TECHNICAL REPORT

RT – ES 746 / 13

Reporting Guidelines for Simulation-Based Studies in Software Engineering

Breno Bernard Nicolau de França
(bfranca@cos.ufrj.br)

Guilherme Horta Travassos
CNPq Researcher
(ght@cos.ufrj.br)

Systems Engineering and Computer Science Department

COPPE / UFRJ

Rio de Janeiro, May 2013

Summary

Abstract

BACKGROUND: In some scientific fields, such as automobile, drugs discovery or engineer simulation-based studies (SBS) have been performed in order to speed up the observation of phenomena and expand knowledge. The benefits have been many and great advancements are continuously obtained for the society. However, the simulation initiatives observed in the context of Software Engineering (SE) do not seem to reach the same lengths, when compared to other fields. In a recent *quasi*-Systematic Review performed to characterize SBS in the context of Software Engineering, we could identify several elements, concerning research protocols, simulation model building and evaluation, used data, quality of reports, among others. AIM: To build a set of reporting guidelines aiming at improving the' understandability and replicability of SBS. METHOD: To carry out a literature review on SE guidelines and simulation guidelines in other research areas. Besides that, compile these findings into the ones captured in the *quasi*-Systematic Review performed, which has the usually reported information regarding SBS. RESULTS: A comprehensive set of 20 reporting guidelines, condensed from general and specific guidelines for empirical research in SE, and also from other disciplines such as computer simulation, statistics, and medicine. Each guideline contains an associated description, examples, and rationale. CONCLUSIONS: The lack of reporting consistency can reduce understandability, replicability, and also compromise their validity. Therefore, an initial set of guidelines is proposed, aiming at improving the quality of SBS reporting, from several points of view, including authors, researchers, practitioners, and reviewers. Further evaluation should be done to assess the feasibility of the guidelines from the experts' point-of-view.

1. Introduction

Simulation-Based Studies (SBS) have been applied since the 1980s in Software Engineering. Many simulation models have been proposed on different Software Engineering (SE) domains. Such models capture, in some sense, knowledge and beliefs acquired over many years of research in these domains. However, it is very hard to find evidence obtained with such studies. Rather, simulation studies rely on proposing specific models, together with initiatives on trying their validation, being SBS performed in an *ad-hoc* fashion.

In order to characterize how the different simulation approaches found in the technical literature have been applied to simulation-based studies in the SE context, we undertook a *quasi*-Systematic Review [1]. Essentially, experimental features concerned with the simulation studies are not reported at all. By 'experimental aspects' we mean clear research goal, hypothesis, experimental design, analysis procedures, and so on. In summary, it seems that the research protocols had not been predefined for these types of studies nor had followed any sort of standard in their organization, indicating a lack of rigour in their performing and reporting.

Therefore, among our findings, it is possible to identify a lack of rigour on reporting simulation-based studies, maybe caused by not performing planning activities or by the absence of compiled guidelines that could support such planning and reporting.

For instance, we identified some published guidelines for empirical studies in SE, such as the one proposed by Kitchenham et al [24]. In their work, authors mentioned the need for specific guidelines for different types of studies. Later, addressing this issue, Höst and Runeson [25] proposed specific guidelines for case studies. However, we could not find guidelines for simulation studies in SE. Therefore, we tried to look for guidelines for simulation studies in different fields such as statistics and medicine research areas.

Ören presents a set of concepts and criteria to assess the acceptability of simulation studies in general [26]. Balci presents guidelines for successful simulation studies [27]. In [28], several experimental designs issues are discussed. In medicine, Burton et al [28] present and discuss a checklist highlighting important issues for designing simulation studies. Essentially, studies should not be different from their reports [25]. Thus, every planned and executed procedure, including the decisions taken during the experimentation process, should be explicit.

Based on the results obtained from the *quasi*-Systematic Review and also from published guidelines in Software Engineering (for general and specific study strategies), Statistics and Medicine, we developed the set of reporting guidelines for SBS described in Section 2 of this document. Section 3 presents the final considerations regarding the use and applicability of the guidelines.

2. Reporting Guidelines

In this section we present a set of guidelines concerning on reporting simulation-based studies in the context of Software Engineering (SE) research. The adopted terminology can be consulted in the Glossary of Terms for Experimental Software Engineering[1].

As a general suggestion, the audience to which the study will be reported on should be considered and the terms should be chosen accordingly. Also, this set of guidelines is organized in chained sections and this organization implicitly suggests a possible organization structure for the report. Finally, email or other contact data should be provided to allow readers to possibly ask authors for further information or details about the study.

[1] http://lens-ese.cos.ufrj.br/wikiese/index.php/Experimental_Software_Engineering_-_Glossary_of_Terms

2.1. Report Identification

At first, a study report should be accessible. In other words, it should be easy to find it in (digital) libraries or through search engines. For that, the report title, abstract and keyword should contain all relevant words regarding the main topic and findings.

Reporting Guideline 1. Proper title and keywords should objectively identify the study report, as well as have a structured abstract summarizing the report contents.

The choice of a proper title has no straightforward rule, but it should address the main topic of the study and also the main research contributions. Keywords generally depend on a glossary of terms used by the publishers.

We suggest the use of structured abstracts, as this eases the identification of the context, problem, goals, used methods, main results, and conclusions. It helps readers to quickly identify whether the study is relevant for their research purposes. An example of structured abstract can be found in IST (Information Software Technology) instructions for authors' page[2].

2.2. From Context to Research Questions

As in any investigation initiative, the research context, problem, and scope are extremely important. It is especially true in SE, where the context of software projects, the human nature of SE activities, and the amount of unknown variables may impact the results of the studies.

Reporting Guideline 2. The context where the research is taking place should be described in full.

