Using Simulation to Aid Decision Making in Managing the Usability Evaluation Process

Nuria Hurtado[1], Mercedes Ruiz[1], Elena Orta[1], Jesús Torres[2]

1_Department of Computer Science and Engineering_

University of Cadiz (Spain)

C/ Chile nº1- 11003 – Cadiz (Spain)

(Phone: +34 956 015206) (fax: +34 956 015139)

{nuria.hurtado, mercedes.ruiz, elena.orta}@uca.es

[2]_Department of Computer Science and Engineering_

University of Seville (Spain)

Avda. Reina Mercedes s/n - 41012 - Seville (Spain)

(Phone: +34 95 4552769) (fax: 95 4557139)

jtorres@lsi.us.es

Corresponding author: nuria.hurtado@uca.es

(Phone: +34 956 015206) (fax: +34 956 015139)

Abstract.

Context: This paper is developed in the context of Usability Engineering. More specifically, it focuses on the use of modelling and simulation to help decision-making in the scope of usability evaluation.

Objective: The main goal of this paper is to present UESim: a System Dynamics simulation model to help decision-making in the make-up of the usability evaluation team during the process of usability evaluation.

Method: To develop this research we followed four main research phases: a) study identification, b) study development, c) running and observation and finally, d) reflexion. In relation with these phases the paper describes the literature revision, the model building and validation, the model simulation and its results and finally the reflexion on it.

Results: We developed and validated a model to simulate the usability evaluation process. Through three different simulations we analysed the effects of different compositions of the evaluation team on the outcome of the evaluation. The simulation results show the utility of the model in the decision making of the usability evaluation process by changing the number and expertise of evaluators employed.

Conclusion: One of the main advantages of using such a simulation model is that it allows developers to observe the evolution of the key indicators of the evaluation process over time. UESim represents a customisable tool to help decision-making in the management of the usability evaluation process, since it makes it possible to analyse how the key process indicators are affected by the main management options of the Usability Evaluation Process.

Keywords: Usability Evaluation, Decision Making, Simulation Modelling, Systems Dynamics, Software Process Improvement

1 Introduction

"Usability" is a quality attribute that assesses how easy a software user interface is to use. It constitutes a necessary precondition for good User Experience [1]. The term usability also refers to methods for improving ease-of-use during the design process. The usability inspection methods are the most-used and best-known methods for evaluating usability. Usability inspection is the generic term for a set of methods based on having evaluators inspect usability-related aspects of a user interface. A usability inspection is aimed at finding usability problems in a user interface, and then treating these problems to make recommendations for fixing the problems and improving the usability of the software product [2].

The number and expertise of the evaluators employed to judge usability are two essential factors that affect the results of an evaluation process. Knowing both the number and expertise of evaluators, it is possible to estimate the number of problems that will be found using the mathematical model presented by Nielsen and Landauer [3]. These two essential factors also affect the costs incurred by the developer in using the evaluation method. Therefore it is possible to estimate the final values for both the number of *problems found*, and *cost*. However, because the time available for developers to produce high-quality usable software is becoming a crucial limitation, *time* must be considered an additional key factor in any software development process.

Since the early 1990's several simulation models have been developed to respond to different questions related to the software process. However, none of these models has addressed issues related to usability and the user experience in software development.

In relation with this context, the present work considers the following general Research Questions:

RQ1. Would it be possible to simulate the process of usability evaluation?

RQ2. Would the simulation of this process help decision-making in the composition of the usability evaluation team?

To be more specific, we considered if, by means of a simulation model, we could provide answers to questions such as "What if …?" For example, what would happen if we mixed evaluators with different levels of experience in the evaluation team? How would this affect the execution of the process? What would the effect be on its cost and length? What would the ideal process configuration be for a given budget to achieve the optimum usability results? How long would all this take?

Dynamic modelling and simulation could be used by developers to make the usability evaluation process visible, and let them observe its behaviour over time; alternative choices and decisions could be simulated and their outcomes analysed before they are made and implemented. Therefore, in the context of usability evaluation, these techniques could greatly aid the decision-making process.

We therefore propose the application of modelling and simulation techniques to help decision-making in the process of usability evaluation. To illustrate the effectiveness of these techniques for this purpose, we present a simulation model, UESim, and three cases. UESim simulates the usability evaluation process using usability inspection methods. This simulation allows us to analyse the effect that alternative different compositions of the usability evaluation team would have on the outcome of the evaluation, namely on *time*, *cost* and *number of problems found*.

The paper has been structured by relating its sections to the phases of the research methodology used. In section 2 we set out the phases of the research conducted, with a brief summary of the work undertaken in each, and referencing those sections of the paper where we present the outcomes of these steps.

2   Materials and Methods

To develop this work we have followed an approach consisting of four main steps. These phases are common in different research methods, such as the Action Research method [4], [5] or [6]. Table 1 shows both the outcomes obtained in each step of the study and the section of the paper in which they are presented. The phases we followed are as follows:

1. Study identification: this phase studies the different usability evaluation methods, centring specifically on inspection methods and how they are carried out, paying special attention to the factors that intervene. This leads us to formulate the conjectures on which we base our simulation model. In addition, in this phase we have also performed a literature review about the different applications of simulation modelling in the scope of software process.

2. Study development: Several different proposals have been set forth for developing simulation models, but they share a series of steps and recommendations [7], [8], [9], [10], [11]. We can summarise them into three main tasks: to build, validate and experiment.

3. Running and observation: To carry out this phase three different experiments have been performed in which we can observe the effects of different compositions of the evaluation team on the outcome of the evaluation process.

4. Reflexion: Finally, having analysed the results obtained following the model simulation process, we reflect on these along with the possible threats to validity. We conclude by contrasting our initial research questions.

To facilitate the task of building and validating UESim, we have used the Vensim® simulation environment [12]. To perform the survey used in the simulation validation phase we have used the online tool *encuestafacil* [13]

Table 1. Summary of research phases

Research phase	Outcome	Paper section
1.-Study identification	Process of usability evaluation description	Section 3
	Modelling and simulation literature revision	Section 4
2.- Study development	Model design and Model building	Section 5
	Model validation	Section 6
3.- Running and observation	Model simulation	Section 7
4.- Reflexion	Discussion	Section 8
	Conclusion	Section 9

3 The Process of Usability Evaluation

Usability inspection is the generic term for a set of methods based on having evaluators inspect usability-related aspects of a user interface. Each method has its particular objectives but they all have the general objective of identifying potential usability problems and providing design-relevant information. This involves employing several evaluators to examine the interface and judge its compliance with: the development of user tasks, recognised usability principles (namely heuristics), usability guidelines or standards.

Figure 1 presents the general process of individual usability inspections formalized using BPMN (Business Process Modelling Notation). The process illustrated is based on Nielsen studies [2], [14] and consists of the following phases:

- **Planning**. First, it is necessary to plan the evaluation session, choose appropriate evaluators, determine the criteria that will be adopted, and prepare the necessary material. In the planning phase the decisions about the make-up (number, expertise and dedication) of the evaluation team are made.

- **Training**: Before beginning the evaluation session itself, the evaluators usually need some training for the particular evaluation proposed. In this training, the person who manages the evaluation, normally named the "observer", will explain the method, the objectives of the evaluation, the criteria to follow, as well as the concepts and

characteristics that the evaluators need to know about the domain of the system to be evaluated.

- **Evaluation session**: during the evaluation session, each individual evaluator will use the interface several times and identify the usability problems. It is very important that the evaluations be performed independently to prevent bias due to the evaluators influencing each other. Finally, each evaluator will provide a list of the usability problems found in relation to the inspection goals.

- **Analysis of results and drafting of the final report**. After all the evaluations have been completed, the problems found are analysed and grouped to obtain a single list of problems. Evaluation of the severity rating of the usability problems found can be made during the evaluation session; these ratings can be reflected in the individual reports to be analysed later, or once the problems have been grouped, in a final debriefing session with all the evaluators.
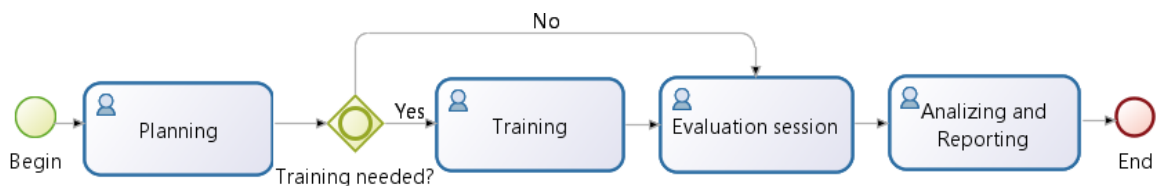

Fig 1. The usability evaluation process

Accordingly, the factors that have been considered as the key aspects for the composition of the evaluation team, are the following:

- Number of evaluators. Generally, usability evaluation is difficult for a single evaluator to perform because it is complicated just for one person to find all the usability problems in an interface. References from several different projects have shown that different people will find different usability problems. This phenomenon has been called the "evaluator effect" [15], [16], [17], [18] It is difficult to find an optimum evaluator, because the same person will not be the best evaluator for every case. Sometimes an evaluator who is not capable of finding all the usability problems where many exist, will however prove to be capable of finding some of the hardest-to-detect. Therefore, it is necessary to use several evaluators in any usability evaluation [19], [2].

In their mathematical model, Nielsen and Landauer state that the detection of usability problems, as a function of the number of evaluators, can be modelled as a Poisson process, represented in Equation 1 [3]:

$$problems\_found(i) = N(1 - (1 - p)^i) \qquad (1)$$

where "problems_found(i)" indicates the number of unique usability problems found by aggregating reports from i independent evaluators, N indicates the estimated total number of usability problems existing in the interface evaluated, p indicates the proportion of all usability problems found by a single evaluator that is the mean probability of detecting a problem by a subject and i indicates the number of evaluators, also called the usability test sample size.

For example, if we consider that p is 0.3, i is 5 and N is 100, we could say, applying Equation 1, that only five evaluators are needed to detect approximately 83% of usability problems. Since 1993, the estimation of p to determine the usability test sample size has been discussed by different studies [20], [21], [22] and currently is an open research problem in the HCI (Human Computer Interaction) community.

- Evaluator's expertise. Several studies have demonstrated that greater evaluator expertise improves the effectiveness of the usability evaluation [23], [24]. Depending on the specific evaluation method used, we might consider different issues affecting the expertise of the evaluators such as: usability experience, heuristic experience, domain experience and training [24]. From a general point of view, the references studied suggest that expertise can usefully be categorised into the following three levels:
    - **Level 1**: corresponds to novice evaluators with no usability experience.
    - **Level 2**: corresponds to regular usability specialists.
    - **Level 3**: corresponds to specialists who combine usability experience with experience of the particular kind of interface being evaluated (domain experience).
- Dedication: this factor represents the amount of time (hours) that each evaluator will spend on the evaluation.

All these factors are crucial for determining the total cost incurred in by the application of the evaluation method. Fixed costs and variable costs must be taken into account. Fixed costs are expenses that are not dependent on the activities carried out in the evaluation itself. The time needed to plan the evaluation, to prepare the materials, and to write up the final report are among the fixed costs of each particular evaluation process. Variable costs are those expenses that change in line with the number and type of evaluators; they include the evaluator's salary as well as the cost of analysing the evaluator's report and the cost of any computers or other resources needed during the evaluation. These costs can greatly vary according to the project and will depend on the company's cost structure as well as on the size and complexity of the interface under evaluation [2].

