TECHNICAL REPORT

RT – ES 747 / 14

Reporting Guidelines for Simulation-Based Studies in Software Engineering

(evolved)

This Technical Report intends to update the RT-ES 746/13

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

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

PESC
Programa de Engenharia de Sistemas e Computação

Systems Engineering and Computer Science Department

COPPE / UFRJ

Rio de Janeiro, Agosto 2014

Summary

Abstract

BACKGROUND: In some scientific fields, such as automobile, drugs discovery or engineering, 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 in the context of SE. METHOD: To carry out a literature review on SE guidelines and simulation guidelines in other research areas. Besides that, to merge 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 planning 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 [2]. In their work, authors mentioned the need for specific guidelines for different types of studies. Later, addressing this issue, Höst and Runeson [3] 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 [4]. Balci presents guidelines for successful simulation studies [5]. In [6], several experimental design issues are discussed. In medicine, Burton et al [6] present and discuss a checklist highlighting important issues for designing simulation studies. Essentially, studies should not be different from their reports [3]. 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 some 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 to 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.

It is also important to highlight that each guideline should be taken not only by the recommendation statement, but also considering the associated discussion and examples. Both discussion and examples often try to bring the perspective of simulation studies and also to Software Engineering research area.

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.

SG1. Proper title and keywords should objectively identify the simulation 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. The term “Computer Simulation” can be identified in many libraries as a general term.

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 and Software Technology) instructions for authors’ page[2].

2.2. From Context to Research Questions

As in any research initiative, the context, problem, goals and scope are extremely important, even when talking about simulation studies.

Simulation-Based Studies may be performed both *in virtuo* and *in silico* environments [7]. *In virtuo* experiments stand for studies where human subjects interact with a computerized environment, while *in silico* experiments stand for studies where both subjects and the environment are represented by computerized (simulation) models. Both alternatives are under the scope of the proposed guidelines.

This kind of study strongly depends on the collected data that supports the simulation model development and calibration. 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.

SG2. The context where the simulation study 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 phenomenon has been observed and data has been collected.

[1]http://lens-ese.cos.ufrj.br/wikiese/index.php/Experimental_Software_Engineering_-_Glossary_of_Terms
[2]http://www.elsevier.com/journals/information-and-software-technology/0950-5849/guide-for-authors#39001

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) are relevant for correctly interpretation of results. Such contextual information can clarify an unexpected behaviour or explain why specific behaviour cannot be generalized. In academic contexts, the background and also the research project goals should be addressed.

In [8], 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 [9] propose a context classification scheme (Table 1), based on two orthogonal factors: incentives and the experiences of subjects. It is particularly applicable for *in virtuo* experiments, where human subjects are present.

Table 1. Context Classification Scheme (adapted from [9])

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 2.6 of this document.

Petersen and Wohlin [10] 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.

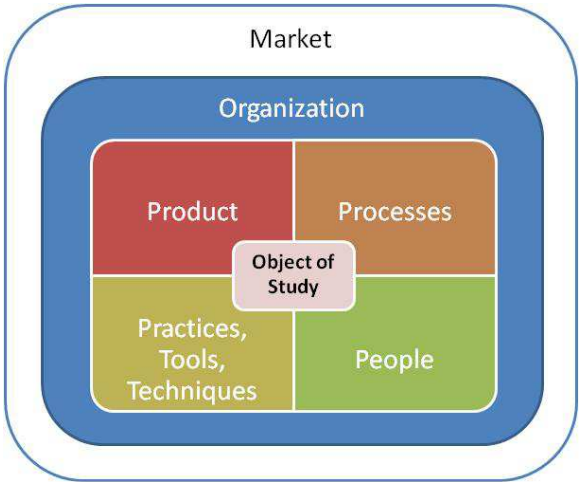

Figure 1. Context Facets (Adapted from [10])

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 SBS.

All these facets are directly related to the practical application of the simulation results. When the solution observed in the simulation study needs to be implemented on real contexts, 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* [11] 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 ecosystem in which the object under investigation takes 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.