Simulation models in SE often come from research initiatives. Both academic and industrial projects are potential environments for simulation studies taking place. In industrial contexts, the description should characterize the organization where the study is being conducted. Information about involved technologies, personal profiles, types of projects performed in the organization, operational procedures, and also non-technical issues (cultural, restrictions imposed by policies, laws, and standards, for instance). Particularly, the (quantitative or qualitative) data collected during the study has to capture some context information. In academic contexts, the background and also the research project goals should be addressed.

Context information is useful when analyzing the simulation results. Such information can clarify an unexpected behaviour or explain why specific behaviour cannot be generalized.

In [12], the context is described in this way:

'My investigation of multiproject dynamics is being conducted within the context of a much broader research effort to study, gain insight into, and make predictions about the dynamics of the software development process. A major part of this effort is devoted to the development of a comprehensive system dynamics model of the process. This was accomplished in two phases. First, a model of a single software project (in isolation) was developed. The model was then extended for the purposes of the current research to model the concurrent execution of two software projects and their dynamic interactions. The model was developed based on field studies in five organizations and supplemented by an extensive database of empirical findings from the literature.'

It is possible to see that the simulation model development is a research initiative immersed in a broader scope. The domain is basically of software project management involving concurrent projects. The model was developed incrementally and based on real-case observations in software organizations, and empirical findings from the technical literature. More details on the context can be obtained from different parts of the text, but the essential one is concentrated in this paragraph.

Höst et al [13] propose a context classification scheme (table 1), based on two orthogonal factors: incentives and the experiences of subjects.

[2] http://www.elsevier.com/journals/information-and-software-technology/0950-5849/guide-for-authors#39001

Table 1. Context Classification Scheme [13]

Incentive	Experience
I1: Isolated artefact	E1: Undergraduate student with less than 3 months recent industrial experience
I2: Artificial project	E2: Graduate student with less than 3 months recent industrial experience
I3: Project with short-term commitment	E3: Academic with less than 3 months recent industrial experience
I4: Project with long-term commitment	E4: Any person with industrial experience, between 3 months and 2 years
	E5: Any person with over 2 years' industrial experience

The Incentive factor is more related to study relevance and environment setting. The Experience factor is strongly related to subject characterization, which is a concern of Section 2.4 of this document.

Petersen and Wohlin [14] presents another set of context information to be reported, where they propose context description based on the six facets related to the object of the study, according to Figure 1.

Figure 1. Context Facets (Adapted from [14])

In SBSs, the object of the study is always related to the simulation model. So, depending on the goal of the study and on model validity, the object of the study may be the simulation model itself or the phenomenon/system/process, which the model abstracts. All the facets in Figure 1 interact with the object of the study in some way. This proposal was made for industrial studies. However, some of these facets can be used to contextualize Simulation-Based Studies.

All these facets are directly related to the practical application of the simulation results. When the simulated solution needs to be implemented, the entire context (environment and pre-requisites) assumed by the simulation model should be guaranteed in the real context. Thus, it is often necessary to change target processes, team training, incorporate new techniques or tools, and apply them to the right kind of systems/applications.

Both proposals for contextual descriptions mentioned above establish discrete variables (such as incentive, experience, processes, people) to describe context information. However, Dybå *et al* (2012) propose the use of a broad perspective approach for the so-called omnibus context. In summary, this proposal describes the context in such a way that the study report allows answering the following type of research question:

'*What* technology is most effective for *whom*, performing *that* specific activity, on *that* kind of system, under *which* set of circumstances?'

For the authors, the object of the study and its context keep a 'mutually reflexive relationship', i.e., with both the object of the study and the context shaping each other in the same intensity. Thus, the definition of the context depends on the situation in which the object investigation is taking place.

Once the context information has been gathered, the problem should then be stated and described as to how it was identified in such context. Problems may arise from a specific critical situation or from repeated situations where the solution has a complex implementation or requires an expensive alternative.

Reporting Guideline 3. Explicitly state the problem that motivates the study, so that research questions can be derived.

One example of problem statement regarding simulation studies can be found in [15] when discussing fault-tolerant systems. Several problems were identified in the context of test and performance evaluation of fault-tolerant systems:

'In order to guarantee that a complex system as a whole will satisfy its fault-tolerance and timeliness requirements, it is necessary to overcome several difficulties in the current state of the art. First, accurate reliability estimations for individual system components (such as networks and workstations) are often not available from their manufacturers. Second, using peak load assumptions for each individual component can lead to overly pessimistic reliability estimations for the system, resulting in excessive complexity and cost. Third, even if each component is well understood in isolation, the overall behavior of a distributed system is the result of subtle interactions between subsystems. It is very difficult to predict how components will perform when combined: scalability problems, unexpected bottlenecks, and exception conditions are often detected only when the system is physically built. This problem is still more acute for fault-tolerant systems, as a random subset of the components can fail at any time. Fourth, the system being designed is frequently not available until the late stages of the project. This creates a circular problem: performance must be taken into account when designing the system, but performance evaluation cannot be done until the system is constructed. Finally, testing and performance evaluation require to observe many aspects of the system's behavior during a series of experiments (e.g. measuring response times to quantify the system's adherence to temporal requirements). Hence, it becomes necessary to instrument the system to observe its evolution, with the resulting perturbation the behavior of the instrumented system will in general differ from uninstrumented runs.'

The underlined parts or the text are the core problem statements. Along with the problem statement, the reason why it happens and the impact it causes is given to clearly present the implications of not solving such problem.

For problem statement, we adopt a template proposal[3] based on the following structure:

Statement 1 (Description of ideal scenario). However (or other adversative conjunction), Statement 2 (The reality of the situation). Thus (or other conclusive conjunction), Statement 3 (The consequences for the involved people).

