

Comparing Value-Driven Methods: an experiment design

Eric Souza
NOVA LINCS,
Departamento de Informática
Faculdade de Ciências e Tecnologia,
Universidade Nova de Lisboa,
Portugal
er.souza@campus.fct.unl.pt

Silvia Abrahão
Departamento de Sistemas
Informaticos y Computacion,
Universitat Politecnica de Valencia,
España
sabrahao@dsic.upv.es

Ana Moreira
NOVA LINCS,
Departamento de Informática
Faculdade de Ciências e Tecnologia,
Universidade Nova de Lisboa,
Portugal
amm@fct.unl.pt

João Araújo
NOVA LINCS, Departamento de Informática
Faculdade de Ciências e Tecnologia,
Universidade Nova de Lisboa, Portugal
joao.araujo@fct.unl.pt

Emilio Insfran
Departamento de Sistemas Informaticos y
Computacion,
Universitat Politecnica de Valencia, España
einsfran@dsic.upv.es

ABSTRACT

A business model is a representation of an organization with a particular point of view. It is common to find different types of models to describe the business. However, methods to create business models representing an economic point of view have only emerged over the last few years in the scientific community. Such methods aid business specialists improving the economics understanding of the business, helping both defining more efficient business strategies and better aligning the information technology systems with the business. This paper aims at describing the design of an experiment to compare two methods to specify economic values (e^3 value and value-driven development). Our experiment design allows predicting the acceptance of a particular method in practice, based on the effort of applying the method, the quality of the artifacts produced, and the user perceptions with regard to the quality of the method.

CCS Concepts

• Information systems → Language models

Keywords

value model; value-driven; experiment design

1. INTRODUCTION

Models to describe the daily behavior of the business, or what the Object Management Group¹ calls “business in motion” [1], are common. This business behavior, in general, is represented by using some well-known description languages such as workflow diagrams [2], UML activity diagrams [3], and BPMN [4]. However, methods to create business models representing an economic point of view have emerged over the last years in the scientific community. In principle, the reason for this is that an organization needs to make money to keep going in a competitive market. In addition, business specialists need to improve the economics understanding to define more efficient business strategies and provide a better alignment of the information technology with the business. These models can then be used to define the requirements of the underlying information systems. In general, these methods are conceptual requirements approaches,

¹ The Object Management Group is an international technology standards consortium.

inspired in business science, requirements engineering, and conceptual modeling techniques [5], whose aim is to show how economic values are created and exchanged in an inter-organizational network [6].

Value means the relationship between satisfying needs and expectations and the resources required to achieve them [7]. Value is the reason for companies and people to trade with each other, offering money to get something in return. Therefore, a value model represents a business model from an economic perspective, and must determine the economic value exchanged and their intervenients [8]. This understanding facilitates the aligning of the software requirements specifications with the value exchanges [9].

There is a number of approaches to represent value. Kundisch and John [10] classify 12 different business model representations: activity system map, business model ontology, causal loop diagram, value map, value net, strategic business model ontology (SBMO), and business model ontology (BMO), among others. However, there is no empirical evidence about which of these methods is more effective under what circumstances. For practitioners to consider adopting a given value-driven method, they must know its effectiveness and how it compares with others.

The goal of this paper is therefore to describe the design of an experiment aimed at comparing two methods to express economic values: e^3 value [11], which is a widely established and applied business model representation, and our own Value-Driven Development method (VDD), which has been proposed recently. This design is also valuable because it can be replicated for other methods.

The remainder of this paper is organized as follows. Section 2 introduces e^3 value and VDD. Section 3 presents the design of a controlled experiment aimed at comparing the effectiveness, efficiency, perceived ease of use, perceived usefulness, and intention to use of several groups of users employing both methods for creating a value model. Finally, Section 4 concludes this paper and summarizes directions for further work.

2. METHODS TO BE COMPARED

This section summarizes the e^3 value [12] and VDD, comparing them with respect to their concepts and processes.