SG3. Explicitly state the problem that motivates the simulation study, so that research questions can be derived.

One example of problem statement regarding simulation studies can be found in [12] 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 required 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.

SG4. Clearly state the simulation study goals and scope.

Defining the goals is the first step, after establishing the problem. It needs to be described in a clear way, leaving no doubt about what is to be achieved, 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 [13]. 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 [14]:

"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 [15]:

"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 be difficult getting the right point, but it is the way in which goals are described in simulation research papers in the SE technical literature. As an example, we present a goal from Araújo et al [16], rewritten in the GQM goal template.

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

Table 2. Goal definition using GQM template (adapted from [16])

Purpose	
Analyze	*E-type* Systems
For the purpose of	Characterization
Perspective	
With respect to	Software quality decay
from the point of view of	Researcher
Environment	
In the following context	Simulated environment (*in silico*): Software Project running an iterative and incremental lifecycle, in a CMMi Level 3-like maturity level, including verification, validation and testing techniques. The product under development consists in a large-scale Web Information System for business process control (in financial and administrative aspects). Thirteen releases were taken as a timeframe. The team is geographically distributed and composed by 12 developers, using Java and JSF as the development platform. Also, using CASE tools like: version control repositories, *bug tracking* and effort spreadsheets.

Therefore, the common goals for SBS shall include: developing a basic understanding (characterization) of a particular simulation model of phenomenon, finding robust or optimum decisions, and comparing the merits of various decisions.

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.

SG5. Present the research questions derived from established goals.

According to Davis *et al* [17], without an intriguing research question, the 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.

SG6. Clearly state the null and alternatives hypotheses from research questions.

The study should also investigate one or more hypotheses. Considering the controlled environment, there is always (at least) a hidden hypothesis. It should describe, 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. From the planning perspective, the SBS goals need to match the simulation model capabilities. It means the simulation model shall support the answers for the research questions through the output data, and its input parameters should allow the description of the desired scenario configuration.

2.3. Simulation Feasibility

It is important to assess the feasibility of simulation as a candidate approach to solve or investigate the problem. As far as we are aware, Balci [5] is the only resource available in the technical literature supporting this kind of analysis. Balci suggests the use of some questions as indicators of simulation feasibility. These questions are mainly driven by context variables such as cost, time, benefits and the relationships among them, what naturally limit the observation field. To overcome this limitation we have added questions to his proposal.

SG7. Present the justifications for considering simulation studies as the ideal or feasible strategy.

The goals of the simulation study should be more than getting a value for an output variable. This is the goal for analytical or regression models. Simulation outputs also comprehend a rationale, an explanation or a chain of changes in the system that results in the output values, often represented by high-order effects. Thus, simulation studies for SE should explain how the phenomenon (events and

variables) occurs and what changes on processes, products or people may give a suitable solution. In this sense, we recommend additional questions to support the decision making about deciding to perform simulation studies. Therefore, to focus on more technical constraints regarding simulation model development and experimentation is necessary. The system or phenomenon under investigation should be observable, in some sense. So what are the available instruments and procedures for data collection? Are the occurrence risks (including loss of money or time, reach an irreversible state of the system, safety) of the real phenomenon high? Also, data should be available in order to accomplish statistical issues and calibration of variables and equations involved in common approaches such as SD and DES.

2.4. 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.

SG8. 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.

2.5. Simulation Model and Validation

The guidelines focus is on the reporting of simulation experiments. Model development issues are out of scope, except those aspects in the frontier between model development and use. For the purpose of planning and reporting such experiments, it is important to know the model in detail. No matter if the model has been developed or not by the experimenter. It is part of the required planning knowledge to understand the underlying simulation approach, the conceptual model, including its variables, parameters and associated metrics, as well as the underlying assumptions and calibration procedures. The lack of knowledge about any of these aspects may impose different types of threats to validity.

SG9. Have a detailed description of both conceptual and executable simulation model, as well as its variables, equations, input parameters 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 identifying the simulation approach. For instance, when describing a system dynamics model, it is possible to infer how simulations are executed and the stocks and flows modelling abstractions. Also, it is expected that the causal relationships and feedback loops would be presented as well.