This structure allows understanding exactly the point where the problem occurs and its possible consequences. This way, the reasons why the simulation study has been proposed can be clearly stated.

Reporting Guideline 4. Clearly state the research goals and scope.

Defining the goals is the first step, after establishing a research question. It needs to be described in a clear way, leaving no doubt about what questions to answer, in the same way it occurs with other study strategies. For instance, it is likely to find, in Software Engineering studies, the definition of the goals using the GQM approach [30]. It seems to be completely useful for defining the goals in simulation studies. Besides, the scope should be explicitly stated, establishing boundaries for the research area, domain, and type of systems or processes under investigation.

One example of goal definition is presented in the study conducted by [17]:

'The goal of this research is to propose a novel integrated modeling framework that will help key stakeholders (in particular, development managers) devise optimal workforce assignments considering both the short-term (productivity) and long-term (robustness of the organization against potential departure of key personnel) needs of the organization in a multi-organizational distributed software development setting considering the employers' positions in their social network.'

Another goal description can be found in [18]:

[3] http://www.personal.psu.edu/cvm115/proposal/formulating_problem_statements.htm

'The main goal of our research was to better understand the XP process and to evaluate its effectiveness. In particular, we aimed to investigate how its key practices influence the evolution of a certain project. To achieve this, we chose the software process simulation technique. We developed a simulation executive to enable simulation of XP software development activities. It is able to vary the usage level of some fundamental XP practices (Pair Programming, Test-first Programming), and to simulate how the modelled project entities evolve as a result.'

Both of these goal definitions are non-structured and it may difficult getting the right point, but it is the way in which goals are described in simulation research papers in the SE technical literature.

The SBS goals should match the capabilities of the simulation model. In other words, the simulation model should be able to support the answers for the research questions through the output data, and its input parameters (variables or constants) should allow the desired scenario configuration.

Reporting Guideline 5. Present the research questions derived from established goals.

Davis et al [16] argue that, without an intriguing question, simulation research relies on 'a fishing expedition, in which the researcher lacks focus and theoretical relevance and risks becoming overwhelmed by computational complexity'. This way, once following the GQM approach in order to drive goals definition, and deriving research questions, the next steps is to define the metrics from which the questions should be answered. The metrics definition allows one to 'ask' the research questions as hypotheses, which should be submitted to statistical tests.

Reporting Guideline 6. Clearly state the null and alternatives hypotheses from research questions.

The study should also investigate one or more hypotheses. It should report, clarifying the null and alternative hypotheses. It is also useful to discuss how such hypotheses were raised, describing the rationale or theory they came from. GQM is just one possible approach to reach hypothesis definition, but other approaches may be used.

2.3. Background and Related Works

Theoretical foundations and background knowledge are essential parts of the study report. Without them, it could be a great barrier for a distant reader or junior researchers.

Reporting Guideline 7. Present only essential background knowledge and also the related works.

On the other hand, presenting all the theoretical foundations may miss the focus on the study results. Essential knowledge should be presented and some important references should be pointed out for detailed understanding. Besides, the same would be applied to related works, presenting just the simulation-based studies closely related to the performed study, i.e., investigating the same or related phenomenon. Any other study can be just referenced. This includes previous related works from one same author.

2.4. Model Description

When describing the simulation model, it is relevant to detail its variables and their relationships, as well as its input parameters and the range of values for each one.

Reporting Guideline 8. Describe the simulation model used in the study through its main variables, constants and the underlying simulation approach.

Model description is useful to supplement the information regarding the experimental design and on how values for input parameters in each simulation run are determined.

Such description should include the underlying simulation approach. It is important to clarify such an approach from the characterization point-of-view. The abstraction and execution mechanisms are immediately understood by presenting the simulation approach. For instance, when describing a system dynamics model, it is possible to infer how simulations are executed; the stocks and flows modelling abstractions; and so, it is expected that the causal relationships and feedback loops would be presented.

The report's reader expects diagrams, equations, and textual descriptions. Diagrams are useful for presenting the whole idea and also the conceptual simulation model. Equations allow the possibility of

replicating the model in other simulation tools. Finally, a text description supplements and clears any doubt about the previous ones.

The model boundaries should also be specified. It is possible to perceive in some reports that simulation models are labelled, for example, as a 'requirements engineering simulation model'. However, such model difficulty encompasses all the activities and variables in requirements engineering. So, unconsidered aspects, assumptions representing simplifications of the real system and limitations, should be included in model description.

2.5. Model Validation

The concern about model validity should be addressed, as SBS validity is highly affected by validity of the simulation model. If the model used cannot be considered valid, invalid results will be obtained regardless of the other possible threats to study validity. In other words, the simulation model itself represents the main threat to study validity.

Reporting Guideline 9. Present all possible evidence regarding the validity of the simulation model (conceptual and execution).

Previous reports or research papers presenting evidence regarding the simulation model validity should be described or cited. In the case where such validation references are absent, verification and validation procedures should be performed to ensure model validity, reporting the results as well as the decisions that guided the validation process.

In a recent *quasi*-Systematic Literature Review [1], we identified nine verification and validation (V&V) procedures (Table 2) applied to simulation models in the context of Software Engineering. Besides, we present the frequency of models (M) and papers (P) where these procedures were applied.

Abdel-Hamid [19] submitted his model to several procedures from Table 2. The basis for developing his Software Project Integrated Model (using the System Dynamics approach) was field interviews with software project managers in five organizations, supplemented by an extensive database of empirical findings from the technical literature.

