An Impact Study of Business Process Models for Requirements Elicitation in XP

Hugo Ordóñez[1], Andrés Felipe Escobar Villada[1], Diana Lorena Velandia Vanegas[1], Carlos Cobos[2], Armando Ordóñez[3(✉)], and Rocio Segovia[1]

[1] Departamento de Ingenieria de Sistemas, Universidad de san Buenaventura, Buenaventura, Colombia
{haordonez,ersegovia}@usbcali.edu.co,
{anfelesvillada,dilove0122}@gmail.com
[2] Departamento de Ingeniería de Sistemas, Universidad del Cauca, Cauca, Colombia
ccobos@unicauca.edu.co
[3] Intelligent Management Systems Group, Foundation University of Popayan, Popayan, Colombia
armando.ordonez@docente.fup.edu.co

Abstract. Many communication problems may appear during requirements elicitation causing that final products do not accomplish client expectations. This paper analyzes the impact of using business processes management notation (BPMN) instead of user stories during requirements analysis in agile methodologies. For analyzing the effectiveness of our approach, we compare the use of user stories vs. BP models in eleven software projects during requirements elicitation phase. Experiments evidence that BPMN models improve quality and quantity of information collected during requirements elicitation and ease that clients specify clearly their needs and business goals.

Keywords: Software · Requirements elicitation · XP (eXtreme Programming) · BPMN (Business Process Management Notation) · User stories

1 Introduction

Software development is an important part of global capital, because it leverages business development and knowledge management [1]. Furthermore, software industry is a white industry that does not pollute and generates well paid jobs [2]. In this scenario, competitive software tools must integrate appropriate methodologies to each project goals [3], Most of the software methodologies must deal with projects with tight deadlines, volatile requirements, and new technologies [4]. One of these methodologies is XP, which is focused on enhancing interpersonal relationships as a key point for successful software development, promoting teamwork, learning developers and good working environment; based on continuous feedback between the client and the development team [5].

In XP, system requirements are gathered in meetings with clients and stakeholders using user stories. User stories describe the functionality of the software to build and customer's needs. However, these user stories are written in natural language, which

O. Gervasi et al. (Eds.): ICCSA 2015, Part I, LNCS 9155, pp. 298–312, 2015.
DOI: 10.1007/978-3-319-21404-7_22

may guide to misinterpretation, given the ambiguity and uncertainty of the collected information [6, 7].

Often in meetings between developers and clients, the dialogue may be not productive enough, causing that requirements are not identified clearly and do not cover all client requirements [8]. Recent studies show that the graphical representation of requirements can contribute to a clearer understanding of the system requirements. This may reduce the ambiguities in the requirements specification. The representations are clear and provide a good overview of the system ("a picture is worth a thousand words") (Pohl et al. 2013). In this paper we evaluate the use of business process (BP) for requirements elicitation. BP can be defined as a set of coordinated activities that aim for achieving a common business goal [9]. In our work, BP are represented using BPMN (Business Process Management Notation) (BPMI, 2006) standard.

For analysis purposes, two strategies (user stories and BPMN models) were applied to eleven software development projects in requirement analysis phase. For each project, a group of analysts evaluated the advantages of the two options for requirements elicitation. This article is organized as follows: in section two some of the most representative works on the subject of research are presented, in section three the proposed method is shown, section four puts forward the validation and results of the proposed technique, finally the conclusions and future work are depicted in section five.

2 Related Work

As previously mentioned, our approach uses user stories. While not all agile methodologies have focus on user stories for requirements elicitation, some of them use these user stories. Consequently, in the present review we analyze some projects using user stories in agile methodologies.

Agile methodologies aims for enhancing interpersonal relationships as a key to success in software development; some of these methodologies promote the use of user stories [6] and constant communication between clients and developers. There are several suggested templates for user stories, however, there is no consensus. In general terms, templates may include one or more of the following parts: name, description, tasks description, and estimate effort in days. Besides, it is recommended that one user story exist for each major functionality; in addition, it is suggested that each developer works in one or two stories each month [10].