As Woolriych et al. [25] defend, the usability evaluation methods should not be used as indivisible wholes. The practitioners must configure, adapt, and complement the method resources to match the specific circumstances of each project. In this line, there are also other factors that are not related to the composition of the evaluation team and that can affect the process behaviour and hence the method effectiveness, such as: the stage of the development lifecycle, the size, type or complexity of the system, and the type of end users [2], [3], [14], [20], [25].

4   Precedents and Related Works

4.1   Software Process Modelling and Simulation

Modelling and simulation techniques are considered valuable for the improvement of processes in many areas of engineering. A simulation model is a computational model that consists of the abstraction or simplified representation of a complex dynamic model. The objectives of simulation models are, commonly, to provide mechanisms for experimenting, predicting, learning, and answering questions such as: *What if ...?* [26].

It is important to bear in mind that a simulation model constitutes an abstraction of the real system, and so it represents only the parts of the system that are considered important to model. The main advantage that modelling and simulation offer is the possibility of trying different decisions and analysing their results in systems where the cost or the risk of real experimentation is prohibitive.

Simulation models can be classified according to different criteria:

- Time: If the variables that describe the model depend on time, and if time constitutes a model variable, then we are dealing with a *dynamic* model; otherwise, it will be termed *static*.

- Evolution and state: If the model can change its state continually over time, and if its variables can assume values within a continuous range, the model is a *continuous* model. If changes in the model happen only at specific points in time or in the process, and the variables defining its state are discrete, then the model is a *discrete* model.

- Results (i.e. if results can be predicted accurately): If different simulations of the model with the same values for the input parameters always yield the same results, the model will be *deterministic*. Otherwise the model will be *stochastic*.

For the purpose of this study we built a deterministic, continuous and dynamic model since we intended to observe the evolution of a process over time. For this kind of model we used the

System Dynamics (SD) paradigm, since it allows for the analysis of cause-effect relationships and feedback links among different variables of the process [7], [27].

Thus, the chosen modelling approach provides the following advantages:

1. A vision of the simulated usability evaluation process as a control system. Developers (i.e. project managers and directors) would be able to control the process by varying inputs and observing the effects of the changes on the process outputs. In our case, the simulated process of usability evaluation can be controlled by modifying the number and expertise of the evaluators, which will have an effect on the *time*, *cost* and number of *problems found*, in the process.

2. It makes visible the evolution of key process indicators *over* time. It is thus possible to obtain not only the final values of the main variables of the process, but also to see how their behaviour varies during the process. This enables corrective or preventive measures to be adopted. This contrasts with the Equation 1 of Nielsen and Landauer [3], which only relates the number of usability evaluators to the number of problems found and offers final values of the variables, without providing visibility of how the process of usability evaluation has evolved.

The discipline of modelling the software process with System Dynamics started with Abdel-Hamid in 1984 [28]. Since then, several simulation models have been developed to respond to different questions related to the software development process. A comprehensive view can be obtained by combining two systematic literature reviews of software process simulation modelling [29] (the initial stage) with [30] (the extended stage). Also it is possible to find an important contribution to the system dynamic modelling body of work in [31].

In this Section we review the main software process simulation models. The models have been classified according to their purpose:

- **Strategic management**: One of the most relevant models that help project managers make decisions on strategic management issues is the model developed by Abdel-Hamid and Madnick [32]. This model is a seminal work and a mandatory reference in this field. Its importance lies in being the first application of SD to software project management. Recently, Uzzafer [33] presented a model to simulate the implications of strategic decisions on factors such as cost, risk, budget and schedule of software projects.

- **Planning, operational management and control**. Among the simulation models that can be used in project planning, the two models devised by Kellner [26] and Pfahl and Lebsant [34], respectively, stand out. Kellner uses deterministic and stochastic models to simulate industrial software processes. Pfahl and Lebsant introduce a new approach named IMMS

(Integrated Measurement Modelling and Simulation) based on industrial experiences, performed with dynamic simulation models.

Dertzbacher et al. [35] presents a simulation model, integrated into a tool that supports project management centred processes. More recently, Zhang et al. [36] introduces the first simulation model of a formal verification process. The model aims to support further understanding and investigation of the dynamic characteristics of the process and to support planning and optimization of future process enactment.

- **Process improvement:** Simulation can help in decision-making in the field of process improvement because it allows the impact of a change in the process to be predicted. Within this field, the dynamic simulation model introduced by Madachy [37] is of great importance because, together with Abdel-Hamid´s original model [32], it is one of the dynamic models that represent in greater detail the whole software development process. Other simulation models are specially designed and oriented towards evaluating the results of various measures for process improvement. Several models have been developed with reference in the Capability Maturity Model (CMM), among which the models proposed by Raffo and Kellner [38] and Ruiz et al. [39] are worth noting. Hsueh et al. [40] presents a study about how to embed CMMI process area goals into the process definition stage to satisfy organization process improvement. In this research, they propose a UML-based approach to define, verify, and validate an organization's process. In the scope of software testing, Saurabh [41] presents an SD model of software development with a special focus on the unit test phase. This work is partially based on Collofello's et al. [42] work about modelling the software testing process under the SD approach.

- **Software evolution:** In the last few years the number of simulation models that address aspects of software evolution has increased considerably, making it one of the most modelled software processes. According to Zhang et al., software process simulation has to evolve to effectively investigate the ever-changing software processes [36]. This author has contributed several studies about the equivalence and model conversion between quantitative and qualitative process modelling. He uses the model transformation scheme to transform a quantitative continuous (SD) software evolution process model into its qualitative form for simulation [44], [45]. Araújo et al. [46] also describe an SD simulation model aimed at supporting the industrial software decay.

- **Software Engineering education**: Simulation can also be used to practise or learn project management. A simulation-based training environment allows us to determine the most likely results of common project management decisions that are often incorrect, for example,

advancing the coding stage, to eliminate revisions or to reduce the effort allocated to testing activities. Shtub [47] presents a new simulation tool for the teaching professional and students of Project Management.

- **Current trends**: Annually, the International Conference on Software and System Process (ICSSP) allocates some specific sessions to the presentation and discussion of the most relevant advances in software process simulation and modelling. Among current trends, we can find the application of simulation and modelling to various activities such as software development with agile methodologies [48], global software development [49] and IT service management processes [50], [51].

- **Application of Modelling and Simulation to Usability Engineering**: The terms 'Simulation' and 'Usability Engineering' can often be found together when searching digital libraries and citation databases of peer-reviewed literature. However, very frequently the works retrieved present results regarding the use of simulation of software interfaces to make early usability evaluations, as in Chatley et al. [52] or in Biswas and Robinson [53]. However, to the best of our knowledge and apart from [54], [55] , there are no previous studies in which simulation has been applied to aid decision-making in the field of usability engineering processes. Thus, we believe that the proposal presented in this paper constitutes an original contribution in this area. In our previous work [54] we developed an SD simulation model for the improvement of the usability of interactive systems through the application of a User Centred Design (UCD) process and its integration into the software development process. In the work presented in [55] we introduced the first version of the UESim model, which has been subsequently extended and validated as shown in the present work.

5 UESim Building

As stated before, several different proposals have been set forth for building simulation models, but they all have in common a series of steps and recommendations [7], [8], [9], [10], [11], Firstly, the purpose of the model needs to be established. This determines key questions to be addressed and the elements to be simulated. The purpose of the model also determines the scope of the intended model. This could be, for example, a portion of the life cycle, a whole development project, etc.

The output variables (i.e. the results) are the information elements needed to answer the key questions that were specified as part of the purpose of the model. Depending on the key questions being asked, different variables could be devised as the intended results of a process simulation.

In the abstraction phase of the construction of the model, the model builder needs to identify the key elements of the process, their interrelationships and behaviour. The focus should be on those aspects of the process that are especially relevant to the purpose of the model, and are believed to have a major influence on the output variables. Input parameters are needed to drive the simulation, and these depend upon the output variables desired and the abstraction process identified.

In the conceptualisation of the model the cause-effect relationships among the main variables of the model are expressed. Once the model has been conceptualised, its formalisation follows by introducing the mathematical equations that govern the behaviour of the variables. Finally, the model must be validated. To facilitate the task of building and validating UESim, we have used the Vensim$^{®}$ simulation environment.

The rest of this section describes in depth the outputs of each of these activities.

5.1 Purpose of UESim

The purpose of UESim is to analyse how the *evaluation team parameters* affect the behaviour of the problem detection phase in the usability evaluation process. This effect will be assessed through the process indicators of *time*, *cost* and *problems found*, which are generally used to assess process performance. One of the objectives of UESim is to serve as a customisable tool for decision-making during the planning stage of the process; it will allow managers to observe variations in process behaviour as a consequence of changes in the *evaluation team parameters* selected.

5.2 UESim Scope

In this study, we focus on simulating the last two phases of the usability evaluation process. To simulate the evaluation session it has been assumed that the analysis and aggregation of problems detected by the various independent evaluators is also carried out during the evaluation sessions. This means that the overlapping of similar problems found by different evaluators is eliminated, and a single list of final usability problems is duly compiled.

5.3 UESim Output Variables

The output variables are those elements that must be known in order to be able to answer the key questions for which the model was created. The performance of the usability evaluation process will be assessed on the basis of *time*, *cost* and *problems found* that are determined by the values of the following output variables:

- *Usability problems found*: this variable represents the number of different usability problems found after the analysis and aggregation of independent evaluations. It will hence determine the evaluation *problems found* indicator.
- *Accumulated variable cost*: this represents the amount of variable cost incurred by implementing the method. It will determine the evaluation *cost* indicator.
- *Time variable*: this represents the time required to complete the usability evaluation.

The behaviour of the *time* variable will be influenced by the dynamic nature of UESim, which will represent the behaviour of the process during the period of time required to complete the usability evaluation. Therefore, the output variables *Usability problems found* and *Accumulated variable cost* will be presented as a function of the *time variable*.

5.4 UESim Input Parameters

The main input parameters of UESim enable us to define the characteristics of the process at any time during the simulation.

- *Number of evaluators E(i)*: This composite parameter includes the three parameters known as *evaluation team parameters*:
  - Number of evaluators;
  - Expertise of the evaluators;
  - Dedication of the evaluators.

  Therefore, this parameter is divided into three other parameters specifying the particular level of expertise employed; it is also possible to specify different levels of dedication, in terms of the amount of time (hours) the evaluator is engaged in the process. Thus, during the simulation these three aspects of the composition of the evaluation team can be changed
  - *No. of Evaluators E1*: Represents the number of novice evaluators or with hardly any expertise in usability (level 1).
  - *No. of Evaluators E2*: Represents the number of evaluators with expertise in usability (level 2).
  - *No. of Evaluators E3*: Represents the number of evaluators with expertise both in usability and in the domain, i.e. with knowledge of the specific system whose usability is to be evaluated (level 3).

  The value for this parameter may be estimated from other similar studies or previous collections of project metrics.

- *Interface usability problems*: Represents the total number of estimated usability problems that exist. This value will depend on the size and type of system to be evaluated and has been obtained from other studies or collections of project metrics.

- *Single evaluation capacity E(i)*: this parameter is also divided into E1, E2, and E3 depending on the level of expertise being employed. It represents the number of usability problems that an evaluator with that expertise can detect and describe per unit of time (per hour). Its value can be estimated from previous projects.