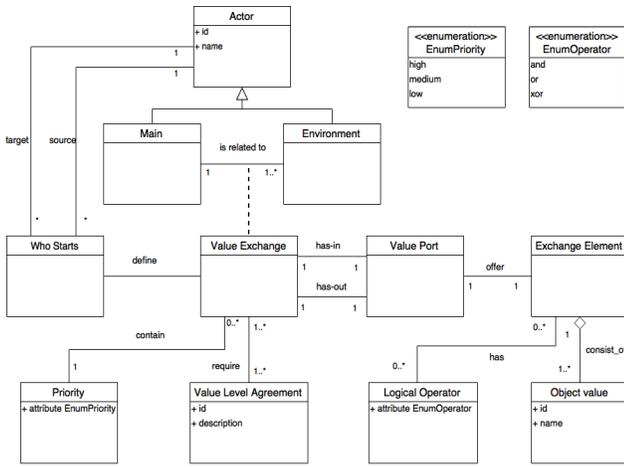


Figure 3. Dynamic Value Description metamodel.

From the actors relationship, a value exchange is performed. It shows economic reciprocity through two value ports (arrows connected to value exchange), one for entry and one for exit, which point to value elements (money, goods, services or information). If there are many value objects in the same value port, the business analyst must use logical operators (“AND”, “OR”, and “XOR”) to detail the relationships among them.

In addition, the business analyst also defines who starts the value exchanges through a configuration of arrows between the main actor and the environment actor. It is important to notice that during the DVD modeling, the business analyst is able to focus on each actor individually in order to give more attention and details to the actor which must be analyzed. Thus, the analyst sets who is the main actor and a given support tool will display it as the central node of the model, dynamically. Each value exchange needs a level of agreement between the ones involved. This level of agreement is a particular business aspect that must be minimally agreed among the actors in order to enable the value exchanges.

Finally, the DVD model describes a prioritization of value exchanges using colors. The red color means high priority, yellow color means medium priority, and blue color means low priority. These priorities are set by business analyst according to the return of investment of the value exchanges in the business.

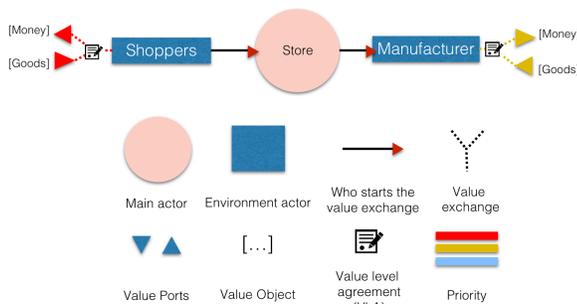


Figure 4. Dynamic Value Description model example.

2.3 Comparing e³value and VDD

This section compares the e³value and the value-driven development methods. Table 1 presents a mapping between e³value and VDD concepts.

Table 1. Concepts mapping.

#	e ³ value concepts	DVD Concepts
1	Elementary actor	Main actor or actor
2	Composite actor	Main actor or actor
3	Market segment	Main actor or actor
4	Value interface	Aggregate in value exchange
5	Value transfer	Aggregate in value exchange
6	Value port	Value port
7	Value object	Value element
8	Value exchange	Value exchange
9	Value transaction	Aggregate in value exchange
10	Value activity	-
11	Start stimulus	Who starts
12	Stop stimulus	Who starts
13	AND element	Logical operators in exchange element
14	OR element	Logical operators in exchange element
15	Connect element	-
16	-	Value level agreement
17	-	Priority

We observed that the DVD model (from the VDD method) describes two new concepts in relation to the e³value: value level agreement (VLA) and priority. In contrast, the e³value model has the value activity concepts, not offered by the DVD model. The VLA defines the business constraints based on the business strategies. For example, a company of the feeding segment provides food fresher than its competitors, as a business strategy. Thus, to provide fresher food, it is essential that its suppliers also deliver fresh ingredients. Therefore, the business analyst can specify a VLA by defining the acceptable time of receipt of these ingredients. Regarding the information system development, the complexity of a software system is determined by its functionality (i.e., what the system does) and by global requirements on its development, such as operational costs, performance, reliability, maintainability, portability, robustness [17]. These global requirements are known as Non-functional Requirements (NFR) and they typically refer to the operational quality of a system, as well as the constraints imposed on a solution [18]. Thus, we can define a VLA as an NFR at the business abstraction level.

In addition, as information systems are usually developed using iterative and incremental processes, the value exchanges prioritization may facilitate the scope definition of each iteration, aligning the system development with the business needs and the time to market. Despite having fewer concepts, the DVD model represents several e³value' concepts but some of these concepts are represented in a partial way or with a different meaning (e.g., UCMs elements). However, for the various case studies developed, the concepts offered by DVD have been proved sufficient.

3. EXPERIMENT DESIGN

This section presents the design of a controlled experiment aimed at comparing the value-driven development method against the e³value method. We followed the guidelines proposed by Wohlin *et al.* [19].

