FACTORS THAT AFFECT THE REQUIREMENTS ADHERENCE TO BUSINESS IN AGILE PROJECTS: AN INDUSTRIAL CROSS-CASE ANALYSIS

Helena Bastos[1], Alexandre Vasconcelos[1], Wylliams Santos[2] and Juliana Dantas[3]

[1]Universidade Federal de Pernambuco, Recife, Brazil
[2] Universidade de Pernambuco, Recife, Brazil
[3] Instituto Federal de Educação Ciência e Tecnologia da Paraíba, Brazil

ABSTRACT

Agile Software Development has advanced in the latest years, but research evidence indicates limitations related to its usage along with Requirements Engineering. One of the reasons for failures in agile projects is the nonconformity to the needs of business processes in companies. This study conducted a cross-case analysis in seven companies to investigate Requirements Engineering in agile projects. Documentation, observation, and interviews were triangulated, analyzed and synthesized by applying techniques of thematic analysis. The aim was identifying factors that affect the requirements adherence to business. The customer business knowledge by the team and the customer availability during elicitation and validation of software requirements are essential to the requirements adherence to business in agile projects. That way, the developed systems (both Information Systems and Intelligent Systems) can better meet the needs of the organization's business processes.

KEYWORDS

Requirements Engineering, Agile Methods, Cross-Case Analysis, Business Process Modelling, Artificial Intelligence

1. INTRODUCTION

In recent years, there has been a growth in the use of Agile Software Development (ASD) [1]. However, even in projects that use agile methods, studies still show many project failures. According to [2], five out of eight main factors of project failures are related to requirements. Although Requirement Engineering (RE) is known as a critical success factor for systems development in the context of ASD, it is also pointed as one of the challenges that have compromised the adoption of agile methods in the software development [3]. Some empirical studies conducted in the industry pointed out to some problems related to RE in the ASD, such as low customer availability and lack of business knowledge by software engineers [4]. Hence, one of the reasons for the failures in the context of ASD is the requirements that do not meet the needs of the business processes [5]. Business Process Modelling (BPM) can contribute to the development of systems so that software requirements reflect business needs, and above all, for the developed systems to be guided by business, not merely by technology [6]. BPM helps requirements analysts understanding the complexity of the business and its integrations [7]. There is a tendency of companies to use BPM as a tool to identify, extract and model software requirements, thus optimizing the RE process [5]. The Business Process Modelling Notation (BPMN) has effectively become the most used standard for processes modelling, currently relying on many tools compatible with its specifications.

DOI: 10.5121/ijsea.2020.11306

In this paper, our purpose is to investigate RE in different project contexts that use agile practices and to identify the factors that must be considered to obtain requirements adherence to the business. Besides this introductory section, this paper is structured as follows: Section 2 introduces the concepts Agile Requirements Engineering and the challenges that affect RE in ASD; Section 3 describes the research methods used; Section 4 presents the design and analysis of seven case studies; Section 5 describes the results of the cross-case analysis of the case studies; Section 6 compares the results of the cross-case synthesis to the findings of literature. This section also presents a discussion on the implications of this study for research and practice; Finally, Section 7 presents the conclusions and open issues for future work.

2. THEORETICAL BACKGROUND

2.1. Agile Requirements Engineering

Since the publication of the Agile Manifesto [8], the adoption of agile practices has been growing. A survey conducted by [1], involving about 4000 people, shows that 45% of the respondents use agile methods in most projects. It has brought some questions to the community on how to deal with RE, in such a flexible and dynamic way. More than a decade has gone since the Agile Manifesto was published and research evidence point out challenges in adopting agile methods regarding the activities of the RE [3]. Studies highlight some implications of this manifesto for agile RE ([9], [10]): software is developed incrementally with requirements being detailed and prioritized just before every iteration, and requirements documentation is reduced in favour of face-to-face communication and prototyping. An advantage of the prototyping technique is that it is simple and do not require additional costs to obtain accurate and complete requirements [11]. In this context, some systematic literature reviews conducted point out several problems related to RE in ASD, such as low customer availability, poor quality of software requirement, among others [3], [10], [12], [13]).

2.2. Challenges that affect RE in ASD

In this research, we have investigated several systematic literature reviews and systematic mapping studies regarding RE in ASD. Table 1 summarizes the challenges which affect the activities of RE in ASD, identified in 15 investigated papers numbered as follows: [3], [10], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24] and [25].

Table 1. Some challenges which affect the activities of RE in ASD.

ID	**Description of the Identified challenges**	**Source**
1	Minimal documentation, lack of documentation or inadequate documentation (Confidence in tacit knowledge of requirements)	[3],[10],[12],[13],[14],[15],[16],[17],[18],[21],[22]
2	Difficulty in creating accurate cost, schedule and performance estimates, budget, and time estimates	[3],[10],[12],[13],[14],[15],[16],[17],[18],[21],[22]
3	Neglected Non-Functional Requirements (NFR)	[3],[10],[12],[14],[15],[16],[17],[18],[19],[21],[22]
4	Inefficiency in the requirements changing control (traceability)	[3],[10],[14],[15],[16][17],[19],[20],[22],[24]
5	Tendency to omit architectural issues, inadequate architecture, architectures not scalable due to constant changes	[3],[10],[12],[13],[14],[15],[16],[17],

ID	**Description of the Identified challenges**	**Source**
		[21],[22]
6	Low customer availability	[3],[10],[12],[13] [14],[15],[16],[17], [21],[24]
7	Customer involvement and interaction (Inadequate user-developer interaction, communication gaps)	[3],[10],[14],[18], [19],[21],[22],[25
8	Difficulties with distributed teams	[3],[13],[14],[15], [16],[20], [25]
9	Much time spent with changes in requirements (scope change and rework, moving targets - changes in goals, business processes and/or requirements)	[13],[14],[19],[20], [22],[23]
10	Customer inability (incomplete domain knowledge) and agreement (consensus among customer groups)	[3],[10],[14],[15], [16]
11	Frequent requirements reprioritization	[10],[14],[15],[18]
12	The team has little knowledge of the client's business domain	[10],[16],[17],[20], [23]
13	Requirements are widely abstract and allow many inconsistent interpretations (incomplete or implicit requirements)	[10],[13],[14],[19], [23]
14	Inadequate automated support to specify and manage requirements	[14],[16],[17],[22]
15	Conflicts due to many sources of requirements (there are stakeholders with different product views)	[3],[14],[15],[17]
16	Misunderstandings due to the absence of key people	[10],[14]

From this investigation, we got an initial theoretical base that mapped the challenges. Next, we detail some of them, presented in Table 1, highlighting the most cited that affect the adherence of requirements to the business in agile projects:

- **Minimal documentation or lack of documentation** - is a challenge that agile methods pose to development teams. Whenever there is a communication lapse due to sudden changes in requirements, unavailability of appropriate client representatives, project complexity and the lack of documentation, several problems arise;