- *Single cost E(i)*: this composite parameter is also divided into E1, E2, and E3 according to the level of evaluator expertise. It represents the remuneration (in Euros) per unit of time (hour) of each type of evaluator.

- *Problems Single proportion E(i)*: this parameter is also divided into E1, E2, and E3 according to the level of expertise. It represents the averaged proportion of the existing total problems that a single evaluator is thought capable of finding according to their level of experience (it has been describes as p in the Equation 1. It may be estimated from other similar studies, previous collections of project metrics, or it can be estimated using Nielsen´s mathematical model [3], [23].

- *Capacity Adjusting factor*: this parameter allows us to customise the model to describe the circumstances under which the evaluation is taking place, by setting some factors which may affect the process of usability evaluation but are not related to the make-up of the evaluation team. The additional factors considered in UESim are the following:
  - *Stage of the development lifecycle:* this represents the degree of influence of the stage in the development lifecycle of the product whose usability is being evaluated; the evaluation may be performed early in the design phase or later, after several iterations of design, test and re-design.
  - *Type of system*: this represents the degree of influence of the type of system whose usability is being evaluated: e.g. web interface, a corporate information system, etc.
  - *System size*: this represents the influence of the size of the system whose usability is being evaluated.
  - *System complexity*: this represents the influence of the complexity of the system whose usability is being evaluated.
  - *Type of users of the final system*: this represents the influence of the context of the prospective system end-users. They may comprise a homogeneous or heterogeneous group; they may be known or unknown to the developers; they may change during the

course of the development; there may be significant cultural differences between users, etc.

- *Set of usability guidelines*: this represents the degree of influence of the set of heuristics, standards or evaluation tasks, employed in the usability evaluation.

The effect of these factors in the evaluation process is modelled as multipliers that increase or reduce the capacity of the evaluation team depending on the circumstances in which the evaluation is being made. This model uses these independent factors to calculate the *capacity adjusting factor* in a similar way that the software cost estimation model COCOMO II uses its *cost drivers* to calculate the *effort adjustment factor* [56]. In this sense, each of the above mentioned factors may take different values within the range ['Very Low', 'Low', 'Nominal', 'High', 'Very High'] depending on the influence that each one is estimated to have in the particular evaluation process being simulated. The quantitative value (weight or rating) corresponding to each qualitative value can be calibrated by the model user depending on the specific project properties. For example, given a certain factor, if the 'Nominal' qualitative value has a quantitative value corresponding to 1, the rest of the qualitative values could be set as follows: Very Low: 0.9; Low: 0.95; High: 1.05; and Very High: 1.1. The final value for the overall *Capacity Adjusting Factor* parameter is obtained by multiplying the values assigned to each of the factors described above, as commonly occurs in other estimation models in Software Engineering as the software cost estimation model COCOMO II [56] or the software estimation method Use Case Points [57] .

5.5 UESim Conceptualisation

The concepts involved in a System Dynamics model are commonly represented in a causal loop diagram. The diagram presented as Figure 2 brings together the various cause-effect relationships found in the usability evaluation process.

The relationships are represented by arrows labelled as positive or negative. A positive causal link means that the two inter-connected variables change in the same direction, i.e. if the variable shown as the origin of the link increases, the other variable also increases. Negative causal links are links in which the variables change in opposite directions (the increase of one causes the decrease of the other linked variable, or the decrease of one variable causes the increase of the other).This diagram also includes the reinforcing or goal-seeking feedback loop that governs the evolution of the detection of usability problems during the running of the usability evaluation. This feedback loop models the transformation of the theoretical *usability problems to be found* into the actual *usability problems found* through the application of a certain *evaluation capacity*.

As mentioned before, this evaluation capacity will mainly depend on the *number of evaluators* and the *evaluators' expertise*. In addition, this evaluation capacity will also be affected by the *Capacity adjusting factor* and the extent of problem *overlapping* that may arise as the various problems found by each of the evaluators employed are aggregated during the course of the evaluation. It is postulated that, as the evaluation is performed, the problems reported are analysed by an observer who eliminates overlapping, and compiles a single list of final problems. To simulate this action, the model assumes an *overlapping* rate which depends only on the number of evaluators, disregarding their expertise. The overlapping rate is obtained using an average overlapping value between two given evaluators [16], [58] and increasing this value as the number of evaluators increase. The overlapping effect acts by decreasing the overall *Evaluation capacity* of the team of evaluators.
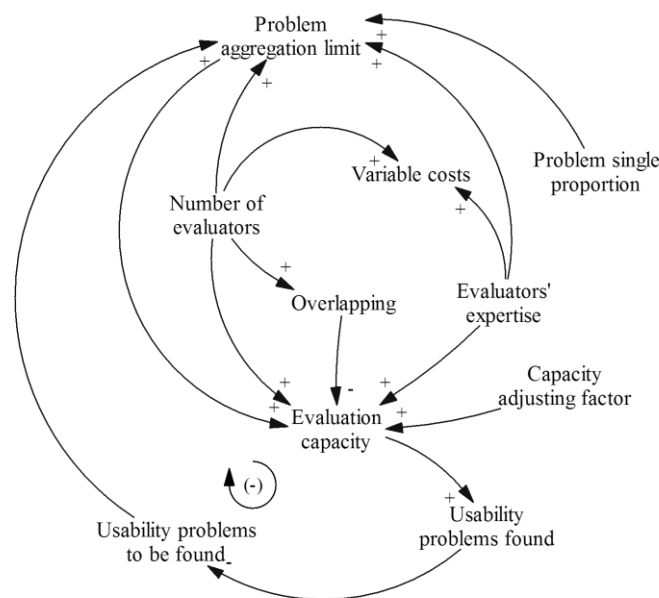

Fig 2. UESim causal loop diagram

In addition, the *problem aggregation limit* determines the total number of problems that is possible to find, in accordance with the initial estimation of *usability problems to be found*, the *number of evaluators* involved, their *expertise*, and the *problem single proportion*. Equation 1 is used to calculate the value of the *problem aggregation limit* that determines the end of the evaluation process. When the u*sability problems found* reaches this value the simulation stops closing the negative feedback loop.

Finally, the *number* and *expertise* of evaluators affects the *variable costs* of the evaluation process as it progresses in time; these costs will depend on the expertise of evaluators since the single evaluator cost input parameter will vary in accordance with the level of expertise.

5.6 UESim Formulation

Once the model has been conceptualised, its formalisation as a mathematical model follows. We have used stock and flow diagrams in this step since they help us to represent the structure of UESim and provide a bridge to simulation modelling by allowing the appropriate equations to be assigned to this structure and the relationships among the variables involved to be identified.

Stock and flow diagrams [27] are the main forms of notation used to represent System Dynamics simulation models. They are made up of three main types of element: level or stock variables, flow variables, and auxiliary variables. Level variables are represented by rectangular boxes and they symbolise cumulative elements (stocks). Flow variables are represented by arrows linking level variables; these arrows have "valves" that symbolise the progression from one level to the next. Flow variables represent the growth or decline of levels when the flow between communicating levels is turned on or off. Auxiliary variables are the rest of the elements in the process that have an influence upon it. It is also necessary to have constants or input parameters that represent external influences on the model and facilitate calibration. Figure 3 shows the stock and flow diagram of UESim where some elements have been intentionally simplified for clarity purposes. It should be noticed that result variables have been represented as the two level variables: *usability problems found* and *accumulated variable cost*. Two flow variables have been added: *problem detection rate* and *cost increase rate*. The first represents the flow between usability problems to be found and usability problems actually found; the second will increase the level variable: *accumulated variable cost* as the simulation progresses. The input parameters have been highlighted by showing their designation in capital letters.
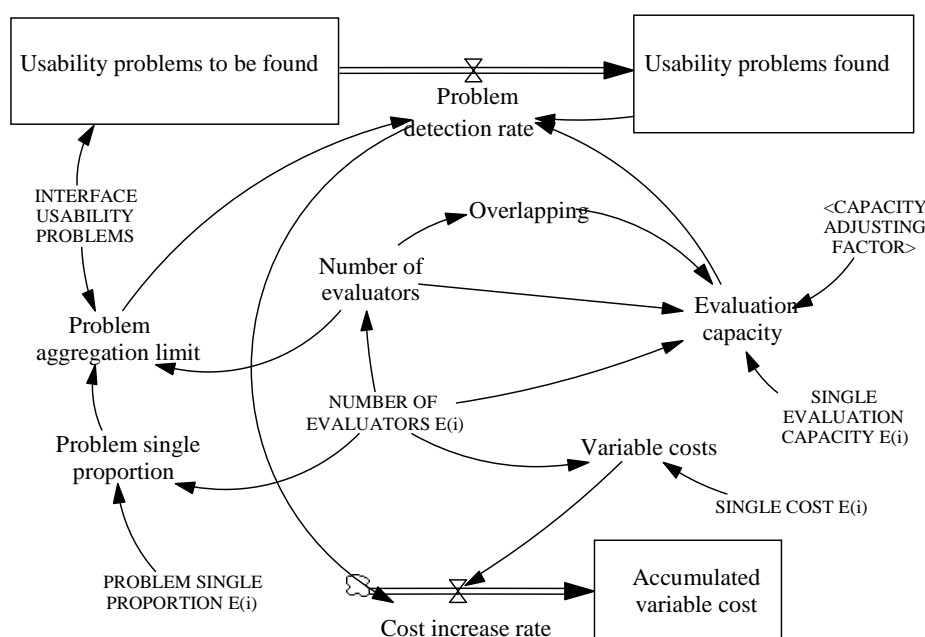

Fig 3. Simplified stock and flow diagram

16

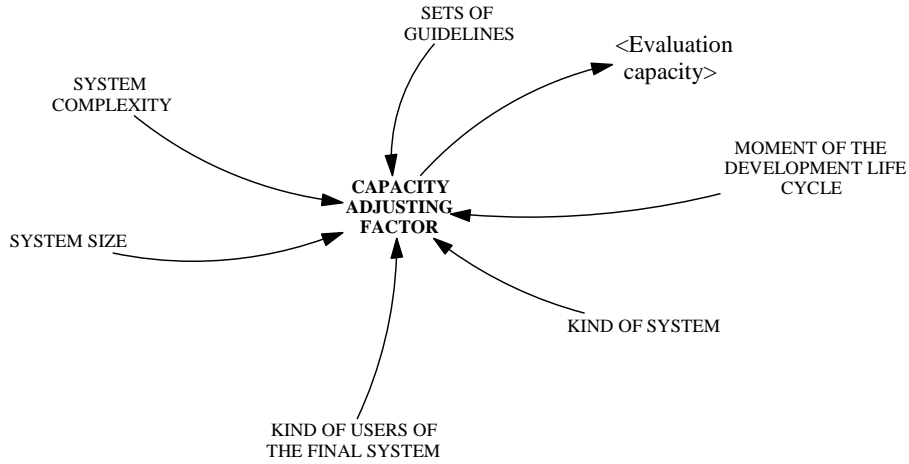Figure 4 shows the detailed diagram of the *Capacity adjusting factor*.