Table 2. Verification and Validation Procedures for Simulation Models [1]

Procedure	Description	M	P
Comparison against actual results	Compares simulation output results against actual output of the same phenomenon. It is likely used for measuring model accuracy.	**10**	**17**
Comparison against data from literature	Compares the simulation output results against output data (performance) from other studies found in technical literature. It is likely used when there is no complete data in hands.	3	3
Comparison against reference behaviours from the technical literature	Compares the simulation output results against trends or expected results often reported in the technical literature. It is likely used when no comparable data is available.	6	12
Comparison against other models results	Compares the simulation model output results one against the other. Controlled experiments can be used to arrange such comparisons.	7	7
Review with experts	Consists of getting feedback from a system or process experts, to evaluate whether the simulation results are reasonable. This review may be done using any method, including inspections. It is likely used for model validation purposes.	5	5
Interview with experts	Gets feedback from a system or process experts through interviews, to evaluate whether the results are reasonable. It is likely used for model validation purposes.	3	9
Survey with experts	Gets feedback from a system or process experts through surveys, to evaluate whether the results seem reasonable. It is likely used for model validation purposes.	3	3
Testing structure and model behaviour	Submits the simulation model to several tests cases, evaluating its responses and traces. It is likely used for model verification purposes.	4	5
Based on empirical evidence from the technical literature	Collects evidence from the technical literature (experimental studies reports) to develop the simulation model.	7	13

Tests were performed to verify the fit between the rate/level/feedback structure of the model and the essential characteristics of the real system (software projects). The software project managers involved in the study confirmed this fit. The procedures for tests and reviews performed were not described in the paper. Besides, the results were not reported either. So one may ask: 'What kind of test was performed? How many discrepancies were identified by the project managers?'

Another procedure performed was the comparison against reference behaviours. In this case, the behaviour was textually and graphically described and the model representation was presented in System Dynamics diagrams. The textual explanation can be observed in [19]:

'Frequently, the manager of such a programmer would note the 90%-complete estimate, proceed to negotiate an assignment on a new project for the programmer after the following week, and schedule the routine to be passed on to the integration team at that time. This meant that a week later, when the job was not done and was now estimated at 95% complete, the manager had to go back to the project manager who was expecting the programmer and renegotiate the transfer for a week later, and similarly with the manager of the integration activity. After a few more such weekly negotiations, people get bitter and don't trust each other, and the entire project control process begins to disintegrate. Further, the blight often spreads to other projects through the uncertainties of interface integration and the delays due to performers who do not show up on time.'

The translation of such behaviour to the model can be characterized as in Figure 2 below.

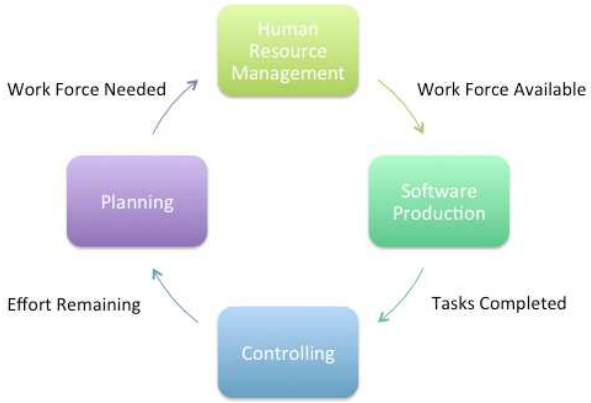

Figure 2.Project Control feedback loop (Adapted from [19])

Also, the simulation results in [19] were plotted in sequence run charts to compare against the expected behaviour. Thus, the results seem to indicate the fit between the reference behaviour and simulation results. Reference behaviours reproduced by the model included a diverse set of behaviour patterns observed both in the organizations studied as well as reported in the literature.

The author also reports extreme condition simulations, i.e., to 'test whether the model behaves reasonably under extreme conditions or extreme policies'. A model that does not behave reasonably under extreme conditions (e.g., zero error density) is suspected as one may not be certain when aspects of extreme conditions may occur in ordinary runs.

Additionally, he performed a case study at NASA. According to him, the DE-A project case study, which was conducted after the model was completely developed, forms an important element in validating model behaviour as NASA was not part of the five organizations studied during model development.

It is important to note that one of these procedures alone could not provide enough validity for this simulation model. However, taking them together can represent a solid group of positive results.

From many simulation models found in Software Engineering, just a few report performance measures. Measures such as bias, accuracy, coverage, and confidence intervals frequently go un-reported. The importance of such measures relies in the possibility of using them as benchmark criteria to compare and choose more accurate simulation models. Also, this will directly impact the risks assigned to SBS conclusions. For instance, interesting outcomes are obtained in a SBS, but the simulation model has a low accuracy or its results are in a very wide confidence interval. How far are these results from reality? This information also brings credibility to the simulation study. Burton et al discuss how to calculate such measures [2].

Lauer et al [20] use the relative error in mean values and confidence intervals to compare different configurations from the perspective of timing problems in the context of an automotive embedded system.

Table 2 shows the opportunity to gather empirical evidence from the technical literature as the last V&V 'procedure'. It is an important step when developing simulation models for experimentation as it does not rely only on expert opinion or ad-hoc observation of the phenomenon under study. Empirical evidence can support the existence of properties in the simulation model, as well as model assumptions. Thus, all the evidence gathered from the technical literature to support the model development or assessment should be cited.

2.6. Subjects

Simulation-based studies may be performed as *in virtuo* or *in silico* categories [3]. In general, SBS makes use of virtual environments. However, it is possible to use individuals or computer programs as subjects.

Reporting Guideline 10. Characterize the subjects involved in the simulation study and report training needs.

When reporting SBS, the study environment should be made explicit. Besides, the characterization of the subjects should be done as it can influence the interpretation of *in virtuo* results. This way, the level of expertise, number of subjects per group (treatment and control, when applicable) and any other relevant characteristic should be included. The description of the subjects' assignment process to the experimental units should be addressed, whether made randomly or not, for example. With computerized subjects, the description of their behaviour model, configuration parameters, and process of assignment should also be informed.