- **Neglected non-functional requirements (NFR)** - NFR are often ill defined and ignored during early development cycles. Customers often focus on core functionality and ignore NFR such as portability, scalability, maintainability, performance, or safety. The tendency to ignore critical issues such as security and performance early in the development process results in major issues as the system matures and becomes ready for deployment;

- **Tendency to omit architectural issues, inappropriate architecture** - Inadequate infrastructure can cause problems during later project phases. Inappropriate architecture finalised by the team in earlier phases of the project becomes inadequate in later phases with new requirements add to the cost and become complicated to deal with;

- **Low customer availability** - Customer availability is advocated by agile methods. Yet, it is seen as challenging and largely scarce. In practice, most of the agile teams have representatives or proxy customers. And, on agile software development, it is not only necessary but crucial for the success of the project. The intense interaction between the developers and the customers is complex because it involves some variables like time, budget allocation and domain knowledge;

- **Customer involvement and interaction** are the main reasons for project success. It is important to identify the customers or representatives of business groups to ensure that requirements are properly defined and prioritized. Failure to identify customer representatives may result in disagreement and differing views on a variety of issues. Thus, agile methods rely on frequent collaboration;

- **Customer inability (incomplete domain knowledge) and agreement (consensus among customer groups)** - The lack of in-depth business process knowledge by the Customers' representatives and their different point of views impairs the definition of the requirements adherent to the business and the decision making process. The disagreement between customer groups affects the performance, especially in short development cycles.

3. RESEARCH METHODS

The overall aim of this study is pursuing answers to exploratory questions. Exploratory questions are designed to gain more profound knowledge about some phenomenon and to discuss useful issues, which help to clarify our understanding of that phenomenon [26]. The phenomenon in question is the requirements adherence to business in agile projects. Thus, our goal is to investigate different contexts of agile projects to represent an important and timely contribution. The philosophical stance chosen for the study affects the methods that must be used to answer the research question and what can be accepted as truth [26]. We chose the constructivist stance which "concentrates less on verifying theories, and more on understanding how different people make sense of the world, and how they assign meaning to actions" [26]. According to Figure 1, initially, a study was performed to investigate the phenomenon in literature, identifying several systematic literature reviews/mapping on RE in agile projects. Therefore, we obtained the initial theoretical background and the key constructs (factors) related to RE in agile projects. These results are summarized and described in Section 2. Based on these findings, we conducted seven case studies to investigate the phenomenon in industrial context and gather data to assess the initial theoretical background. The case studies were chosen because they offer the opportunity to get a thorough understanding of how and why the phenomenon occurs in practice. In each case study, we used thematic analysis techniques [27]. To conduct the cross-case analysis, the outcomes of each case study were analyzed, searching for patterns. The idea behind searching for cross-case patterns is forcing the investigators to go beyond initial impressions by using structured and diverse lenses on the data. Meta-ethnography procedures were used to translate concepts and propositions during the cross-case synthesis. The design of the case studies, the thematic analysis procedures, and the steps to conduct the cross-case analysis and synthesis are described in Section 4. As a result, the factors found in the research were compared with the findings of the literature investigation.

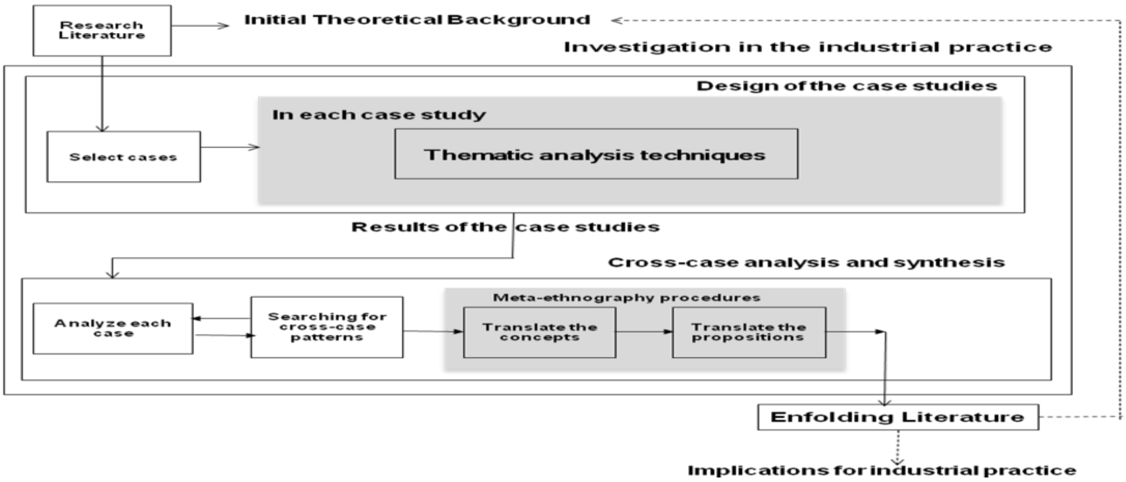

Research Literature
Initial Theoretical Background
Investigation in the industrial practice
Design of the case studies
In each case study
Select cases
Thematic analysis techniques
Results of the case studies
Cross-case analysis and synthesis
Meta-ethnography procedures
Analyze each case
Searching for cross-case patterns
Translate the concepts
Translate the propositions
Enfolding Literature
Implications for industrial practice

Figure 1. Research design

4. DESIGN AND ANALYSIS OF THE CASE STUDIES

We use a qualitative research approach, as we want to understand the research problem in a specific context, as described by [28]. This qualitative research has an approach through ethnographic case studies in areas of software development of companies. The use of case studies was selected, because they can be conceptualized as an intensive description and analysis of a phenomenon or social unit, such as a group, an individual, institution or a community [28].

4.1 Participants Sampling

The unit of analysis of the study is made up of software engineers. According [28], the sample can be characterized as non-probabilistic, since the selection of the elements of the population to compose the sample, depends partly, on the researcher's judgment. The case studies are classified as instrumental because our aim is understanding the constructs related to the phenomenon. The phenomenon was investigated from the software engineers’ point of view. In each case study, we followed the thematic analysis procedures earlier illustrated in Figure 1. Two levels of sampling are often needed in qualitative case studies: the cases which will be investigated and the participants [29]. The condition for the projects be selected was the companies having the business processes alignment with the RE as an objective and adopting agile software development in their projects. To increase data diversity, we sought for companies with different characteristics. Seven companies that fit these prerequisites were chosen. The profile of the companies is summarized later in Table 2. The second level of sampling was the software engineers that took part in each case study.

4.2 Procedures for collecting data

We have conducted semi-structured interviews, observations, and document analysis to gather data for 12 months (October/2018 to December/2019).

4.2.1 Interviews

The interviews were carried out face-to-face at the company's site, after obtaining the authorization of the companies to contact each participant directly. We used an interview script to guide the interviews with the software engineers. Most of the questions were open-ended. The questions were presented in a funnel format, with general questions first and moving towards more specific ones. We also conducted a pre-test with two pilot interviews. This was important to refine the interview scripts. The participants were supposed to answer the questions, based on produced requirements in the project that they were working with. We also conducted retrospective interviews to explain and complement the information identified in the data analysis.

