



JISTEM: Journal of Information Systems
and Technology Management

E-ISSN: 1807-1775

tecsi@usp.br

Universidade de São Paulo
Brasil

Kalb Roses, Luís; Windmöller, Arno; Almeida do Carmo, Eliana
FAVORABILITY CONDITIONS IN THE ADOPTION OF AGILE METHOD PRACTICES
FOR SOFTWARE DEVELOPMENT IN A PUBLIC BANKING
JISTEM: Journal of Information Systems and Technology Management, vol. 13, núm. 3,
septiembre-diciembre, 2016, pp. 439-458
Universidade de São Paulo
São Paulo, Brasil

Available in: <http://www.redalyc.org/articulo.oa?id=203249448005>

- How to cite
- Complete issue
- More information about this article
- Journal's homepage in redalyc.org

redalyc.org

Scientific Information System
Network of Scientific Journals from Latin America, the Caribbean, Spain and Portugal
Non-profit academic project, developed under the open access initiative

FAVORABILITY CONDITIONS IN THE ADOPTION OF AGILE METHOD PRACTICES FOR SOFTWARE DEVELOPMENT IN A PUBLIC BANKING

FAVORABILIDADE NA ADOÇÃO DE PRÁTICAS DE MÉTODOS ÁGEIS NO DESENVOLVIMENTO DE SOFTWARE EM UM BANCO PÚBLICO

Luís Kalb Roses

Financial Ministry, Catholic University of Brasília, Brasília, Distrito Federal, Brazil

Arno Windmüller

Brasília Catholic University, Brasília, Distrito Federal, Brazil

Eliana Almeida do Carmo

University of Brasília, Brasília, Distrito Federal, Brazil

ABSTRACT

The main purpose of this paper is to propose and test a model to assess the degree of conditions favorability in the adoption of agile methods to develop software where traditional methods predominate. In order to achieve this aim, a survey was applied on software developers of a Brazilian public retail bank. Two different statistical techniques were used in order to assess the quantitative data from the closed questions in the survey. The first, exploratory factorial analysis validated the structure of perspectives related to the agile model of the proposed assessment. The second, frequency distribution analysis to categorize the answers. Qualitative data from the survey opened question were analyzed with the technique of qualitative thematic content analysis. As a result, the paper proposes a model to assess the degree of favorability conditions in the adoption of Agile practices within the context of the proposed study.

Keywords: Software; development methodology; Traditional method; Agile method; Agile practices; Software engineering.

Manuscript first received/Recebido em: 21/07/2015 Manuscript accepted/Aprovado em: 26/04/2016
Address for correspondence / Endereço para correspondência:

Luís Kalb Roses, PhD in Business Administration at Federal University of Rio Grande do Sul (Porto Alegre, Brazil) and information technology consultant at Financial Ministry of Brazil (Brasília, Brazil). E-mail: lkroses@gmail.com.

Arno Windmüller, Master in Knowledge and Information Technology Management at Catholic University of Brasília (Brasília, Brazil) and information technology consultant. E-mail: arnowind@yahoo.com.br.

Eliana Almeida do Carmo, Master in Agrobusiness at University of Brasília (Brasília, Brazil). E-mail: eliane.adm@gmail.com.

1 INTRODUCTION

Nowadays, organizations rely heavily on Information Technology (IT), which in a broad sense involves a series of software supported by computational resources for storage, access, processing and use of information. In turn, software is a set of programs executed in computers owned by the organization or interconnected by means of communication networks. The business processes of organizations use software, some of which developed internally by themselves, considering their characteristics and needs (Turban, Leidner, Wetherbe & Mclean, 2010, p. 35).

The time it takes for this development has become a critical success factor for organizations, as the software provides speed for organizational business. The development of software requires the use of adequate methods with the best practices in software engineering. The use of these methods can be the solution or poison in the organizations, as they can bring competitive advantages, but the incorrect use can mean significant business losses (Sanders & Curran, 1994, pp. 34-35).

Furthermore, organizations have required greater agility in their software development (Mohan, Ramesh & Sagumaran, 2010, p. 48) to address market evolution. Therefore, greater involvement and commitment from the user/client segment becomes necessary in this development (Misra, Kumar & Kumar, 2009, p. 1869). However, increased participation of these clients provokes changes in the way software is developed, demanding adaptability (Hanssen & Faegri, 2008, p. 843), dynamism and agility (CAO, 2010, p. 12) in the use of methods for this activity.

In this scenario, a new software method class called agile emerged as an alternative, which was very different from traditional methods (Misra, Kumar & Kumar, 2009, p. 1884; Petersen & Wohlin, 2009, pp. 1480-1481). Agile methods can afford market changes and fluctuations better (Fogelström, Gorschecht, Svahnberg & Olsson, 2010, p. 54), besides demanding greater proximity and participation of the clients in the software development process (Alzoubi, Gill & Al-An, 2016).

Notwithstanding the advantages gained with the agility of these methodologies, we must also consider the situations in which they are best applied. Organic organizations – having reduced or no hierarchical structure – with flexible internal communication lines and projects impacted by frequent changes benefit more from agile methods. However, mechanistic organizations, with structured and hierarchical powers, in their turn, obtain greater advantages with stable projects supported by traditional methods (Nerur, Mahapatra & Mangalaraj, 2005, p. 75). The integration of these two methodological approaches – agile and traditional – for software development can solve the constraints found in each of them (Black, Boca, Bowen & Hinchey, 2009, p. 38; Mohan, Ramesh & Sagumaran, 2010, pp. 48-49; Spundak, 2014; Silva, Soares, Peres & Meira, 2013).

Anyway, even if this integration can optimize software development, there is still the need to identify the conditions where agile methods are best suited, especially in situations where traditional methods predominate, due to the time of existence and the size of the organizations. In this scenario, the following question is interesting for this research: How can we assess favorability in the adoption of agile method practices for software development in organizations, where primarily traditional method practices are used? To answer this question, the objective of the study is to propose and test a model to assess the degree of conditions favorability in the adoption of agile methods to develop software where traditional methods predominate.

2 SOFTWARE DEVELOPMENT METHODS

Software engineering proposes several approaches for the development of software, reflecting how organizations structure themselves for this activity. The methods currently used for these approaches are divided into traditional and agile ones.

2.1 TRADITIONAL METHODS

Traditional methods are characterized by prior, strict and extensive planning as well as focus on documentation, repetitive processes (standardization) and predictability (Fogelström et al., 2010, pp. 55-56). Standardization allows for comparability and repeatability, which strengthens the use of traditional methodologies (Boehm & Turner, 2004, p. 12). When processes are well-defined, documented and there is no question about the execution of each job, organizations have no problems in allocating their professionals.