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 rarely encompasses the whole requirements engineering process, including all possible activities and variables. So, unconsidered aspects, assumptions and limitations representing simplifications of the real system should also be included.

The concern about model validity should be addressed, as SBS validity is highly affected by the validity of the simulation model. It is reflection of the nature of computer-based controlled environment, where the phenomenon under investigation is observed essentially through the simulation model execution. This way, the only possible changes are on the input data or the simulation model. Consequently, the validity aspects concentrate on both the simulation model validity and data validity. Thus, if the model used cannot be considered valid, invalid results will be obtained regardless the mitigation actions applied to deal with other possible validity threats. In other words, the simulation model itself represents the main threat to study validity.

SG10. Gather as much evidence as possible regarding the simulation model (conceptual and execution) validity.

Evidence regarding model validity means the experimenter should be aware about the initiatives (previous reports and research papers) of submitting the simulation model to V&V procedures, and understanding their results. In the case where such validation references are absent, these procedures should be performed to ensure model validity, exposing the results as well as the decisions that guided the validation process.

Such procedures have been extensively discussed in the technical literature about computer simulation. Besides, we identified nine V&V procedures applied to simulation models in the context of Software Engineering and merged this list with the one presented by Sargent [18], which are twelve V&V procedures often performed for discrete-event simulation models in several domains, excluding three useful instruments to perform V&V activities, rather than procedures or techniques. This way, the merge from the remaining thirteen with the procedures identified in the systematic literature review are presented in Table 3. In [19] the reader can find discussions on how this set of procedures can reduce of some threats to validity.

As an example of application of such V&V procedures, Abdel-Hamid [8] submitted his model to several of them. The basis for developing his Software Project Integrated Model (using the SD approach) was field interviews with software project managers in five organizations, supplemented by an extensive database of empirical findings from the technical literature. Additionally, tests were performed to verify the fit between the rate/level/feedback structure of the model and the essential characteristics of the real software projects dynamics. The project managers involved in the study confirmed this fit. However, the procedures for tests and reviews performed were not described in the report. Besides, the results were not reported either. So, one may ask among other questions, "*What kinds of test were performed? How many discrepancies were identified by the project managers?*"

Another procedure performed was the comparison against reference behaviors. In this case, the behavior was textually and graphically described and the model representation was presented in System Dynamics diagrams. The reference behavior in this case is the 90% syndrome, where developers use to miscalculate the required effort for a task, and always underestimate it.

Also, the simulation results in [8] were plotted in sequence run charts to compare against the expected behavior. Thus, the results seem to indicate the fit between the reference behavior and simulation results. Reference behaviors reproduced by the model included a diverse set of behavior 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" [8].

Additionally, the author conducted 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 behavior as NASA was not part of the five organizations studied during model development. Also, as also pointed out by the author, none of these procedures alone may provide enough

validity for this simulation model. However, taking them together can represent a solid group of positive results [8].

