Research Inventy: International Journal of Engineering And Science
Vol.6, Issue 7 (August 2016), PP -59-69
Issn (e): 2278-4721, Issn (p):2319-6483, www.researchinventy.com

Conceptualization of a Domain Specific Simulator for Requirements Prioritization

[1]Timóteo Silva, [1]Maria Lencastre, [1]João Pimentel, [1]Sérgio Galdino, [1]Cinthya Cavalcanti

[1]*Universidade de Pernambuco (UPE), Caixa Postal 50.720-001, Recife, Brazil.*

Abstract: *This paper conceptualizes a domain specific simulator for requirements prioritization; its aims at helping to identify appropriate prioritization strategies for a project in hand. The possible existing scenarios are difficult to analyze; they involve different variables, like the selection of: stakeholders (their availability, expertise, and importance); prioritization criteria; and prioritization methods. To demonstrate the feasibility of the proposed simulator elements, a well established general purpose simulator, called Arena, was used. The results demonstrate that, it is possible to build the suggested scenarios in order to study and make inferences about the prioritization strategies.*

Keywords: *Requirements Prioritization, Requirements Engineering, Software Engineering, Prioritization Techniques, Simulation.*

I. Introduction

Most often, the prioritization of requirements is an afterthought in software development projects [1]. Usually this is due to the difficulty and complexity in configuring a requirement prioritization process, caused by: the existence of several techniques, the participation of different stakeholders, the possibility of choosing different selection criteria, effort/time spent and lack of tools support. In this paper, we use simulation to allows managers to test and analyze hypothesis and implications of an implementation of a given strategy for requirements prioritization, without disturbing the real world; the simulation also helps to reduce the uncertainties and risks associated with the possible application, in addition to providing support for re-planning of the project in progress [2].

However, the key success of a simulation also depends on the use of an appropriate simulation tool. According to [3], among the criteria, for the evaluation and selection of simulation software, one must consider: ease of use, time required for learning the software, time required for building models, customized reports, model size, and the configuration of attributes. To facilitate the simulation of specific areas, domain specific simulators have emerged; generally, they supply already built components that model complex domain objects (its structure and behavior) [4]. In software engineering, some examples of domain specific simulators are Software Engineering Process Simulation (SEPS) [5], the Software Project Management Simulator [2] and The Incredible Manager [6].

This paper considers the lack of simulator tools to support the decision making in requirements prioritization, resulting in a higher level of difficulty to make this kind of studies. In order to provide a step towards the development of these tools, this paper presents the conceptualization of a specific purpose simulator to assist in the analysis and evaluation of scenarios in the field of requirements prioritization. The construction of its main structure was based on a general purpose simulator tool, well established in the market, called Arena [7]. This tool was selected since it's well-known and very used and it also supports the creation of completely customized environments through graphical modeling, without requiring the writing of any programming code. In this paper, the Arena not only serves as the base for defining the structure for the simulator's conceptualization, but also it allows the implementations of specific models related to requirements prioritization simulation, with the objective of concept proof regarding the proposed conceptualization.

This paper is organized in the following way. Section 2 presents theoretical basis for understanding our proposal. Section 3 details the conceptualization of the domain specific simulator – called 4RPSimulation. Section 4 illustrates the use of the simulator concepts, implementing them through Arena models examples (which can be simulated, using different scenarios). Section 5 presents some related work. Finally, section 6 presents the conclusion and future works.

II. Background

2.1 Prioritization of Requirements

The quality of a software product is usually determined by its ability to meet the needs of customers and users, which requires making a plan for the implementation and development of the product [8]. The selection of the subset of requirements by a required criterion to be implemented earlier in the project development – the requirements prioritization, is a complex activity; but, it is important for the success of a project [9]. However, it is difficult to define a metric and a process to evaluate the prejudice caused by not doing so.

According to [1], four preparatory activities can be defined for prioritizing requirements: select stakeholder, select requirements, define the prioritization criteria, and select the prioritization technique. The requirements prioritization can be seen as the act of defining the order of requirements development, according to some criterion. The criterion is how to measure the required value, importance, development costs, benefits, etc. Finally, prioritization technique is the process of decision criteria is used to set the order of the requirements.

Several requirements prioritization techniques have been proposed and studied over the years. Table 1 describes five of the most cited literature: Moscow, $100, AHP, WiegersMatrix and Genetic Algorithm (GA). However, to choose the appropriate/effective technique, for a particular organizational situation, remains a complex task for most companies [10]. To [11], despite the prioritization of requirements being an important task in software development, it is usually performed in an ad hoc form, that is, without clear guidelines.

The reason for this difficulty in requirements prioritization is because it involves issues such as conflicts of interest among stakeholders, budget constraints, customer expectations awaiting the final product, resource constraints, such as: human and technological, and risks inherent in the development project software [12].