Fig 4. Detailed stock and flow diagram of the *Capacity adjusting factor*

Equations 2, 3, 4, 5 and 6 are the main behaviour-governing equations of UESim. Mathematically, the basic structure of a formal system dynamics computer simulation model is a system of coupled, nonlinear, first-order differential (or integral) equations. Simulation of such systems is easily accomplished by partitioning simulated time into discrete intervals of length *dt* and stepping the system through time one *dt* at a time. Level variables are always defined as differential equations containing the integration of all the flow variables that affect them using the standard notation:

$$\text{level} = \text{level}_{t=0} + \int_0^t (\text{inflow} - \text{outflow}) \, \text{dt}$$

In this case the level variables are affected only by the *Problem detection rate* flow variable. Equations 2 and 3 correspond to the level variable: *usability problems to be found (upbf)*. Equation 3 indicates the initial value for Equation 2. Equations 4 and 5 correspond to the level variable: *usability problems found(upf)*. Equation 5 indicates the initial value for Equation 4.

$$\text{upbf} = \text{upbf}_{t=0} - \int_0^t \text{Problem detection rate} \, \text{dt} \qquad (2)$$

$$\text{upbf}_{t=0} = \text{Interface Usability problems} \qquad (3)$$

$$\text{upf} = \text{upf}_{t=0} + \int_0^t \text{Problem detection rate} \, \text{dt} \qquad (4)$$

$$\text{upf}_{t=0} = 0 \qquad (5)$$

Equation 6 corresponds to the flow variable: *problem detection rate*. The value for this flow variable is conditioned by the most important decisions represented in UESim. In this case, it takes its value from the variable: *evaluation capacity* until the evaluation is completed. The *evaluation capacity* will vary depending on the input parameters and depending on the *overlapping* detected during the evaluation session.

$$
\text{Problem detection rate} = \begin{cases} 0 & \text{if usability problems found} \geq \text{problem aggregation limit} \\ \text{Evaluation capacity} & \text{if usability problems found} < \text{problem aggregation limit} \end{cases} \tag{6}
$$

Once UESim has been developed and the behaviour-governing equations have been introduced, the model validation phase follows.

6 UESim Verification and Validation

As Barlas stated [59], the validation of a System Dynamics model is technically difficult, because there are no established formal tests, such as statistical tests, to decide if the structure of a model is close enough to the real structure of the system. Furthermore, even standard statistical tests cannot be used in validating the behaviour of a system dynamics model, because of problems of autocorrelations and multicollinearity [59], [9].Therefore, some of the most relevant literature in the field of System Dynamics model validation [31], [60], [61], [62], [63], focus on the importance of model assessment in the validation of its internal structure using tools of informal and qualitative natures, such as expert reviews, inspections, walkthroughs, etc. "For dynamic models, validity primarily means validity of the internal structure of the model, not its output behaviour, and validation of the internal structure cannot be made entirely objective, formal and quantitative" [59]. "Structural validity of a simulation model (i.e. right behaviour for the right reasons) is a stringent measure to build confidence in a simulation model regardless of how well the model passes behaviour validity tests"[64]. Then, the verification and validation are focused on building confidence on the model as a reasonable representation of the system and in its usefulness in the provision of results [65].

The following paragraphs describe the validation scheme used, the techniques applied and the results obtained.

6.1 Validation Scheme

According to the proposals of Barlas [59] and Sterman [7], the validation scheme followed should pay special attention to aspects regarding the structural validation of the model. In this study, the scheme proposed by Barlas and Sterman is used to validate the model. Furthermore, some efforts have also been applied to validate the model behaviour by adding the validation of the simulation results in various cases.

The validation scheme followed consists of the following types of validation [66]:

1. Computerized **Model verification**: The test of model verification confirms that the model implementation is error-free.

   - *Dimensional consistency test:* This test verifies that the dimensions of the variables used are consistent and the units are correct.

   - *Syntax validation test*: This test verifies that the behaviour-governing equations of the model are free of syntax errors.

   - *Semantic validation test:* This test verifies that the behaviour-governing equations of the model are free of semantic errors.

2. **Structure validation** or conceptual model validation: these tests assess the model structure, by direct comparison with knowledge about real system structure.

   - *Structure confirmation test*: This test attempts to compare the form of the equations of the model directly with the form of the relationships that exist in the real system.

   - *Parameter confirmation test*: this kind of test involves evaluating the constant parameters against knowledge of the real system, both conceptually and numerically.

   - *Extreme conditions test:* these tests involve assigning extreme values to selected parameters and comparing the model-generated behaviour to the observed (or anticipated) behaviour of the real system under the same "extreme condition".

   - *Behaviour sensitive test*: these tests consist of determining those parameters to which the model is highly sensitive, and asking if the real system would exhibit similar high sensitivity to the corresponding parameters.

3. Behaviour validation or operational validation:

   - *Behaviour patterns tests*: these tests are oriented towards measuring how accurately the model can reproduce the major behaviour patterns exhibited by the real system. It is very important to note that the emphasis is on pattern prediction (periods, frequencies, trends, phase lags, amplitudes), rather than point (event) prediction.

6.2 Validation Techniques

Table 2 summarises the techniques used to perform the validation of UESim in relation to each test in which they were applied.

Table 2. Summary of tests performed and techniques used

Type of validation	Test name	Technique used
Model verification	Dimensional consistency test	Units check
	Syntax validation test	Check syntax
	Semantic validation test	Check model
Structure validation	Structure confirmation test	Face validity (Expert survey) and literature review
	Parameter confirmation test	Face validity (Expert survey) and literature review
	Extreme conditions test	Model simulation and literature review
	Behaviour sensitive test	Model simulation and literature review
Behaviour validation	Behaviour patterns tests	Face validity (Expert survey), Model simulation and Reference model comparison

The validation techniques used in this study are the following:

- Vensim® tools: the Vensim® simulation environment provides useful functionality for helping in model verification. The "Check Syntax" function checks that the behaviour-governing equations are free of syntax errors. The "Check Model" function checks that the behaviour-governing equations are free of semantic errors. The "Units Check" function verifies that the model is consistent, i.e. the units assigned to the different variables are consistent.

- Face validity: Among the main validation techniques, various sources in the field of simulation model validation recommend the use of interviews, inspections and/or opinions of experts [60], [7], [66]. In this case, to obtain opinions from experts about UESim, we carried out an on-line validation survey. This survey was distributed to several international communities of usability experts: ACM SIGCHI (Special Interest Group on Computer-Human Interaction), AIPO (Spanish Association of Human-Computer Interaction), UPA (Usability Professionals' Association), Interaction (specialist HCI group of the British Computer Society (BCS)), and the Centre for HCI Design (City University, London).

- Model simulation and literature review: some of the tests carried out involve the simulation of UESim to compare the simulation results with knowledge available on real system behaviour reported in the related literature.

- Reference model comparison: Using the model simulation technique referred to above, we compared if the results obtained are similar to the results provided by the reference model used [3], [23].

6.3 Validation Results

This section describes the results obtained in each of the types of validation conducted. Successful results were obtained for all tests performed.

1. Computerized **Model verification:** all the tests performed using the Vensim® simulation environment concluded that the implementation of UESim is error-free.

2. **Structure validation**: For this type of validation, the techniques used are: literature review, model simulation and expert survey.

   a. Results from the literature review: the structure and parameters of UESim are consistent with the information provided in the related literature [2], [3], [19], [20], [67], [68], [69], [70], [71].

   b. Results from the model simulation: for the simulation of UESim the tests of *Extreme conditions* and *Behaviour sensitive* were performed. The results of the simulation, in each test, were compared with the information extracted from the literature observing similar trends from a qualitative point of view.

   c. Results from the expert survey: approximately 100 respondents answered the questionnaire. The answers were filtered to take into account only the questionnaires corresponding to the answers of the respondents who described themselves as experts in usability evaluation (50 respondents); that is, persons who have chosen one of the following options: a) "I am familiar with usability and I have carried out a usability evaluation" or b) "I frequently carry out usability evaluations".

   The survey was divided into three parts: the first part contains general questions to classify the population, the second part contains questions regarding UESim structure, and the third part contains questions about UESim behaviour. The survey contained a total of twenty questions in which we asked for the degree of the respondent´s agreement on a five point Likert scale. Some questions also included an explanatory text field to get feedback from the expert.

The survey has been verified using the parameter *quality of answer*. If the *quality of answer* is less than 60%, the survey should be considered unsatisfactory. In this case, the value obtained for this parameter is 91% and has been calculated using Equation 7:

$$QA = \left(\frac{\sum QAQ}{\text{Number of survey responses received}} \right) \times 100 \quad (7)$$

where QAQ is defined as the quality of each of the survey response, which is calculated using Equation 8:

$$QAQ = 1 - \left(\frac{\text{Number of answers : "do not know or do not answer"}}{\text{Number of questions}} \right) \quad (8)$$

To analyse the results of the survey, for the *Structure confirmation test* and the *Parameter confirmation test,* we identified a number of indicators whose values are obtained from Equation 9

$$IV = \left(\frac{\sum \text{score for the indicator}}{\text{Number of survey responses received}} \right) \quad (9)$$

Table 3 shows the scores associated with each type of answer.

Table 3. Scores associated with each type of answer.

Type of answer	Associated score
Completely disagree	-2
Partially agree	-1
Quite agree	1
Very much agree	2

The indicators selected for structure validation are the following:

- ELA (Expertise Level Agreement): this indicator represents the level of agreement with the expertise level considered.
- IPA-IUP (Input Parameter Agreement-Interface Usability Problems): this represents the level of agreement with the *interface usability problems* input parameter.
- IPA-NE (Input Parameter Agreement- Number of Evaluators): this represents the level of agreement with the *number of evaluators* input parameter.
- IPA-EC (Input Parameter Agreement- Evaluation Capacity): this represents the level of agreement with the *single evaluation capacity* input parameter.

- IPA-VC (Input Parameter Agreement- Variable Costs): this represents the level of agreement with the *single variable costs* input parameter.

- IPA-PP (Input Parameter Agreement- Problems Proportion): this represents the level of agreement with the *problem single proportion* input parameter.

- AFA-SUG (Adjusting Factor- Set of Usability Guidelines): this represents the level of agreement with the *set of usability guidelines* adjusting factor.

- AFA-MDLC (Adjusting Factor- Moment of the Development Life Cycle): this represents the level of agreement with the *moment of the development lifecycle* adjusting factor.

- AFA- KS (Adjusting Factor- Type of System): this represents the level of agreement with the *type of system* adjusting factor.

- AFA-KU (Adjusting Factor-Type of Users): this represents the level of agreement with the *type of users* adjusting factor.

- AFA-SS (Adjusting Factor-System Size): this represents the level of agreement with the *system size* adjusting factor.

- AFA-SC (Adjusting Factor-System Complexity): this represents the level of agreement with the *system complexity* adjusting factor.

Table 4 summarises the values obtained for each indicator as stated in Equation 9.

Table 4. Values for the structure validation indicators.

Indicator Name	Indicator value
ELA	1
IP-IUP	1.5
IPA-NE	0.8
IPA-EC	0.8
IPA-VC	0.7
IPA-PP	0.7
AFA-SUG	1.1
AFA-MDLC	1.3
AFA-KS	1
AFA-KU	0.9
AFA-SS	0.9
AFA-SC	0.9

The answers quoted as "do not know, or do not answer" were not taken into account to calculate the indicator values shown here. If the *indicator value* is negative it is considered unsatisfactory. It can be seen that all the values are positive. Values less than

one are the result of differences of opinions among experts, expressed by means of the respondent opinion text boxes included in the survey.

3. **Behaviour validation**: the techniques used in this type of validation are: model simulation, survey of experts and comparison with the reference model.

   a) Results from the model simulation: The simulation of the model enabled the *Behaviour patterns test* to be performed. In this test the outputs of the simulation have been compared with the information extracted from the reference model [3] and both models follow similar trends.

   b) Results from the survey of experts: to analyse the results of the survey, to perform the *Behaviour pattern test,* we identify a number of indicators, whose values are obtained from Equation 9.