Table 3. Verification and Validation Procedures for Simulation Models

Procedure	Description
Face Validity	Consists of getting feedback from individuals knowledgeable about the phenomenon of interest through reviews, interviews, or surveys, to evaluate whether the (conceptual) simulation model and its results (input-output relationships) are reasonable.
Comparison to Reference Behaviors	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.
Comparison to Other Models	Compares the results (outputs) of the simulation model being validated to results of other valid (simulation or analytic) model. Controlled experiments can be used to arrange such comparisons.
Event Validity	Compares the "events" of occurrences of the simulation model to those of the real phenomenon to determine if they are similar. This technique is applicable for event-driven models.
Historical Data Validation	If historical data exist, part of the data is used to build the model and the remaining data are used to compare the model behavior and the actual phenomenon. Such testing is conducted by driving the simulation model with either sample from distributions or traces, and it is likely used for measuring model accuracy.
Rationalism	Uses logic deductions from model assumptions to develop the correct (valid) model, by assuming that everyone knows whether the clearly stated underlying assumptions are true.
Predictive Validation	Uses the model to forecast the phenomenon's behavior, and then compares the phenomenon's behavior to the model's forecast to determine if they are the same. The phenomenon data may come from the real phenomenon observation or be obtained by conducting experiments, e.g., field tests for provoking its occurrence. Also, data from the technical literature may be used, when there is no complete data in hands. It is likely used for measuring model accuracy.
Internal Validity	Several runs of a stochastic model are made to determine the amount of (internal) stochastic variability. A large amount of variability (lack of consistency) may cause the model's results to be questionable, even if typical of the problem under investigation.
Sensitivity Analysis	Consists of changing the values of the input and internal parameters of a model to determine the effect upon the model's output. The same relationships should occur in the model as in the real phenomenon. This technique can be used qualitatively— trends only — and quantitatively—both directions and (precise) magnitudes of outputs.
Testing model structure and behavior	Submits the simulation model to tests cases, evaluating its responses and traces. Both model structure and outputs should be reasonable for any combination of values of model inputs, including extreme and unlikely ones. Besides, the degeneracy of the model's behavior can be tested by appropriate selection of values of parameters.
Based on empirical evidence	Collects evidence from the technical literature (experimental studies reports) to develop the model's causal relationships (mechanisms).
Turing Tests	Individuals knowledgeable about the phenomenon are asked if they can distinguish between real and model outputs.

From many simulation studies 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 [6].

As an example of performance measures, 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 3 shows also 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 such evidence 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* experiments [7], making use of virtual environments. *In virtuo* experiments stand for studies where human subjects interact with a computerized environment, while *in silico* experiments stand for studies where both subjects and the environment are represented by simulation models. Both alternatives are under the scope of the proposed reporting guidelines.

SG11. Characterize the subjects involved in the simulation study as well as their training needs.

The study environment should be made explicit when planning and reporting SBS. Besides, the characterization of human 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 in the study plan and considered in the subjects' assignment process to the experimental units whether made randomly or not, for example. Additionally, the training sessions and its costs should be planned as well. With computerized subjects, their behavior model, configuration parameters, and process of assignment should also be considered when preparing the experimental design, in case that such behavior can be clearly identified in the simulation model. Also, it is possible that subjects' behavior may be implicitly embedded in the simulation model when dealing with *in silico* environments. An example of subjects' description can be found in the experiment by Pfahl et al [21] on software project learning, involving twelve computer science graduate students, who were enrolled in the advanced software engineering class lasting one semester. Besides, they captured information about personal characteristics, education, background regarding experience in software development and project management, 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 performing the tasks planned for the study. The training procedure as well as its costs should be reported.

2.7. Experimental Design

Basically, experimental design issues involve the definition of a causal model establishing a relationship between independent (or factors) and dependent variables, in a cause-effect nature. During the experiment, the design factors may be held constant or allowed to vary. So, interest factors may be: controllable, which are possible to measure and vary; uncontrollable, possible just to measure; and noise factors, the ones we cannot measure and they naturally vary.

The causal model should be derived from the research questions and should reflect part or the whole simulation model. Also, it involves the arrangement of factors and the definition of the respective treatments (or levels) for each factor. According to Montgomery [22], levels correspond to the range of interest over which the factors will be varied. This way, the experimenter should have practical experience and theoretical understanding on the domain. For characterization studies, it is recommended to keep a low number of levels per factor, but covering a high region of interest.

Here, the importance of describing the model and its variables is clear. Once they are described, the comprehension of the experimental design can be made easier. 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.

The experimental design is often represented by a **design matrix** that includes the factors and treatments for each factor. In this matrix, every row is called a design point or a scenario, which is a combination of different levels for each factor [23].

SG12. Experimental design (design matrix), including independent and dependent variables and how levels are assigned to each factor should be reported.

Also, it is important to identify control and treatment groups 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.

Clearly, the use of scenarios in simulation experiments can be viewed as a consequence of selecting a proper experimental design. However, it can also be a cause of it, since it is common to make use of scenarios [24] even when an *ad-hoc* experimental design is adopted. In this case, the experimenter plans the scenarios of interest and then derivate the design. By adopting the last strategy, the relevance and adequacy of each chosen scenario is important to be described, and should also be tied to the study goals. Although both strategies are possible, choosing a known experimental design at first is recommended to avoid biased designs, especially for non-experienced experimenters.