3.1 Experimental Planning

3.1.1 Experiment Goal

According to the Goal-Question Metric (GQM) approach [20], the goal of this experiment is to analyze VDD and e³value **for the purpose of comparing them with respect to their effectiveness, efficiency, perceived ease of use, perceived usefulness and intention to use in order to obtain high-quality value models from the point of view of novice business analysts and software engineers, in the context of undergraduate and postgraduate students in Business Management and Computer Science.**

The broad research questions addressed by the experiment are:

- RQ1: Is the actual efficacy of VDD higher than the actual efficacy of e³value?
- RQ2: Is the perceived efficacy and intention to use of participants applying VDD higher than that of e³value?

The context of the experiment is the creation of a business value model for specific software systems. This context is determined by the product to be developed and the subjects' selection.

3.1.2 Experimental Objects

The software systems to be developed were selected from the literature [21], [22]. Two experimental objects were selected from the requirements specifications of the following two systems:

- **Waste management (O1):** It describes the business where waste is traded between an exporter and an importer. In the majority of cases the exporter has to pay the importer for the waste handling. However, there are some cases where the waste can be traded like a regular good, for example, when the waste is recycled.
- **Wireless access provisioning (O2):** It describes the business where a hotel would like to offer wireless connectivity to businessmen as an additional service.

To assess the complexity of the models used and to identify possible mistakes, we plan to carry out a pilot experiment with a small group of PhD students at UPV.

3.1.3 Participant's Selection

The context of this experiment is the evaluation of value-driven development methods from the perspective of novice modelers. Although experienced modelers and practitioners are desired, we focus on the profile of novice modelers since one of our goals is to provide a value-driven development method that will help less experienced modelers to specify value models. In addition, according to the Technology Transfer Model proposed by Gorschek *et al.* [23], it is recommended to first perform initial evaluations in lab environments before the realization of realistic evaluations in industrial environments.

Value models can be produced by business analysts or software engineers. The following groups of participants are therefore identified in order to facilitate the generalization of results:

- Undergraduate students, all Computer Science students from the Software Engineering intensification at the Universitat Politècnica de València. These students will attend the "Requirements Engineering" course from September 2016 to January 2017, during this time they will have 8h of lectures on business modeling and value-driven development.
- Master's students, enrolled on the Master's Degree in Engineering and Technology of Software Systems at the Universitat Politècnica de València. These students will

attend the "Empirical Software Engineering" course from September to November 2016.

- Undergraduate students, all Computer Science students from the Software Engineering intensification at the Universitat Politècnica de València. These students will attend the "Software Quality" course from February to June 2017. One of the main topics of this course is to evaluate the quality of models obtained through the software development process. A teaching unit on the evaluation of value-driven development methods will be added to the course program.
- Master's students, enrolled on the Master's Degree in Software Engineering at the Universidade Nova de Lisboa. These students will attend the "Software Engineering" course from September 2016 to January 2017.
- Undergraduate students, all business management students at the Universidade Federal de Pernambuco. These students will attend the "Information System" course from January to July 2017.

These courses were selected because the preparation and training and the experimental task itself fit their scope. We take a *convenience sample* (i.e., all the students available in the class). The original experiment will be conducted in the Requirements Engineering course and the other groups will be exact and/or differentiated replications. This will allow us to build a body of knowledge about these value-driven development methods. As Basili *et al.* [24] suggested, relevant and credible results can only be obtained by replicating the experiments since single studies rarely provide definitive answers.

3.1.4 Selection of Variables

The independent variable of interest is the use of each value-driven method with nominal values: VDD and e³value. Hence, the experiment use two treatments: the creation of a value model for two software systems using VDD and the creation of a value model for the same systems using e³value. The experimental data collected allows comparing the effects of both treatments.

There are two types of dependent variables in which the treatments are compared: performance-based and perception-based variables. Performance-based variables assess how well the participants perform the experimental task. They are used to evaluate the *actual efficacy* of the value-driven development methods. Perception-based variables assess the participants' perceptions of their performance and their subsequent intention to use VDD or e³value. These variables are used to evaluate the *perceived efficacy* of these methods, as well as their likely adoption in practice.