4.2.2 Data analysis from documents

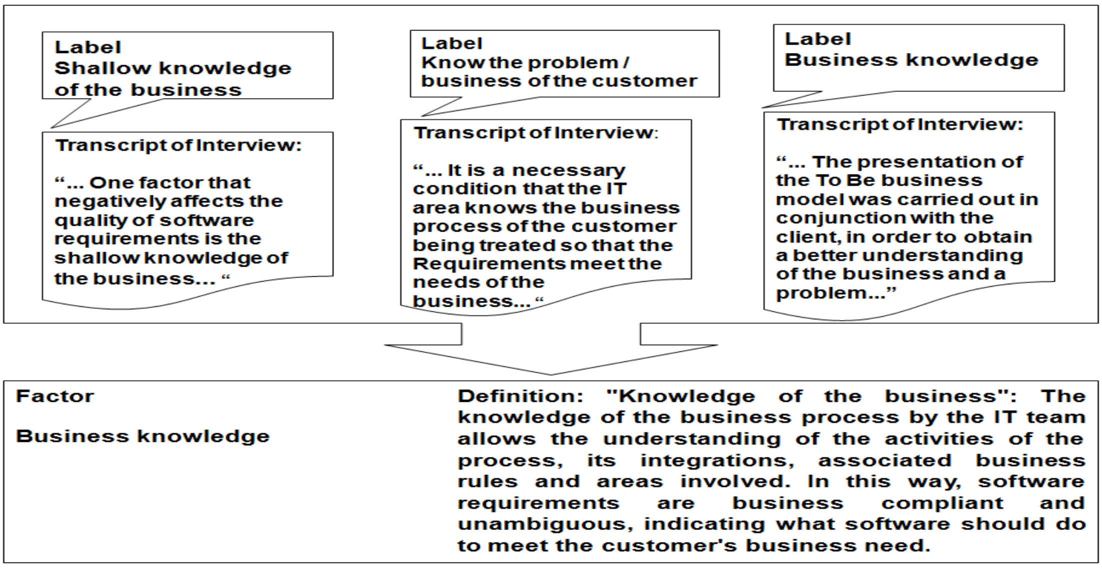
Documents are an essential source of data for qualitative research [28]. Several documents used by the software engineers to elicit, validate, manage, and specify requirements were analysed. All documents analysed during the studies were produced by the project software engineers interviewed. They helped to obtain a better understanding of the difficulties and facilities pointed out by the team.

4.2.3 Observations

Initially, observations were made seeking to follow the workday of the development teams to understand their culture and behaviour in everyday activities in their workplace. Agile projects were observed, and daily team meetings were held when necessary, at least once a week. They tried to share best practices, problems, and difficulties faced by the team. Researchers took part in some meetings as observers, recording and observing how software engineers were handling the requirements and their difficulties. We did not interfere in the way the project was managed.

4.2.4 Data analysis in each case study

Data analysis is the process of making sense of the data [28]. It involves interpreting, consolidating, and reducing what people said and what the researchers have seen and read. The practical aim of data analysis is finding answers to the research questions. We performed the data collecting and analysis simultaneously, in an incremental and iterative way using thematic analysis. At first, we used the open coding procedure, which involves attaching codes (labels) to segments of text extracted from the interview transcripts that were important to the study. The resulting codes from each interview were constantly compared against codes from the same interview, and from other interviews to identify duplication, similarities, and inappropriate codes, as proposed by the constant comparison method. The data gathered in the interviews were triangulated with those obtained from the analysis of documents and observations, to increase its reliability. The triangulation technique avoids the influence of individual analysis based on the interviewer's opinion. From the constant comparisons of the codes, they were grouped into categories representing factors that affect the requirements' adherence to the business in agile projects. "Business Knowledge" factor, for example is the result of grouping some labels as shown in Figure 2. Then, we used thematic analysis to identify the relationships among factors.


Label
Shallow knowledge of the business
Label
Know the problem / business of the customer
Label
Business knowledge

Transcript of Interview:

"... One factor that negatively affects the quality of software requirements is the shallow knowledge of the business... "

Transcript of Interview:

"... It is a necessary condition that the IT area knows the business process of the customer being treated so that the Requirements meet the needs of the business... "

Transcript of Interview:

"... The presentation of the To Be business model was carried out in conjunction with the client, in order to obtain a better understanding of the business and a problem..."

Definition: "Knowledge of the business": The knowledge of the business process by the IT team allows the understanding of the activities of the process, its integrations, associated business rules and areas involved. In this way, software requirements are business compliant and unambiguous, indicating what software should do to meet the customer's business need.

Figure 2. Example of factor building

4.3 Analysis of each case study

We interviewed 87 software engineers in public and private companies of different sizes. The investigated companies have a different level of experience regarding BPM and had as a goal the business processes alignment with the RE. Different companies have been selected to have a maximum sampling variation strategy. All the investigated projects in the companies used an ASD method based on Scrum. Other relevant aspects are summarized in Table 2.

Table 2. Contrasting organizational contexts.

Characteristics	C1	C2	C3	C4	C5	C6	C7
Years of Existence	53	20	25	42	10	15	60
Size	Very large	Small	Medium	Large	Micro	Large	Very Large
Area of operation	Brazil	Brazil	Brazil	Brazil	Brazil	Brazil	150
Nature of Sector: Public (Pub) Private (Pri)	Pub	Pub	Pub	Pub	Pri	Pri	Pri
Type of Customer	Pub	Pub	Pub	Pub	Pri	Pri	Pub/ Priv
Nature of Products: Information Systems (IS) and Artificial Intelligence (AI)	IS and AI	IS and AI	IS	IS	IS	IS	IS
Nº of Employees in the Organization	10374	200	600	3.800	8	400	20.000
Nº of Software Engineers in the Case study	20	9	10	12	8	14	11
Experience in ASD (years)	8	3	2	7	7	3	10
Communication with the customer	Team	PO	PO	Team	PO	Team	Team
Frequency of validation of requirements by software (client)	Every month	Every month	4 weeks	8 weeks	Every month	Every month	Every month
Validation Type: Remote (RE) or Presential (PRE)	Pre	Re	Pre	Re	Re	Re	Re

In the interviews, to fulfil the overall objective of the investigation, we formulated the following research questions: (I) RQ1. What are the techniques and practices used by the teams to perform RE activities? and (II) RQ2. Which are the problems faced by the teams when performing RE activities?