SG 13. Describe the selected simulation scenarios and the criteria used to identify them as relevant.

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

In Ambrósio et al [25], 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 parameters settings.

Houston et al [26] selected four published deterministic system dynamic models, from the technical literature, to perform a characterization experiment. Their two-level fractional factorial designs (2^{k-p}, where k is the number of factors and p is called power of the fraction, in which 2^{k-p} is greater than k) are described in Table 4.

Table 4. Experimental Designs used in Houston et al (adapted from [26])

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 [27] 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 outcome, with a total 40 runs. The 2 factors and the 2 levels, for each factor are shown in Table 5.

Table 5. Experimental design from (Adapted from [27])

	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.

The number of simulation runs should be based on the selected simulation scenarios and on the simulation model deterministic or stochastic nature. Each selected scenario consists of an arrangement of experimental conditions where possible factors are assigned to one specific level. 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. So, if three factors and two treatments are considered, we have a design 2^3 with 8 simulation runs required. A detailed discussion on how to determine the number of simulation runs, based on factorial designs, can be found on [26] and [27].

SG14. The number of runs together with the rationale to determine it should be reported.

Houston [26] and Wakeland [27] explain their reasoning to determine the number of runs, based on the number of factors and levels the simulation experiment takes into account. Both studies use 2^k factorial designs. Additionally, this reasoning is true for simulation experiments using both deterministic and stochastic simulation models. However, when using stochastic models, the use of random variables should also be taken into account as a confidence interval should be estimated from the sample size to determine the number of simulation runs (or replications). Such calculation can be found in [6]. **Replication** is achieved by using different pseudo-random numbers (PRNs) to simulate the same scenario. In this case, the output is a time series, which has auto-correlated observations [23].

2.8. Intermediate Experimental Trials

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 instance. Both the plan and report should contain information on how these measures are stored, if they are stored in a database or external file. Also, it should be considered how such data will be used in the analysis, if plotted on charts, used as threshold values to support decision-making, and so on.

SG15. Describe which and how intermediate measures are stored among simulation trials to be used in the final analysis.

Specific or customized simulation environments should be concerned with these capabilities, while it is already supported by commercial tools.

2.9. Supporting Data

When planning SBS it is important to check the availability of supporting data. Simulation models need to be calibrated, requiring data for the generation of equations and parameters, and to determine the random variables distribution. Therefore, it is important to determine the type of data: real or synthetic data [4]. If synthetic data has been used, some evidence should be presented to guarantee data's validity, 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 real and synthetic samples could be.

SG16. Assess, whenever possible, the data used to support the simulation model development or SBS.

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 consistence and accuracy, avoiding including outliers or incomplete data. If the simulation model need to be calibrated, it is important to report whether it was calibrated or not, including the procedure used to accomplish the task and its results.

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. For instance, Araújo et al [16] 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.

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.

2.10. 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 considered.

SG17. 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 the engine's implementation. Moreover, the process used to translate the conceptual simulation model description to the simulation language offered by the package should be considered. Information should also be provided on how such translation was performed and if any model characteristic could not be implemented due to technological constraints. In stochastic models, the use 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 questions, assumptions, and the theoretical logic of the conceptual model with those of the simulation approach [26]. It is an important decision as the simulation approach may impose a theoretical logic, type of research questions, or related assumptions.

Garousi et al [28] 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 taken into consideration.

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.

2.11. Output Analysis

In the context of Software Engineering, the output analysis of simulation-based studies is mostly performed using charts [1]. On the other hand, there are fewer cases where we can find statistical (hypothesis) tests or descriptive stats.

SG18. Procedures and instruments for output analysis should be reported, as well as the underlying rationale.

The simulation 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 and Balanced Relative Error are examples of such measures [29]. Charts often assume the data is organized in a particular way; for example, Sequential Run Charts [30] 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 given.