Jaqueira et al. affirm that the greatest weakness of user stories is that managers, clients and developers describe non-essential elements instead of describing essential functionalities [11]. The user story should only describe external behavior of the system, which can be understood by the client [12]. As an alternative to the textual requirements elicitation, some studies have used visual models that help to understand to the client how the system works, equally these models also facilitate communication, understanding, problem detection and exploration of scenarios.

Zheng et al. present the adaptation of an agile methodology for implementing a Business Process Management Systems [15], the authors analyze the overall project implementation as well as the impact in the organization. This work uses Scrum methodology and a team formed by developers and customers. The team defines requirements and test deliverables at the end of each iteration. At the end of each iteration, assessment of deliverables is performed. With this assessment, business analysts begin to implement BPMN processes.

Avner et al. compare use case templates vs. graphical notation of BPMN business process that represents the system requirements [16]. This study shows that business analysts and a clients have different visions of processes inside the organization. The experiment concluded that the graphical notation enables faster cognitive understanding. Moreover, clients and analysts understanding improves when use cases templates are used first before BPMN models.

Existing approaches evidence that the use of natural language between clients and developers may produce that that requirements or priorities are not well defined, creating delays in development review and redefinition of the requirements for each user story. Furthermore during user stories creation some information considered "obvious" may be lost. Some approaches propose using BPMN for requirements elicitation but they do analyze the impact on agile methodologies.

3 Our Proposal

Our proposal replaces user stories by BP models during requirements elicitation in agile development methodologies such as XP. BP models allows client and developers to adopt a simple and standard notation to facilitate requirements capture. BP models represent may be created between development team and clients. For a description of the proposal, each element of the user story is taken and the correspondence is established within a model BP, in the BPMN notation. For this, the following considerations are taken into account:

3.1 Title of the Story

Title describes the main objective is clearly propounded. It also defines the interactions between users and the system. In addition the title of the story contains descriptions of all functions (each interaction) of the System. In BPMN the title of the story is represented by the name of the business process model.

3.2 Description of the Story

It describes the sequence of activities identified that make up the flow model for the development process. It is then, the step by step to follow and the description of the activities. In BP, is defined by the name and description attached to each one of the activities or tasks that make up the logical sequence of the BP model.

3.3 Historical Review

The historical revision allows to trace changes made to the requirements and establish responsible and change dates. These changes and its traceability are represented in BP models through the versioning process.

3.4 Reference Documents

Reference documents are a helpful way to elicit new requirements or as an input or output products. In the BP, reference documents are represented as data objects, however, if the document is not part of the process it is represented as an annotation in BP.

3.5 Actors

These are actors perform the sequence of actions that the system must perform to meet each of the set requirements. In the BP, actors are represented by the roles within each process. Roles are defined by Pools and lanes. A pool represents the main participants in a process, and can be divided according to the organizations they belong. A pool contains one or more lanes. The lanes are groups of tasks by area or participant involved in the process.

3.6 Dependencies

Within the user stories some dependencies may be established. For example, some user stories must be previous to some activities, this dependence allows establishing order constraints between stories. In the BP, these dependencies are determined by sending messages between processes and creating threads to indicate the sequence of activities between different roles and tasks. The sequence of actions is structured in terms of steps expressed consistently and sequentially generating the complete specification of the process through the control flow. Consequently, from the description of the individual processes and the use of messaging between processes, pools and lanes may be established dependencies of processes and responsibility of the actors involved in them. Figure 1 provides a graphic representation of the relationship between each of the elements of a user story and the elements within a business process model.

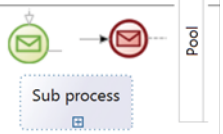
USER STORY	BPMN NOTATION	SYMBOL
Title of the story	Name of Process	
Historical Review	Process versioning	
Reference Documents	Data Objects, but if the document is not part of the process but to elicit reference is treated as annotation.	
Actors	Roles. Task Allocation via Lanes.	
Dependencies	Related threads or processes. Sending messages between processes.	
Description of the Story	Sequence of Activities	

Fig. 1. User Story vs. BP notation

3.7 Adaptation of BP in the XP Methodology

BP generated models are integrated into the later stages of the XP methodology. This section shows how this integration is performed. XP proposes a dynamic life cycle supported on short development cycles (called iterations) with functional deliverables at the end of each cycle [13]. In each iteration a full cycle of analysis, design, development and testing is done (see Figure 2). Next the integration of BP models in the XP methodology is described.