Among the traditional software development methods the waterfall model can be highlighted. This method is guided towards the conclusion of the work in strictly separated stages and in an irreversible sequence, as if the stages of each project occurred naturally, each one carried out only after the conclusion of the previous one (Boehm & Turner, 2004, p. 10). The initial stage brings a detailed and complete survey of the scope of work, thus a lot of effort is spent on the definition and planning of the project (Fogelström et al., 2010, p. 56). When the planning is completed, every change in requirement demands a new estimate and renegotiation between the client and the vendor (Hansson, Dittrich, Gustafsson & Zarnaket, 2006, p. 1299).

However, there was much dissatisfaction with the incorrect manipulation of the client's requirements and the lack of feedback to the latter (Guntamukkala, Wen & Tarn, 2006). This caused the emergence of more flexible methods such as the spiral development one. For the first time, they recognized the need for development risks and the incremental (partial) delivery of products in this development. It was possible to define the following increments more precisely (Sommerville, 2007) as from the client's assessment of the partial software deliveries and also from the tests.

Large projects were better developed using traditional methods, considering that the processes, plans and documents generated by them supplied the possibility of better communication and coordination between the multiple and large working teams. However, the question remains whether planning does not mean the absence of problems. In the view of Boehm and Turner (2004, p. 13), if the application of plans and processes is very strict, there can be deviation from the objective. Thus, professionals can become excellent document producers instead of obtaining excellence in software development.

2.2 AGILE METHODS

Nowadays, agile methods represent a new approach to software development (Misra, Kumar & Kumar, 2009, p. 1869) and they are used by a large number of IT professionals, at a time when software has increasingly become part of the most diverse organizational processes, services and products offered to society. This requires development speedy, mainly in a competitive scenario (Petersen & Wohlin, 2009, p. 1479). The word agile itself already represents flexibility and speedy in generating answers to business challenges in organizations (Chow & Cao, 2008, p. 962; Nguyen,

2016).

According to research from 2013 in VersionOne (2014), among the existing agile methods, the most used one is called Scrum (55% of the respondent organizations), which was created in 2004 by Schwaber (2004, p. 3). It is important to highlight that with its hybrid versions – Scrum/XP hybrid (11%), personalized Scrum Hybrid (10%) and Scrumban (7%) – it is used in 83% of the surveyed organizations that use agile methods (VersionOne, 2014). Besides Scrum, there is also Extreme Programming - XP and the Dynamic Systems Development Methodology (DSDM), among others (Strode, Huff & Tretiakov, 2009).

In these methods, the scope of the project comes up in the development process and is not planned beforehand (Fogelström et al., 2010, p. 56). To guarantee quick software development in a flexible and adaptive manner with high value, work planning is done without being absolutely sure of the software functionalities (Nerur, Mahapatra & Mangalaraj, 2005, p. 77). Within this context, Hoda, Noble and Marshall (2011, p. 521) note that the collaboration of the client is a vital resource and an important factor for the success of the projects. This is because as there is intense use of implied knowledge, as opposed to documents and respective explicit knowledge, aiming at establishing and prioritizing the requirements of the software to be developed (Boehm & Turner, 2004, p. 17; Iivari & Iivari, 2011, p. 3; Yu & Petter, 2014).

The guiding principles of agile methods are: a) individuals and interactions become more significant in the development of software than processes and tools; b) it is more important to respond to new demands than to follow a plan; c) the visibility of a project can be better achieved by delivering the software code developed; d) the assumption of counterproductive documentation; and e) self-organized teams, in order to have autonomy to determine the best way of performing each software delivery.

Conboy and Morgan (2011, p. 535) criticize agile methods when they suggest the latter have little rigor, an insufficient theoretical basis and little cumulative tradition. There is also a blank as to what constitutes innovation, in a general manner, in the development of software, as well as to what extent these methods in fact make innovation easier. Furthermore, Hoda, Noble and Marshall (2011, p. 521) noticed problems arising from the distance factor and the absence of client commitment, difficulty to use them in large organizations, demands from clients for fixed contracts with vendors and representatives from inefficient clients.

2.3 DIMENSIONS AND THE MODEL OF ADEQUACY AND COEXISTENCE

Empirical evidence indicates that in practice software development teams' use a little of each approach – traditional and agile. In projects executed using agile methods there is planning in the beginning and concern with application architecture. In projects executed using traditional methods there is also iterative development and the possibility of changes in the scope during the projects, notably based on the spiral model. This evidence suggests greater integration of agile and traditional approach practices (Vinekar & Huntley, 2010, p. 89).

In a research with software developers, Falessi, Cantone, Sarcia, Calavaro, Subiaco and D'Amore (2010, pp. 23-25) concluded the practices of agile and traditional methods rather than being opposite or neutral, have complementary characteristics. Agile practices are already broadly widespread in the software industry and although

conflicts persist among the agile and traditional communities, studies indicate they are closer (Boehm & Turner, 2004; Falessi et al., 2010; Hanssen & Faegri, 2008; Mohan, Ramesh & Sagumaran, 2010; Vinekar & Huntley, 2010).

The adaptation of a software development method must be based on compliance with the reality of organizational processes and best practices. One way of doing this is identifying the company's behaviors, practices, mental models, already used methodologies and its area of activity, comparing them to different perspectives associated to traditional and agile dimensions. Adequate choices are made based on this. A dimension can refer to an area from a set of associated processes that define the limits of analysis proposed by the study, based on software engineering theory, which are three: knowledge, administration and processes.

Compliance of the organizations with these dimensions not only helps to choose which methods are most adequate to the situation the organization is in, but also allows us to identify if parts of distinct methods can be combined to create a hybrid method, which is exclusive for a certain organization.

2.3.1 Knowledge Dimension

The formation of perspectives that characterize the knowledge dimension deals with how the information is exhibited to all those involved in the IT area and its projects. Knowledge which is acquired, perceived and managed in the stages of the IT projects is treated by traditional methods as well as agile ones in software development (Dyba & Dingsoyr, 2008, p. 837), according to the perspectives presented in Table 1.

Table 1 – Development Perspectives in the Knowledge Dimension

Perspectives	Traditional Methodologies	Agile Methodologies
Characteristics of IT Professionals	Knowledge from specialists in different distributed teams. Well-defined and documented processes. Little flexibility for distribution of team activities. Loyalty, obedience and few conflicts.	Different exchange roles (generalists) who work in the same environment. Proximity to clients. Committed to the objective and tolerant towards conflicts and dialectics. Problem solution in the team with a project leader.
Knowledge Management	Documental (explicit). Easy substitution of professionals. Communication with a well-defined process, with formal registration of meetings. Communication plans. Formalization of agreements and contracts.	Implied and motivated exploration of participant knowledge. Importance of maintenance of professionals, considering little documentation. Lateral and verbal communication, with records prior to the meetings.