   - IPV-EC (Input Parameter Value – Evaluation Capacity): this represents the level of agreement with the values assigned to the *single variable costs* input parameter.
   - IPV-VC (Input Parameter Value- Variable Costs): this represents the level of agreement with the values assigned to the *single variable costs* input parameter.
   - IPV-PP (Input Parameter Value- Proportion of Problems): this represents the level of agreement with the values assigned to the *problem single proportion* input parameter.
   - GAB (General Agreement with Behaviour): this indicator represents the level of agreement with the general trend of the output variables.
   - EBA-UPF (Experiment Behaviour Agreement- Usability Problems Found): this represents the level of agreement with the results of the output variable: *usability problems found* in the experiments.
   - EBA-AVC: (Experiment Behaviour Agreement- Accumulated variable cost): this represents the level of agreement with the results of the output variable: *accumulated variable cost* in the experiments.
   - MU (Model Utility): this represents the degree of usefulness of UESim.

   Table 5 shows the values obtained for each indicator. It can be seen that all the values are positive and, therefore, the general level of agreement is high.

Table 5. Values for the behaviour validation indicators.

Indicator Name	Indicator value
IPV-EC	0.8
IPV-VC	0.9
IPV-PP	1.2
GAB	1.5
EBA-UPF	0.9
EBA-AVC	0.8
MU	1.1

c) Results from the comparison with the reference model: we can compare - from a qualitative point of view - the results of the simulation of UESim to the expected behaviour. For example, in both models to find 60% of the estimated problems, it would be necessary to use either just one expert evaluator (level 3), from 2 to 3 medium evaluators (level 2) or about 5 novice evaluators (level 1). In Figure 5 it can be seen that UESim follows a similar trend to those of the reference model used [23]. The simulation results are consistent with perceived system behaviour, especially in the top two levels of expertise.
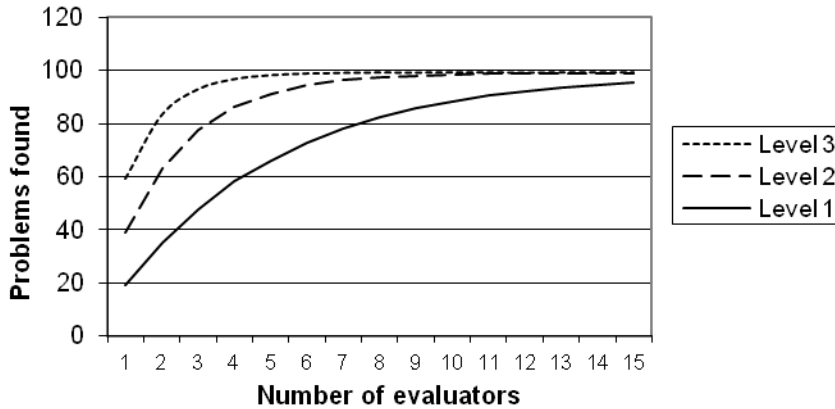
Fig 5. Number of problems found, according to the number of evaluators employed

7 UESim Simulation

To illustrate the application of UESim, three different experiments have been performed in which we analyse the effects of different compositions of the evaluation team on the outcome of the evaluation process, namely the *time*, *cost* and *problems found*. The *evaluation team parameters* are aggregated into the *number of evaluators E(i)* input parameter as explained in Section 5.4.

The rest of the input parameters of UESim will be set with the same values in the three cases, as described in the following section.

7.1 Setting the Values for UESim Input Parameters

Data from several different projects published in the related literature [2], [23], [72] were used to establish the values for the *problem single proportion* parameter. The values of the *single capacity* parameter were obtained from usability evaluation experiments performed with undergraduate students. The values for the *single variable cost* parameter were obtained from local companies. Table 6 shows the values adopted for the input parameters that will not be modified during simulation

As we commented in Section 5.4, the value of the *Capacity Adjusting Factor* parameter is obtained as a multiplier of the quantitative values of the several factors that make it up. Since our intention is to analyse the effect of different compositions of the evaluation team, to limit the complexity of the case, the input parameters corresponding to the *Capacity adjusting factor* were set at their nominal values (1) in all cases. Thus, in the experiments reported in this paper, these parameters have no effect on the simulation results.

To calculate the overlapping rate, we considered an average overlapping of 9% between two given evaluators [16]. It is possible for the user to set other values [58] for this input parameter.

Table 6. Input parameters established.

Interface Usability Problems (problems)	Parameter E(i)	E1	E2	E3
	Problem single proportion (% problems)	0.2	0.4	0.6
100	Single capacity (problems/hour)	2	4	6
	Single variable costs (euro/hour)	50	100	150

It is important to note that, before using UESim in an organisation, the values for the input parameters have to be set according to the organisation's historical data so that UESim describes the specific context and nature of the project. Thus, the model parameters can be easily calibrated to available empirical data and expert knowledge.

Once the base case for the evaluation process has been established, the effect of the make-up of the evaluation team on this process can be studied by setting up different scenarios. In this, study, each one of them is based on different decisions that could be made by an organisation planning to implement usability evaluations within its software development process. The following cases are based on actual problems encountered by real companies.

7.2 Case 1

7.2.1 Case Description

In the first experiment the goal is to show the evolution of the process when all the evaluators of the evaluation team have the same level of expertise and full-time dedication to the evaluation. To this end, we chose level 2 expertise, corresponding to a moderate degree of expertise in usability evaluation. Four different scenarios were established, each having a different number of evaluators. Table 7 shows the simulation scenarios for Case 1.

Table 7. Scenario Definition for Case 1

Number of Evaluators	Level of Expertise	Scenario Name
1	2	Sc1
4	2	Sc2
8	2	Sc3
12	2	Sc4

7.2.1 Simulation Results

Figure 6 shows the evolution over time of the two result variables: *usability problems found* and *accumulated variable cost* for evaluators with a moderate level of expertise (level 2). Looking at the results, it is possible to observe the evolution of time-related process behaviour and to estimate, for example:

  a) the proportion of problems found at any chosen moment, or

  b) the time it takes for the evaluation to be completed, or

  c) how long it takes to achieve some results.

This allows the developers to make decisions regarding:

  a) the optimum moment to stop the evaluation, or

  b) which combination of evaluators is most suitable.
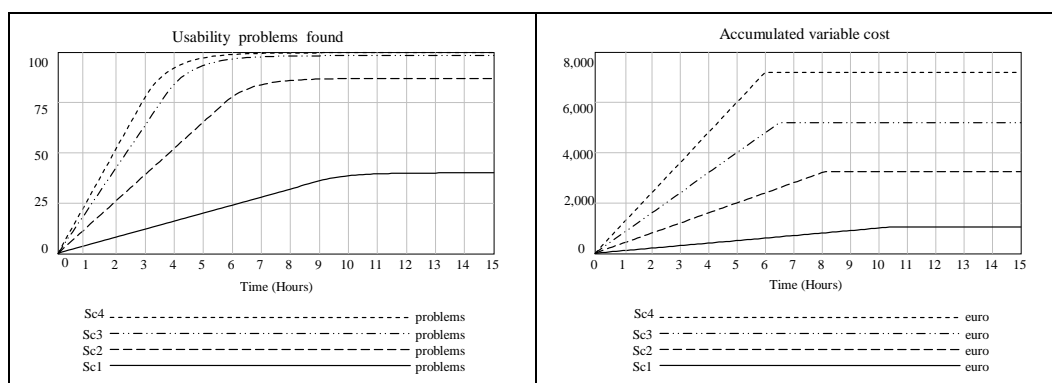
Fig 6. Comparing different numbers of expert evaluators (level 2)

The final decision will depend on the organisation's resources, on how crucial usability is for the system, and on how much benefit is expected to be achieved by improving usability.

The evolution over time of the result variables can be complemented with a representation that includes the final values for the key factors of different scenarios in a single view. For that purpose, we use a type of representation in the shape of a triangle, where each axis represents a key factor of the evaluation process: *cost*, *time* and *problems found*. These indicators coincide with the output variables of UESim explained in Section 4.3. To facilitate comparison between the four scenarios, the final values have been normalised between 0 and 1. Figure 7 shows the triangle of normalised final values for case 1. The number of *problems found* in scenario Sc2 compared with Sc1 increases by approximately 45%, while *time* decreases by 20%. It can be seen that the only factor that undergoes meaningful changes in the last two scenarios is *cost*, which increases by approximately 30% between scenarios 3 and 4. In this case, the organisation's decision would rule out Sc4. The final team make-up would be determined by the aims and limitation of the organisation or the project in particular.
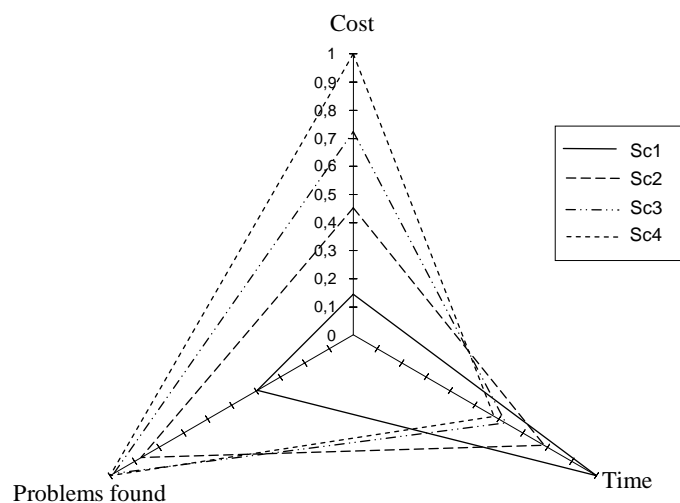

Fig 7. Triangle diagram for comparing final values (Case 1)

7.3 Case 2

7.3.1 Case Description

To set up Case 2, we will make the assumption that the aim of the organisation is to integrate the usability evaluation into its development process incurring only the minimum cost. For this objective, the three levels of usability expertise will be compared, using the lowest possible number of evaluators to achieve similar results in terms of *problems found*. Once again, in this

28

case we consider that all the evaluators work full-time in the evaluation. Table 8 depicts three simulation scenarios.