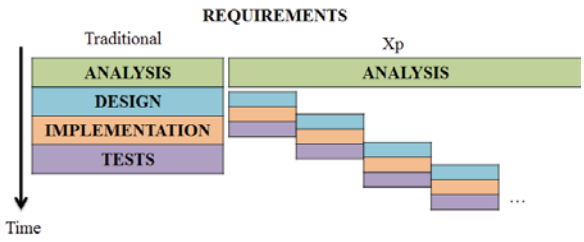


Fig. 2. Stages of development Traditional methodology vs. XP

Exploration: In this phase, the client defines what he needs by writing "user stories" at a high level. Developers estimate the development time of each activity described in the story [14]. This estimation may vary later when the requirements are analyzed in more detail. With the BP model, more details are specified, because this is created in the interview with the client or stakeholder and the BP model is created. Therefore, in this phase, low detail level is maintained as methodology proposes. Functionalities in BP models should be programmed by developers in less that three weeks. If time is greater, the process should be divided into two or more processes.

Planning: In this phase the stakeholder group and the developers agree the order in which user stories should be implemented, and, associated with them, its deliverables. These deliverables are specified in a document called the Delivery plan, or Release Plan. Furthermore at this stage, when developers develop small test programs called "spikes" to validate the client functionality, spikes are created in order to reduce risks and effort estimation. These spikes are used exclusively during requirements phase, and subsequently discarded for further phases; however developing of such spikes may be time consuming. The BP model specifies the order, priority and dependency of the functionality to be developed, therefore, the implementation and deliverables order is defined in advance. Additionally, in this phase, there are some BP tools such as: BonitaSoft and BizAgi that automatically generate small prototypes from defined process models. In these prototypes the developer does not invest more time than the necessary for requirement modelling, he also validates the requirement immediately, reducing the time it might take for validation.

Iterations: in this phase each user story is translated into specific tasks scheduling software functionality, generating deliverables. Because user stories are not defined in sufficient detail in the exploration phase, then a process of analysis is performed with

the client to specify items as constraints or data necessary for the required features. With the BP data details and restrictions are already defined from the exploration phase, this allows estimation of time for the developer to be more accurate, therefore at this stage the process with the client is more a validation process and not about specification.

Deployment: this phase includes Implementation of functionalities and testing, these test may require modifications or adjustments, these changes are known as "fine tuning". Table 1 summarizes the use of User Stories vs. BP models.

Table 1. BP models vs. user stories in XP methodology

User Stories	BP Models
During exploration some aspects has are not defined such as conditions and data. It requires "spikes"	More detail to facilitate the estimation of time and risks, it becomes possible to avoid the generation of spikes
Release Plan based on the prioritization of the stories, agreed between developers and customers.	Release Plan based on prioritized BP models modeled with stakeholder.
They are written by the client, in their own language	Modeled in BPMN, understandable graphic language.
Minimum Detail, programmers make an estimate of the time that it will take to be development	Greater detail, it facilitates estimation, i.e. it needs more time for modeling, but reduces time in following phases.
If calculated effort is higher to 3 weeks by story, so it is divided in 2 or more stories. If one story take less than a week, it is combined other one.	If the estimation is more than 3 weeks, it is divided into two or more processes. The division is done using threads, avoiding the most complex model.
The customer order and group user stories and defines the dependencies.	The customer defines priority and the model gives dependencies over pools and threads.
Each user story is translated into specific tasks.	The sequence and priority of tasks are already defined in the BP flow.
Spikes help to estimate the risk	Some BPMN tools generate quick prototypes which permit validate in the moment when a model is defined.

4 Hypothesis

Some hypotheses were defined in our research:

- H1: requirements elicitation using BPMN improves domain understanding compared to user stories
- H2: The use of BP models after performing user stories in requirements elicitation increases domain understanding.