An example of subjects' description can be found in experiment by Pfahl et al [21] on software project learning, as it can be seen in the text:

'The participants of the study were computer science students at the University of Kaiserslautern, Germany, who were enrolled in the advanced software engineering class lasting one semester. While the course was running, subjects were asked if they would be interested in participating in an experiment related to software project management issues that would involve a simulation model. The subjects knew that they would have to participate in a self-learning training session, that they would have to pass a test, and that the test scores would be analyzed to evaluate the training session. Twelve students expressed their interest in participation.

As the German system allows students to take different classes at different times during their studies, information on their personal background with regard to experience in software development and software project management was captured before passing the pre-test.

… Questions about personal characteristics (age, gender), university education (year, major, minor), practical software development experience, software project management literature background, and preferred learning style.'

When the study involves human subjects, it is common to submit them to training sessions in order to see them perform the tasks planned for the study. The training procedure as well as its costs should be reported.

2.7. Simulation Scenarios

When investigating a system or process through SBS, it is common to make use of scenarios [4]. The relevance and adequacy of each chosen scenario is important to be described.

Reporting Guideline 11. Describe the selected simulation scenarios and the procedure used to identify them as relevant.

To choose and report on the most representative scenarios, including those that both check best and worst cases, can help foreseeing behaviour in normal circumstances and in exceptions. The description of the scenarios should be as precise as possible, clarifying all the context information, and characterizing the roles involved in each scenario step. Also, input parameters should reflect the scenario.

In Ambrósio et al [22], the authors use three scenarios (optimistic, baseline, and pessimistic) in two sets of simulations by changing the value of model components related to risk factors in a model concerned

with requirements activities: requirements errors and volatility, and workforce turnover. These scenarios are described as three different model input parameter settings.

2.8. Experimental Design

Balci [5] mentions four techniques for the design of experiments: Response-surface for maximization or minimization of the values for response variables by optimizing the combination of parameters values; Factorial Design for determining the effect of input variables on response variables; Variance reduction techniques to get better statistical accuracy for the same number of simulations; Ranking and selection techniques to compare alternative systems (or system configurations). From these four techniques, only factorial design was found in our review regarding SBS in Software Engineering. It cannot be claimed all SBS in SE apply this design, but these were the only ones reporting their experimental design with enough detail.

Reporting Guideline 12. Experimental design, including independent and dependent variables and how treatments are assigned to each factor should be reported.

Basically, experimental design issues involve the definition of a causal model establishing a relationship between independent and dependent variables, in a cause-effect nature. Such a model should be derived from the hypothesis definition and should reflect part or the whole simulation model. Also, it involves the arrangement of independent (or factors) variables and the definition of the respective treatments for each factor. Here, the importance of describing the model and its variables is clear. Once they are described, the experimental design can be easily understood. As seen in [5], different values and types of system parameters, input variables, and behavioural relationships - as they constitute the statistical design factors - may represent system variants.

Houston et al [6] selected four published deterministic system dynamic models, found in technical literature, to perform an experiment. Their two-level fractional factorial designs (2^{k-p}, where k is the number of factors and p is the power of the fraction) are described in Table 3.

Table 3. Experimental Designs used in Houston et al (adapted from [6])

Model	Number of factors	Design	Number of runs
Abdel-Hamid & Madnick	65	2^{65-57}	256
Madachy	21	2^{21-15}	64
Tvedt	103	2^{103-95}	256
Sycamore	30	2^{30-24}	64

Wakeland et al [23] performed an experiment arranged in 2 factors (re-inspections and inspection effectiveness) X 2 levels (perform or skip re-inspection and the percentage of project effort allocated to inspection activity: 5% is low and 15% is high) factorial design with 10 replications per response, with a total 40 runs. The 2 factors and the 2 levels, for each factor are shown in Table 4.

Table 4. Experimental design from (Adapted from [23])

	Skip Re-inspections	Perform Re-inspections
Effectiveness of Inspections [and re-inspections] is High (15%)	Conventional wisdom says this would probably make the most sense – do it right and do it once.	Quality zealots would probably advocate this scenario – in order to minimize escaped errors.
Effectiveness of Inspections [and re-inspections] is Low (5%)	Time to market zealots might advocate this scenario – do it quickly and forget it.	It is not likely that anyone would advocate doing ineffective inspections twice.

Control and treatment groups should be identified when performing controlled experiments using simulation models as instruments. For instance, validated models under known conditions can be assumed as control and the new model (or new versions) to be evaluated or experimented (under the same conditions) can be assumed as the treatment. Another possibility is to use distinct datasets as factors, with the simulation model remaining constant. This way, different calibrations representing the different simulation scenarios can be compared and should be reported.

The number of simulation runs should be based on the selected simulation scenarios and on the experimental design. Each selected scenario consists of an arrangement of experimental conditions where

possible factors are assigned to one specific treatment. The more simulation scenarios involved in the study, the more simulation runs are needed. For instance, factorial designs usually require one simulation run for each combination of factors and treatments. For a discussion on how to determine the number of simulation runs, based on factorial designs, we suggest the reading of [6] [7].

Reporting Guideline 13. The number of runs together with the rationale to determine it should be reported.

Houston and Wakeland explain their reasoning to determine the number of runs, based on the number of factors and levels the simulation experiment takes into account. Both of them use 2^k factorial designs. This kind of design requires a run for each combination of the factor settings. For instance, if three factors are considered, so $2^3 = 8$ simulation runs.

Additionally, this reasoning is true for simulation experiments using deterministic simulation models. However, when using stochastic models, the use of random variables should be taken into account as a confidence interval should be estimated from the sample size to determine the number of simulation runs. Such calculation can be found in [2].