Source: Dyba and Dingsoyr (2008, p. 836), Dyba and Dingsoyr (2009, p. 7), Fogelström et al. (2010, p. 59), Nerur, Mahapatra and Mangalaraj (2005, p. 75), Vinekar and Huntley (2010, p. 88) and Sheffield and Lemétayer (2012), adapted by Authors.

2.3.2 Administration Dimension

Another determining point in the choice of the software development method is the administration style practiced in the IT area. According to Nerur, Mahapatra and Mangalaraj (2005, p. 77), certain administration styles are more adequate for the adoption of agile methods and/or practices. Agile methods are capable of dealing with unpredictability, as they are based on people and creativity, instead of processes (Nerur, Mahapatra & Mangalaraj, 2005, p. 75). In the Administration dimension, Table 2

explains the perspectives of agile and traditional methods.

Table 2 – Development perspectives in the Administration dimension

Perspectives	Traditional Methodologies	Agile Methodologies
Style	Command, control and direction in the running of projects by the manager, interfering in the planning and distribution of work for the teams. Managers participate in the decisions with clients.	Leadership, collaboration and communication. The manager is a facilitator. Decentralization. Support to team leaders aiming at results. Managers act as facilitators. Coordination of work using team components.
Focus	Centered in the process. Approach directed to planning in software development with a standardized process.	Centered on people. Mixed action. Caution with minimum execution processes and support for people executing roles.
Hierarchy	Authority hierarchy. Clear definition of subordination between manager and subordinate. Mechanistic. Bureaucratic with the formalization of repeated functional structures in different projects.	Team network. Consulting without a defined command. Self-organization in the distribution of team member roles. Stimulating collaboration. Organic, flexible and participative. Cooperation activity stimulating social area.

Source: Dyba and Dingsoyr (2008, p. 836), Dyba and Dingsoyr (2009, p. 7), Fogelström et al. (2010, p. 59), Nerur, Mahapatra and Mangalaraj (2005, p. 75), Vinekar and Huntley (2010, p. 88) and Sheffield and Lemétayer (2012), adapted by Authors.

2.3.3 Processes Dimension

Dyba and Dingsoyr (2008, p. 837) state certain scholars believe agile methods will not discard traditional methods; instead, agile and traditional methodologies will have a symbiotic relationship. In this relationship, factors such as the number of people working in a project, application domain, criticality and capacity for innovation will determine what parts of the software development process methods the organization will choose. In the Processes dimension, Table 3 exposes the differences between the method perspectives.

Table 3 – Development perspectives in the processes dimension

Perspectives	Traditional Methodologies	Agile Methodologies
Project cycle	Planning precedes the beginning of the projects. A waterfall, spiral or mixed life cycle. Rigorous planning and control. Delayed diagnosis and heavy tests. Changes in requirements affect new planning.	Tasks executed by team components according to needs. Progressive, emerging, iterative, informal and quick delivery model. Development and review of artefacts along the course of the project. Continuous control and testing. Client reviews scope of project.
Business context	Many potential clients who focus on one product or a family of products. Development centered on the product through several projects.	Focus on client in a specific project. The project is fundamental. Gradual delivery of functionalities. Prioritization of basics in the project.
Problem solution process	Selection of the best way to achieve a certain purpose using formal well-planned activities.	Learning from experimentation and introspection. Constant reformulation of problem and solution. Stimulus towards general view.
Work process	Standard operational rules and procedures. Clear definition of roles and responsibilities.	Fluidity of work processes. Execution of activities without prior artefacts.

Perspectives	Traditional Methodologies	Agile Methodologies
Requirements	Previously specified software built with meticulous and extensive planning. Previous specification and approval of requirements with clients before beginning of projects. Complex and known requirements from market studies and business intelligence.	Software development by small teams. Tolerance towards quick changes. Tests based on feedback. Inseparable conception and development. Continuous tests with evidence artefacts in the homologation. Conception of project coexists with its beginning. Simple and partially known (incremental) requirements.

Source: Dyba and Dingsoyr (2008, p. 836), Dyba and Dingsoyr (2009, p. 7), Fogelström et al. (2010, p. 59), Nerur, Mahapatra and Mangalaraj (2005, p. 75), Vinekar and Huntley (2010, p. 88) and Sheffield and Lemétayer (2012), adapted by Authors.

3 METHODOLOGY

This exploratory research under the survey strategy is based on a mixed method – quantitative and qualitative – in the study of a contemporary phenomenon related to the favorability conditions in the use of agile software development methods.

3.1 LOCATION OF RESEARCH

The location of the research was a Brazilian public financial institution, henceforth known as BANK, which operates with practically all existing banking products and services, and is often innovative when launching new products and services. Most of the software used by this institution in processing its business is developed internally. Its internal clients define and present the demands for software development and they are mostly located in the same city where the IT area is installed.

The target population of the research is approximately two thousand employees who work in the BANK's IT area, as executives, managers and analysts, responsible for the development and maintenance of software used in the Institution. Its software development process is identified in this study as the BANK Software Development Method (BSDM), which foresees the following four traditional forms of software development: complex projects (creation of new software or high risk complex interventions); simple projects (complex, but with a smaller number of interventions and less risk); express service (small interventions, without any significant changes); and incidents (correction or prevention of failures). Thus, the BSDM is inspired in engineering disciplines, regulation norms (NBR, ISO/IEC 12207, 15504[5], 9126, IEEE 829 and ISO/IEC 14764) and software maturity models (MPS-BR, p. ex.). It also supports the maintenance stage.

The software interventions in the maintenance stage can be corrective (defects generated by an incident), preventive (imposed when detecting potential defects), adaptive (changes necessary to accommodate an environment undergoing change) and for enhancements (provide improvements in pre-existing functionalities). It also involves the documentation identified by the artefact term, which is used to: a) transfer the work between the teams responsible for different stages; b) provide guidance for intervention work; c) promote future iterations; d) supply knowledge management; and e) exhibit verifications and validations. Also, it is used to show compliance with legislation and rules, as well as helping to minimize risks and negative impacts.

3.2 RESEARCH MODEL AND INSTRUMENT

By positioning process analysis in the day to day routine of the software development area of an organization, we can broaden the universe of research and leverage the analysis to a more extensive and assertive scope in the adoption of software development practices, either traditional or agile ones. For this purpose, it becomes necessary to trace a conceptual and operational model for research, aiming at analyzing the phenomenon of the empirical point of view (Gil, 2009, p. 43). In this study, the model was composed of variables originating from the practices foreseen in each perspective of the knowledge, administration and software development processes dimensions, aiming at assessing adequacy and coexistence of agile and traditional methods.