TABLE 1. Five Well Known Prioritization Techniques

$100: Each stakeholder distributes $ 100 among the requirements to be prioritized; the result indicates how much a requirement is more or less important than another one in proportional terms [12].
Moscow: Stakeholders assign relative importance to the requirements, which are prioritized by words meaning: must should, could, won't [13].
AHP: Stakeholders compare pairs of requirements, using a relative scale of importance: uses 1 (for the same priority) to 9 (extreme difference); and then derives the priority vector, determining the overall value for each requirement; it ends with a measurement of consistency of results [14].
Wiegers Matrix: It includes the items to be evaluated by the criteria that will be used to evaluate; the possible values range (which can be used by the criteria), and the weighting being assigned to each criterion. In the end, the value of each item is obtained by a formula based on the criteria and weightings [15].
Genetic Algorithm (GA): a bio-inspired search algorithm based on the evolution of the collection of individuals resulting from natural selection and natural genetics. A GA usually applies a crossover operator to two solutions which play a major role, plus a mutation operator that randomly modifies the individual contents to promote diversity [16].

In [12] a systematic review of the literature, regarding the requirements prioritization techniques, presents the analysis of 108 papers. Based on the results, some important aspects were identified in planning a prioritization approach for a specific project: prioritization technique to be used, context variables that influence the process, response variables to analyze. See Table 2 for more details.

TABLE 2.Prioritization Aspects [12]

Prioritization technique	Prioritization Techniques have different levels of sophistication, more simple techniques are easier to use and require less time to implement (e.g. S100); complex ones can provide a more sensitive analysis, but require longer time to implement and are more difficult to use (e.g. AHP).
Context variables	System type (industrial, academic, or other), kind of process development, software tools used; experience of stakeholders on the system domain; number of requirements; and category of requirement (functional and non-functional).
Response variables	accuracy of the result, time and cost required to prioritize the requirements, etc.

2.2 Computer Simulation

Computer Simulation can be defined as the use of a computer model as a basis for the exploration and experimentation of real or imagined world [17]. The simulation involves the generation of an artificial story of a system (real or imagined) to be studied, from the creation of an abstract model. Through experimentation, the control variation (in the input variables) allows the analysis of their impact in the results performance (output variables or responses). Thus, it is possible to assess what "would happen if" certain action/ decision occurs in the actual system, observing the result of changing various input parameters, and comparing the different scenarios. One of the greatest advantages, of the study and analysis through simulation, is that once a valid simulation model is developed, one may incorporate modifications evaluating different scenarios without concerning effective costs or disruption of the real system [18]. The terminology used in the simulation includes the concepts: simulation objective, system, model, entity, resource, attribute, process (activity), state variable, event, scenario, controllers (input variable), and response variables [3]. See first column of Table 4.

Arena is one of the general purpose simulation systems, which provides an interactive environment for building, animating, verifying and analyzing simulation models. As it is used in this paper as the base for the proposed simulator and also for modeling and implementing our example, it is detailed next.

The Arena is divided into: Flow modules, that once interconnected describe the visually logic simulation process; and Data Modules, which represent objects that define the characteristics of various process factors such as: resources and entities. The first Modules have visual representation and the Data ones do not. Table 3 describes each module main elements, and also generic pre-defined variables and entity attributes provided to help analyzing simulations.

Arena allows the design and incorporation of news templates that are specific to a particular project, company, or industry. The template is built on Arena's structure, enabling the creation of new simulation tools in a graphical environment. So, new complete simulation building blocks can be developed. One can take the systems being study into Arena´s basic elements, and then combine these basic elements into more complex ones to be simulated. The built modules are collected into libraries, referred to as templates. This is one of the main reasons why Arena was used in this paper.

TABLE 3. Arena Modules: Flowand Data

FLOW MODELS (blocks that can be connected)		
Create		Starting point of the model is where the entity appears in the system. Entities are the triggers of the process, they move in the system and make the processes work.
Process		Action involving a range of time and/or resource, where the bodies go by.
Dispose		Does the disposal of entities.
Assign		Assign a value to an attribute or variable.
Decide		Allows decision making.
Batch		Grouping mechanism (e,g: can wait unitl all the requireemtns are prioritized).
DATA MODULES BLOCKS		
Entity		System elements that can be: distinguished individually and characterized/identified by their attributes.
Resource		The Resource is a static entity that provides a service for the other entities.
Variable		Hold values that are available for the entire model.
Attribute		Keep individual values for each entity.
Schedule		Usedto define an operating schedule for a resource
PRE-DEFINED ATTRIBUTES and VARIABLES (examples)		
Attribute	Resource.Busy/hour: reflects the cost per hour of occupation of a given resource.	
	Entity.Type: controls the type of an entity in the simulation.	
Variable	ResBusyCost: calculates the cost of the sum per hour occupation of a given resource.	

III. The Simulator Conceptualization

The conceptualization of a special purpose simulator for prioritization of requirements, the 4RPSimulation, aims at assisting in the planning of a requirements prioritization process in a given organizational context. However, in this section we focus only on presenting the main concepts chosen for the proposed e simulator. The 4RPSimulation's elements are based onsimulation terms exposed in [3] and also requirementsprioritizing concepts, including aspects raised by [12] (see Table 2), such as number of requirements, experience of stakeholders, prioritization technique used. Table 4 presents the correspondence between simulation elements and the prioritization domain, demonstrating how the prioritization can be modeled in a simulation context.