- H3: Requirements elicitation using user stories after performing the BPMN model increases the understanding of domination by those involved.
- H4: The generation of release plan from BPMN process model specifies in greater detail the deliverables, unlike those that use user stories.
- H5: The activities of plan iterations from BPMN process model specified better, instead of the use of user stories.
- H6: The percentage of implemented tasks in the first iteration of release plan that accomplish customer requirements, modeled with BPMN is greater than the percentage of tasks implemented from user stories

The aim of the first hypotheses (H1 to H3) is to identify which technique improves the understanding between the client and developers. The last three hypotheses aimed to determine if the use of BP models for requirements elicitation, instead of the user story improves the phases release execution plan and plan iterations of the XP methodology.

For the hypothesis validation, an experimental approach was used; this validation is based in the survey by sampling proposed by Liu et al. [17]. For performing the survey, each technique (user stories and BP) where tested separately for requirements elicitation. Then a comparative analysis of the results with each technique is done. Finally the BP based approach was tested during planning and iterations phases

4.1 Experimentation

Both techniques (BP and user stories) were applied for requirements elicitation in 11 projects in analysis phase, with the participation of 58 analysts (software engineers) in the software building department of SENA - Latin American Minor Species (LCMS) Valley Regional and 25 organizations stakeholder from different companies.

To begin the research, participants were trained in XP methodology, user stories creation and BPMN Once analyst groups for each project were trained, two groups were created for the purpose of applying the two techniques in each project. In these projects, both techniques were applied to compare the quantity and quality of information collected. In each of the techniques, the analysts interacted with stakeholders involved in each project through focus groups that collect data using a semi-structured group interviews [18]. The main purpose of the focus group is to arise concerns and reactions in stakeholder and analysts. Moreover, the focus group focuses on the user story that is being generated or the BP model being defined.

Table 2 shows the projects and participants for each technique. For requirements elicitation with user stories we worked with a template based in the work of Quasaimeh [13], for the elicitation of requirements under BP models, BonitaSoft tool was used (see Figure 3). As an example we take one of the user stories and one BP model defined in project A. This project is focused on staff evaluation. The A project aims to create a tool that allows the evaluation of employees performance when a contract ends. This system will help determine the performance of the employees based on a predetermined performance factors. It also allows the user to register and establish assessment criteria and elements to be considered during the assessment.

Table 2. Number of projects and Stakeholders by Hypothesis

Hypothesis	Projects	Number of Analysts		Number of Stakeholder
		User stories	BP Models	
H1	A	2	2	2
	B	2	2	2
	C	4	3	3
	D	4	3	3
H2	E	6	6	2
	F	6	6	5
	G	6	6	4
H3	H	4	4	3
	I	4	4	3
	J	4	4	3
	K	4	4	3
H4	H	6	6	3
	I	6	6	3
	J	6	6	3
	K	4	4	3
H5	H	6	6	3
	I	6	6	3
	J	6	6	3
	K	4	4	3

This in order to generate performance reports and identify weaknesses and strengths of their employees. Table 3 shows the user story created for one of the requirements (R1) Project A. The requirement (R1) consists of recording the performance evaluation of an employee by the manager of human resources. This registration process is performed after the close of each employee and is input to the renewal of the contract each year or semester.

Table 3. Template of user story for Project A, R1

Title: Project A – R1. Register staff evaluation						
Historical review						
Version.	Change description	Author	Date			
Reference document						
Document number:		title:				
Word Format – staff evaluation template						
GENERAL INFORMATION						
Actor:	Human capital manager					
Dependencies:	Staff must be registered.					
Priority:	Low		Medium		High	X
Description of the story						
Human talent manager can record performance evaluation of staff. To do so, he must enter the employee's id and if it is registered with a valid contract, the last active evaluation will be displayed… the system must register the scores per criterion, the date of the assessment and the overall score						
Observations: None						