These variables are related to the 30 closed questions in the research instrument (see summary concept of each of them in Table 4), formulated so that agreement could suggest a favorable condition in the use of the agile or traditional method in software development. These questions were validated by a software development expert and segmented in the following topics: a) BSDM (questions 1 to 7); b) the addressed projects and demands (questions 8 to 15); c) about the BANK and internal clients (business areas) if its IT area (questions 16 to 23); and d) the professionals and teams in the IT area (24 to 30). Regarding collection of answers, the five-point Likert scale was used to verify the level of agreement (1 – disagree; 2 – partially disagree; 3 – neutral; 4 – partially agree; and 5 – agree).

Table 4 – Items from the closed questions of the research instrument

Question	
Q1	I notice that sometimes the BANK's software development projects are not executed according to the BSDM formally defined by the IT area.
Q2	For urgent projects, the knowledge of what and how the software was developed can be documented partially, as the team analysts will be able to maintain it.
Q3	I know BSDM well.
Q4	I think the BANK's software development process, formalized in the BSDM, is very good, well-standardized and I don't see the need to change it.
Q5	When we receive an urgent demand, our managers agree to develop the software without complying with all the BSDM stages and requirements in effect.
Q6	I assess that my work colleagues know BSDM well.
Q7	If necessary, using the instructions from BSDM I feel capable of guiding any IT professional in how to develop software in the BANK.
Q8	In the BANK, for project success, I consider it necessary to have permanent participation of clients during practically all the software development stages.
Q9	I notice that for the urgent project, after the delivery of the new functionality or new software in production, IT documents what was done.
Q10	All our software development projects, even the urgent ones, must be developed following traditional software engineering forms, i.e., one stage after the other.
Q11	I notice that for the demands that our IT receives, the projects are initiated without the client having completely defined the software requirements to be adapted or developed.

Question	
Q12	I judge that there are few success cases of software development in the BANK when the BSDM rules are not followed.
Q13	I believe that many of the BANK's software development projects could result in more success if the production delivery were made gradually, in an incremental manner.
Q14	I assess that in the BANK we develop our own software projects with good productivity and we deliver within the necessary deadlines, even the urgent ones.
Q15	I think there is no waste of time at the BANK in the initial stages of the software projects, as the clients always manage to say what they want from IT.
Q16	I notice that there is no difficulty in the effective participation of our main clients in the projects due to work schedule problems or physical distance.
Q17	The business performed by the BANK depends very much on software.
Q18	I notice in the projects where I participate that our clients always accept the work schedules we present to them to address their demands.
Q19	I think the BANK is too big and complex to have software projects being developed in it quickly, in an informal manner and with little systematization.
Q20	Our processes and software are frequently audited and we need to observe and follow many regulations and laws to develop them.
Q21	When necessary, the managers of several IT areas agree with the formation of multidisciplinary teams for the development of software projects to address the new business.
Q22	Our IT promotes frequent training and provides clear guidance on the software development processes that we adopt.
Q23	When necessary, IT clients collaborate, show motivation and are available to participate at any moment in all the software project stages.
Q24	I see our IT professionals as well-disposed and motivated to learn and work with new technologies and new work processes.
Q25	I have no doubt that our IT professionals agree and believe that the documentation of software development is very important.
Q26	I believe that the way services are distributed in the different IT teams at the bank for software development is adequate.
Q27	I believe that developing software is an art and that the IT analyst must have freedom to develop it in the way he believes is most appropriate.
Q28	IT analysts can participate in development projects for any BANK software, without needing to have prior knowledge of processes and businesses that the software must automate and control.
Q29	IT analysts are able to take on responsibilities in the software projects to decide on sequencing of the work and to make decisions with the clients without involving IT managers.
Q30	For the success of projects, I believe it is better for IT teams to be specialized in certain technologies, IT processes, the banking market segment or an internal BANK process.

3.3 DATA AND SAMPLE COLLECTION PROCEDURES

Before the research, the instrument was pre-tested with four managers and four developing analysts, leading to improvements related to the reduction of the closed question scale from seven to five points, inclusion of an opened question for comments about the BANK's development process (“[...] you are invited to say something about the BANK's software development process. It can be your opinion about something you believe to be important to reflect upon and eventually change in the BSDM”) and the better understanding of some closed questions. Besides the pre-test, a pilot test was carried out with the instrument, without any new enhancements being observed in it, which was made available to respondents personally using forms in an internet site.

The research sample was non-probabilistic with critical cases, i.e., with key participants being focused on (Freitas et al., 2000, pp. 106-107), and with approximately 2,000 IT professionals involved in software development at the BANK, distributed in two departments composed of 20 structured managements in 64 divisions with 164 teams, each one with approximately 11 employees, on average. The professionals from the division responsible for structuring processes and methods used in the BANK for software development were also invited to answer the questionnaire.

The research had senior management support from the BANK's IT area, which communicated its importance to the population of respondents. The questionnaire was made available for gathering the answers between 11.23.2011 and 12.07.2011, being requested two opportunities for participating in the research. 24 instruments were answered, according to the distribution of the sample on Table 5, which represented approximately 12% of the target public. The proportion between the number of cases and the quantity of variables must exceed five or more for each question (Hair, Anderson, Tatham & Black, 2005, p. 98). In this study, the requirement was addressed by the proportion of questionnaires answered compared to the number of questions closed (30), i.e., in a proportion equal to 8 (240/30). With regard to the opened question in the instrument, 60 respondents expressed their views.

Table 5 – Questionnaire respondent profile

Function performed in IT	Quantity	Quantity of respondents
Executive Manager	20	5
Division Manager	64	41
Team Manager	164	39
Software Analyst	1.752	155
Totals	2.000	240

3.4 QUANTITATIVE AND QUALITATIVE ANALYSIS

The quantitative data arising from 30 closed questions from the research instrument were analyzed through two statistical techniques: exploratory factorial analysis (EFA), a multivariate technique recommended for new research models, until then having no empirical activity; and frequency distribution analysis (FDA), a univariate technique. The first one (EFA) was applied in the validation of the structure of perspectives related to the agile practices of the assessment model proposed, using

SPSS™, version 20, as application support for the statistics calculations, the statistic application Minitab™, version 16, and the Excel™ application. The perspectives presented in the theoretical part of this study served as a basis to qualify the factors identified, following a recommendation regarding the last stage of the EFA (factor nomination). The second statistical technique (FDA) was used in the frequency analysis of the answers to each question, which served as a calculation basis for the degree of favorability to the adoption of agile practices at the BANK.

The qualitative data arising from the opened question in the research instrument was analyzed using a thematic qualitative content analysis technique (Bardin, 1977), which showed the perspectives of agile practices foreseen in the research model. In this manner, these practices constituted a prior category system to the application of the box procedure foreseen in this type of analysis. This supported the results of the frequency distribution analysis on the answers to the closed questions, and, consequently, in the interpretation of the Degree of Favorability (DF), as the categorization of content allows for inferences and interpretations (Bardin, 1977, p. 101).