The proposed simulator was built considering Arena's Basic Blocks; these include Flow Blocks and Data Blocks presented in Table 2. However some specific Data blocks were created, in order to give a more abstraction for the prioritization domains. They include: Stakeholders (defined as a resource), Project, Releases and Requirement (defined as entities). In this paper we only present Stakeholders and the Requirement data blocks. The basic Flow blocks are from Arena, and need to be configured with proper data elements. We do

not find relevancy in creating domain specific flow blocks yet. In future work we preview to detail some of them in a higher level of abstraction to prioritization context.

TABLE 4. Simulation concepts and their application to prioritization domain

Element	Description [3]	Requirements prioritization domain
Simulation Objective	Suggest questions that should be answered by a simulation study. The project plan should include the establishment of some scenarios to be investigated.	It is difficult to establish the best strategy for requirements prioritization (in terms of costs and time spent). The objective is helping in this analysis, by considering the possible scenarios, and answering questions such as: What is the final cost and time spent, when a specific scenario is applied?
System	World under observation, represented by set of entities (including resources) that cooperate to produce a certain goal. The goal is who determines what will be the entities involved.	In software development, for requirements prioritization it is necessary the selection of stakeholders participation (each with different availabilities, costs, and expertise). Ideally, each stakeholder must prioritize requirements that address their immediate sphere of knowledge and concern; but this is not always possible, the requirements can be numerous, varied in nature, consume many resources, and impose different costs and experts.
Model	Abstraction of the system, which represents its approach. There is loss of properties, but has lower cost and enables the study of the desired simulation system according to the objective set.	The abstract model represents a set of requirements to be prioritized by a set of stakeholders, using existing prioritization techniques (AHP, $100, Moscow, Wiegers), and following a specific process.
Entity	The element that moves within the system and interacts with resources, participating in activities to assist in achieving a certain goal.	Requirements to be prioritized. They can be grouped by releases, type, etc.
Resource	Static entities that are allocated and which provide services to other entities. They have a finite capacity, and may be in different states (busy, free, unavailable, etc).	Stakeholders that will participate in prioritizing a set of requirements; they have profiles, costs, experience levels, availability, etc.
Attribute	Property of and entity or of a resource, i.e., a characteristic that defines it.	Requirement type (functional or non-functional) cost; Stakeholders information, Project information etc.
Process (activity)	Action taken on the entities along the simulation; it has a previously known finishing time.	Requirements prioritization techniques: AHP, $ 100, MoSCoW, Wiegers Matrix, and their sub processes. Understanding Requirements, Build Ranking.
State variable	All variables that provide information about what is happening in the system at a given time. Include monitoring data.	At a given time: the number of prioritized requirements, the number of stakeholders occupied and idle, time spent, remaining time, costs per stakeholder until the moment, etc.
Event	Event, programmed or not, when it occurs causes a state change in the system.	The arrival of a new release to be prioritized; prioritization start, end of prioritization, etc.

Controllers (input)	Input variables that can significantly affect the model's response variables; they are used in different scenarios.	Quantity of requirements, type and number of stakeholders (cost, level of expertise, etc).
Scenario	Collection of controllers and response variables used in the experimentation of a given simulation model.	See an example of scenario in Table 7 and Table 8.
Response variables	Values observed from the system behavior (response) represent measures of how the model behaved during simulation.	Total time spent on priority by all stakeholders; utilization rate of stakeholders related to performance measures; Total cost of the requirements and total cost of Prioritization.

As the Arena simulator offers the opportunity to create fully customized environments without writing any programming code, the proposed simulator was structured considering building blocks of the Arena. The intention to create specific blocks of the field of prioritizing requirements is to give a more abstract response to mounting the models within that domain. The next section describes the data blocks and flow blocks created specifically for prioritizing requirements.

2.3 Blocks Data

This section describes the data blocks designed for the field of priority, as follows: Stakeholder Data, Requirement Data and Project Data.

STAKEHOLDER DATA
Stake Profile:can be a Project Manager, Developer, Process Analyst, Test Engineer, Customer, or User.
Cost per hour: Sets the cost/time that each stakeholder has to performthe evaluation.
Experience Level: Classifies the stakeholder in beginner, intermediate or senior.
Availability: Time stakeholder has available for prioritizing requirements.
Time factor:1.0 (beginner); 0.8 (intermediate); 0.6 (senior).
REQUIREMENT DATA
Requirements Id:requirements identification.
Requirement Type:if the requirement is functional, nonfunctional.,or not determined.
Implementation cost:the cost of implementation of each requirement.
Benefit:defines the benefit of implementing such requirement.The values are scaled from 1 to 9, based on [14], considering that the higher the value, the more beneficial it will be the implementation;
Penalty:defines the penalty of e implementing such requirement. The values are scaled from 1 to 9, based on [14], considering that the higher the value, the more painful will be the non-implementation.
Dependency:if the requirement is dependends or not to another one.

PROJECT DATA
Project Name: inform the project name to be analyzed.
Release quantity: tells you how many releases the project has.
Feature quantity: tells you how many features the project has.
Benefit: defines the benefit of implementation; the values are scaled from 1 to 9, based on [14], considering that the higher the value, the more beneficial it will be the implementation of the project;
Penalty: defines the penalty of not implementing the project; the values are scaled from 1 to 9, based on [14], considering that the higher the value, the more painful will be the non-implementation of the project.
Project cost: sets the cost of the project.

