficulties in their functioning should be taken seriously in consideration.

REFERENCES

[1] Vishal Chandra, Comparison between Various Software Development Methodologies. *International Journal of Computer Applications (0975 – 8887) Volume 131 – No.9*, December2015.

[2] Manifesto for Agile Software Development, http://agilemanifesto.org/, Accessed 15 June 2017.

[3] Mike McLaughlin, *SAFe Program Consultant, CSP, PMI-ACP, PMP, What is Agile Methodology.*

[4] Agile Project Management, https://www.versionone.com/agile-project-management/, Accessed 15 June 2017.

[5] M.L. Despa, Comparative study on software development methodologies, *Database Systems Journal*, Vol. V 3/2014.

[6] Krešimir Fertalj. An Overview of Modern Software Development Methodologies // Proceedings of the 19th Central European Conference on Information and Intelligent Systems.

[7] E.M.Zajkovska, V. Fustik. *Survey for project team workforce environment in IT business*, Faculty of El. Engineering, Skopje, 2016.

[8] V. Fustik. Proceedings of the International Conference on Information Technologies (InfoTech-2017) 20-21 September 2017, Bulgaria.

[9] The benefits you get by doing agile project management, 03/21/2017 |In Agile web and app development, (https://apiumhub.com/tech-blog-barcelona/category/agile-web-and-mobile-development/, 9.10.2017).

Information about the author:

Vangel Fustik - full professor at Faculty of Electrical Engineering and Information Technologies, Skopje, Republic of Macedonia. His field of interest is project management, risk management and design and control of power plants and systems.

Manuscript received on 12 October 2017