Table 8. Scenario definition for Case 2

Number of Evaluators	Level of Expertise	Scenario Name
4	1	Sc1
2	2	Sc2
1	3	Sc3

7.3.2 Simulation Results

Figure 8 shows the results in relation to time, for the output variables: *Usability problems found* and *Accumulated variable cost*. No significant difference is observed in the number of *problems found*, although Sc2 is slightly better than the other two. For the minimum *Accumulated variable cost*, the best scenario would be Sc3.
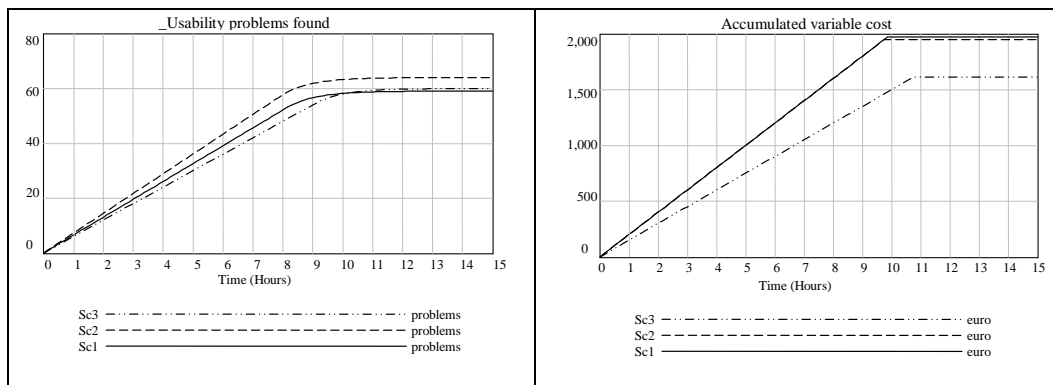
Fig 8. Results for output variables in Case 2

Figure 9 shows the triangle of normalised values for case 2.
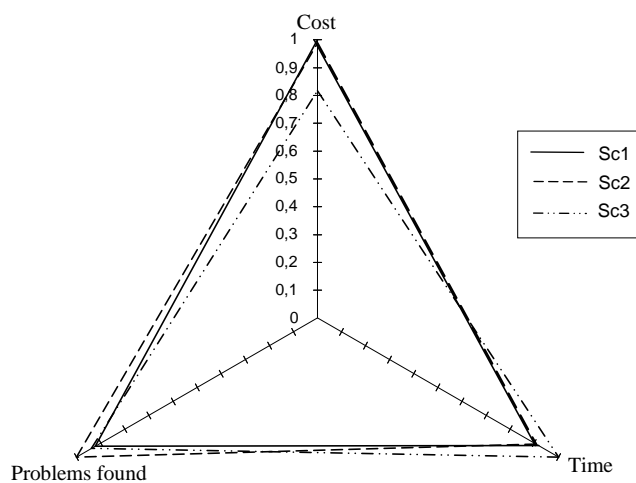
Fig 9. Triangle diagram for comparing final values (Case 2)

Figure 9 shows in more detail how scenario Sc3 improves *cost* results by 20% compared to the other two scenarios, between which there is hardly any difference (only 1.5%). However, scenario Sc2 improves *problems found* and *time* results by 10%. Depending on which factors the organisation considers most important, the choice would be between scenarios Sc2 and Sc3.

7.4 Case 3

7.4.1 Case Description

In Case 3, we assume, as in Case 2, that an organization is interested in minimizing the cost of the usability evaluation that is being planned. Let us say, for instance, that the maximum cost that the organisation can afford for the evaluation is 3,000 euros. In this case, we are interested in finding what team configurations can find a higher number of problems but keeping the cost of the evaluation under the economical restriction. To do that, we ran different scenarios each containing a different team configuration. It should be noted that, in this case, the composition of the evaluation team is mixed. That is, each team is composed of evaluators that have different levels of expertise and dedication to the evaluation. Table 9 shows the settings for each scenario simulated. The first three columns of the table show the values for the number and level of expertise of the evaluators that have full dedication to the evaluation. The following three columns set the values for the evaluators that do not have full dedication to the evaluation: number of evaluators, level of expertise and the amount of time they spend in the evaluation. For instance, in Scenario 2 (Sc2) the team of evaluators is composed of five evaluators that work full-time in the evaluation; one of them has Level 2 expertise and the other four have Level 1 expertise. In addition, there is one evaluator more that has Level 2 expertise and works in the evaluation only during the first two hours of the scheduled time. Once that time has passed, the evaluator stops working in the evaluation.

Table 9. Scenario definition for case 3

Full Dedication			Not Full Dedication			
No. Ev Level 1	No. Ev Level 2	No.Ev Level 3	No&Dedic Ev Level 1	No&Dedic Ev Level 2	No&Dedic Ev Level 3	Scenario Name
6	0	0	0	0	0	Sc1
4	1	0	0	1(2h)	0	Sc2
5	0	0	0	0	1(4h)	Sc3
2	2	0	0	0	1(2h)	Sc4
0	0	2	0	0	1(3h)	Sc5

7.4.2 Simulation Results

Figure 10 shows the results, in relation to *time*, for the output variables: *Usability problems found* and *Accumulated variable cost* for Case 3. It can be seen that all the scenarios have similar values in the *Accumulated variable cost*, since having a cost no higher that 3,000 euros was set as a restriction.
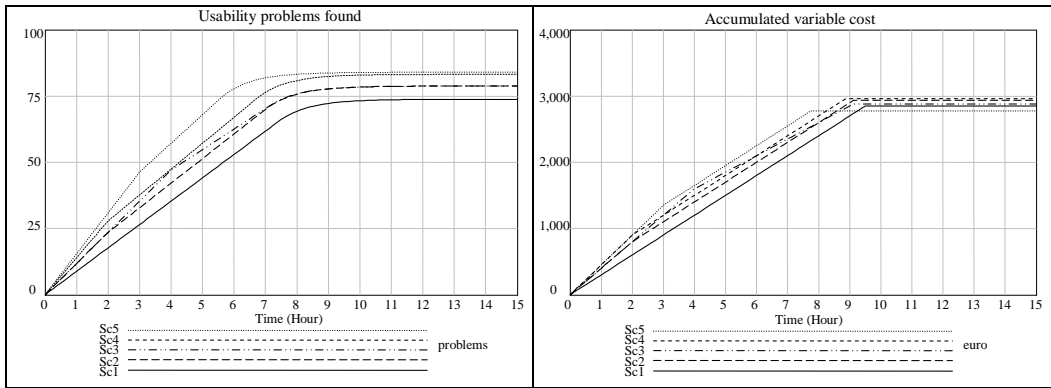

Fig 10. Results for output variables in case 3

Figure 11 shows the triangle of normalised values for Case 3. Changes in *cost* are small, about 2% or 3%, but the *time* factor improves almost by 20% between scenario Sc5 and Sc1 and problems found improves by 10%. In this case the best scenario is the Sc5 because it reaches approximately the same amount of problems found as the Sc4 but using about 10% less time and around 7% less cost.
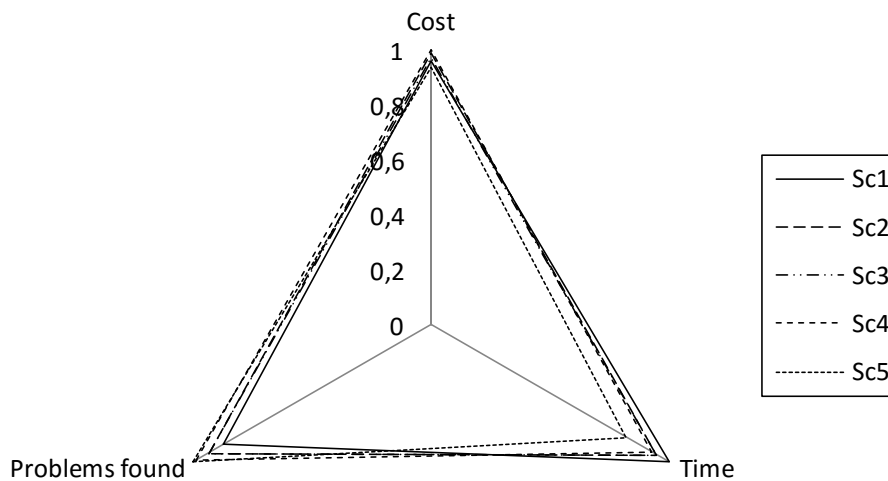
Fig 11. Triangle diagram comparing final values (case 3)

8 Discussion and Threats to Validity

In the previous section, we saw how the results of the simulation model changes in each case depending on the composition of the evaluation team. The simulation of the model shows us the time evolution of the main variables. These results could be of benefit to managers of user evaluation processes, especially concerning the make-up of the team. The ultimate decision regarding composition will, nonetheless, be conditioned by the specific goals and limitations of the organisation or project in hand.

As we discussed at the beginning of Section 6, it is difficult to fully validate a simulation model due to, among other reasons, the lack of historical data of projects and the diversity of organisations and project environments [65]. In that section we also illustrate, using many references, that the importance of the validation of a simulation model lies in the confidence that the model is able to generate in its users in terms of its internal structure.

The results presented about the validation of UESim's structure show a high level of agreement of the expert community about the assumptions modeled and the input parameters selected.

Regarding the validation of the model behaviour, the values for the simulation model input parameters used in the experiments presented in this work were extracted from the limited project data provided from related literature and therefore, we have to question the accuracy of quantitative simulations results.

To use UESim as a predictive tool in an industrial context, it is necessary to calibrate it to the specific process and features of the organisation. It is important to remember that the model is a customisable tool, providing an extensive set of configuration parameters that, once calibrated, describes the project or organisation in particular. This is a common feature in most of the simulations models found in the literature [65], [73], [74]. Nevertheless, the results of the experiments presented in this paper were included in the expert survey to validate the model behaviour. This survey also addressed the overall assessment of the model usefulness. The results obtained for these indicators are shown in the validation section (6.3). These results together with those obtained for the structure validation of the model are considered sufficient to conclude that the model generates a successful level of confidence in the HCI expert community.

9. Conclusions

At the outset of the paper, we established two general research lines to be addressed in this research.

Firstly, we posed whether it was possible to simulate the usability evaluation process, as has been carried out in the field of software (RQ1). Our research has shown that it is also possible to apply modelling and simulation techniques to the field of usability engineering.

Secondly, and more specifically, in RQ2 we endeavoured to see whether modelling and simulation could be of benefit to decide the make-up of the evaluation team for usability evaluation. Our results show that UESim not only allows us to simulate evaluation behaviour under different team compositions, but also to alter these parameters during the process and observe the results.

Consequently, this work illustrates the benefits that the application of modelling and simulation techniques provides for the management of the usability evaluation process. The process has been conceptualised and formalised under the System Dynamics approach, and a simulation model named UESim has been built for the main phases of this process. In addition, the paper shows the validation process of the model and its results. Three different cases have been set forth to illustrate how UESim might be used. For each of these cases, the evolution over time of the output variables of UESim has been depicted graphically and analysed.

Simulation modelling in the context of usability evaluation is a very useful technique that improves the knowledge of the usability evaluation process allowing us to visualize the effect that its key input parameters have on the main process indicators. One of the main advantages of this technique is the possibility to experiment with different policies for managing the usability evaluation process. Thus, it could be considered a valuable tool to help decision-making on the implementation of the usability evaluation.

This research establishes the main benefits of using UESim in the management of the usability evaluation process. Some of the benefits identified are:

- UESim makes it possible to extend Nielsen's conclusions [3], [23] by giving an insight on how the evaluation process evolves over time, on the effect of different process management decisions, and on the variable costs associated with the options chosen. These are essential factors for any organisation that intends to implement any usability inspection method.

- It enables managers to see the effect that different compositions of the evaluation team have on the key process indicators, in terms of *cost*, *time* and *problems found*.

- It is a customisable tool allowing to adapt it to the specific circumstances of the project or the organization through its input parameters and the *Capacity Adjusting Factor*

- It illustrates the effects of mixing evaluators with different degrees of expertise and different levels of dedication to the evaluation.

- It allows us to change the values of the input parameters dynamically during the simulation, helping the decision-making even when the process has already started

To overcome the threats to validity, one of our main goals is to extend the model validation by evaluating the model in a real organization using the *Capacity Adjusting Factor* to adapt the model to the specific project circumstances.