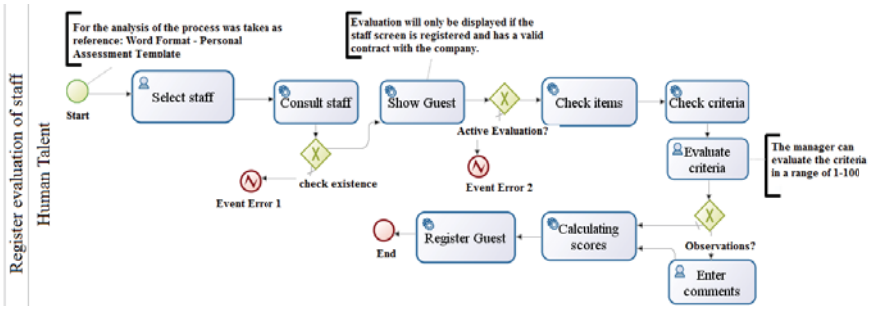


Fig. 3. Example of model for project A

Figure 3 shows the BP model created for requirements (R1) of Project A. This process consists of the evaluation record of the organization's staff. To do this the manager should select human talent through the identification, the employee to whom the assessment will be applied; in this case the system must consult if the employee exists and has a valid contract with the company. Assessment that will be applied, provided such active displays. If the assessment is active for being applied, so the elements and the evaluation criteria can be evaluated in a range of 1-100 points. If necessary observations may be recorded. Eventually the system will calculate and display the scores and the assessment will be recorded.

Upon completion of the requirements elicitation stage, a survey for each applied technique was performed. The survey allows evaluating how each analyst experienced information capture. For more detailed and accurate information on the surveys, the sample was defined considering different groups of analysts working on development projects, in which the hypotheses were validated. The end of the sample is to avoid partial or biased results. For example, Table 4 shows the survey to the evaluation process in user stories.

Table 4. Survey for user stories application

Question
Does the template used to describe user stories allow you to define a concise and complete customer requirements?
Does the template used to describe the user story allowed you to establish if the requirement outlined by the user was required for the system or the organization?
Does the template used to describe user stories allow you to define whether the requirements were consistent? This means, if it was established that the requirements were not contradictory?
When the review of user stories was performed written by another analyst, did you find that these were not ambiguous and that its interpretation was that the stakeholder had said?
Are the requirements documented in the user stories are verifiable, i.e. could do tests to validate their compliance once implemented?
With the template used to write user stories could set the priority or urgency of the requirement according to the objectives of the stakeholder?
It is clear the language used by the stakeholder when the system requirements were specified?
Highlight of the percentage of the total system information that could capture using user stories.
It was adequate the disposition of the customer to specify their requirements?

Later a survey for Stakeholders was also applied. This survey has the following purposes: 1) To measure the percentage of needs and objectives that the stakeholder consider that are specified correctly with the two applied techniques, 2) to determine how the analyst is dealing with customers and/or users communication in order to unify criteria, 3) to identify how stakeholder are included in the process of requirements elicitation and how the analyst makes use of information collected in the interview with the stakeholders.

To validate the hypothesis H4- H6 both groups used the requirements elicitation techniques and subsequently they created iterations plan and release plan both for user stories and BP models approaches. For generating the plan from user stories and BP models, it was used a template generated from Valkenhoef et al. [19]. The release plan in the first iteration includes: priority, that may be low, medium or high, estimated effort at a range of 1-10, where 10 is the largest effort, delivery date. Each process or user story results in specific programming tasks.

Each module of the project was developed by a couple of developers who from the user story or BP model generated activities for each iteration. At the end of this first iteration, an interview with the developers and stakeholder was performed to verify the status of each task and acceptance by the user. The interview allowed the validation of hypotheses H1 to H6 in the following items: 1) Measure the percentage of requirements that were implemented according to the needs of the stakeholder, after the first iteration with each of the techniques applied, 2) determine whether release plan generation from BP model specifies in greater detail the deliveries, unlike that with the use of user stories, 3) the activities of the iterations plan from the BP model specify more completeness, unlike when using user stories. Table 5 shows the questions asked in the survey to the evaluation.

Table 5. Survey for evaluation of release plan

Does the technique provides enough detail assigned for software development?
The selected technique allows you to know restrictions in some requirements during development time?
Did you take into account other sources of requirements different to the assigned technique? Which one?
Does the assigned technique allow you identify other requirements during development?
Does the release plan modified structurally at the end of the first iteration?
The esteemed effort in the release plan is consistent with what you experienced during development?
Does the assigned technique allowed you to establish a precise number or tasks? During development these tasks were increased / decreased?
Does the estimated delivery date of each user story / BP correctly established?