There are two performance-based variables:

Effectiveness: It is calculated with the Jaccard index (see formula (1)) that measures similarity between sample sets and is defined as the size of the intersection divided by the size of the union of the sample sets. Given two models, A and B, the Jaccard index measures the overlap that A and B share with their elements. In our case, we will calculate the Jaccard index between an agreed solution among experts (A) (for the value models obtained with VDD and e³value) and the solution given by each participant (B). A Jaccard index of 0 represents no overlap between the solutions, while 1 indicates that they contain the same results.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

Efficiency: It is the time required to apply the method.

There are also three perception-based variables, which are based on the Technology Acceptance Model (TAM) [25], since TAM is one of the most widely applied theoretical models when analyzing user acceptance and usage behavior of emerging information technologies, and has empirical support through validations and replications [26]. This model has been also applied previously to evaluate requirements modeling methods [27]. The perceived efficacy [25] of the method can be broken down into the following subjective dependent variables:

Perceived Ease of Use (PEOU): It refers to the degree to which a person believes that learning and using a particular value-driven method would be free of effort.

Perceived Usefulness (PU): It refers to the degree to which a person believes that using a specific method will increase his or her job performance within an organizational context.

Intention to Use (ITU): It refers to the extent to which a person intend to use a particular method. This last variable represents a perceptual judgment of the method's efficacy – that is, whether it is cost-effective and is commonly used to predict the likelihood of acceptance of a method in practice.

These three subjective variables will be measured by using a Likert scale questionnaire with a set of 12 closed-questions: 3 questions for perceived ease of use (PEOU), 6 questions for perceived usefulness (PU) and 3 for intention to use (ITU). The questionnaire can be found here: <http://bit.ly/2ak1wLS>. The closed-questions were formulated by using a 5-point Likert scale, using the opposing statement question format. In other words, each question contains two opposite statements representing the maximum and minimum possible values (5 and 1), where the value 3 is considered to be a neutral perception. The aggregated value of each subjective variable will be calculated as the arithmetical mean of the answers to the questions associated with each perception-based variable.

3.1.5 Hypotheses Formulation

We formulated several null hypotheses, which were formulated in a one-tailed manner, since we want to analyze the effect of the use of VDD on the variables. Each null hypothesis and its alternative are presented as follows:

- **H1₀:** There is no significant difference between the effectiveness of VDD and e³value / H1a: VDD is significantly more effective than e³value.
- **H2₀:** There is no significant difference between the efficiency of VDD and e³value / H2a: VDD is significantly more efficient than e³value.
- **H3₀:** There is no significant difference between the perceived ease of use of evaluators applying VDD and e³value / H3a: VDD is perceived as easier to use than e³value.
- **H4₀:** There is no significant difference between the perceived usefulness of VDD and e³value / H4a: VDD is perceived as more useful than e³value.
- **H5₀:** There is no significant difference between the intention to use of VDD and e³value / H5a: VDD is perceived as more likely to be used than e³value.

3.1.6 Factors to be Controlled

Although Method is the only factor of interest in this empirical investigation, other factors may affect the participants' performance in an undesirable way, thus confounding the Method effect. These factors have to be controlled so that only the effect of the Method factor, if there is any, is observable:

System Domain. The complexity of the software requirements considered in the tasks may have a confounding effect on the results. The application domain of the tasks could also be a confounding factor that could affect the subjects' comprehension.

Order of Methods. The order in which the subjects apply the methods may produce learning effects, which may bias the results.

3.1.7 Design of the experiment

The experiment is planned as a balanced within-participant design with a confounding effect, signifying that the same participants apply both methods with both experimental objects in a different order. We plan to establish four groups (each of which will apply one method to one experimental object) and the participants will be randomly assigned to each group. Table 3 summarizes the design of the experiment.

The within-participant experimental design is intended to minimize the impact of learning effects on the results, since none of the participants repeat any treatment or experimental object during the execution. Other factors which may also be present need to be controlled, since they might influence the results, i.e., the complexity of experimental objects. The comprehension of the software systems requirements to be modeled may also affect the application of both methods. We attempted to alleviate the influence of this factor by selecting two representative software systems with software system requirements of a reasonable complexity. The complexity of the software systems selected made them suitable for application in the time slot available for the execution of the experiments (2 hour sessions).