3.5 DEGREE OF FAVORABILITY - DF

A method was formulated to obtain a score between 0 and 10, aiming to identify the DF in the use of agile method practices in the development of software at the BANK, as from its segmentation per question, dimensions and respective perspectives. Each of the 30 closed questions, excluding those that did not address the presumed minimum statistical values, had their answers totaled per degree of agreement and calculation of their percentage distribution from 0 to 100.

The questions were formulated in a way to make agreement regarding the use of the agile or traditional method. This is why in the questions where agreement was associated to the use of the agile method (questions 1, 2, 5, 7, 8, 11, 13, 16, 17, 21, 22, 23, 24, 27, 28, 29 and 30), agreement was multiplied by a weight of 10, partial agreement was multiplied by a weight of 7.5, neutrality by 5, partial disagreement by 2.5 and disagreement by zero. In the other questions, where agreement was associated to the use of the traditional method (questions 3, 4, 6, 9, 10, 12, 14, 15, 18, 19, 20, 25 and 26), in contrast, the attribution of these weights was inverted (zero for agreement, 2.5 for partial agreement, 5 for neutrality, 7.5 for partial disagreement and 10 for disagreement).

Thus, the formula to calculate the DF in the group of questions with agreement associated to the agile method is given by $D_1 = ((X_2 * 2,5) + (Y * 5) + (Z_1 * 7,5) + (Z_2 * 10)) / 100$, where: D_1 = Degree of agreement per question; X_2 = Percentage of answers with partial disagreement in the question; Y = Percentage of neutral answers in the question; Z_1 = Percentage of answers with partial agreement in the question; and Z_2 = Percentage of answers with agreement in the question. The percentage regarding answers with disagreement (1 point on the scale) will not be used to calculate the DF in terms of favorability of the agile method.

The formula for the calculation of DF in the group of questions with disagreement associated to the agile method and, consequently, associated to agreement with the traditional method, is given by $D_2 = ((X_1 * 10) + (X_2 * 7,5) + (Y * 5) + (Z_1 * 2,5)) / 100$, where: D_2 = Degree of agreement per question; X_1 = Percentage of answers with disagreement in the question; X_2 = Percentage of answers with partial disagreement

in the question; Y = Percentage of neutral answers in the question; and Z_1 = Percentage of answers with partial agreement in the question. The percentage regarding the answers with agreement (5 points on the scale) will not be used to calculate the DF in terms of favorability of the agile method.

After calculating the DF per question, using the groups of questions in the research instrument – or common factors resulting from the EFA or perspectives –, where different correlated questions were grouped, the DF values were also calculated per perspective, per dimension and in general. In order to verify the DF of each perspective, the simple arithmetic average of the DF of questions associated to it was calculated. In the same manner, the verification of DF per dimension was calculated. In the verification of the general DF, the simple arithmetic average of the DF of all dimensions was calculated.

Boehm and Turner (2004, p. 150) proposed a model similar to the assessment, where they suggest that organizations use a set of dimensions as a measure to choose the most adequate methodological approach to conduct each software development project. In the model used in this study, with the DF being verified according to the use of agile method practices, we intend to assess if favorable conditions are perceived for the use of both agile and traditional practices in the organization studied.

4 RESULTS AND ANALYSIS

The results and analysis of the data were segmented into stages of EFA and DF analyses.

4.1 EXPLORATORY FACTORIAL ANALYSIS

In order to validate the EFA, the assumptions of normality, homoscedasticity and linearity were assessed. Normality was not observed in the sample. Homogenous behavior was verified in the distribution of the set of questions used, conferring homoscedasticity. The scatterplot graphs between both variables proved the linearity. As the objective of this study was to find standards among the groups of questions, the application of EFA is justified, despite the normality assumption not having been achieved.

To transform the data from this study, an electronic Excel™ spreadsheet was used together with the SPSS™ application. Cronbach's Alpha was calculated and the EFA performed. The result was better using the potentiation technique.

Five rounds of EFA in the SPSS™ were necessary to match the questions that did not achieve the MSA, smaller than 0.5. As the questions with smaller indexes were removed, the index came closer to the accepted level. The questions excluded were: Q15, Q16, Q19, Q21 and Q30. As a result, 10 factors with commonalities bigger than 0.5, MSA's also higher than 0.5 and Cronbach's Alpha equal to 0.603. The results were obtained using the Rotated Factor Matrix, as this allows for better interpretation of the factors and the correlation structure is simple.

Hair et al. (2005, pp. 90, 101-102) highlights that in social science studies the recommendation is to select a quantity of factors that account for at least 60% of variance. In this study, the total variance explained for the ten factors represented 63.24%.

To validate the adequacy of the data and the size of the sample, the tests used were the Kaiser-Meyer-Olkin (KMO) and Bartlett's Test of Sphericity (BTS). The

objective was to assess if the EFA results are valid for the questions chosen. The KMO and BTS tests are considered reasonable, as KMO = 0.646 and BTS presented Chi-squared equal to 917.470 and 300 degrees of freedom, resulting in statistical significance < 0.000 .

An orthogonal rotation was used as it is simpler and more comprehensive (HAIR et al., 2005, p. 107). A cutoff point of 0.40 was adopted for each question. As to the correlation standard between the variables, the Correlation Matrix must show most of the coefficients with a value above 0.30 (Hair et al., 2005, p. 98) to apply the EFA, according to the suggestion of Hair et al. (2005, p. 92). All the commonalities are higher than 0.5, which means that the issues are statistically significant to explain the first factor.

4.2 ANALYSIS OF DEGREE OF FAVORABILITY

As a result of the FDA statistical technique application on the answers of closed questions and of the two formulas to calculate the DF related to them, Table 6 presents the DF of practices, perspectives and dimensions, as well as the general DF (4.39), which did not indicate ample favorability to the adoption of agile practices in the BANK's software development. This took place mainly due to practices regarding Knowledge dimension perspectives, although part of the Administration dimension practices are seen as favorable for adoption.