5 Discussion of Results

5.1 Validation of the Hypotheses

H1: According to information collected in the survey, requirements documentation in natural language creates ambiguities and different interpretations by the analyst when other person review the description, furthermore it was not possible to establish whether the requirements contradicted each other, for example when multiple stakeholder were involved in the process of requirements elicitation. Moreover, in the survey of analysts, the percentage of the total system information that could be captured with this technique was between 50% and 80%, indicating that the requirements are not defined completely in this phase. However it was established that the client's participation was 75% favoring validation of requirements. In addition language used by clients to specify his needs was clear, but could be improved.

A second survey with the second analysts group who applied requirements identification using BP models in projects was done. This survey allowed to establish the utility of the use of BP models to represent the client's needs. According to information collected in the survey, documentation requirements through model BP generates more clarity and a single interpretation by the analyst when reviewing models made by another analyst, equally, the BP model, allowed to identify consistency between the requirements using relationships / dependencies between processes. It also clearly defines the assignment of responsibilities to each role or an actor through the Lanes. From the results of this survey it was highlighted that 87% of analysts indicated that the requirements modeled through BP allowed to validate information with stakeholder, verifying that the sequence of activities was adequate, and besides if the information requested in each activity was complete.

H2: eliciting requirements through BP after performing user stories increases domain understanding by those involved. A group of analysts used user stories for requirements elicitation in 3 projects. It was established that 73.33% of the analysts found that user stories written by another analyst were ambiguous and that its interpretation was not the same of the stakeholder. In these surveys it was established that 56% of the analysts said that they captured between 50 and 70% of the requirements. After application of user stories, the same group use the BP model to raise the requirements, at this point it was found that 95% of analysts affirm that BP Model allow them to define a concise and complete requirements client. Now, from the previous survey increased from 22% of analysts who say that when reviewing a model BP has made another analyst team are not ambiguous and its interpretation was that the same that stakeholder had expressed. The rate of requirements capture was between 78% and 89% with this technique, demonstrating that BP model allows to increase by 19% the capture and identification of requirements.

H3: eliciting requirements with user stories after making BP models increases domain understanding by those involved. A group of analysts applied first BP for requirements elicitation in 4 projects, through a survey, I was established that for 80% of the

analysts, to understand the notation of the model was easy, considering they pass first through a training process. Now for the stakeholder, it was found that 83.33% said that the notation was easy to understand, because analyst generates diagrams with stakeholders in real time, which allows both parties to resolve doubts immediately. The validation of this hypothesis also found that 73.33% of analysts indicated that when reviewing a model of BP was made by another analyst, no ambiguities arise and that his interpretation was that the same that the stakeholder said. After applying the BP model, the same group used user stories for requirements elicitation. In this process it was found that 83% of analysts used models that BP had done before, to write the story. In addition, it was noted that 85% of analysts indicated that when they conducted the review of user stories written by another analyst, these were not ambiguous and that its interpretation was that the stakeholder had said. The results therefore validate the four hypotheses allowing to evidence that the BP model can increase by 19%, the number of identified requirements, besides the ambiguity in the requirements specification is decreased by 25%. This demonstrates that using BP models increases the level of understanding between the different actors in the process of building software in XP.

H4 – H5 – H5: Generation Release Plan: with the projects worked in the previous stage, it was evaluated the result the first iteration. The results are listed in Table 6. The results of the first iteration may identify that the technical proposal (BP) reaches 98.33% in the tasks approved by the stakeholder and implemented by the development team, which exceeds 23% of the requirements elicitation technique with user stories in the first iteration, which achieves 75.85 of the Release Plan.

Table 6. Execution results of the first iteration (P: Project T: Technique used, TT: Total specified tasks, NT: Number of implemented tasks, NS: Number of tasks approved by the stakeholder, NI: Number of approved and implemented tasks

P	T	TT	NT	%	NS	%	NI
H	BP	15	15	100	15	100%	100%
I	BP	18	17	94,44	17	94,44	100%
J	BP	15	15	100	14	93,33%	93,33%
K	BP	14	12	85,77	12	85,71%	100%
H	HU	16	13	81,25	13	81,25%	92,86%
I	HU	19	14	73,68	16	84,21%	100%
J	HU	11	9	81,81	9	75,00%	90 %
K	HU	12	8	66,66	6	54,55%	75%

Additionally at this stage, it was used some metrics that allowed to establish the value of client needs. With these metrics, the client will know if in each iteration functionalities are being implemented according to the needs and according to the established time without making many changes between. For this, a set of different

metrics of different related aspects (see Table 8). It can be seen that with the use of BP model, the averaged percentage of requirements that were not approved by the client at the end of the first iteration was 20% lower than the average of the requirements with user stories.