2.4 Blocks Flow

This section describes the flow of blocks designed for the field of priority, as follows: Block Templates Prioritization Process and Block Templates Decision Prioritization.

BLOCK TEMPLATES PRIORITIZATION PROCESS
Name: the user can select any of the predefined names (Understanding Requirement; Mount rank; Distribute Dollar; Somar Dollars; Assign Importance; Assign Value; Decomposing Requirements Levels; Mounting Matrix).
Scan Type: Individual or Group.
Type Unification: specifies the Unified type of prioritization made individually, ie, whether it is via calculation or talk. If unification is via calculation, the Action parameter is always of type "Delay".
Distribution Type: Specifies the type of distribution, ie, whether it is performed individually for each stakeholder or group.
Stakeholders Participants: defines which one (s) of registered stakeholders in block stakeholders participate in that process.
BLOCK TEMPLATES DECISION PRIORITIZATION
Pre-Defined name: The sum of dollars is less than 100 ?; The assignment of importance was satisfactory ?; You want to change the criteria ?; There was arrival of new requirements?
Condition: Yes or No.

IV. Example Of Use

This section illustrates an example of requirements prioritization using the proposed blocks and concepts. The objective is to model and simulate the application of the $100 prioritization technique. Fig. 1 presents the model created to simulate $100 technique, in Arena tool. The main blocks include the processes to Understand Requirements, Distribute 100 Dollars, and Build Ranking. Each used block is detailed in Table 5 but only the configuration's data for stakeholder 1 is considered.

The Table 6 shows the configuration of the data block Resource which was inserted the stake resources 01 and 02 stake with their respective settings.

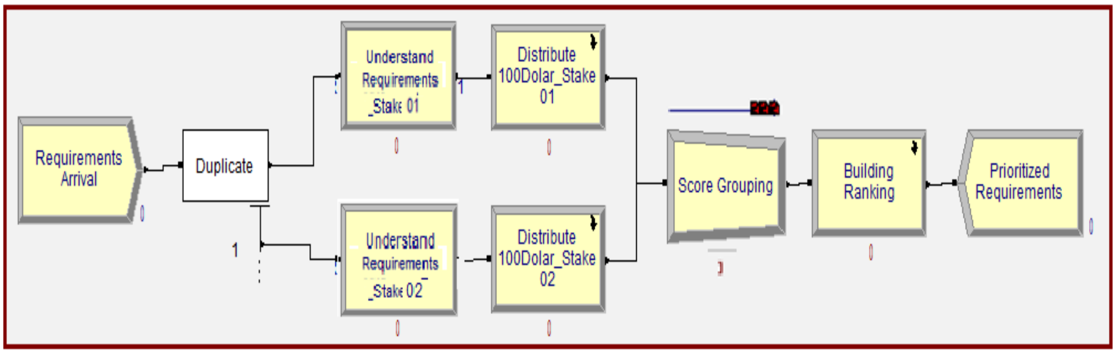
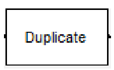
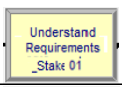


Figure 1. Model for simulation of $100 Prioritization Technique Using ARENA

TABLE 5. Blocks used in the $100 Simulation Model

FLOW BLOCK	**DESCRIPTION**	**CONFIGURATION**
	Begins the simulation process and creates requirement entities to be prioritized.	Entity type = Requirement; Entities per arrival = QtdReq; Max Arrivals =1.
	Makes a copy of the requirements for each stakeholder, allowing each one to make its own analysis.	Entities =2.
	Process were a stakeholder reads all the requirements and try to understand them.	Resource: Stake 01; Delay = timeStake01; Value= 4,8 min.
	Process where each stakeholder spends time giving each requirement a value, that is a percentage of the receive dollars.	Resource: Stake01; Delay = timeStake01; Value= 6,4 min.

Score Grouping	Score Grouping: unifies the values of the scores given by the stakeholders for each requirement. So, it has to compare the serial number of the two entities (requirements) that come from the two stakeholders.	Batch Size= 2; Entity Type = Requirement; Rule = compare Entity.Serial Numbers.
Building Ranking	Organizes requirements by the achieved order of prioritization This process spends a constant delay.	Delay= 20 seconds.
Prioritized Requirements	Final disposal of prioritized requirements.	Name.
DATA BLOCKS	**DESCRIPTION**	**CONFIGURATION**
Entity	Entities which pass through the process flow.	Requirements (to be prioritized).
Resource	Two resources participate in the prioritization.	Stake01; Stake02.
Variable	• Input variable: number of requirements to be prioritized; • Monitoring variables: Time spent by each stakeholder: to Understand requirements and to Distribute100Dollars, Total Cost spent for each stakeholder, Evaluated Requirements, Available time, Days elapsed, Dollar's spent; • Response variables: Total Costs and Total Time Simulation (length of time to complete the prioritization process).	Qtd Req = 32.
Schedule	Schedule related to the degree of availability of the stakeholders.	Low availability (2h/day); Medium Availability (3h/day); High Availability (4h/day).
PREDEFINED VARIABLE	For the calculation of the final results several pre-defined variables were used, including: • ResBusyCost: calculates the cost of the sum per hour occupation of a given resource, that is, the value of the variable Stake Cost spent for each; • TNOW: Current simulation time. TNOW records the simulation clock time as the model progresses (Total Time).	