Table 6 – Degree of favorability in the agile method

Dimensions	Perspectives	Methodological Practices	DF			
			Pract.	Persp.	Dim.	Gen.
Administration	Style	Use of Methodology	7.73	7.75	6,08	
		Methodological Flexibility	7.77			
	Focus	Incentive for Learning	6.01	5.59		
		Motivation to Learn	6.36			
		Belief in Documentation	4.41			
	Hierarchy	Emphasis on Implied Knowledge	4.28	4.89		
		Autonomy for Planning	4.56			
		Subordination to Rules	5.82			
Knowledge	Characteristic of IT Professionals	Guidance for Work	2.41	3.06	2.46	4.39
		Professional Rotation	2.88			
		Work Control	3.89			
	Knowledge Management	Software Documentation	1.04	1.86		
		Ad Hoc Development	2.68			
Processes	Project Cycle	Methodology Revision	5.34	4.49	4.64	
		Projects with Vagueness	5.93			
	Business Context	Business Automation	3.80	2.71		
		Participation of Clients in Projects	1.63			

Dimensions	Perspectives	Methodological Practices	DF			
			Pract.	Persp.	Dim.	Gen.
	Problem Solution	Incremental Development	6.50	5.89		
		Compliance with Deadlines	5.27			
	Work Process	Methodological Culture	3.39	4.69		
		Team work	5.31			
		Promoting Collaboration	5.36			
	Requirements	Flexibility to Review the Scope	4.14	5.43		
		Rigor in Definition of Requirements	6.22			
		Multidisciplinary Teams	5.95			

4.2.1 The Administration Dimension and its three perspectives

This dimension presented reasonable favorability in the agile methodologies (DF 6.08), strongly influenced by the favorability of its Style perspective (DF = 7,75). This arises from the fact that the BANK is adopting other practices not formally foreseen in the BSDM, including agile method practices, or it is necessary to review the current method to formalize the required practices or the ones that are already in use in certain types of projects. According to Bajec et al. (2007, p. 345), even with formal software development methods in the organizations, they can be underutilized. Besides, respondents show flexibility in the development of urgent software.

According to Fogelström et al. (2010, p. 54), Agile methods support market changes better as they focus on quick solutions with less of a view to the future (Mohan, Ramesh & Sagumaran, 2010, p. 49); while Qumer and Henderson (2008a, p. 281) state they are flexible, quick (iterative with product development in small deployments), light (focusing on time reduction, cost and quality) and have a response capacity.

With regard to the Focus perspective, favorability of the agile method is not expressive (DF = 5,59). The training culture is a positive factor in the adoption of a new approach, as software developers are more liable to accept agile methods if they have adequate knowledge, according to that foreseen by Chan and Thong (2008, p. 811). Furthermore, respondents perceive IT professionals as being motivated and there is an atmosphere for the adoption of new practices. Several studies highlight human factors as important for project success, such as competence and learning of the professionals (Misra, Kumar & Kumar, 2009, pp. 1881-1883; Tolfo, Waslawick, Ferreira & Forcellini, 2009, pp. 1-3).

Nevertheless, there is no consensus on the documentation necessary for the projects. Traditional methods require more extensive documentation; while agile methods focus on quick solutions with less of a view to the future (Hanssen & Faegri, 2008, p. 843; Mohan, Ramesh & Sagumaran, 2010, p. 49), minimizing the need for planning (Black et al., 2009, p. 40). The capacity of the team to use and maintain the implied knowledge is fundamental, considering that documentation is minimally necessary (Boehm & Turner, 2004, p. 10, 17; Iivari & Iivari, 2011, p. 3).

The Hierarchy perspective (DF = 4.89) does not indicate favorability towards agile methods. There is more disagreement than agreement as to the non-documentation

of software developed in urgent projects. Documented plans are practices considered to be fundamental for software development success (Fogelström et al., 2010, pp. 55-56; Misra, Kumar & Kumar, 2009, p. 1872). On the other hand, agile methods, which are adequate for urgent projects, suggest the intense use of implied knowledge, as opposed to documents (explicit knowledge), so that the software requirements to be developed can be established, prioritized and verified (Boehm & Turner, 2004, p. 17; Iivari & Iivari, 2011, p. 3).

Besides, reasonable comfort of the clients with regard to the deadline established for projects with IT professionals is not observed in their perception. Organizations do not accept more and longer waiting periods to receive new software or updates (Bassi Filho, 2008, p. 3; Mohan, Ramesh & Sagumaran, 2010, p. 248; Pfleeger, 2001, p. 56). Furthermore, the establishment of deadlines, in the agile perspective, presuppose joint participation of the business and IT areas. Finally, the respondents do not highlight the need to comply with the rules and laws as a requirement for audits to justify them. Agile methods are characterized by improvisations (Mohan, Ramesh & Sagumaram, 2010, p. 49), while the traditional ones use formal techniques and several well-defined processes to guarantee quality (Cao, 2010, p. 18).

4.2.2 The Knowledge Dimension and its two perspectives

This dimension presented low favorability towards the agile methods (DF = 2.49), due to its perspective Professionals' Characteristics (DF = 3.06) and Knowledge Management (DF = 1.86). In the first, a certain aversion towards freedom and autonomy of the IT professional is observed, which stimulates the use of traditional methods and rejects agile methods, as in these analysts have freedom to act with creativity in the development of software (Beck, Beedle, Bennekum, Cockburn, Cunningham, Fowler, Grenning, Highsmith, Hunt, Jeffries, Kern, Marick, Martin, Mellor, Schwaber, Sutherland & Thomas, 2001, p. 1).

The respondents also disagreed with the possibility of the IT professionals, even with good technical knowledge, to participate in different projects at any time, also suggesting specialization of these professionals in the rules of the business. Moreover, respondents do not agree with the inexistence of managerial control over the IT professional's activities, suggesting that self-management advocated by agile methods is not applicable (Moe, Dingsoyr & Dyba, 2010, p. 480; Nerur & Balijepally, 2007).

Regarding the Knowledge Management perspective, respondents highlight the document due for urgent projects that are implanted, which grants an explicit characteristic in this management, typical of the traditional methods. In the agile approach, the documentation for software development is not considered to be very useful (Boehm & Turner, 2004, p. 17; Iivari & Iivari, 2011, p. 3). There is also the perception that the success of the projects is related to compliance with the practices established in the BSDM, which is traditional. After all, this non-compliance is not a synonym of the use of agile practices. Besides, the use of traditional practices does not mean that a project cannot have flexibility (Vinekar & Huntley, 2010, p. 89).

4.2.3 The Processes Dimension and its five perspectives

This dimension presented itself as slightly unfavorable towards agile methods (DF = 4.64) and as a result slightly favorable towards traditional methods, due to its five perspectives – Project Cycle (DF = 4.49), Business Context (DF = 2.71), Problem Solution (DF = 5.89), Work Process (DF = 4.69), Requirements (DF = 5.43). About the first of them (Project Cycle), the respondents are neutral on the need to change the development process foreseen in the BSDM, although some have expressed themselves in the opened questions about the need to change this process.

The search for a consensus on the software development method has occurred since 1968 (Eischen, 2002, pp. 36-38), as the technologies and processes evolve towards making quick, precise and reliable information available (Bassi Filho, 2008, p. 3), an example of which is the most recent emergence of agile methods (Chow & Cao, 2008, p. 961). Within this context, aligned to these methods is the perception of respondents that normally the BANK projects start without a complete definition of the demanding client requirements, although the BSDM has a traditional bias.