Table 7. Validation Metrics for hypotheses H4-H6

Metrics evaluated by iteration	User story	BP Model
Average requirements completed in iteration	12	14,75
Average incorporated changes and added requirements on the initial scope of the project	4	1
Average completed requirements for total iteration requirements.	75,85	98,43
Average conditions not approved by the client	26,25	6,63

Results obtained when evaluating the hypothesis H4 –H6 demonstrate that proposed technique allows 23% more fully developed requirements by iteration, consequently the functionality defined by the client is widely covered, which significantly reduces the time of software development, because the client does not need to go back to the redefinition of the requirements that were not approved. Furthermore clear and understandable communication between the development team, client and stakeholder in the identification and definition of the model using BP reduces changes in the identified requirements. The reduction in changes avoid delays and allows the development times stays in line with client needs.

5.2 Results of Stakeholder and Customers Surveys

Using BP models for requirements elicitation in an agile methodologies such as XP instead of using user stories, according to the data provides some advantages: improved communication between the analyst and the stakeholder, allowing to establish requirements more clearly. The BP models allow to model the processes in a unified and standardized way, representing the sequence of the process just as it does the stakeholder in your organization. Furthermore, understand a model through this notation by the client does not require much time and additional training. The stakeholder process and describes the BP model and may validate at real time, allowing clarify some requirements that are not understood at the time of the interview. In addition it was found in the study that 87% of stakeholders and customers (see Figure 5) indicated that the notation used by the analyst to specify their requirements were clear and for easier to understand compared to the use of user stories. With BP models, customers can actively participate in the validation requirement, due to the fact that they can add or define clearly the sequence of activities to develop, also assign users to specific tasks as well as the data required in each task.

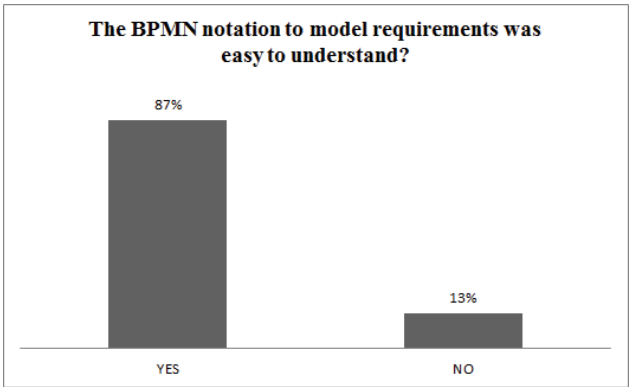


Fig. 4. Understanding of BP model by stakeholder

6 Conclusions and Future Work

Errors in requirements elicitation stage are caused mainly by the ambiguity of the communication between users and the development team. Using BP models for requirements elicitation improved the communication between the analyst, stakeholder and clients. BP notation used was easier to understand that natural language. Furthermore, using BP models was much more productive and easy to understand by 80% of analysts, which represents a major improvement in this activity. Analysts also emphasize that BP models can be more productive, because they elicit a greater number of requirements compared to user stories. BP models improve interpretation and understanding, upon review of the requirements that have been elicited by other analyst team. This is because the BP models are defined a common and standardized language allowing to clearly identify the sequence of activities and what the user expects the system.

In relation to the support among the techniques of elicitation of requirements, we can say that the user story itself did not allow fully clarify requirements compared to BP models. The description of requirements in user stories created from BP models increased understanding in 11.67 % due to the fact that BP models allowed to make clearer the redaction of the user story by analysts. However, it is convenient to use two techniques for requirements elicitation, this is because this will add an additional activity that does not contribute more to the process would and this will affect one of the agile manifest that supports on simplicity. The generation and implementation of release plan from BP model is faster and generates a higher percentage of implemented and approved requirements at each iteration, this allows that the development to be best fit to the client needs. We can conclude that BPs is appropriate to give a top-level view of the processes supported by one or more organizations; on the other hand, user stories may be used in XP for describing the user-centric requirements of the system under design and development. That means that BPs and user stories should be better considered as complementary techniques.