Table 3. Experiment Design

Sessions	1st Session			
	Detailed training with VDD and e ³ value			
	2nd Session		3rd Session	
	VDD and e ³ value quick training			
Group of participants (sample size = 4n participants)	G1: VDD in O1	G2: VDD in O2	G1: e ³ value in O2	G2: e ³ value in O1
	G3: e ³ value in O2	G4: e ³ value in O1	G3: VDD in O1	G4: VDD in O2
	Post-experimental questionnaire		Post-experimental questionnaire	

3.1.8 Instrumentation

The experimental task was structured to allow the comparison of both methods. Depending on the method, the task was composed of the method activities that help to achieve its purpose. After applying the method, the participants have to fill in a post-experimental questionnaire with subjective questions regarding the method.

We have defined only one experimental task (create the value model) of which its steps vary according to the value model that the subjects will create. We will offer a training session to explain the concepts of value models and how they are created. During the experimentation, we will offer the requirements specifications to the subjects (see Section 3.1.2). The requirements specifications describe how the business works. With these specifications in their hands, the participants will create a particular value model (DVD or e³value) and will register the start time and the end time for each step performed.

Figure 5 presents the steps to create the e³value model. In this case, the steps are:

Step 1 - Identify scenarios: Scenarios are short textual sentences, meaning the product, service, or experience expected by a customer. Therefore, the goal of this step is that the participants write a scenarios list.

Step 2 - Identify actors: The participants will identify who offers and who receives the product, service or experience expected from the scenarios list and they will create a list of actors.

Step 3 - Create value model: With the scenarios list and the actors list in their hands, the subjects will create the initial e^3 value model by using the products and services mentioned in the scenarios list and the actors described in actors list.

Step 4 - Identify UCMs: The participants will insert the UCM's elements, representing the paths of all scenarios in the e^3 value model. In other words, they will insert the start stimulus, stop stimulus, AND element, OR element, and connect element in the e^3 value model.

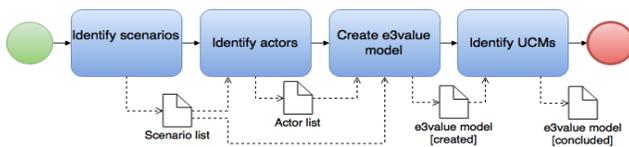


Figure 5. e^3 value process.

In the case of creating the DVD model, the participants will follow the steps from the VDD process (see Figure 6). These steps are:

Step 1 - Specify actors: Participants start the DVD model by describing the main actor (the focus of their analysis) and their related environment actors. Thus, the participants will create a DVD model like a mind map, where the main actor is the central node and the environment actors are the leaf nodes. Due to this “main actor” focus, the DVD model shows only the environment actors who directly interact with it. Thus, the participants will be required to create as many DVD models as necessary to represent the whole business.

Step 2 - Set value exchanges: Participants will update the model by adding the value exchanges. During this activity, participants define the value element related to each value port.

Step 3 - Set who starts each value exchange: Participants will define which actor starts the value exchange. Here, it is important to check if the value elements are specified in the correct value port.

Step 4 - Set value level agreement: Participants define the criteria required for value exchanges to be performed. This step is very important that participants understand the business constraints related to each value exchange.

Step 5 - Prioritize the value exchanges: Participants prioritize each value exchange according to the expected return of investment (ROI). This is a subjective prioritization as participants will set the value exchanges priority in relation to other value exchanges without the use of any mathematical model.

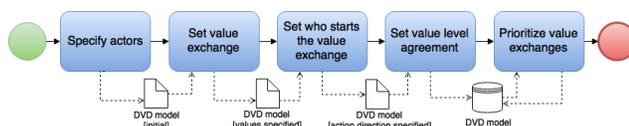


Figure 6. Dynamic Value Description process.

The reason for this is that there is a lack of economics results at this moment of the analysis.

Once the value model is created, the participants will answer the post-experimental questionnaire. Hence, we will be able to evaluate the performance-based variables (effectiveness and efficiency) by comparing the value model they created against the value model created by experts⁴ and by analyzing the time registered to perform each experimental step. In addition, we will evaluate the perception-based variables (perceived ease of use, perceived usefulness, and intention to use) from the responses received in the post-experimental questionnaire.

The experimental material is composed of a set of documents required to support the experimental tasks and the training sessions, along with the post-experimental questionnaire. The training materials include: i) a set of slides containing an introduction to business modeling and value-driven development; ii) a set of slides describing the VDD method, along with an example of its application; and iii) a set of slides describing the e^3 value method, with an example of its application.