2.9. Storage of Experimental Trials

Reporting Guideline 14. Describe which and how intermediate measures are stored among simulation trials to be used in the final analysis.

SBS involving multiple trials and runs often need to use the information of each intermediate trial for the final output analysis. Means, standard deviations, and other measures are likely to be applied to summarize the whole simulation run (including all trials) and to determine confidence intervals, for example. The report should contain data on how these measures are stored, if they are stored in a database or external file. Also, how such data is used in the analysis, if they are plotted on charts, used as threshold values to support decision-making, and so on.

2.10. Data Support

When planning SBS it is important to check the availability of supporting data and determine its type: real-system or simulated data [10]. If simulated data has been adopted, some evidence should be presented to guarantee the validation of such data, i.e., the report should answer questions such as 'How far the simulated data is from real-system data?' and show indicators of this gap. Here, statistical tests can be applied to verify how close both samples could be.

Reporting Guideline 15. The data used to support the simulation model development or SBS should be assessed and reported, whenever possible.

Simulation models often require time-sensitive data. Hence, in order to avoid biased observations and an exposure to risk (i.e. undetected seasonal data); the data collection time period should represent both transient and steady state behaviours.

Araújo et al [11] present a system dynamics model for the observation of software evolution that requires time-sensitive and real-system data. For their model, the time when the data is collected is important since it is desirable to observe how the successive maintenance cycles impact software quality. The study presents the observations made over a 2-year large-scale software project executed in the industry. So, real data was treated and analyzed accordingly.

Data collection should be planned to also avoid measurement mistakes, promoting the collection of data as soon as they are made available. After the collection, quality assurance procedures ought to take place in order to verify their quality. If the simulation model can be calibrated, it is important to report whether it was calibrated or not, including the procedure used to accomplish the task and its results.

Another important aspect related to the collected data (or used datasets) relies on the raw data publication. However, it is rarely reported, basically for two reasons: (1) most papers report that it was not possible to present the raw data as it is confidential and (2) since simulation studies usually involve a large amount of data and it may not fit in conference or journal papers.

Even so, the raw data should, when possible, be reported or made available by consulting the authors or publishing it at a downloadable source.

2.11. Simulation Supporting Environment

The simulation environment consists of all the instruments needed to perform the study. It encompasses the simulation model itself, datasets, data analysis tools (including statistical packages), and simulation tools/packages. As the simulation model and datasets have already been previously discussed, here the supporting tools are the focus as an important feature to be reported.

Reporting Guideline 16. Describe the simulation environment, including the supporting tools, associated costs, and decision for using a specific simulation package.

The simulation package should support not only the underlying simulation approach, but also the experimental design and output data analysis. Simulation packages often differ on how they implement the simulation engine mechanism. So it is possible to get different results depending on how the engine is implemented. Moreover, the process used to translate the conceptual simulation model description to the simulation language offered by the package should be reported. Information on how this translation was performed and if any model characteristic could not be implemented due to technological constraints should be given. In stochastic models, the report of random number generators and on how the starting seeds were selected is fundamental.

The choice of a simulation package should depend on the fit of the research question, assumptions, and the theoretical logic of the conceptual model with those of the simulation approach [16]. It is an important decision as the simulation approach may impose a theoretical logic, type of research question, or related assumptions.

Garousi et al [8] justify the choice for Vensim (www.vensim.com) for a Software Process system dynamic model according to the sentence: 'It was decided to use Vensim in this work because of two major reasons: (a) its capability of working with external dynamic link libraries (DLLs) to support organization-specific heuristics, e.g., developer allocation algorithms, and (b) the ability to provide rich analysis features during a simulation, e.g., graphing of variables.'

Raw input data always requires an extra effort to understand its properties (such as data distribution and shape, trends, and descriptive stats) and perform the transformations (such as scale transformations and derived metrics) needed to fit the model parameters and variables. Similarly, the simulation output data needs specific analysis techniques such as statistical tests and accuracy analysis. For both input and output data there is a need for other supporting tools like statistical packages or even other specific ones. These tools form the whole simulation environment that should be mentioned in the report.

Another important perspective is related to the computational infrastructure. The settings used to run the simulations need to be reported so that one can understand the requirements for replicating the study, for example. Processor capacity, operating system, amount of data, and execution time interval are relevant characteristics to estimate schedule and costs for the study.

2.12. Output Analysis

In the context of Software Engineering, the output analysis of simulation-based studies is mostly performed using charts. On the other hand, there are fewer cases where we can find statistical (hypothesis) tests or descriptive stats.

Reporting Guideline 17. Procedures and instruments for output analysis should be reported, as well as the underlying rationale.

The study protocol should contain the procedures and instruments to be used in the analysis of simulation results. Simulation runs often produce large volumes of data, distributed in different output variables. The output data analysis procedure and instruments should be properly chosen, as statistical instruments (such as charts) and methods have many assumptions and restrictions.

Assumptions on the independence of variables and on how data is distributed should be carefully observed to adopt the correct charts, statistical measures, and tests. Simulation experiments use such statistical measures for accuracy, for instance. Mean Magnitude of Relative Error (MMRE), Balanced

Relative Error (BRE), andPred(25) are examples of such measures [31]. Charts often assume the data is organized in a particular way; for example, Sequential Run Charts [32] assume data is chronologically ordered. Specific hypothesis tests assume normally distributed data or homoscedastic distributions. These properties should be assured in order to use such instruments when performing the output analysis. Also, evidence that support how these properties are reached should be reported.

It is also important to note the analysis through different simulation runs (or replications). Simulations from different replications are usually independent from each other, so it is possible to use measures such as mean, standard deviation, and confidence intervals across replications.