The traditional software development methods are used a lot and are more applicable where the project requirements are fixed and defined from the beginning (Qumer & Henderson, 2008a, p. 289; Williams & Cockburn, 2003, p. 39). In the traditional methods, requirement specification occurs as the projects are developed (Boehm & Turner, 2004, p. 17; Iivari & Iivari, 2011, p. 3). Thus, traditional methods can benefit from the adoption of agile practices in situations that demand greater flexibility, when the dynamics of needs imposes instability in the requirements (Mohan, Ramesh & Sagumaran, 2010, p. 49). This situation can be favorable in the BSDM, in the face of normal incompleteness of the requirements from the beginning of the projects.

In the perspective of the Business Context, the respondents did not notice the high dependence the BANK's business has on software, suggesting there are other aspects involved, such as maybe business conceptions and respective rules that are automated using software. The respondents also do not perceive collaboration, motivation and involvement of the clients in the software projects they demand, which generates conflict in what is expected of the use of agile methods, when the involvement of a client is a fundamental practice. Clients accustomed to the traditional forms of software development can be resistant to change and involvement in the collaboration for agile projects (Hoda, Noble & Marshall, 2011, pp. 521, 525-526). Finally, within the Business Context, favorability is not observed in the adoption of agile methods.

Regarding the Problem Solution perspective, respondents agree that the incremental delivery of solutions (software functionalities) in a project would potentialize its success, something foreseen in the agile approach, which suggests appropriateness of practice integration with the ones from the traditional method (Vinekar & Huntley, 2010, p. 89). After all, "Dividing a large project into delivery stages allows for the application of the "spiral" model, preventing any urgent interruptions from making a strong impact on the negotiation process", according to the answer given by one of the respondents in the opened question.

This is probably why there is a lack of consensus among respondents regarding productivity and achieving the deadlines defined in the projects. Pfleeger (2001, p. 56) indicates that delays are not tolerated anymore, within a context of continuous demand

for new software in shorter terms (Bassi Filho, 2008, p. 3). The traditional methods focus more on the organizations' strategic objectives and on long-term solutions, while agile methods benefit more from tactical ambitions that address more needs for quick solutions on a smaller scale, thus solving different problems (Hanssen & Faegri, 2008, p. 845).

A good culture in the use of the BSDM is observed under the Work Process perspective. According to one respondent of an opened question, although it is "based on the best existing practices and rules, it has a very conceptual and academic bias, very distant from the organizational culture [...]", which would not be adequate to the level of urgency in the demanding business areas. However, there is no consensus among respondents about good knowledge of the BSDM in the working team. A good software development process will not be sufficient if the team has a low working capacity and knowledge, while a good team can work with a process that is not so good (Hansson et al, 2006, p. 1298). In this sense, BSDM's capacity to facilitate learning on its use among the users is relative, according to the respondents, which suggests the opportunity for revision. Finally, the more mature the software development method, the easier it is to adopt and disseminate (Chan & Thong, 2008, p. 811).

Finally, in the Requirements perspective, the perception of respondents is that client participation in all the stages of the Project is not necessary, which suggests low favorability towards this agile practice. After all, in the agile approach to deliver software in a shorter timeframe, clients must be permanently involved during the whole project (Cao, 2010, p. 12; Hanssen & Faegri, 2008, p. 843; Misra, Kumar & Kumar, 2009, p. 1869). On the other hand, the respondents do not perceive that it is necessary to have a waterfall development model, i.e., sequential and mandatory in all the stages (survey and analysis of requirements; development; tests; homologation; and implantation) of the project (Boehm & Turner, 2004, p. 10), which suggests favorability towards the agile approach, which does not require this obligation. Furthermore, the multidisciplinary interaction of different teams is not perceived as adequate, an aspect that also favors the agile approach, where the multidisciplinary characteristic is in the project team.

5 FINAL REMARKS

The main objective of this study was to propose and test a model to assess the degree of conditions favorability in the adoption of agile methods to develop software where traditional methods predominate. The model was tested in the software development sphere of a Brazilian public banking institution, not only with regard to its factorial structure, but also regarding the degree of favorability in the adoption of agile practices, segmented into perspectives and dimensions. In general, the model indicates a slight favorability towards the adoption of agile practices in the institution.

As a contribution to this study, we notice in theoretical terms it presents a research model for the assessment of the opportunity for adoption and coexistence of agile and traditional method practices in software development. This allows extension of research on the opportunity and condition of use of such methods. In the practical aspect, this study contributes to the model that serves as an instrument for software

development managers to better assess the conditions for the coexistence of agile and traditional practices, due to the institutional environment of their organizations.

The constraint in this study is related to the fact that it researched in a single organization, which prevents it from generalizing the results externally, nevertheless the model can be assessed and adjusted to other organizational contexts, thus allowing the use of the theory contained in it. Another constraint is in the quantity of perspectives adopted, that ends up restricting the operability of the corresponding variables, in face of the impact in terms of extending the research instrument. Finally, a third constraint is in the conceptual association of the model perspectives with the dimensions foreseen in theory, as this was not the object of the exploratory factorial analysis.

Considering these limitations, some future studies are foreseen. The first, in the sense of contemplating the application of the model in organizations with different activity contexts, which suggests different cultures for software development and practices that can be adopted in this development. Another study to reduce the quantity of perspectives, aiming at simplification of the model. Maybe this simplification can happen in the dimension of the perspectives themselves. A last study can validate the practices considered to be favorable in the application of the model developed, in a second moment of research in the same organization.

REFERENCES

- Alzoubi, Y. I., Gill, A. Q., & Al-An, A. (2016). Empirical Studies of Geographically Distributed Agile Development Communication Challenges: A Systematic Review. *Information & Management*, 53(1), 22-37.
- Bajec, M. & Krisper, M. (2007). Practice-driven approach for creating project-specific software development methods. In: *Information and Software Technology*, 49(1), 345–365.
- Bardin, L. (1977). *Análise de conteúdo*. (L. A. Reto & A. Pinheiro, Trad.). São Paulo: Edições 70, Livraria Martins Fontes (Original work published in 1977).
- Bassi Filho, D. L. (2008). *Experiências com desenvolvimento ágil*. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Ciência da Computação, USP, São Paulo.
- Beck, K., Beedle, M., Bennekum, A. V., Cockburn, Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). Manifesto for agile software development. Available at: <<http://www.agilemanifesto.org/>>. Accessed on: July 10, 2010.
- Black, S., Boca, P., Bowen, J., & Hinchey, M. (2009). Formal versus agile: survival of the fittest? In: *IEEE Computer Society*, 42(9), 37-45.
- Boehm, B., & Turner, R. (2004). *Balancing agility and discipline*. Boston, MA, USA: Addison – Wesley.
- Cao, L. (2010). Dynamic capability for trustworthy software development. In: *Journal of Software Maintenance and Evolution: Research and Practice*, 23(5), 1-14.
- Chan, F. K.Y.; & Thong, J. Y. L. (2009). Acceptance of agile methodologies: A critical review and conceptual framework. In: *Elsevier Journal – Decision Support Systems*, 46(4), 803-814.
- Chow, T., & Cao, D. (2008). A survey study of critical success factors in agile software projects. In: *The Journal of Systems and Software*, 81(6), 961-971.