TABLE 6. Resources Configuration

Resource - Basic Process

	Name	Type	Capacity	Schedule Name	Schedule Rule	Busy / Hour
1	Stake 02	Based on Schedule	Low	Low availability	Wait	60
2	Stake 01	Based on Schedule	Disponibili	Medium availability	Wait	45

The example, that are part of the Fig. 1 model, was developed building on the description of the literature on the use of the technique $ 100 associated with the practical application of the technique with two students of Masters in Engineering Computer - Requirements Engineering Research Line, who prioritized a real project, called SDO (Budget Documents System), which has 32 requirements, design this belonging to a construction company. The practical application also aimed to verify / validate the process steps and thus build the model of the proposed example and served to check the time taken by such students (are also professionals, less than 5 years experience) at each stage process. Also took part in the survey information for construction of the proposed interview with 3 experience level professionals characterized as senior working on 3 different companies (government, mixed and private sector) where the peculiarities were discussed, weaknesses and needs in the process of prioritization of requirements in the respective companies.

In this example two scenarios are analyzed using the proposed model: Scenarios 1 and Scenario2 (see Table 7 and Table 8). Both scenarios use the same amount of requirements as input QtdReq = 32, and the same number of stakeholders (2 per scenario).The difference is in the experience's level of each stakeholder (beginner, intermediate and senior). The level of experience impacts in the time they spent for realizing tasks, and also in stakeholder's costs, based on a predefined percentage.

The cost for beginners is also a system´s input (30$), and the cost for the intermediate and senior, is 50% and 100% more than for the beginners. The time for beginners to execute understanding of each requirements is 6 min and distribute dollars is 8 min. The time spent by the senior activity and the intermediate is fixed, respectively, in 60% and 80% of a beginner's time. To estimate the time taken for the activities (processes) that are part of the Fig. 1 model, prioritizing requirements of a project called SDO (Budget Documents System), which has 32 requirements, was carried out by two students of Masters in Engineering Computer - Requirements Engineering Research Line, both Beginner profile, and from the values found, the percentages for intermediate and senior profiles were calculated, as described above, these percentages based on [19]. The available time of each stakeholder is fixed for both scenarios: 4/h day for beginners, 3h/day for intermediate and 2h/day for seniors. The time available for execution prioritization of requirements is 3 days. The cost of implementation of each requirement is 50$.

The input configuration and resources representing the scenarios 1 and 2 respectively, the processes involved, the authorities and response variables are detailed in Tables 7 and 8, respectively. Only some parameters of the data blocks were used. Fig. 2 and Fig. 3 have the results screens of scenarios 1 and 2 respectively. The results after the model execution for scenarios 1 and 2 are presented in Table 9.

TABLE 7. Elements for the simulation of Scenario 1

Scenario 1	**Analyzing the prioritization of requirements with technical $ 100, with the experience level of Stake 01 (intermediate) and 02 Stake (senior).**					
Resources	**Stakeholder**	**Experience Level**	**Cost $/h**	**Time to understand Requirements**	**Time to Distribute Dollars**	**Available Time**
	1	Intermediate	45	4.8 min	6.4 min	3h/day
	2	Senior	60	3.6 min	4.8 min	2h/day

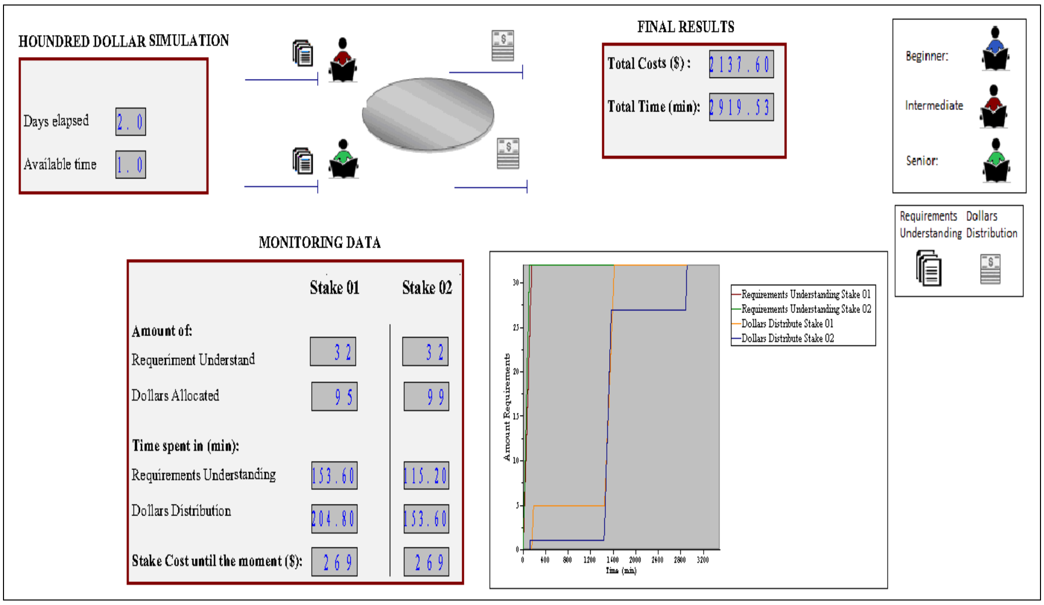


Figure 2. Visual Execution of the Simulation Model using the proposed Scenario 1 (input data) in ARENA