An interview with 34 questions was performed to collect data. These questions were divided into the following groups: (I) demographics; (II) agile RE artefacts, (III) techniques and practices of requirements validation, elicitation, and management; and (IV) problems faced during requirements elicitation, specification, validation, and management. The first group aims at capturing data about the respondents and their organizations. With this data, it is possible to understand, for example, the experience with ASD and the roles played by the participants, as well as the size of the organization and its maturity in the adoption of agile methodologies. The questions of group two were designed to collect data about the artefacts used by the agile teams to document the requirements; and which of these artefacts are the focus of validation, elicitation, and management. The questions of the third group gather data about the techniques and practices used to elicit, validate, and manage the requirements. Besides, they also focus on finding out the stakeholders who usually participate of requirements elicitation and validation sessions. The last group of questions focuses on collecting data about the problems faced on running requirements validation and elicitation sessions.

In the case studies, we investigated whether the factors listed in Table 1 affect the requirements adherence to business, how they are related to each other and how software engineers perceive them. Moreover, we attempted to identify other factors in addition to the ones presented in Table 1.

The identified factors in each case study and their impact (positive or negative) are listed in Table 3. The factors which positively affect and improve the requirements adherence to business are represented as "+". The factors which negatively affect the requirements adherence to business are represented as "-".

Table 3. Factors by case studies.

Factors	**Impact**	**Case studies**						
		1	**2**	**3**	**4**	**5**	**6**	**7**
Business Knowledge	+	✓			✓	✓	✓	✓
BPM and RE Tools Integration	+							✓
Inadequate Experience with BPM	-	✓		✓	✓	✓	✓	
To-Be Model	+	✓	✓	✓	✓	✓	✓	✓
Outdated To-Be model	-	✓		✓	✓			✓
Customer-Team integration	+	✓			✓	✓		✓
Sponsorship	+	✓			✓	✓	✓	✓
Lack of Sponsorship	-		✓	✓				
Automated Support	+	✓			✓	✓		✓
NFR	+	✓	✓		✓		✓	✓
Lack of NFR	-			✓		✓		
Traceability	+	✓		✓	✓			✓
Change History	+	✓		✓	✓			✓
Acceptance Criteria	+	✓		✓	✓			✓
Inadequate Acceptance Criteria	-		✓			✓	✓	
Prototyping	+	✓			✓	✓		✓
Business Rules	+	✓			✓			✓
Inadequate Customer Availability	-	✓						✓
Low Customer collaboration to validation	-		✓	✓		✓	✓	

Factors	**Impact**	**Case studies**						
		1	**2**	**3**	**4**	**5**	**6**	**7**
Inadequate Experience with ASD	-	✓		✓	✓	✓	✓	✓
Late validation by SW	-		✓	✓				
Excessive changes	-		✓	✓		✓		
Documentation used to requirements validation	+	✓	✓		✓		✓	✓
Contract agreements	-	✓	✓	✓	✓		✓	✓

Participants were asked about their understanding of factors that affect the requirements adherence to business. Some findings are presented here. For each quote, the following format was adopted: (Cn: Case study number, Pn: Participant number). Participants emphasized some factors, for example, Business Knowledge, as seen in the following quotes.

“The team needs to know the customer's business process to be addressed so that the requirements meet the business needs.” (C1P4)

“The analysts did not have business knowledge and had difficulty to find out which requirements were more relevant and so prioritize them. I should not waste time trying to understand the meaning of the requirements and waiting for its prioritization.” (C2P11)

“... One factor that negatively affects the quality of software requirements is the shallow knowledge of the business...” (C3P16)

“The presentation of the To Be model was made by the client, with the purpose of passing on business knowledge so that the team had a broader view of the business and the problem. We understand that this activity should be performed during the requirements gathering to help establish a broader product view and system requirements that add value to the customer's business.” (C4P18)

“... It is a necessary condition that the IT area knows the business process of the customer being treated so that the requirements meet the needs of the business...” (C5P25)

“The Knowledge of the business process and its problems faced by the customer make the requirements adherent to the business and the proposed solution meets the customer's needs and expectations. In the particular case of Intelligent Systems, an in-depth understanding of the business process and its data is fundamental for the development team propose the most appropriate solution. “(C2P4)

“The knowledge of the business process by the team allows the understanding of the activities of the process, its integrations, associated business rules and areas involved. In this way, software requirements are business compliant and unambiguous, indicating what software should do to meet the customer's business need.” (C6P29)

“...the presentation of the To Be process model was carried out by the client in order to share business knowledge with the team...” (C7P30)

5. Cross-Case Analysis and Synthesis

We performed a cross-case analysis to reach the synthesis by categorizing the concepts identified in the case studies, following the steps earlier shown in Figure 1. The key to useful cross-case comparisons is looking at the data in many different ways. Furthermore, the cross-case analysis enhances the probability of capturing new findings, which may already exist in the data. The cross-case analysis could explain similarities and contradictions among different cases. At first, we analyzed each case to identify and understand the concepts and their relationships. Then, we sought for cross-case patterns interpreting and summarizing the key similarities and differences among the cases to extend our understanding of the phenomenon in different contexts, and to explain it. Concepts and propositions of the case studies have been translated during the cross-case synthesis by using meta-ethnography procedures which entails some degree of induction and interpretation [29]. To predict possible situations that could occur when translating the concepts from one case to another, we applied the strategy used by [29] based on the principles of meta-ethnography to generalize and rename concepts, as shown in Table 4.

Table 4. Translation strategies for first level concepts.

Type of translation	**Situation**	**Strategy**
Identical	The identical label and definition were used for a concept in cases studies	We merely repeated the concept as the translation
Renaming	Distinct labels were used for a concept in each study, but the definitions were similar	We selected the label that better expressed the meaning, consulting the thesaurus, dictionaries, and the literature to support the choice
Generalization	Different concepts were found in each case study, with distinct names and definitions, but one concept could be interpreted as a generalization, or abstraction, that included one or more concepts in the other case studies	We used the more general concept as the translation whenever it expressed the findings of case studies
Localization	A concept was found in one case study but not in the others.	We maintained the concept as the translation with a restriction that it was context dependent and associated the concept to its context.
Refutation	A concept present in one case study contradicted another one in other case study	We attempted to understand and explain the Refutation based on contextual data and included the explanation to the translation.

After the unification of concepts, the relationships among the factors (propositions) were analyzed to translate them across the case studies. The propositions were also translated using the strategies proposed in meta-ethnography, and the Localization type was added, as described in Table 5.

Table 5. Translation strategies for propositions.

Type of translation	Situation	Strategy
Reciprocal (RTA)	Propositions related the concepts by similar or comparable causal relationships	We identified the common aspects among the cases studies and translated into a consistent proposition with them
Line-of-argument (LOA)	Propositions related a distinct set of concepts, with sufficient intersection and without refutation, allowing the construction of a line-of-argument that could explain the different situations	Higher degree of inductive inference and interpretation was used to build the translations than in the reciprocal case
Localization (LOC)	Proposition not identified in all cases studies where the concepts were presented. Thus, it is context dependent.	The proposition was kept with a remark that represents a relationship context dependent.
Refutational (RFA)*	Propositions related the concepts in opposing or contradicting relationships	Contextual information would be used to clarify the refutations enhancing our understanding of the varying conditions under which the local propositions would hold or fail

5.1. Searching for cross-case patterns

To get a maximum variation sampling strategy, we selected different organizations, as illustrated in Table 2. All the organizations are headquartered in Brazil and provide customers services, located in the different Brazilian States. One of them has branches in several countries (the seventh case study – C7) and the development team works entirely distributed. The investigated organizations operate in different business segments. In most of them, software development is the organization's core activity. In four of them, the customer is in other city different from the development team. Therefore, software validations were performed either face-to-face or remotely. All organizations adopt short sprints, ranging from 1 to 4 weeks. But the software validation frequency is different in each organization.