UESim is part of a general simulation-based framework. We plan to add other models with a view to complete the simulation of all the processes of the usability engineering lifecycle. Our subsequent work will be aimed at the full development of simulation models of other methods and processes, to extend the application of these techniques in the field of usability engineering.

Acknowledgements

Appendix

This appendix includes the questionnaire for the experts' survey performed during the model validation. To improve understanding of the survey, it was originally accompanied by an explanatory text of the model and its performance. In order not to increase the extent of the paper, in this appendix we have decided to include only the questions that were performed in each one of the three parts of the experts' survey.

1 Introduction

1. Choose the option (just one) that you consider most appropriate to describe your degree of expertise in usability.

- o   I have neither experience nor any knowledge of it.
- o   I just know the concept.
- o   I know the concept, the different methods of evaluation and their application process.
- o   I am familiar with usability and I have carried out a usability evaluation.
- o   I frequently carry out usability evaluations.
- o   Others (please specify).

2. Choose the options (more than one can be chosen) that you consider most appropriate to describe your working relationship with usability and its evaluation:

- o   Related teaching dedication
- o   Specific teaching dedication
- o   Related researching dedication
- o   Specific researching dedication
- o   Related professional dedication
- o   Specific professional dedication

o   Others (please specify)

2   Model structure validation

3. Say how much you agree with the proposed classification of expertise levels of evaluators

   o   Completely disagree

   o   Partially agree

   o   Quite agree

   o   Very much agree

   o   Do not know or do not answer

4. Say whether you believe it necessary to bear in mind other aspects in relation to evaluators' expertise to classify them in levels of expertise.

5. Say how much you agree with the mathematical model described by Equation 1:

   o   Completely disagree

   o   Partially agree

   o   Quite agree

   o   Very much agree

   o   Do not know or do not answer

6. Choose the degree of importance you would give to each one of the input parameters commented in usability evaluation 1.None   2. Little 3. Quite a lot 4. A lot 5. Does not know or does not answer

- Interface usability problems   ☐
- Number of evaluators   ☐
- Evaluation capacity   ☐
- Variable costs   ☐
- Individual problem %   ☐

7. Choose the degree of importance that you assign to each of these parameters that make up the "adjusting factor" in usability evaluation: 1.None   2. Little 3. Quite a lot 4. A lot 5. Does not know or does not answer

- Stage the development life cycle at which the evaluation is carried out   ☐
- Type of system   ☐
- Set of usability guidelines   ☐
- type of users the final system will have   ☐
- System size   ☐
- System complexity   ☐

8. If you consider any input parameters to be missing, please point it out, give reasons and grade its importance.

9 . Choose the value you consider most appropriate, according to the level of expertise of evaluators, for the parameter "evaluation capacity" that represents the average number of problems per hour that an evaluator can find. (2,4, 6,8,10,12, more than 12)

- Level 1(beginner) ☐
- Level 2(expert) ☐
- Level 3(double expert) ☐

10. Choose the value you consider most appropriate, according to the level of expertise of evaluators, for the parameter "variable costs" that represent the amount of euro per hour paid to an evaluator. 2,4, 6,8,10,12, more than 12)

- Level 1(beginner) ☐
- Level 2(expert) ☐
- Level 3(double expert) ☐

11. Look at the graph above. Supposing an evaluation session of two hours, regardless of the expertise level considered, ¿Which curve do you think that best depicts the evolution of the percentage of problems that an evaluator encounters along this session? The x-axis shows the time elapsed during the evaluation session and the y-axis shows the amount of encountered problems.

- red curve
- green curve
- blue curve
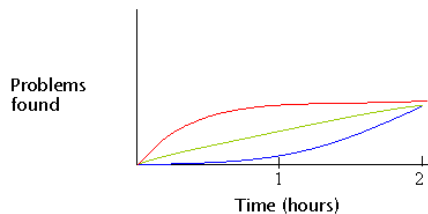
12. In case you consider that none of the above curves approximately describes the evolution of the number of problems encountered by an evaluator during the evaluation session, describe such behaviour.
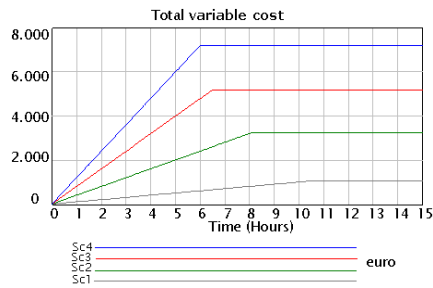
3 Model behaviour validation

13. The graph below shows the number of problems encountered in relation to time for the various simulation scenarios of experiment 1. Assess their level according to the results shown in the graph

- Completely disagree
- Partially agree
- Quite agree
- Very much agree
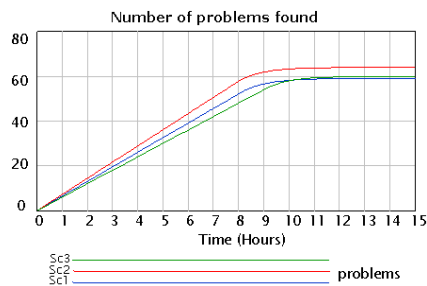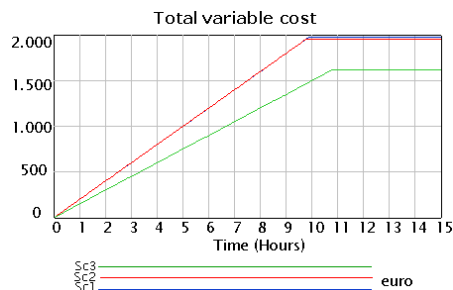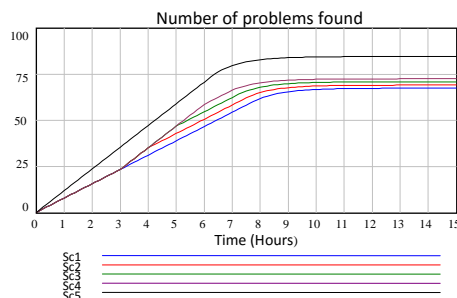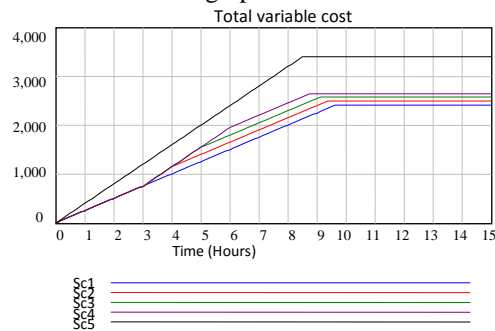- Do not know or do not answer
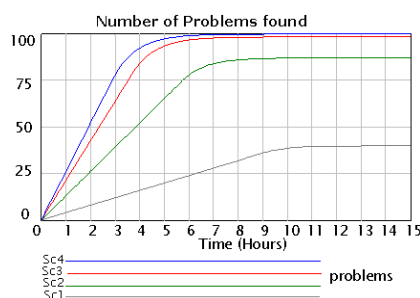
14. The graph below shows the total variable cost in euro in relation to time for the various simulation scenarios of experiment 1. Assess their level according to the results shown in the graph.

- o Completely disagree
- o Partially agree
- o Quite agree
- o Very much agree
- o Do not know or do not answer

15. The graph below shows the number of problems encountered in relation to time for the various simulation scenarios of experiment 2. Assess their level according to the results shown in the graph

- o Completely disagree
- o Partially agree
- o Quite agree
- o Very much agree
- o Do not know or do not answer

16. The graph below shows the total variable cost in euro in relation to time for the various simulation scenarios of experiment 2. Assess their level according to the results shown in the graph.

- o Completely disagree
- o Partially agree
- o Quite agree
- o Very much agree
- o Do not know or do not answer

17. The graph below shows the number of problems encountered in relation to time for the various simulation scenarios of experiment 3. Assess their level according to the results shown in the graph

- o Completely disagree
- o Partially agree
- o Quite agree
- o Very much agree
- o Do not know or do not answer

18. The graph below shows the total variable cost in euro in relation to time for the various simulation scenarios of experiment 3. Assess their level according to the results shown in the graph.

- o Completely disagree
- o Partially agree
- o Quite agree
- o Very much agree
- o Do not know or do not answer

Total variable cost

19. In case you do not agree with the results shown in the experiments, please explain why.

20. Evaluate the usefulness of the proposed model

- o Not useful at all
- o Not very useful
- o Quite useful
- o Very useful
- o Do not know or do not answer

References

[1] E. Law, ; V. Roto; M. Hassenzahl, A.Vermeeren, J.Kort, Understanding, scoping and defining user experience: a survey approach, Proceedings of CHII. User experience, 2009, pp. 719-28.

[2] J. Nielsen, R.L. Mack, Usability Inspection Methods, J. Nielsen, and R.L. Mack (eds). John Wiley & Sons, New York, NY, 1994, pp. 25-62.

[3] J. Nielsen, T.K. Landauer, A Mathematical Model of the Finding of Usability problems, Proceedings of the ACM INTERCHI'93 Conference. Amsterdam, the Netherlands, 1993, pp. 206-213.

[4] D.E. Avison, F. Lau, M.D. Myers, P.A. Nielsen. Action research. Communications of the ACM 42, 1999, pp 94–97.

[5] R.Mc. Taggart. Principles for participatory action research. Adult Education Quarterly, 41, 1991, pp168-187

[6] B.J. Oates. Researching information systems and computing. Sage, 2005.

[7] J. D. Sterman, Business Dynamics. Systems Thinking and Modeling for a Complex World. McGraw-Hill. 2000.

[8] I.J. Martinez, G. P. Richardson, Best Practices in System Dynamics Modeling. Plenary paper In Proceedings of the 29th International Conference of the System Dynamics Society, Hines, .H., Diker, V.G. (Eds.), Atlanta, GA, USA, 2001, pp: 1-22.

[9] A. M. Law, How to build valid and credible simulation models, Proceedings of the 2009 Winter Simulation Conference, 2009, pp 24-33.

[10] N. B. Ali, K.A. Petersen, A consolidated process for software process simulation: State of the art and industry experience, Proceedings of the 38th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2012 , art. no. 6328171 , 2012, pp. 327-336.

[11] M.I. Kellner, R. J Madachy, D. Raffo, Software Process Simulation Modeling: Why? What? How?. The Journal of Systems and Software, (46: 2), 1999, pp. 91-105.

[12] Vensim (Ventana Systems.inc) http://www.vensim.com [last accessed: January 2014]

[13] Encuesta Fácil, S.L. http://www.encuestafacil.com/Home.aspx [last accessed: January 2014]

[14] J. Nielsen, Usability Engineering. Morgan Kaufmann, 1993.

[15] M. Hertzum, N.E. Jacobsen, The Evaluator Effect: A Chilling Fact about Usability Evaluation Methods, International Journal of Human-Computer Interaction, 2003, 15:1, pp. 183-204.

[16] M. Hertzum, N.E. Jacobsen, R. Molich, Usability Inspections by Groups of Specialists: Perceived Agreement in Spite of Disparate Observations, In Extended abstract of the ACM CHI 2002 Conference on Human Factors in Computing Systems, CHI '02, 2002, pp. 662-663.

[17] E. L-C. Law, E.T. Hvannberg, Analysis of combinatorial user effect in international usability tests, In Proceedings of the Conference on Human Factors in Computing Systems, 2004 , pp. 9-16.

[18] M. Hertzum, R.Molich, N.E. Jacobsen. What you get is what you see: revisiting the evaluator effect in usability test. Behaviour & Information technology (2013) pp 1-32.