The documents supporting the experimental tasks include:

- Two kinds of booklets covering the two possible combinations of both the value-driven development method and the experimental objects (VDD-O1, VDD-O2, e^3 value-O1, e^3 value-O2). The purpose of these booklets is i) to describe the experimental tasks to be performed; ii) to describe the software system requirements; and iii) to gather the data from the experimental task.
- An appendix containing a guideline to help the participants to apply the value-driven development method.

The post-experimental questionnaire contains a set of closed-questions that allows participants to express their opinion on the ease of use, usefulness, and their intention to use of the method in the future. We also include two open questions to obtain the participants’ feedback regarding the changes they would make to improve the method and their reasons for using a given method in the future (if any). This questionnaire will be online, using Google Forms and the data collected will be kept anonymously. All the experimental material will be created in Spanish and Portuguese, since these are the participants’ native languages.

3.1.9 Threats to Validity

We must consider certain issues which may threaten the validity of this experiment. With regard to *internal validity*, the main threats are: learning effect, participant experience, information exchange among participants, and understandability of the documents.

The learning effect is alleviated by ensuring that groups of participants will apply the two methods to different experimental objects in a different order. We also plan to assess the effect of order of system domain and order of methods by using statistical tests. Participants’ experience is alleviated as none of the participants have any experience in value-driven development. We plan to confirm this fact by asking the participants about their experience with value-driven development methods.

To minimize the information exchange among participants, they will be monitored by the experimenters to avoid communication

⁴ In the case of e^3 value, the value models which will be used in comparison were found in literature.

biases while performing the tasks. However, this might affect the results since the experiment will take place over more than one day, and it is difficult to be certain whether or not the participants will exchange any information. To alleviate this situation, at least to some extent, participants will be asked to return all the material at the end of each task. Finally, understandability of the material will be alleviated by performing a pilot study. In addition, we will clear up all the misunderstandings that may appear in each experimental session.

With regard to *external validity*, the main threats are: representativeness of the results and the size and complexity of the tasks. The representativeness of the results may be affected by the software systems used and the participant's context selected.

With regard to the selection of software systems, we attempted to alleviate this by considering a set of artifacts with similar size and complexity, and which contains representative artifacts of an existing value-driven development method (i.e., e³value).

Despite the fact that the planned experiments will be performed in an academic context (undergraduate and Master's students), the participants' performance can be considered to be representative of novice modelers since the kinds of students involved will be soon integrated into the industry's market. As further work, we plan to conduct more experiments involving practitioners in order to assess how the experience level would impact on the obtained results. Also, since only internal replications will be conducted, more external replications need to be conducted by other experimenters in other settings to confirm these results. In order to address the aforementioned limitations, these external replications will involve participants from different contexts and also with different levels of experience in value-driven development.

The size and complexity of the tasks may also affect the external validity. We use relatively small tasks that would be applied in a few representative software artifacts since a controlled experiment requires participants to complete the assigned tasks in a limited amount of time.

With regard to *construct validity*, the main threats are: the measures that will be applied in the data analysis and the reliability of the questionnaire. We attempt to alleviate this threat by using measures that are commonly applied in other software engineering experiments. In particular, the Effectiveness was measured using the Jaccard index, also known as the Jaccard similarity coefficient, which has commonly been used to measure the similarity and diversity of sample sets. The subjective variables are based on the Technology Acceptance Method (TAM), a well-known and empirically validated model for the evaluation of information technologies [25], [26]. The reliability of the questionnaire will be tested by applying the Cronbach test.

With regard to *conclusion validity*, the main threats are: the data collection and the validity of the statistical tests applied. With regard to the data collection, we plan to apply the same procedure in each individual experiment in order to extract the data, and ensure that each dependent variable is calculated by applying the same formula. With regard to the validity of the statistical tests proposed, we chose the most common tests that are employed in the empirical software engineering field due to their robustness and sensitivity [28].

3.2 Operation and Execution

This section describes the experimental operation, including the preparation, execution, data recording and data validation.

With regard to the operation of the experiment, the experiment is planned to be conducted in three sessions (Table 3 shows the details for each session). On the first session, the participants will be given a detailed training on the methods to be applied and also on the tasks to be performed in the execution of the experiment. In this session, they will perform a practical session in which they will specify a value model using both methods.

On the second and third sessions, the participants will be given an overview of the training before applying each value-driven development method to the experimental objects (O1 or O2). We will establish a slot of 90 min with no time limit for any of the methods to be applied. In addition, we will allow the participants to continue the experiment even though these 90 min is not enough in order to avoid a possible ceiling effect.