5.2 Similarities between the studies

The collected data were analysed to identify similarities and differences, and to find ways of explaining them. The similarities in the findings of the case studies increase their reliability and external validity [29]. Although the diversity in the organizations' context and in the software engineers' profile, we identified several similarities among the case studies.

The customer integration with the team brings harmony between the people, which helps overcome the challenges that arise during the project. It was also observed that the lack of integration between the teams generates a hostile environment, which hampers overcoming the problems found in the project. All the interviewed teams considered that communication between the customer and software engineers was fundamental for project success.

The most used techniques during requirement elicitation were interviews, prototyping and facilitated meetings. Some companies employed observations. The models To Be, in BPMN notation, as one of the artefacts used in the software requirement elicitation made the business understanding easier and improved the team productivity, avoiding rework. The lack of To Be model or the lack of its update implied understanding difficulties of the problem generating requirements not adherent to the business.

The functional requirement documentation was described with user stories, business/domain process models, data models, and prototypes. The lack of NFR, technical aspects, such as project constraints and interface validation rules were regarded as a negative factor. The changing history and the dependency relation between the requirements were relevant to analyse the changing request impact. The lack of acceptance criteria or their inadequacy were also identified as negative factors in the RE process.

All software engineers highlighted that customer availability is important to detail requirements, clarify doubts, and validate the features. The customer must be available when the team needs him/her. The most critical problems faced by requirements validation teams were the lack of availability of the clients or of their representatives and stakeholders with different viewpoints of the product. In two of the companies, the customer availability was inadequate, and that was identified as harmful for the requirement elicitation activity, for the document content, and for the software validation. Insufficient experience in ASD affected the way in which the requirements were elicited, described, and validated, as well as their content.

The most critical problems faced by requirements validation teams were the lack of availability of the clients or of their representatives and stakeholders with different viewpoints of the product. The most used artefacts in requirements validation were prototypes, user stories and acceptance criteria. In all investigated projects, the interviewees mentioned the importance of top management sponsorship is fundamental for the integration of the RE and BPM to get requirements adherent to business.

5.3 Explaining differences between the case studies

The diversity of organizations, processes, and the kind of software developed variety open opportunity for explanations, based on contextual differences, are thus enriching the phenomenal understanding in largely different environments. The main differences between the case studies are, shortly, described as follows:

- **Software Requirement Documentation used to validate requirements -** five companies used documentation to validate the software to be developed instead of doing validations through software frequent deliveries. Their contracts established that a specification should be validated by the customer before coding;
- **Software validations -** we found some differences regarding the procedures of software validations. In two companies, validations were conducted only remotely by the customers, without the participation of the development team (C2, C3). In these companies (C5, C6), the customer had limited availability to clarify doubts, detailing requirements, and validating software. Thus, the remote validations entailed problems for the project because the customer often took a long time to verify the partial versions. The remote validations also did not affect the project execution when the customer had an adequate collaboration (C1, C4, C7);
- **Specialized automated support -** there was unanimity among the interviewees about the importance of using an automated tool for BPM, integrated with the tool for RE, but we only found this situation in the C7. In the case studies (C2, C6), it was pointed out that the lack of a specialized tool to specify and manage requirements affected the requirements adherence to the business. We observed that the importance of the tool depends on the size and level of the software project complexity.

5.4 Translating the concepts

After analyzing the context, the storyline of each study, the similarities, and differences among them, we performed the translation of the concepts obtained, to unify the nomenclature of the factors. Concepts of the case studies have been translated during the cross-case synthesis by using meta-ethnography procedures, which involves some degree of induction and interpretation. To anticipate possible situations that could occur when translating the concepts from one case to another, we adopted the strategy used by [29]. Renaming and Generalization types required a bigger challenge. Identical and Localization translation types were trivial. For Identical, the names and meanings of the concepts were the same in several case studies. Moreover, for location, there was no translation because the concept is context dependent. In Table 6, we present some translations performed.

Table 6. Examples of concept translation.

Concepts	Type	Translation	Id
Inadequate Experience with RE (-)	Identical	Inadequate Experience with RE (-)	#1
Business Knowledge (+), Superficial Business Knowledge (-)	Renaming	Business Knowledge (+)	#2
Sponsorship (+) and Lack of Sponsorship (-)	Renaming	Sponsorship (+)	#3
Acceptance Criteria (+), Inadequate Acceptance Criteria (-), Lack of Acceptance Criteria (-)	Renaming	Acceptance Criteria (+)	#4
TO BE Model (+), Outdated TO BE Model (-)	Renaming	TO BE Model (+)	#5
Low Customer collaboration to validation (-), Low Customer Availability to detail and test requirements (-)	Generalization	Inadequate Customer Availability (-)	#6
User Stories, Conceptual Model, Prototypes of UI and Business Rules (+)	Generalization	Functional Requirements (+)	#7

5.4.1 Factors that impact the requirements adherence to business in agile projects (identifying the Constructs)

According to the cross-case synthesis, the factors that affect positively on the requirements adherence to the business are:

- **Business knowledge:** The business process knowledge by of the team allows the understanding of the business process, their integrations, associated business rules, and affected areas. As a result, we have unambiguous software requirements adherent to the business, showing what the software can do to meet the needs of the customer;
- **Customer-Team integration:** it allows a common vision of the business goals shared by all the project stakeholders what generates an excellent organizational environment that motivates everyone to reach the project goals;

- **To-Be Model:** Business process modelling using BPMN notation allows the team to understand the business process, its data and its integrations. Identifies which areas are involved in the processes and which business rules are associated to them;
- **Sponsorship:** Top management should sponsor the participation, in the requirements specification group, of key people who have knowledge of the business processes prioritized by the customer and who have decision-making power. In addition, it must sponsor the integration of BPM and RE processes, as well as the integration of tools, which support these processes and the integration of the teams involved;
- **BPM and RE tools integration:** the tools used in BPM must be integrated with the automated tool used in the RE in order to ensure traceability and facilitate the updating of requirements;
- **Prototyping:** it is perceived as a simple and straightforward way to review requirements with clients and to gain timely feedback prior to moving to subsequent iterations. Prototyping starts with simple requirements that are thoroughly understood and have high priority. It promotes quicker feedback and enhances the product anticipation to the customer;
- **Acceptance Criteria:** It is the description of criteria that support the acceptance, or not, of a requirement;
- **Architecture:** consists of the description of the project's architecture and any other technical information for operationalization the requirements;
- **Functional requirements:** It consists of the description of the functions or tasks to be performed by the system. Requirements can be represented using a textual format, for example, by business rules, user stories; or using a visual representation, such as conceptual models or other representations;
- **NFR:** it is the description of the requirements for operating the system, such as performance, security, among others;
- **Traceability**: it is the description of the source of each requirement, as well as, the dependency relationships between them;
- **Change History:** It consists of the history of the changes made in the requirements;
- **Automated Support:** The BPM and RE activities should be carried out with the support of a specialized tool.