TABLE 8. Elements for the simulation of Scenario 2

Scenario 2	**Analyzing the prioritization of requirements with technical $ 100, with the experience level of Stake 01 (intermediate) and 02 Stake (beginner).**					
Resources	**Stakeholder**	**Experience Level**	**Cost $/h**	**Time to understand Requirements**	**Time to Distribute Dollars**	**Available Time**
	1	Intermediate	45	4.8 min	6.4 min	3h/day
	2	Beginner	30	6 min	8 min	4h/day

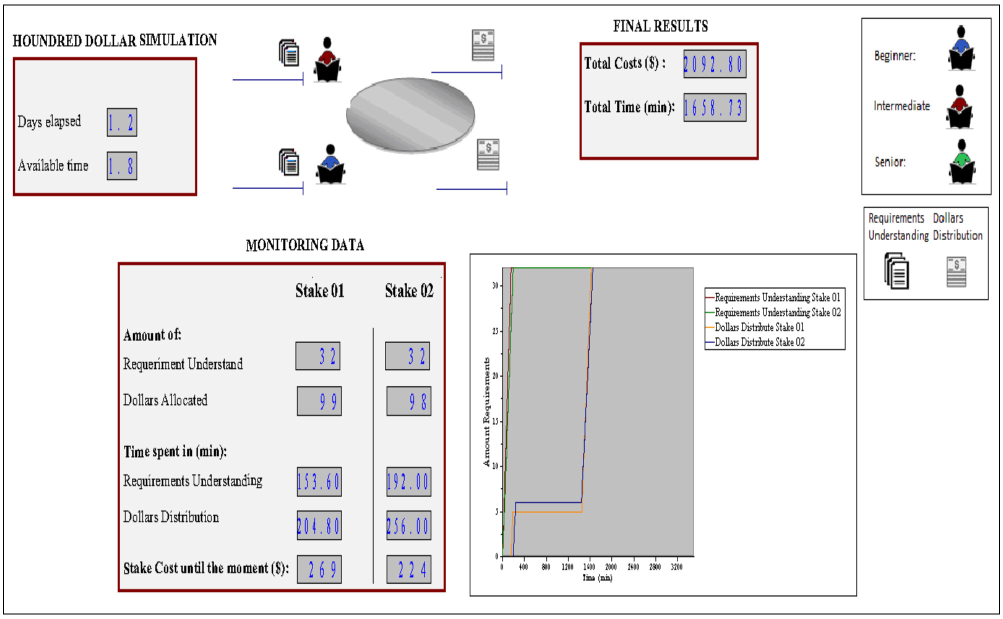


Figure 3. Visual Execution of the Simulation Model using the proposed Scenario 2 (input data) in ARENA

TABLE 9. Report from the scenarios simulation

	Scenario 1			Scenario 2		
Tasks	**Stake 01**	**Stake 02**	**Total**	**Stake 01**	**Stake 02**	**Total Time**
Understand Requirements	153.6 min	115.2 min	268.8 min	153.6 min	192 min	345.6 min
Distribute 100Dolar	204.8 min	153.6 min	358.4 min	204.8 min	256 min	460.8 min
Cost Implementation	-	-	$1600,00	-	-	$1600,00
Cost Stakeholders	$269,00	$269,00	$ 538,00	$269,00	$224,00	$ 493,00
Prioritized Requirements	32	32	32	32	32	32
Final Cost	-	-	$2137,60	-	-	$2092,80
Total Time Simulation	-	-	2919,53 min	-	-	1658,73 min

By evaluating the results, we can observe that, in both scenarios, all requirements are prioritized within the designed period for the prioritization implementation. However, we realized that unlike what intuition would lead us to believe, Scenario 2 is more time consuming, as the stakeholder with experience level equal to Beginner takes 40% more time to perform each activity, compared to the stakeholder with experience level equal to Senior; observe that the simulation shows a result contrary to this intuition. It is noticed that the availability factor was decisive for the results; as presented in Table 9, Scenario 2 is significantly more advantageous from the point of view of time spent in prioritization, since, in this scenario the elapsed time for prioritization execution is approximately 1 day and 4 hours, the time required for this activity in Scenario 1 was approximately 2 days and 1 hour. From the final cost point of view, Scenario 2 is also slightly more advantageous, that is economical. Faced with this two response variables (time and cost), we can infer that Scenario 2 is the best one. However, we can consider that the efficacy (accuracy) of that priority is questionable, since, Scenario 2 that has proven to be the most advantageous, may not be the most efficient, since one of participating stakeholders experience level is equal to Beginner.