2.13. Threats to Validity

SBS reports should, as any other empirical study, discuss the threats to study validity. Here, we concentrate our perspective on threats to validity regarding implications of experimenting with simulation models. All of the common aspects of experimental validity are strongly related to the validity of the simulation model. It should be valid to assure the study does represent actual system scenarios.

Reporting Guideline 18. Always report the threats to study validity and limitations.

According to Davis et al [16], simulation improves *construct* and internal validity, by accurately specifying and measuring constructs (and the relationship among them) and the theoretical logic that is enforced through the discipline of algorithmic representation in software, respectively.

Garousi et al [8] consider that model validity is mainly affected by three factors: (a) proper implementation of cause-effect structures, (b) proper representation of real-world attributes by model parameters, and (c) proper calibration. So, each of these three factors is associated to an aspect of validity: (1) structural validity: related to the building blocks and elements of the corresponding real-world process capture; (2) parametric validity: verification of model parameters suitability for the instrumentation and analysis of the real-world process, and; (3) empirical validity: related to model calibration using data from the corresponding real-world process.

Raffo [9] mentions other model validity aspects: face, output, and scenario validity. Face validity tests the fit between the model structure and the real system essential characteristics, often performed by system experts. It seems to be strongly related to what Garousi et al call ‘structural validity’. Output validity verifies the accuracy of the output data; it involves performance measurements and/or statistical tests. Finally, scenario validity evaluates the meaning of simulation scenarios used in the study, also performed by experts.

External and conclusion validity should be accomplished with the application of adequate statistical tests over the model outputs. However, the validity of the conclusion also relates to sample size, number of simulation runs, model coverage, and the degree of representation of the simulated scenarios for all possible situations.

One important threat to validity is publication bias. It is related to funding or vested interests, which should be mentioned in an explicit way. The threat is the publication of only successful results and also to hide details regarding the solution and the study, considered strategic for the organization.

2.14. Conclusions and Future Works

By the end of the report, the results/findings/ express the main contributions in summary. The conclusions should be drawn upon the findings, establishing a link from the goals, using methods to achieve results that allow making conclusions.

Reporting Guideline 19. Main results/findings should be identified and summarized, as well as the conclusions arising from results.

The final discussion should include implications about the applicability of the solution in real scenarios, e.g., use in practice. How to implement the solution? What is the knowledge required, as well as the capabilities and training needed? Also, the associated risks in adopting the solution should be explicitly stated. The risks are closely related to context description (facets), so it means that changes do occur not only in processes and methods, but also with personnel, IT infrastructure, financial costs, need for consultancy, and so on.

Reporting Guideline 20. Applicability issues should be addressed in the report, considering organizational changes and associated risks.

Finally, the way ahead should be mentioned in the report, pointing out further work and research challenges. It may also include hot topics and possible roadmaps for future research.

3. Final Remarks

The main motivation for this work arose from the opportunity in organizing a set of guidelines that could promote the quality of reported studies, as indicated by the performed *quasi*-Systematic Review. Our expectation is that these guidelines guide authors, researchers interested in simulation results, practitioners, and reviewers, whose information should be presented when reporting simulation-based studies in the context of Software Engineering.

Specifically for authors, the contextual and planning information recommended by the guidelines indirectly motivate them to observing some specific features when planning simulation-based studies in Software Engineering. Researchers and practitioners can be aware of core information concerning the SBS results that may be used in their research work, respectively. Examples of such information are context information (Reporting Guideline 2), threats to validity (Reporting Guideline 18), conclusions (Reporting Guideline 19), and applicability (Reporting Guideline 20). Reviewers, members of conference and in the editorial boards of journals should be able to quickly find the relevant contributions, as well as the evidence confirming the contributions and the possible limitations of the EBS.

Even the topics in the reporting guidelines may seem the same of other disciplines, their content presenting some discussion on how they are and should be presented for Software Engineering studies. Some particularities can be observed since Software Engineering, at least as a science field, is not in a mature stage.

These guidelines were proposed aiming at increasing orientation on the reporting of SBS. It is out of their scope to explain how these studies should be conducted. In other words, these guidelines do not mean to be a process or methodology to perform SBS. Processes for selecting the suitable simulation approach, V&V procedure or analysis instruments are beyond the purpose of the guidelines. The specifics of any SE domain or simulation approach are not covered either, as this work has a general purpose.

As our next steps, we are planning a further evaluation of these guidelines to assess their contents from the perspective of SBS experts in Software Engineering who need to report and get information from simulation-based studies. We are also currently working on an evidence-based process to conduct SBS oriented to produce results according to these guidelines.

4. References

[1] França, B.B.N.; Travassos, G.H. Are We Prepared for Simulation Based Studies in Software Engineering yet? In: Experimental Software Engineering Latin American Workshop, ESELAW, Buenos Aires, Argentina. 2012.

[2] A. Burton, D. G. Altman, P. Royston, R. L. Holder, “The design of simulation studies in medical statistics,” Statistics in Medicine, vol. 25, pp. 4279-4292, 2006.

[3] G. H. Travassos, M. O. Barros, “Contributions of in virtuo and in silico experiments for the future of empirical studies in software engineering,” WSESE03. Fraunhofer IRB Verlag, Rome, 2003.

[4] M. O. Barros, C. M. L. Werner, G. H. Travassos, “Applying system dynamics to scenario based software risk management,” International System Dynamics Conference, Bergen, Norway, 2000.

[5] O. Balci, “Guidelines for successful simulation studies,” Proc. Winter Simulation Conference, pp. 25-32, 1990.

[6] D. X. Houston, S. Ferreira, J. S. Collofello, D. C. Montgomery, G. T. Mackulak, D. L. Shunk, "Behavioral characterization: Finding and using the influential factors in software process simulation models," Journal of Systems and Software, vol. 59, pp. 259-270, 2001.