- Conboy, K., & Morgan, L. (2011). Beyond the customer: Opening the agile systems development process. In: *Information and Software Technology*, 53(5), 535-542.
- Dyba, T., & Dingsoyr, T. (2008). Empirical studies of agile software development: A systematic review In: *Information and Software Technology*, 50(9), 833–859.
- _____. (2009). What do we know about agile software development. In: *IEEE Software*, 25(5) 6-9, set./out.
- Eischen, K. (2002). Software Development: An Outsider's view. In: *IEEE Computer*, 35(5), 36 – 44.
- Falessi, D., Cantone, G., Sarcia, S., Calavaro, G., Subiaco, P., & D'amore C. (2010). Peaceful coexistence: Agile developer perspectives on software architecture. In: *IEEE Software*, 27(2), 23-25.
- Fogelström, N. D., Gorschekt, T., Svahnberg, M., & Olsson, P. (2010). The impact of agile principles on market-driven software product development. In: *Journal of Software maintenance and Evolution: Research and Practice*, 22(1), 53-80.
- Freitas, H., Oliveira, M., Saccol A. Z., Moscarola, J. (2000). O método de pesquisa survey. In: *Revista de Administração*, 35(3), 105 – 112.
- Gil, A. C. (2009). Como elaborar projetos de pesquisa. São Paulo: Atlas.
- Guntamukkala, V., Wen, H. J., & Tarn, J. M. (2006). An empirical study of selecting software development life cycle models. *Human System Management*, 25(4), 265-278.
- Hair Jr, J. F., Anderson, R. E., Tatham, R. L., & Black, W. C. (2005). *Análise multivariada de dados*. 5. e. Porto Alegre: Bookman.
- Hanssen, G. K., & Faegri, T. E. (2008). Process fusion: An industrial case study on agile software product line engineering. In: *The Journal of Systems and Software*, 81(6), 843–854.
- Hansson, C., Dittrich, Y., Gustafsson, B., & Zarnaket, S. (2006). How agile are industrial software development practices? In: *The Journal of Systems and Software*, 79(2), 1295–1311.
- Hoda, R., Noble, J., & Marshall, S. (2011). The impact of inadequate customer collaboration of self-organizing agile teams. In: *Information and Software Technology*, 53(5), 521-534.
- Iivari, J., & Iivari, N. (2011). The Relationship between organizational culture and the deployment of agile methods. In: *Information and Software Technology*, 53(5), 509-520.
- Misra, S. C., Kumar, V., & Kumar. U. (2009). Identifying some important success factors in adopting agile software development practices. In: *The Journal of Systems and Software*, 82(2), 1869-1890.
- Moe, N. B., Dingsoyr, T., & Dyba, T. (2010). A teamwork model for understanding an agile team: A case study of a Scrum project. In: *ScienceDirect – Information and Software Technology*, 52(5), 480-491.
- Mohan, K., Ramesh, B., & Sagumaran, V. (2010). Integrating software product line engineering and agile development. In: *IEEE Software*, 27(3), 48-55.
- Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. In: *Communications of the ACM*, 48(5), 72-78.
- Nerur, S., & Balijepally, V. (2007). Theoretical reflections on agile development methodologies. In: *Communications of the ACM*, 50(3), 79-83.

- Nguyen, D. S. (2016). Success Factors for Building and Managing High Performance Agile Software Development Teams. *International Journal of Computer*, 20(1), 51-82.
- Petersen, K., & Wohlin, C. (2009). A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. In: *The Journal of Systems and Software*, 82(1), 1479–1490.
- Pfleeger, S. L. (2001). *Software Engineering: theory and practice*. NJ, USA: Prentice-Halol, Inc. Upper Saddle River.
- Qumer, A., & Henderson, B. (2008a). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. In: *Information and Software Technology*, 50(4), 280–295.
- _____. (2008b). A framework to support the evaluation, adoption and improvement or agile methods in practice. In: *The Journal of Systems and Software*, 81(12), 1899-1919.
- Sanders, J., & Curran, E. (1994). *Software Quality*. Harlow, England: Addison-Wesley.
- Schwaber, K. (2004). *Agile Project Management with Scrum*. Washington: Microsoft Press. 192 p.
- Sheffield, J., & Lemétayer, J. (2012). Factors associated with the software development agility of successful projects. *International Journal of Project Management*, 31(3), 459-472.
- Silva, F. S., Soares, F. S. F, Peres, A. L., & Meira, S. (2013). Using CMMI Together with Agile Software Development: a Systematic Review. *Information and Software Technology*, 58(2), 20-43.
- Sommerville, I. (2007). *Engenharia de software*. 8. e. São Paulo: Pearson Addison – Wesley, 552 p.
- Spundak, M. (2014). Mixed Agile/Traditional Project Management Methodology - Reality or Illusion? *Procedia - Social and Behavioral Science*, 119, 939-948.
- Strode, D. E., Huff, S. L., & Tretiakov, A. (2009). A. The impact of organizational culture on agile method use. In: *42nd International Conference on system sciences. Proceedings...* Hawaii: IEEE, 1-9.
- Tolfo, C., Wazlawick, R. S., Ferreira, M. G. G., & Forcellini, F. (2009). A. agile methods and organizational culture: Reflections about cultural levels. In: *Software Process Improvement and Practice -Wiley InterScience*, 23(8), 1-19.
- Turban, E., Leidner, D. E., Wetherbe, J. C., & Mclean, E. (2010). *Tecnologia da informação para gestão: Transformando os negócios na economia digital*. Porto Alegre: Bookman.
- Versionone. (2014). State of agile development. 2014. Available at: <http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf>. Accessed on: April 9, 2014.
- Vinekar, V., & Huntley, C. L. (2010). Agility versus maturity: Is There Really a Trade-Off? In: *IEEE Computer Society*, 43(5), 87-89.
- Williams, L., & Cockburn, A. (2003). Agile software development: It's About Feedback and Change. In: *IEEE Computer Society*, 36(6), 39-43.
- Yu, X., & Petter, S. (2014). Understanding Agile Software Development Practices Using Shared Mental Models Theory. *Information and Software Technology*, 56(8), 911-921.