During our study, other 54 scenarios were modeled and analyzed, considering variations on $100, and also including other prioritization techniques (such as MoSCoW and AHP), modifying the amount, availability and the profile of defined stakeholders. Unfortunately, due to lack of space in this paper, we could not include them here. However, it is worth noting that in other scenarios (varying the amount of stakeholders) with the very technical $ 100 and AHP, analyzing the results it was found that, even with the increased number of stakeholders, the total time for implementation of the prioritization virtually It does not change. This possibly happens because these techniques have an element of assigning numerical values to each requirement, facilitating the unification of prioritization and definition of the end of each requirement value. As for the MoSCoW technique, the increased amount of stakeholders triggered a significant increase in the time to execute the prioritization. Possibly, this increase was due to the fact that MoSCoW technique, being the nominal type, requires the unification of prioritization, a discussion among stakeholders in order to reach a common denominator regarding the final classification of each requirement.

The simulation results, implemented through a general purpose simulator, showed the advantages: carrying out the assessment of different scenarios without having to spend time and cost for a real implementation of these scenarios, and the possibility of analyzing the results. Also, being able to reuse the proposed model with different parameters (other scenarios). Moreover, when one has to deal with different variables it is difficult to evaluate results only with intuition; the decision could not necessarily reflect the most advantageous scenario to be adopted, due to the lack of more specialized support for making experimentations and analysis.

The implementation of the models in Arena were costly and labor-intensive, both in the sense of learning curve for Arena (which is pretty long) and in the sense of the difficulty of representing certain processes, variables, attributes and calculations. This happened also due to the fact that there are no representative elements within Arena to treat directly prioritizing requirements and concepts, as well as the complexity of dealing with some subjective aspects.

The involved complexity justifies the creation of some new blocks, and specific processes through the RE4Prioritization, in order to simplify the prioritization modeling process and incorporating existing knowledge in this specific domain.

V. Related Work

Regarding the tools in requirements engineering, there are several domain specific tools, as indicated in the survey [4] and in the systematic review [20]. Specific examples of simulator are detailed next.

In [5] a simulation tool for Software Engineering Process Simulation (SEPS) is presented. It is a planning tool, which allows to examine the trade-offs of costs, schedule and functionality, and to test the implications of different management policies on a project’s outcome. Moreover, SEPS allows managers to get a better understanding of the software project development dynamics and perform evaluations. Among the assessments that can be made, for example, one can check the impacts caused by the changes in requirements that affect the: project cost, schedule, availability, completion time of a task, project completion time, and the productivity. When performing tests in a given project, nine input parameters are considered in SEPS, including: level of experience of staff (experienced and inexperienced), project size, estimated timeline (considering availability and effort) and team size.

The Software Project Management Simulator [2] considers, for a software business model, the effects of volatility of requirements on the cost, schedule and quality of a project. For such, it considers:

- Project information, such as:
  - The project size is defined in function points.
  - The cost is considered to be the human effort consumed.

During the project (person per day); this cost does notinclude other project costs.

  - Time for execution.
- Stakeholder information (type, experience, productivity, schedule of activities, etc.). It considers that a requirement engineer can have two levels of experience (beginner and experienced). The productivity is measured in terms of function points/person-day. The schedule of project activities is measured in days.

The Incredible Manager is a game based on simulation, motivated by the frequent inappropriate use of software project management techniques [6]. This game can be used to provide experiential learning in managing software projects. The game considers 3 types of stakeholders: the Manager, the Head and the developers; it also represents the stakeholders’ level of experience, availability and cost of working hours per day (only for the developers). The system process is divided into five steps: Initial Phase, Project Planning, Planning Acceptance, Project execution and Finals. The variables considered in these steps include: tasks and their function points, the schedule, budget, time available for performing the process, stakeholders added to the process. During the execution, the user should be aware of the project's behavior and take corrective actions when necessary; visual effects and project reports show the current conditions of the project such as the developers which are exhausted, tasks that are late and are without funds in the project.

The 4RPSimulation, compared to the simulator tools described above and indicated in the reviews of [6] and [20], focus on a new distinct domain – the requirements prioritization. Some common concepts can be found among 4RPSimulation and the other simulators like: the definition of stakeholders, availability, level of expertise, cost per activity/stakeholder, time spent by activity, and cost estimate of the possible implementation. The purpose of 4RPSimulation is for planning like [5, 2], but different from [6] which focus on education. As 4RPSimulation is based on Arena, it inherits many simulation concepts which are generic and give more flexibility and expressiveness to this proposal.

VI. Conclusion

Prioritizing requirements allows minimizing conflicting demands and planning the software product. The existence of several prioritization techniques, together with various possible scenarios, raises the need to analyze which technique and scenario are the most appropriate to be applied in a project. Although many works deal with prioritizing requirements [11], there is a gap in tools that aid the evaluation of prioritization strategies, without disturbing the real environment.In addition, the growing awareness to improve software quality motivates the software industry to come up with tools that help in the management and decision making. We believe that this paper's proposal can contribute to the decision-making by allowing the analyzes of consequences of different pertinent scenarios, without having to disturb the real environment process.