On the other hand, the factors that jeopardize the requirements adherence to business are:

- **Inadequate customer availability:** the inadequate availability of the customer to cope with the developers or to validate the software;
- **Inadequate experience of the team with agile RE:** the developers has little experience with the agile RE;
- **Outdated Requirements:** the description of the requirements is outdated, no longer corresponds to the needs requested by the customer;
- **Excessive changes:** the development team receives many requests for changes in partial versions of the application.

5.4.2 Revealing the outcomes

The presence or absence of these factors affect the software requirement quality and its adherence to business; consequently, affect the ASD in a positive or negative way. The cross-case analysis revealed some outcomes, as follows:

- **Effort required to elicit:** Effort required for the development team to elicit the software requirements so that the requirements are adherent to the business;
- **Effort required to document:** the effort required for the development team to elaborate the software requirements documentation, so that the requirements are adherent to the business;

- **Dependency between stakeholders:** the degree of dependency that exists among the stakeholders due to the requirement content;
- **Non-conformity in the software:** the amount and type of non-conformities detected in the software, due to requirement problems;
- **Knowledge transfer:** The capacity of the requirements documentation to be used as a tool for knowledge transfer among the stakeholders, especially when there is a turnover, as well as in distributed teams;
- **Impact analysis:** the capacity of the requirements documentation to support the impact analysis of requests for changes to requirements or business processes.

5.5 Translate the propositions

After the concepts unification, the relationships among the factors (propositions) were analyzed to translate them across the case studies. The meta-ethnography procedures [29] were used in this process. As a result of a cross-case analysis and synthesis, 9 propositions were extracted as shown in Table 8. The propositions identified only in the local context (LOC) of a specific case study were maintained. For example, the proposition number eight (#P8) in Table 7, in the first case study (C1P4). The reciprocal translations (RTA) were also trivial since we only needed to rewrite the propositions after the construct's unification. Translations of Line-of-Argument (LOA) type required a greater effort of interpretation and induction. For example, #P6, the inadequate automated support contributed to the lack of traceability and changing history, which impairs the software requirement update (C4P5). However, the automated support makes the traceability and changing history feasible. The integration of the BPM and RE tools enables the requirement update (C7P1). Analyzing the data collected and the central stories, we have identified a new proposition related to the automated support and the integration of BPM and RE tools that avoids excessive changes.

Table 7. Examples of proposition.

Proposition	**Proposition**	**Translation**	**Type**	**Nº**
C1P1: The **TO-BE model** facilitates the **Business knowledge** by the team. The Knowledge of the business process and its problems faced by the customer make the requirements adherent to the business and the proposed solution meets the customer's needs and expectations. In the case of Intelligent Systems, an in-depth understanding of the business process and its data is fundamental for the development team propose the most appropriate solution.	C2P2: The lack of knowledge or superficial knowledge of the business by the team implies ambiguous software requirements and does not adherent to the business.	The **TO-BE model** facilitates the **Business knowledge** by the team. The Knowledge of the business process and its problems faced by the customer make the requirements adherent to the business and the proposed solution meets the customer's needs and expectations. In the case of Intelligent Systems, an in-depth understanding of the business process and its data is fundamental for the development team propose the most appropriate solution.	RTA	P#1

Proposition	Proposition	Translation	Type	Nº
C7P4: The **Sponsorship** of top management for the **Customer-Team integration** allows the participation, in the Inception meeting, of people who have essential knowledge of the business process and its integrations, as well as people who have decision-making power. Thus, the requirements obtained are correct and complete, avoiding rework.	C3P5: The lack of **Sponsorship** from the top management for the **Customer-Team integration** generated communication failures that result in incomplete, O**utdated** and **Excessive changes** requirements.	The **Sponsorship** of top management for the **Customer-Team integration** allows the participation, in the Inception meeting, of people who have essential knowledge of the business process and its integrations, as well as people who have decision-making power. Thus, the requirements obtained are correct and complete, avoiding rework.	RTA	P#2
C3P6: The **Sponsorship** of top management for **BPM and RE tools integration** contributes for improving the quality of requirements, keeping them updated, reliable and adherent to the business process.	C2P5: The lack of top management **Sponsorship for BPM and RE tools integration** implies **outdated** requirements.	The **Sponsorship** of top management for **BPM and RE tools integration** contributes for improving the quality of requirements, keeping them updated, reliable and adherent to the business process.	RTA	P#3
C1P11: The adequate documentation of the requirements is made up of **Functional requirements**, A**rchitectural model**, **NFR**, **To Be model** and **conceptual model**.	C6P4: **Business process knowledge** and of its data by the team right in the beginning of the project allows the development of a scalable A**rchitecture** for the project and the definition of **NFR**. In this way, rework is avoided, and customer needs are met.	The adequate documentation of the requirements is made up of **Functional requirements**, A**rchitectural model**, **NFR**, **To Be model** and **conceptual model**. **Business process knowledge** and of its data by the team right in the beginning of the project allows the development of a scalable A**rchitecture** for the project. In this way, rework is avoided, and customer needs are met.	RTA	P#4
C4P5: The inadequate **Automated Support** contributes to the lack of	C7P1**: Automated Support** enables	The **Automated Support** makes the **Traceability** and **Change History**	LOA	P#6

Proposition	Proposition	Translation	Type	Nº
Traceability and **Change History** which hinders the software requirements updating.	**Traceability** and Change **History**. The **BPM and RE tools integration** make it possible to update the requirements.	feasible. The **integration of the BPM and RE tools** enables the requirements updating and avoids **Excessive changes**. That way, improves the developed software adherence to the business process, meeting the customers' expectations.		
C1P9: **Prototyping** is perceived as a simple and straightforward way to review requirements with clients and to gain timely feedback prior to moving to subsequent iterations.	C2P8: **Prototyping** promotes quicker feedback and enhances the product anticipation to the customer.	**Prototyping** is perceived as a simple and straightforward way to review requirements with clients and to gain timely feedback prior to moving to subsequent iterations.	RTA	P#7
C1P4: **Inadequate experience of the team with agile RE** implies long-winded and ambiguous requirements that hinder their clarity, making it difficult to understand them.		C1P4: **Inadequate experience of the team with agile RE** implies long-winded and ambiguous requirements that hinder their clarity, making it difficult to understand them.	LOC	P#8
C2P9: **Inadequate customer availability** hinders the clarification of doubts and the validation of partial versions of the requirements. As a result, the developer lacks technical aspects**,** the solution does not meet customer expectations and **Excessive changes** are requested.	C7P3: Adequate customer availability prevents **Excessive changes** and guarantees updated and adherent requirements to the business.	Adequate customer availability at the inception meeting, to clarify doubts and validate partial versions of the requirements, prevents **Excessive changes**, and guarantees updated and adherent requirements to the business.	RTA	P#9