2.12. Threats to Validity

SBS protocols need, as any other empirical study, to mitigate and discuss possible threats to the study validity. Common types of experimental validities are closely related to the simulation model validity [19]. So, such model should be valid to assure the study can represent actual phenomena. The SE community has discussed threats to validity, and most of the reported threats concerned with *in vitro* or *in vivo* experimentation have already been described in [31]. Most of them have to be considered when planning simulation studies, especially considering *in vitro* experiments, in which the human nature may impose risks to the study. Still, new situations emerge for *in silico* experiments. Either known threats appear in a different outlook, or specific threats of such environment affect the results validity. Here, we concentrate our perspective on these new situations

SG19. Always report the threats to study validity, limitations and non-verified assumptions.

According to Davis *et al* [17], 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 [28] and Raffo [32] mention model validity in a similar way. They take several perspectives into account, such as: model structure, supporting data, input parameters and scenarios, and simulation output. We understand that these aspects are extremely relevant, but are not the only ones, since the study validity goes beyond the simulation model [19]. It is also important to consider the simulation experiment design.

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

2.13. 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.

SG 20. 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 are the required knowledge, 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.

SG21. 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.

SG22. Point out future research directions and challenges after current results.

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 (SG2), threats to validity (SG19), conclusions (SG20), and applicability (SG21). 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 SBS.

Although the topics in the reporting guidelines may seem the same of other disciplines, their content offers 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.

The proposed guidelines aim 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] de França, B. B. N., and Travassos, G. H. 2013. Are We Prepared for Simulation Based Studies in Software Engineering Yet? CLEI electronic journal, 16, 1 (Apr.), paper 8. Available at: http://www.clei.cl/cleiej/papers/v16i1p8.pdf

[2] Kitchenham, B., Pfleeger, S. L., Hoaglin, D. C., El Emam, K., Rosenberg, J. 2002. Preliminary Guidelines for Empirical Research in Software Engineering. IEEE Trans. Soft. Eng., 28, 721-734.

[3] Runeson, P., Höst, M. 2009. Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering, 14, 131–164.

[4] Ören, T. I. 1981. Concepts and criteria to assess acceptability of simulation studies: a frame of reference. Simulation Modeling and Statistical Computing, 24, 4 (Apr.), 180-189.

[5] Balci, O. 1990. Guidelines for successful simulation studies. In Proc. Winter Simulation Conference (Dec. 9–12) 25-32.

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

[7] G. H. Travassos, M. O. Barros, “Contributions of In Virtuo and In Silico Experiments for the Future of Empirical Studies in Software Engineering,” in WSESE03, Fraunhofer IRB Verlag, Rome, 2003.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] M. A. Araújo, V. Monteiro, G. H. Travassos, “Towards a Model to Support in silico Studies regarding Software Evolution,” in ESEM 2012, sep. 2012.

[17] 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.

[18] Sargent, R. G. 1999. Validation and Verification of Simulation Models. In Winter Simulation Conference.

[19] B. B. N. de França, G. H. Travassos. Simulation Based Studies in Software Engineering: A Matter of Validity. In CIbSE/ESELAW. April, 2014. Pucón, Chile.

[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] Montgomery, D.C. Design and Analysis of Experiments. 5th edition. John Wiley & Sons. 2000.

[23] Jack P. C. Kleijnen, Susan M. Sanchez, Thomas W. Lucas, Thomas M. Cioppa, (2005) State-of-the-Art Review: A User's Guide to the Brave New World of Designing Simulation Experiments. INFORMS Journal on Computing 17(3):263-289. http://dx.doi.org/10.1287/ijoc.1050.0136.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] Florac, W.A., Carleton, A.D. "Measuring the Software Process", Addison-Wesley, 1999.

[31] Wohlin, C., Runeson, P., Host, M., Ohlsson, C., Regnell, B., & Wesslén, A. (2000). Experimentation in software engineering: an introduction.

[32] D. Raffo. Software project management using PROMPT: A hybrid metrics, modeling and utility framework. IST, 47, 1009-1017. 2005.