[7] J. P. C. Kleijnen, "Statistical design and analysis of simulation experiments," Informatie, 17, no. 10, pp. 531-535, Oct. 1975.

[8] V. Garousi, K. Khosrovian, D. Pfahl, "A customizable pattern-based software process simulation model: design, calibration and application," SPIP, vol. 14, pp. 165 – 180, 2009.

[9] D. Raffo, "Software project management using PROMPT: A hybrid metrics, modeling and utility framework," Information and Software Technology, vol. 47, pp. 1009-1017, 2005.

[10] T. I. Ören, "Concepts and criteria to assess acceptability of simulation studies: a frame of reference," Simulation Modeling and Statistical Computing, vol. 24, n. 4, pp. 180-189, April 1981.

[11] M. Araújo, V. Monteiro, G. H. Travassos, "Towards a Model to Support in silico Studies of Software Evolution," In 6th International Symposium on Empirical Software Engineering and Measurement (ESEM), Lund, Sweden.

[12] T. Abdel-Hamid, "A multiproject perspective of single-project dynamics," *Journal of Systems and Software*, vol. 22, pp. 151-165, 1993.

[13] M. Höst, C. Wohlin and T. Thelin, "Experimental Context Classification: Incentives and Experience of Subjects", IEEE Conference Proceedings International Conference on Software Engineering, pp. 470-478, St. Louis, USA, 2005.

[14] K. Petersen and C. Wohlin, "Context in Industrial Software Engineering Research", Proceedings 3rd International Symposium on Empirical Software Engineering and Measurement, pp. 401-404, Orlando, USA, October 2009.

[15] F. Alvarez, Guillermo A., Cristian, "Applying simulation to the design and performance evaluation of fault-tolerant systems," in Proc. of the IEEE Symposium on Reliable Distributed Systems, Durham, NC, USA, 1997, pp. 35–42.

[16] J. P. Davis, K. M. Eisenhardt, C. B. Bingham, "Developing Theory Through Simulation Methods," Academy of Management Review, vol. 32, no. 2, 2007, pp 480-499.

[17] N. Celik, H. Xi, D. Xu, Y. Son, "Simulation-based workforce assignment considering position in a social network," in *Proc. of Winter Simulation Conference*, Baltimore, MD, United states, 2010, pp. 3228 – 3240.

[18] M. Melis, I. Turnu, A. Cau, G. Concas, "Evaluating the impact of test-first programming and pair programming through software process simulation," Software Process Improvement and Practice, vol. 11, pp. 345 – 360, 2006.

[19] T. Abdel-Hamid, "Understanding the "90% syndrome" in software project management: A simulation-based case study," Journal of Systems and Software, vol. 8, pp. 319-33, 1988.

[20] C. Lauer, R. German, J. Pollmer, "Discrete event simulation and analysis of timing problems in automotive embedded systems," in *IEEE International Systems Conference Proceedings*, SysCon 2010, San Diego, CA, United states, 2010, pp. 18 – 22.

[21] D. Pfahl, M. Klemm, G. Ruhe, "A CBT module with integrated simulation component for software project management education and training," *Journal of Systems and Software*, vol. 59, no. 3, pp. 283 – 298, 2001.

[22] B. G. Ambrosio, J. L. Braga, and M. A. Resende-Filho, "Modeling and scenario simulation for decision support in management of requirements activities in software projects," *Journal of Software Maintenance and Evolution*, vol. 23, no. 1, pp. 35 – 50, 2011.

[23] W. W. Wakeland, R. H. Martin, D. Raffo, "Using Design of Experiments, sensitivity analysis, and hybrid simulation to evaluate changes to a software development process: A case study," *Software Process Improvement and Practice*, vol. 9, pp. 107–119, 2004.

[24] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. El Emam, J. Rosenberg, "Preliminary Guidelines for Empirical Research in Software Engineering," *IEEE Transactions on Software Engineering*, v. 28, n. 8, Aug, 2002.

[25] M. Höst, P. Runeson, "Guidelines for conducting and reporting case study research in software engineering," *Empir Software Eng*, vol. 14, pp. 131–164, 2009.

[26] T. I. Ören, “Concepts and criteria to assess acceptability of simulation studies: a frame of reference,” *Simulation Modeling and Statistical Computing*, vol. 24, n. 4, pp. 180-189, April 1981.

[27] O. Balci, “Guidelines for successful simulation studies,” Proc. Winter Simulation Conference, pp. 25-32, 1990.

[28] J. P. C. Kleijnen, “Statistical design and analysis of simulation experiments,” *Informatie*, 17, no. 10, pp. 531-535, Oct. 1975.

[29] A. Burton, D. G. Altman, P. Royston, R. L. Holder, “The design of simulation studies in medical statistics,” *Statistics in Medicine*, vol. 25, pp. 4279-4292, 2006.

[30] V. R. Basili. “Software Modeling and Measurement: The Goal/Question/Metric Paradigm”. Technical Report. University of Maryland at College Park, College Park, MD, USA, 1992.

[31] T. Foss, E. Stensrud, B. Kitchenham, I. Myrtveit. “A Simulation Study of the Model Evaluation Criterion MMRE”. *IEEE Trans. Softw. Eng*. v. 29, i. 11, pp. 985-995, November , 2003.

[32] Florac, W.A., Carleton, A.D. “Measuring the Software Process”, Addison-Wesley, 1999.

[33] Dybå, T., Sjøberg, D.I.K., Cruzes, D.S. “What Works for Whom, Where, When, and Why? On the Role of Context in Empirical Software Engineering,” In: ESEM’12. Sep 19-20, Lund, Sweden, 2012.