6. DISCUSSION AND THREAT TO THE VALIDATION

6.1 Comparison of results with literature

An essential characteristic of the theory building is the concepts comparison, propositions, or emerging hypothesis with the existing literature. This includes asking what is similar, what contradicts and why. The findings of the cross-case synthesis were compared to the literature review made in this research. The results of the cross-case analysis confirmed that all the factors in the investigation of literature, which affect the requirements' adherence to the business in agile projects were also identified in the case studies. In the case studies performed in this research, it was observed that the business knowledge by the team implies requirements adherent to the business. In some case studies, software engineers pointed to problems deriving from low availability and inadequate iteration with the customer. Excessive changes in requirements have been identified as a factor that contributes to the requirements becoming outdated, which may impair the adherence of requirements to the business. The inadequate control of the changes was also mentioned as a negative factor. The customer's availability to respond to questions, detail requirements and validate the partial versions according to the team's demand is more relevant. The investigation of the phenomenon in practice introduced new factors that had not been pointed out in the investigation of the literature: the top management sponsorship for BPM and RE tools integration contributes to improving the quality of requirements, keeping them updated and reliable, adherent to the business process; and in the particular case of Artificial Intelligence Systems, a in-depth understanding of the business process and its data is fundamental for the development team to propose the most appropriate solution.

6.2 Implications for the industrial practice

The cross-case synthesis revealed 7 findings (F) that may have implications for software organizations in their quest to improve the quality of requirements and their adherence to the business in agile projects, as follows:

- **F1:** BPM can be understood as an activity for building models to represent business processes. These models help identify the system requirements to be developed, prioritizing the most relevant processes and their integration. In the particular case of Intelligent Systems (Predictive and Artificial Intelligence), the in-depth understanding of the business process and its data is fundamental for the development team to propose the most appropriate solution;

- **F2:** The participation in the business process modelling or the presentation of the business process To Be model makes the gain of knowledge easier for the team in the requirement elicitation phase. The initial goal of the BPMN notation is promoting an easy-to-understand notation, so that all the stakeholders could have a common sense of the business processes;

- **F3:** The top management sponsorship for the "integration of the team and the customer" allows the participation, in the Inception meeting, of people who have essential knowledge of the business process and its integrations, as well as people who have decision-making power;

- **F4:** The lack of the customers' knowledge to define the requirements and their inability in terms of decision making when a customers' group is involved can directly affect the project performance and the requirements adherence to the business;

- **F5:** Business process knowledge and its data by the team right in beginning of the project allow the development of a scalable architecture for the project and the definition of non-functional requirements. In this way, rework is avoided, and the customer's needs are met;

- **F6:** Changes on the requirements or incomplete requirements are stated as problems and should encourage the use of agile practices. Those changes lead to the specified requirements become quickly obsolete. Overall, the effects might be small if the stakeholders agree on the requirement changes. Therefore, the project will not have problems related to budget and schedule because everyone knows that the requirements are flexible. This way, stress to the customer and to the development team is avoided;

- **F7:** BPM and RE tools integration enables the requirements updating and avoids excessive changes. Automated support enables traceability and the requirement change history, thus, making these changes impact analysis easier, mainly in complex projects, in large teams or even distributed teams.

6.3 Limitations and threats to validity

The investigated companies did not provide historical data on quality and productivity. The outcomes presented in the paper reflect the results of the analysis considering the opinion of software engineers, artefacts and the comments that were made during the period of observation in the development environment of the projects. However, we did not collect metrics to assess the extent of effectiveness of these outcomes.

7. CONCLUSIONS

This paper presented the activities performed to investigate the phenomenon of the RE in ASD and provide an in-depth description of the factors, which should be considered in the RE in ASD with the aim of improving requirement adherence to the business in ASD. In the case studies investigated in this research, in terms of the present state of the practice related to the elicitation, the agile RE is not so different from the traditional RE. The backlog is the central mean to deal with the changes in the requirements. We observed that the BPM makes the knowledge gain easy by the team in the requirement elicitation phase. BPM can be understood as an activity for building models to represent business processes. These models help identify the system requirements to be developed, prioritizing the most relevant processes and their integration. In the specific case of Intelligent Systems (Predictive systems based on Artificial Intelligence), the in-depth understanding of the business process and its data is fundamental for the development team propose the most appropriate solution. As part of our future research, we aim to propose new practices to improve the quality of requirements adherence to business in ASD.

REFERENCES

[1] VersionOne., 11th Annual State of Agile Survey. Available at:https://versionone.com/pdf/VersionOne-11th-Annual-State-of-Agile-Report.pdf. Accessed in: 18/12/2018, 2018.

[2] Standish Group, “The Chaos Report. Available: <http://www.projectsma.co.uk/white-papers/chaos-report.pdf>, 2017.

[3] Schön, M., Thomaschewski, J., Escalona, J., "Agile Requirements Engineering: A Systematic Literature Review". Comput. Stand. Interfaces http://dx.doi.org/10. 1016/j.csi., 2016.

[4] Daneva, M, "Agile requirements prioritization in large-scale outsourced system projects: An empirical study". J. Syst. Soft. 86, 5, 1333-1353, 2013.

[5] Przybylek, A.,"A Business-Oriented Approach to Requirements Elicitation", Conference Paper, ENASE 2014, DOI: 10.5220/0004887701520163, 2014.

[6] Thiemich, C., Puhlmann, F., "An Agile BPM Project Methodology", In: Daniel, F., Wang, J., Weber, B. (eds), Business Process Management, Springer Berlin Heidelberg, pp. 291–306, 2013.

[7] Pastor, O., Sánchez, J., La Vara, L., "Business Process Modelling and Purpose Analysis for Requirements Analysis of Information Systems, CaiSE, 2008.

[8] Agile Manifesto. Manifesto for Agile Software Development. Agile Aliance, 2001.

[9] Sillitti, A., Succi, G., "Requirements Engineering for Agile Methods", In A. Aurum and C.Wohlin, editors, Engineering and Managing Software Requirements, Springer, page 315, 2005.

[10] Inayat, I., Salim, S., Marczak, S., Daneva, M., Shamshirband, S., "A systematic literature review on agile requirements engineering practices and challenges", Computers in Human Behavior. 51, pp. 915-929, 2014.

[11] Tawfig, A., Abdelsalam, M., Fatma R., “An approach to improvement the usability in software products”, International Journal of Software Engineering & Applications (IJSEA), Vol.7, No.2, March 2016.