[19] J. Nielsen, R. Molich. Heuristic Evaluation of User Interfaces, In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Seattle, WA, USA, 1990, pp. 249-256.

[20] C.W. Turner, J.R. Lewis, J. Nielsen, Determining Usability Test Sample Size. International Encyclopedia of Ergonomics and Human Factors, 2006, Second Edition, Volume 3 Edited by Waldemar Karwowski, Boca Raton, pp 3084-3088.

[21] W. Hwang, G. Salvendy, Number of people required for usability evaluation: The 10±2 rule. Communications of the ACM 53 (5) , (2010), pp. 130-133.

[22] M. Schmettow, Sample size in usability studies. Communications of the ACM, Volume 55, Issue 4 (2012) pp 64-70

[23] J. Nielsen, Finding Usability Problems through Heuristic Evaluation, In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM CHI'92. Monterey, CA, USA, 1992, pp. 373-380.

[24] S. Kirmani, Heuristic Evaluation Quality Score (HEQS): Defining Heuristic Expertise. Journal of Usability Studies, Volume 4, Issue 1, November 2008, pp. 31-48.

[25] A. Woolrych, K. Hornbæk, E. Frøkjær, G. Cockton, Ingredients and Meals Rather Than Recipes: A Proposal for Research That Does Not Treat Usability Evaluation Methods as Indivisible Wholes. International Journal of Human-Computer Interaction, 27, 10 (2011), pp 940-970.

[26] M. Kellner, Software process modeling support for management planning and control, Proceedings of the First International Conference on the Software Process, Redondo Beach, California. IEEE Computer Society Press, Los Alamitos, CA, 1991, pp. 8-28.

[27] J.W. Forrester, World Dynamics. Cambridge MA, Productivity Press, 1973.

[28] Abdel-Hamid, T., The dynamics of software project staffing: An Integrative System Dynamics Perspective, Ph. D. dissertation, Massachusetts Institute of Technology, 1984.

[29] H. Zhang, B. Kitchenham, D. Pfahl, Reflections on 10 Years of Software Process Simulation Modelling: A Systematic Review, in Proceedings of International Conference on Software Process, ICSP 2008;Leipzig. Volume 5007 LNCS, 2008, pp 345-356.

[30] H. Zhang, B. Kitchenham, D. Pfahl, Software Process Simulation Modeling: An Extended Systematic Review, in Proceedings of International Conference on Software Process (ICSP'10), Paderborn, Germany: Springer (LNCS 6195), 2010, pp.309-320.

[31] R. J. Madachy, Software Process Dynamics, Wiley-IEEE Press, 2008.

[32] T. Abdel-Hamid, S. Madnick, Software Project Dynamics: An Integrated Approach, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[33] M. Uzzafer, A simulation model for strategic management process of software projects, The Journal of Systems and Software. volume 86, (2013) pp 21-37.

[34] D. Pfahl, K. Lebsant, Integration of system dynamics modelling with descriptive process modelling and goal-oriented measurement, The Journal of Systems and Software, 46, (2/3), 1999, pp 135-150.

[35] J. Dertzbacher, A.L.R. De Sousa, D.J. Nunes, A simulation model for process-centered software engineering environments using sensitivity analysis, In proceedings of 2011 Workshop-School on Theoretical Computer Science, WEIT 2011, 2011, pp 74-80.

[36] H. Zhang, G. Klein, M. Staples, J. Andronick, L. Zhu and R. Kolanski, Simulation Modeling of A Large Scale Formal Verification Process, in Proceedings of International Conference on Software and Systems Process (ICSSP'12), Zurich, Switzerland: IEEE, 2012, pp.3-12.

[37] R.J. Madachy, A Software Project Dynamics Model for Process Cost, Schedule and Risk Assessment. Ph.D. Dissertation. University of Southern California, Los Angeles, CA.1994.

[38] D. Raffo, M. Kellner, Chapter 16. Modeling Software Processes Quantitatively and Evaluating the Performance of Process Alternatives. K. En El Eman, N. Madhavji, (Eds.), Elements of Software Process Assessment and Improvement. IEEE Computer Society Press, Los Alamitos, CA. 1999.

[39] M. Ruiz, I. Ramos, M. Toro, A Dynamic Integrated Framework for Software Process Improvement, Software Quality Journal, 10. Kluwer Academic Publishers,(2002) pp. 181-194.

[40] N. Hsueh, W. Shen, Z. Yang, and D. Yang, Applying UML and software simulation for process definition, verification, and validation, Information and Software Technology, vol. 50, no. 9, (2008), pp. 897-911.

[41] K. Saurabh. Modeling unit testing processes a system dynamics approach ICEIS 2008. Proceedings of the 10th International Conference on Enterprise Information Systems, 1 ISAS, (2008), pp . 183 - 186.

[42] J.S Collofello, Y. Zhen, J.D Tvedt., D. Merrill, and I. Rus, Modeling software testing processes Conference Proceedings. International Phoenix Conference on Computers and Communications, (1996), pp. 289-293.

[43] H. Zhang. Simulation Modeling of Evolving Software Processes. Proceedings of the International Conference on Software and Systems Process. (ICssP 2012); Zurich, Switzerland: IEEE, 2012, pp.228-230.

[44] H. Zhang, B. Kitchenham, R. Jeffery, R. Qualitative vs. quantitative software process simulation modeling: Conversion and comparison, Proceedings of the Australian Software Engineering Conference, ASWEC, 2009, pp 345-354.

[45] H. Zhang, B. Kitchenham, R. Jeffery, R. Toward trustworthy software process models: An exploratory study on transformable process modeling. Journal of Software Maintenance and Evolution, Volume 24, Issue 7, (2012), pp 741-763.

[46] M.A.P. Araújo, V. F. Monteiro, G. H. Travassos, Towards a model to support in silico studies of software evolution. International Symposium on Empirical Software Engineering and Measurement, 2012, pp 281-289.

[47] A. Shtub, Project management simulation with PTB project team builder, Proceedings of the 2010 Winter Simulation Conference, 2010, pp 242-253.

[48] M. Wu, H. Yan, H. Simulation in Software Engineering with System Dynamics: A Case Study. Journal of Software, Vol 4, No 10 (2009), pp 1127-1135.

[49] S. Setamanit, D. Raffo, Identifying Key Success Factors for Globally Distributed Software Development Project Using Simulation: A Case Study. Proceedings of the International Conference on Software Process. (co-located with ICSE 2008), Leipzig, Germany, 2008, pp:320-332.

[50] E. Orta, M. Ruiz, M. Toro, Analyzing Strategic Business Rules Through Simulation Modeling, In Proceedings of the IFIP Conference on E-Business, E-Services and E-Society – I3E'09. Nancy, France, 2009, pp. 357-368.

[51] E. Orta, M. Ruiz, M. Toro, A System Dynamics Approach to Web Service Capacity Management, In Proceedings of the IEEE Seventh European Conference on Web Services – ECOWS'09, Eindhoven, The Netherlands, 2009, pp. 109-117.

[52] R. Chatley, J. Kramer, J. Magee, S. Uchitel, Model-based Simulation of Web Applications for Usability Assessment, Proceedings of the Workshop on Bridging the Gaps between Software Engineering and Human-Computer Interaction. (co-located with ICSE 2003), Portland, OR, 2003, pp: 5-11.

[53] P. Biswas, P. Robinson, Evaluating interface layout for visually impaired and mobility-impaired users through simulation, Universal Access in the Information Society, 2011, pp. 1-18.

[54] N. Hurtado, M. Ruiz, J. Torres, Towards Interactive System Usability through Simulation Modeling, In: Proceedings of the 6th International Workshop on Software Process Simulation and Modeling, ProSim 2005, St. Louis, MO (in CD- ROM)

[55] N. Hurtado, M. Ruiz, J. Torres, Using a Dynamic Model to Simulate the Heuristic Evaluation of Usability, In: Proceedings of the 12th IFIP TC13 International Conference on Human-Computer Interaction. INTERACT 2009. Uppsala (Sweden), 2009, pp: 912-915.

[56] B. Boehm, B. Clark, E. Horowitz, R. Madachy, R. Shelby, C. Westland, Cost Models for Future Software Life Cycle Processes: COCOMO® 2.0. Annals of Software Engineering, (1995), pp. 57–94.

[57] G. Karner, Metrics for Objectory. Diploma thesis, Linköping University 1993.

[58] E. Lai-Chong Law, E. Thora Hvannberg. Analysis of strategies for improving and estimating the effectiveness of heuristic evaluation. In Proceedings of the third Nordic conference on Human-computer interaction (NordiCHI '04) (2004). ACM, New York, NY, USA, pp 241-250.

[59] Y. Barlas, Formal Aspects of Model Validity and Validation in System Dynamics. Systems Dynamics Review. John Wiley & Sons. Vol. 12 n° 3, 1996, pp 183-210.

[60] J.W. Forrester, Industrial Dynamics. Portland, OR Productivity Press.1961.

[61] J.W. Forrester, P.M. Senge . Tests for Building Confidence in System Dynamics Models. In System Dynamics models. TIMS Studies in the Management Sciences, 14, (1980), pp:209-228.

[62] G.P. Richardson, A.L. Pugh, Introduction To System Dynamics Modeling With DYNAMO. Portland, OR: Productivity Press.1981.

[63] D.W. Peterson, R.L, Eberlein, Reality Check: A Bridge Between Systems Thinking and Systems Dynamics. Systems Dynamics Review 10(2-3), (1994), pp:159-174.

[64] H. Qudrat- Ullah, Structural Validation of Systems Dynamic and Agent Based Simulation Models, Proceedings of the 19th European conference on Modelling and Simulation. Riga. Latvia. 2005.

[65] K. Choi, D. Bae, Dynamic project performance estimation by combining static estimation models with system dynamics, Information and Software Technology. (51), (2009), pp 162–172.

[66] R. G. Sargent, Verification and validation of simulation models. Proceedings of the 2011Winter Simulation Conference, 2011, pp 183-198.

[67] O. Daly-Jones, N. Bevan, C. Thomas, Handbook of User Centered Design. IE2016 INUSE Deliverable D6.2.1 Version 1.31. Serco Usability Services, National Physical Laboratory, Teddington, Middx, UK. 2001.

[68] J. S. Dumas, J.C. Redish, A Practical Guide to Usability Testing. Rev. ed. Exeter, England, Portland, Or. , Intellect Books, 1999.

[69] G. Lindgaard, Usability Testing and System Evaluation. London: Chapman & Hall.1994.

[70] R. Molich, J. Nielsen, Improving a Human-Computer Dialogue. Communications of the ACM, (33:3), (1990), pp: 338-348.

[71] R. A. Virzi, Refining the Test Phase of Usability Evaluation: How Many Subjects is Enough. Human Factors, (34:4), (1992), pp. 457-468.

[72] R. G. Bias, R.G., D. J. Mayhew (eds), Cost-Justifying Usability. An Update for the Internet Age, Morgan Kaufmann, San Francisco. 2005.

[73] V. Garousi, K. Khosrovian, D. Pfahl, A customizable pattern-based software process simulation model: Design, calibration and application. Software Process Improvement and Practice. Volume 14, Issue 3, (2009), pp 165-180.

[74] K.G Kouskouras, A.C. Georgiou, A discrete event simulation model in the case of managing a software project. European Journal of Operational Research. Volume 181, Issue 1, 16 August 2007, (2007), pp 374–389.