References

1. Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: Software development in startup companies: A systematic mapping study. Information and Software Technology **56**, 1200–1218 (2014)
2. Nguyen-Duc, A., Cruzes, D.S., Conradi, R.: The impact of global dispersion on coordination, team performance and software quality – A systematic literature review. Information and Software Technology **57**, 277–294 (2015)
3. Christensen, H.B., Hansen, K.M., Kyng, M., Manikas, K.: Analysis and design of software ecosystem architectures – Towards the 4S telemedicine ecosystem. Information and Software Technology **56**, 1476–1492 (2014)
4. Losada, B., Urretavizcaya, M., Fernández-Castro, I.: A guide to agile development of interactive software with a "User Objectives"-driven methodology. Science of Computer Programming **78**, 2268–2281 (2013)
5. Thiyagarajan, P.S., Verma, S.: A Closer Look at Extreme Programming (XP) with an Onsite-Offshore Model to Develop Software Projects Using XP Methodology. In: Berkling, K., Joseph, M., Meyer, B., Nordio, M. (eds.) Software Engineering Approaches for Offshore and Outsourced Development. LNBIP, vol. 16, pp. 166–180. Springer, Heidelberg (2009)
6. Blom, M.: Is Scrum and XP suitable for CSE Development? Procedia Computer Science **1**, 1511–1517 (2010)
7. Domann, J., Hartmann, S., Burkhardt, M., Barge, A., Albayrak, S.: An Agile Method for Multiagent Software Engineering. Procedia Computer Science **32**, 928–934 (2014)
8. Wood, S., Michaelides, G., Thomson, C.: Successful extreme programming: Fidelity to the methodology or good teamworking? Information and Software Technology **55**, 660–672 (2013)
9. Chinosi, M., Trombetta, A.: Computer Standards & Interfaces BPMN : An introduction to the standard. Computer Standards & Interfaces **34**, 124–134 (2012)
10. Tong, K.I.: Chapter 8 Managing Software Projects with User Stories. In: Essential Skills for Agile Development, pp. 217–252 (2010)
11. Jaqueira, M.L.A., Aranha, E., Alencar, F., Castro, J.: Using i Models to Enrich User Stories Objectives of the research. iStar. In: Proceeding ICSE 2002 Proceedings of the 24th International Conference on Software Engineering, pp. 695–696 (2013)
12. Layman, L., Williams, L., Damian, D., Bures, H.: Essential communication practices for Extreme Programming in a global software development team. Information and Software Technology **48**, 781–794 (2006)
13. Qasaimeh, M.: Extending Extreme Programming User Stories to Meet ISO 9001 Formality Requirements. Software Engineering and Applications, pp. 626–638 (2011)
14. Kent Beck, A.-W.: Extreme Programming Explained: Embrace Change, 2nd edn., pp. 178–200. Addison Wesley (2012)
15. Zheng, G.: Implementing a business process management system applying Agile development methodology: A real-world case study. Thesis. Erasmus Universiteit Rotterdam (2012)
16. Ottensooser, A., Fekete, A., Reijers, H.A., Mendling, J., Menictas, C.: Making sense of business process descriptions: An experimental comparison of graphical and textual notations. Journal of Systems and Software **85**(3), pp. 596–606 (2012)
17. Liu, Y., Tian, G.-L.: A variant of the parallel model for sample surveys with sensitive characteristics. Computational Statistics & Data Analysis **67**, 115–135 (2013)
18. Singh, G.N., Priyanka, K., et al.: Estimation of population mean using imputation techniques in sample surveys. Journal of the Korean Statistical Society **39**(1), 67–74 (2010)
19. Van Valkenhoef, G.T., Tervonen, T., de Brock, B., Postmus, D.: Quantitative release planning in extreme programming. Information and Software Technology 53(11), 1227–1235 (2011)