[12] Cao, L., Ramesh, B. "Agile requirements engineering practices: An empirical study", IEEE Software, vol. 25, no. 1, Jan./Feb. 2008, doi:10.1109/MS.2008.1, 2008, pp. 60-67, 2008.

[13] Batra, M., Bhatnagar, A. “A Research Study on Critical Challenges in Agile Requirements Engineering”, International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 06 | June, 2019.

[14] Medeiros, J., Alves, D., Wanderly, E., Vasconcelos, A., Silva, C., "Requirements Engineering in Agile Projects: A Systematic Mapping based in Evidences of Industry", 12th Workshop on Experimental Software Engineering (ESELAW) in XVIII CIBSE Ibero-American Conference on Software Engineering, 2015.

[15] Heikkilä, V., Damian, D., Lassenius, C., Paasivaara, M., "A Mapping Study on Requirements Engineering in Agile Software Development". Software Engineering and Advanced Applications (SEAA), 41st Euromicro Conference on. IEEE, pp. 199-207. http://dx.doi.org/10.1109/SEAA.2015.70., 2015.

[16] Curcio, K., Navarro, T., Malucelli, A., Reinehr, S.,"Requirements engineering: A systematic mapping study in agile software development", The Journal of Systems and Software 139 32–50, 2018.

[17] Eghariani, K., Kama, N., "Review on Agile requirements engineering challenges". In: 2016 3rd Int. Conf. Comput. Inf. Sci., pp. 507–512. doi:10.1109/ICCOINS.2016.7783267, 2016.

[18] Ramesh, B., Baskerville, R., Cao, L. "Agile requirements engineering practices and challenges: An empirical study". Information Systems Journal, 20(5),449–480, 2010.

[19] Wagner, S., Méndez, D., Feldere, M., Kalinowski, M "Requirements Engineering Practice and Problems in Agile Projects: Results from an International Survey". Peer Preprints https://doi.org/10.7287/peerj.preprints.2038v1 | CC-BY 4.0 Open Access | rec: 10 May 2016, publ: 10 May 201, 2016.

[20] Kasauli, R., Liebel, G., Knauss, E., Gopakumar, S., Kanagwa, B., "Requirements engineering challenges in large-scale agile system development", in: IEEE 25th International Requirements Engineering Conference (RE), pp. 352–361, 2017.

[21] Hess, A., Diebold, P., Seyff, N., "Understanding information needs of agile teams to improve requirements communication", International Workshop on Just-in-time Requirements Engineering (JITRE), Lisbon, 2017.

[22] Sehrish, A., Shahid, B., Shah, S., Jadi, A., “Impact and Challenges of Requirement Engineering in Agile Methodologies: A Systematic Review”, International Journal of Advanced Computer Science and Applications, Vol. 8 No. 4, 2017.

[23] Fernández, M..,"Naming the pain in requirements engineering: contemporary problems, causes, and effects in practice". Empirical Software Engineering, doi:10.1007/s10664-016-9451-7, 2016.

[24] Alsaqaf, W., Daneva, M., Wieringa, R. "Quality requirements in large-scale distributed agile projects – a systematic literature review", in: P. Grünbacher, A. Perini (Eds.), Requirements Engineering: Foundation for Software Quality, Springer International Publishing, Cham, pp. 219–234, 2017.

[25] Ayman S., Abd-El-Kader S. "Culture Effect on Requirements Elicitation Practice in Devel- oping Countries", International Journal of Software Engineering & Applications (IJSEA), AIRCCSE, 2017, 8 (1), pp.49-58. 10.5121/ijsea.2017.8105. hal-01703239

[26] Easterbrook, S., Singer, J., Storey, M., Damian, D. "Selecting empirical methods for software engineering research". In: Shull, F., Singer, J., Sjoberg, D.I. (Eds.), Guide to advanced empirical software engineering. Springer, London, Chapter 11., pp. 285-311, 2008.

[27] Cruzes, D. S., & Dybå, T. "Recommended Steps for Thematic Synthesis in Software Engineering". In International Symposium on Empirical Software Engineering and Measurement, pp.275–284, 2011.

[28] Merriam, S.,"Qualitative Research: A Guide to Design and Implementation". ISBN: 978-1-118-94701-2. Jossey-Bass, 2009.

[29] França, A. C. C., Silva, F. Q. B., Felix, A.L. C., Carneiro, D. E. S., "Motivation in software engineering industrial practice: A cross-case analysis of two software organizations", Inf. Softw. Technol. pp. 79-101, 2014.

AUTHORS

Helena Bastos is a PhD student at Universidade Federal de Pernambuco (CInUFPE), since 2016; Master in Computer Science from Universidade Federal de Pernambuco (1999); She has more than 30 years of experience with management and development of research projects, and innovation in software companies. She has a specialization in Data Science and Analytics and another in Information Technology Management. Helena is particularly interested in Requirements Engineering, Agile Methods and Intelligent Systems.

Alexandre Marcos Lins de Vasconcelos is Graduated (BA) in Computer Science at Universidade Federal de Pernambuco (1987); Master in Computer Science at Universidade Federal de Pernambuco (1989); Ph.D. in Computer Science (Dphil in Computer Science) at the University of York, GB (1993); Postdoctoral Fellow in Software Engineering at Universidad Politécnica de Valencia, Spain (2011); Associate Professor at the Center of Informatics at Universidade Federal de Pernambuco (CIn-UFPE) since 1995; Coordinator of partnership projects between EPSON and CIN-UFPE: 1st Project - Test Method for Embedded Applications for Fiscal Printers, between December 2006 and November 2007, 2nd project - Test Automation (Test Framework for Fiscal Printers), between December 2007 and April 2010; Director of Information Technology Center from march 2013 to october 2015. He is currently an instructor, implementer and assessor of MPS.BR (Model for Software Quality created by the Society for Promotion of Brazilian Software Excellence - SOFTEX) and has experience in Software Engineering, acting on the following research topics: software quality, methodologies and software processes, environments and CASE tools, software testing and requirements engineering.

Wylliams Santos is an adjunct professor at the University of Pernambuco (UPE), where he leads the REACT Research Labs. PhD in Computer Science (2018), Informatics Center (CIn) at Federal University of Pernambuco (UFPE), Brazil. MSc in Computer Science (2011), Informatics Center at Federal University of Pernambuco, Brazil. He undertook his sandwich PhD (2015-2016) research at the Department of Computer Science and Information Systems (CSIS) of the University of Limerick, Ireland and in collaboration with Lero - the Irish Software Research Centre. His research areas of interest includes: management of software projects, agile software development and empirical software engineering.

Juliana Medeiros is a Professor and Researcher of the Federal Institute of Education, Science and Technology of Paraiba (IFPB), since 2006. She has a Ph.D. in Computer Science from Universidade Federal de Pemambuco. She has more than 20 years of experience with management and development of research projects, and innovation in software companies. Juliana is particularly interested in Requirements Engineering, Agile Methods and Empirical Software Engineering.