With regard to the experiment execution, the experiment will take place in a single room, and no interaction among participants will be allowed. The experimenter will clarify possible questions that may arise during the sessions.

With regard to the data validation, we plan to verify that the participants complete the two experimental sessions. Data points containing only one session will be discarded. If this occurs, other data points may also be discarded in order to maintain the balanced design shown in Table 3 (i.e., having exactly the same number of participants in each group).

3.3 Data Analysis & Interpretation

This section introduces the statistical tests that will be used to analyze the data collected: the influence of the method on the dependent variables and the effect of system domain and order of method. These tests have been chosen because they are very robust and sensitive, and have been used in previous experiments similar to ours, e.g., [29], [30]. As usual, in all the tests we have decided to accept a probability of 5% of committing a Type-I-Error [19], i.e., of rejecting the null hypothesis when it is actually true.

3.3.1 Influence of Method

We plan to use boxplots and statistical tests to analyze the data collected. In particular, we will test the normality of the data distribution by applying the Shapiro-Wilk test. The results of the normality test will allow us to select the proper significance test in order to test our hypotheses. When data is assumed to be normally distributed ($p\text{-value} \geq 0.05$), we will apply the parametric one-tailed t-test for independent samples [31]. However, when data could not be assumed to be normally distributed ($p\text{-value} < 0.05$), we will apply the non-parametric Mann-Whitney test [32].

3.3.2 Influence of Order of System Domain and Order of Method

To test the influence of order of system domain and order of method (both independent variables), we plan to use a method similar to that proposed by Briand *et al.* [30]. We will use the Diff function:

$$\text{Diff}_x = \text{observation}_x(A) - \text{observation}_x(B) \quad (2)$$

where x denotes a particular participant, and A, B are the two possible nominal values of an independent variable. We plan to create Diff variables from each dependent variable e.g., Effectiveness_Diff(VDD) will represent the difference in effectiveness of the subjects who used VDD first and e³value second. On the other hand, Effectiveness_Diff(e³value) will represent the difference in effectiveness of the participants who

used e^3 value first and VDD second. The aim is to verify that there are no significant differences between Diff functions since that would signify that there is no influence in the order of the independent variables. We also plan to apply the Shapiro-Wilk test to prove the normality of the Diff functions. The hypotheses related to the Diff functions are two-sided since we do not make any assumption about whether one specific order would be more influential than another. We plan to verify these hypotheses by applying the parametric two-tailed t-test for independent samples or the non-parametric Mann-Whitney test depending on the results of the normality test.

4. CONCLUSIONS AND FURTHER WORK

We have presented two early requirements modeling methods to represent a business from an economic point of view: value-driven development method and e^3 value. Moreover, we also have presented an experiment design aimed at comparing these two methods. Our experiment design allows predicting the acceptance of a particular method in practice, based on the effort of applying the method, the quality of the artifacts produced, and the user perceptions with regard to the quality of the method. In future work, we plan to perform a family of experiments by using the proposed experiment design.

5. ACKNOWLEDGMENTS

This research is supported by the Value@Cloud project (MINECO TIN2013-46300-R), NOVA LINCS Research Laboratory (Ref. UID/CEC/04516/2013), and programa Ciência sem Fronteiras - CAPES.