The interviews confirmed what the literature says about the resistance to apply a formality in prioritizing requirements, precisely because they have the idea that the use of certain technique associated with established criteria for prioritization will delay the progress of the project and will be costly [14]. The simulation within this priority area, which by the way, is something innovative, comes just break this paradigm, to build on the basis of guesses. This paper proposes the conceptualization of a domain specific simulator, facing the simulation of prioritization scenarios. The objective is enabling the study, via simulation prioritization, of different configurations, taking as output values (responses) cost and time spent; these values can help deciding which prioritization plan (scenario) should be adopted. The simulator viability was analyzed by developing and running experiments, using the Arena simulator, considering the prioritization in software development context. However, to build this staff it was laborious.

As [7] states, one of the problems with simulation tools is the lack of illustration of a real environment of the activity being simulated; since the interaction with the project environment does not resemble an actual situation, the user's motivation to use simulation tools may be limited.

Future work includes the implementation of the 4RPSimulator as a specific template in Arena, followed by verification and validation of it in an academic environment; the evaluation of the adequacy of existing elements, consistency of results and efficiency. Also, other parameters could be incorporated into 4RPSimulation as productivity[2, 6] and dependence [2].The application of the simulator to industry, verifying the effectiveness of the application. Other future studies could analyze the use of other simulator tools which can eventually reduce the involved complexity identified in Arena for modeling the prioritization domain.

References

[1]. Pohl, K.: Requirements Engineering: Fundamentals, Principles and Technologies. Springer, 2010.

[2]. Ferreira, S., Collafello, J., Shunk D., Mackulak, G.: Understanding the effects of requirements volatility in software engineering by using analytical modelling and software process simulation." Journal of Systems and Software, 2009.

[3]. Freitas Filho, P.: Introdução à Modelagem e Simulação de Sistemas com Aplicações em Arena. 2ed.: Visual Book, 2008.

[4]. De Gea, J. M. C., Nicolás, J., Alemán, J. L. F., Toval, A., Ebert, C., & Vizcaíno, A."Requirements engineering tools:Capabilities, survey and assessment." Information and Software Technology54.10: 1142-1157, 2012.

[5]. Lin, C., Abdel-Hamid, T., Sherif, J: Software-Engineering Process Simulation Model (SEPS)". The Journal of Systems and Software, Vol. 38, p. 263 – 277,1997.

[6]. Dantas, A., Barros, M., Werner, C.: A Simulation-Based Game for Project Management Experiential Learning. In: International Conference on Software Engineering & Knowledge Engineering, p. 19-24, Alberta, 2004.

[7]. Law, A.: Simulation Modeling & Analysis, 4th ed., McGraw-Hill, p.70, 2007.

[8]. Berander, P.: Evolving Prioritization for Software Product Management. Doctoral Dissertation Series Blekinge Institute of Technology, School of Engineering, Department of Systems and Software Engineering, Sweden, 2007.

[9]. Karlsson, L., Dahlstedt, Å. G., Regnell, B., och Dag, J. N., & Persson, A. "Requirements engineering challenges in market-driven software development–An interview study with practitioners." Information and Software technology 49.6: 588-604, 2007.

[10]. Greer, D.; Ruhe, G. Software Release Planning: An Evolutionary and Iterative Approach. Journal Information and Software Technology, (S.l.), v. 46, n. 4, 2004.

[11]. Balushi, T., Sampaio, P., Loucopoulos, P.: Eliciting and prioritizing quality requirements supported by ontologies: a case study using the ElicitO framework and tool, Expert Systems, v.30, n.2. 2013.

[12]. Khan, K.: A systematic review of software requirements prioritization. Master Thesis MSE-2006-18 – Blekinge Institute of Technology, School of Engineering, Department of Systems and Software Engineering, Sweden, 2006.

[13]. Oracle. Custom Development Method Fast Track (CDM Fast Track) Method Handbook, Release 1.2.0. Parkway, California, USA: Oracle Corporation, 2000.

[14]. Saaty, T. : Theory and Applications of the Analytic Network Process: Decision Making with Benefits, Opportunities, Costs, and Risks. Pittsburgh: RWS Publications.2005

[15]. Wiegers, K., and Beatty, Joy.: Software requirements. Pearson Education, 2013.

[16]. Pitangueira, A. M., Maciel, R. S. P., de Oliveira Barros, M., & Andrade, A. S. "A systematic review of software requirements selection and prioritization using SBSE approaches."International Symposium on Search Based Software Engineering. Springer Berlin Heidelberg, 2013.

[17]. Pidd, M. "Computer simulation in management science." 1998.

[18]. Gupta, A., Verma, R., & Singh, K.; Smart Sim selector: a software for simulation software selection. International Journal of Engineering (IJE), 3(3), 175, 2009.

[19]. Isotton, E.: Scrumming - Ferramenta Educacional para Ensino de Práticas de SCRUM. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Pontifícia Universidade Católica do Rio Grande do Sul, 2008

[20]. Zhang, J., Kitchenham, B. and Pfahl D.: Reflections on 10 Years of Software Process Simulation Modeling: A Systematic Review. International Conference on Software Process, 2008.