6. REFERENCES

- [1] R. Veryard, *Six Viewpoints of Business Architecture*. Leanpub, 2015.
- [2] R. Passonneau, K. Kukich, and J. Robin, "Generating summaries of workflow diagrams", in *International Conference on Natural Language Processing and Industrial Applications*, Moncton, Canada, 1996.
- [3] M. Dumas and A. H. M. ter Hofstede, "UML Activity Diagrams as a Workflow Specification Language", in *UML: Modeling Languages, Concepts, and Tools*, vol. 2185, no. 7, Toronto, Canada, 2001, pp. 76–90.
- [4] S. A. White, "Introduction to BPMN", IBM, 2004.
- [5] A. Rasiwasia, "Meta Model for Business Model Design: Designing a Meta model for E3 value model based on MOF", Stockholm, Sweden, 2013.
- [6] J. Gordijn and H. Akkermans, "Designing and evaluating e-business models", *IEEE Intell. Syst.*, vol. 16, no. 4, 2001.
- [7] The Institute of Value Management, "What is Value?." [Online]. Available: <https://ivm.org.uk/what-is-value-management>. [Accessed: 21-Jul-2016].
- [8] J. Gordijn and J. M. Akkermans, "Value-based requirements engineering: exploring innovative e-commerce ideas", *RE*, vol. 8, no. 2, pp. 114–134, Jul. 2003.
- [9] V. Kartseva, J. Gordijn, and Y.-H. Tan, "Inter-organisational Controls as Value Objects in Network Organisations", in *Experiences with Model Reuse: Non-Functional Requirements Catalogues for Ubiquitous Systems*, vol. 4001, no. 23, 2006, pp. 336–350.
- [10] D. Kundisch and T. John, "Business Model Representation Incorporating Real Options: An Extension of e^3 -Value", in *45th HICSS*, 2012, pp. 4456–4465.
- [11] J. Gordijn, "E3-value in a Nutshell", International Workshop on e-Business Modeling, 2002.
- [12] J. Gordijn, "Value-based Requirements Engineering", Phd Thesis, 2002.
- [13] R. J. A. Buhr and R. S. Casselman, *Use case maps for object-oriented systems*. Prentice-Hall, Inc., 1995.
- [14] V. Pijpers, P. de Leenheer, J. Gordijn, and H. Akkermans, "Using conceptual models to explore business-ICT alignment in networked value constellations", *RE*, vol. 17, no. 3, pp. 203–226, Oct. 2011.
- [15] F. Wanderley, A. Silva, and J. Araújo, "Evaluation of BehaviorMap: A user-centered behavior language", in *9th RCIS*, 2015, pp. 309–320.
- [16] S. Ambler, *Agile modeling: effective practices for extreme programming and the unified process*. John Wiley & Sons, Inc., New York, 2002.
- [17] L. Chung, B. A. Nixon, E. S.-K. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*. Boston, MA: Kluwer Academic Publishers, 1999.
- [18] I. Jacobson, J. Rumbaugh, and G. Booch, *The Unified Modeling Language Reference Manual*. Reading, MA: Addison-Wesley, 1999.
- [19] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Springer Berlin Heidelberg, 2012.
- [20] V. R. Basili and H. D. Rombach, "The Tame Project - Towards Improvement-Oriented Software Environments", *IEEE Trans. Software Eng.*, vol. 14, no. 6, 1988.
- [21] C. Huemer, A. Schmidt, and H. Werthner, "A UML profile for the e^3 -value e-business model ontology", in *BUSITAL held in conjunction with CAiSE'08 Conference*, 2008.
- [22] Z. Derzsi and J. Gordijn, "A Framework for Business/IT Alignment in Networked Value Constellations", in *BUSITAL*, 2006.
- [23] T. Gorschek, C. Wohlin, P. Garre, and S. Larsson, "A model for technology transfer in practice", *IEEE Software*, vol. 23, no. 6, pp. 88–95, 2006.
- [24] V. R. Basili, F. Shull, and F. Lanubile, "Building knowledge through families of experiments", *IEEE Trans. Software Eng.*, vol. 25, no. 4, pp. 456–473, 1999.
- [25] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology", *MIS quarterly*, vol. 13, no. 3, 1989.
- [26] W. R. King and J. He, "A meta-analysis of the technology acceptance model", *Information & Management*, vol. 43, no. 6, pp. 740–755, Sep. 2006.
- [27] S. Abrahao, E. Insfran, J. A. Carsí, and M. Genero, "Evaluating requirements modeling methods based on user perceptions: A family of experiments", *Information Sciences*, vol. 181, no. 16, pp. 3356–3378, Aug. 2011.
- [28] K. Maxwell, *Applied statistics for software managers*. Software Quality Institute Series, Prentice Hall, 2002.
- [29] S. Abrahao, C. Gravino, E. Insfran, G. Scanniello, and G. Tortora, "Assessing the Effectiveness of Sequence Diagrams in the Comprehension of Functional Requirements: Results from a Family of Five Experiments", *IEEE Trans. Software Eng.*, vol. 39, no. 3, 2013.
- [30] L. C. Briand, Y. Labiche, M. Di Penta, and H. Yan-Bondoc, "An experimental investigation of formality in UML-based development", *IEEE Trans. Software Eng.*, vol. 31, no. 10, pp. 833–849, Nov. 2005.
- [31] N. Juristo and A. M. Moreno, *Basics of Software Engineering Experimentation*, 1st edition. Springer Publishing Company, Incorporated, 2010.
- [32] W. J. Conover, *Practical Nonparametric Statistics*, 3rd ed. Wiley India Pvt. Limited, 2006.