The Role of Simulation-Based Studies in Software Engineering Research

Breno Bernard Nicolau de França and Nauman Bin Ali

Abstract Several decades ago, inspired by other knowledge areas, simulation was introduced as a research method to Software Engineering. Motivated by potential benefits achieved in other areas, the software engineering community has used simulation-based studies for planning, controlling, and improving software development. However, unclear expectations from simulation-based studies, a lack of methodological support, as well as dispersed knowledge to support model building and calibration have hindered widespread adoption of simulation-based investigations. In this chapter, we delineate the role of simulation in software engineering research and compile processes and guidelines into a comprehensive life cycle. This chapter aims to guide software engineering researchers to conduct effective simulation-based studies in real-world settings.

1 Introduction

Computer simulation is used as a research tool in several areas, such as Medicine (Burton et al. 2006), Engineering (Babuska and Oden 2004), Social Sciences (Eck and Liu 2008), and others (Law and Kelton 2000). In Software Engineering (SE), the community presented several initiatives on simulating different kinds of phenomena, ranging from software products, processes to team behavior.

Among these different areas, the use of the term *simulation* varies substantially. To frame our perspective on simulation for this chapter, we have adopted the following definition from Banks (1999): "Simulation is the *imitation* of the operation of a real-world process or system *over time*. Simulation involves the generation of

B. B. N. de França (✉)
Universidade Estadual de Campinas, Campinas, SP, Brazil
e-mail: breno@ic.unicamp.br

N. B. Ali
Blekinge Institute of Technology, Karlskrona, Sweden
e-mail: nauman.ali@bth.se

M. Felderer, G. H. Travassos (eds.), *Contemporary Empirical Methods in Software Engineering*, https://doi.org/10.1007/978-3-030-32489-6_10

an *artificial* history of the system, and the observation of that artificial history to draw *inferences* concerning the operating characteristics of the real system that is represented."

In this chapter, empirical investigations of a system of interest are referred to as simulation-based studies (SBS) if they use simulation to numerically evaluate a mathematical model that imitates the real-world behavior of the system.

As a supporting tool for research in SE, SBS are not meant to replace other types of empirical investigations such as controlled experiments or case studies (we refer to chapter "Guidelines for Conducting Software Engineering Research" for a more comprehensive view on research methods). Rather, it is useful to support knowledge acquisition and decision-making during the research process. Simulation-based studies also require previous empirical knowledge from in vivo[1] or in vitro[2] studies for modeling SE phenomena or behavior (Travassos and Barros 2003). Such modeling fosters large-scale observations, using a controlled and computational environment (in virtuo[3] and in silico[4]), to understand better the phenomenon and possibly explain it through simulation traces and diverse scenarios that could be difficult to observe in an in vivo or in vitro environment. Therefore, simulations are recommended for studies involving a combination of many factors (with possible interactions) and alternatives, as well as long-term observations.

Besides modeling, SBS also require data for calibration, validation, and experimentation. Usually, such data come from observations and, consequently, are limited to their original context. When insufficient data and theories are available, simulation can still be used. In such cases, the role of simulation is limited to a tool that models our assumptions and approximations. The likely outcome of our proposed solutions is then judged objectively under these assumptions. However, we should be aware of the reliability of the results and take extreme care not to overinterpret the results.

Building on lessons learned from the existing literature on simulation in SE, this chapter proceeds in Sect. 2 with a discussion of the claimed benefits to motivate the adoption of simulation in SE. Then, in Sect. 3, we present what we understand are the actual benefits and describe the role of SBS in SE research. In Sect. 4, we motivate the need for methodological support for conducting SBS. Later, in Sect. 5, we present a comprehensive and consolidated life cycle for simulation-based studies in SE. Section 6 presents two practical examples of SBS conducted based on several of the recommendations presented in this chapter. Furthermore, additional readings are suggested in Sect. 7. Finally, we conclude the chapter in Sect. 8.

[1] In vivo studies are the ones occurring in real-life environments.

[2] In vitro studies are the ones occurring in controlled environments with human participants.

[3] In virtuo studies are the ones occurring in computational environments driven by human participants.

[4] In silico studies are the ones occurring completely in computational environments with no human intervention.

2 Motivation for Simulation-Based Studies

Software development is a dynamic activity that involves several people, working with various tools and technologies, guided by policies and processes to develop solutions that fulfill user requirements. The interaction and interdependence between human, technological, and organizational factors make it difficult to confidently assess the potential impact of a proposed change in software development. Furthermore, introducing a change in the development practice is a time and resource-intensive undertaking. Therefore, we want to have high confidence in the likely impact of a change before changing the actual process.

Figure 1 depicts several ways of studying the impact of introducing a change in a system. Broadly, we can (1) manipulate the actual system and investigate the effect of an intervention or (2) study the effects on a model of the system in controlled settings. The main trade-off between the two choices is that of realism vs. control in conducting the investigation. An additional concern is that of the cost of conducting the investigation and the risk if the introduced change does not produce the expected results or has unforeseen consequences.

Due to the cost of manipulating the actual system and the lack of control in real-world settings, we tend to rely on developing and manipulating models of the system. It is only after gaining more confidence in solution proposals that we begin moving towards changing the actual practice. Even in that case, the investigations in real-world settings (through case study or action research) often try to reduce the cost of such studies by incorporating change on a smaller scale. For example, investigating by introducing change in one of the teams or projects and observing the effects before adopting a change throughout the organization.

Simulation is proposed as an inexpensive (Wu and Yan 2009; Melis et al. 2006; Kellner et al. 1999; Madachy 2002) and proactive means to assess what will happen before actually committing resources for a change (Kellner et al. 1999). McCall et al. (1979) made the first suggestion for its use in SE in 1979. It has since been used to study various aspects of software development, e.g., effort estimation, project planning, risk assessment, and training. The simulation models developed over the

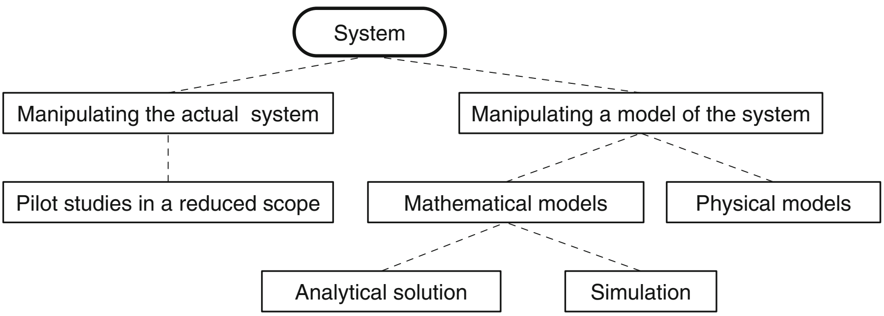

Fig. 1 Ways of studying a system (adapted from Law 2007)

years have varied in scope (from parts of the life cycle to long-term organizational evolution), purpose (including planning and training), and approach (e.g., system dynamics or discrete-event simulation) (Zhang et al. 2010).

3 Limitations of Simulation-Based Studies in SE

As discussed in the previous section, the range of claimed potential benefits coupled with occasional claims of industrial application and impact (Zhang et al. 2011) gives an impression that simulation is a panacea for problems in SE. However, in the following sections, we discuss the limitations of SBS in SE. This discussion draws on an analysis of the existing literature (Ali et al. 2014) and our experience of using simulation in industrial settings (Ali and Petersen 2012; Ali et al. 2015). In the SE literature, the use of simulation has two implied purposes, which are:

1. Simulation as a problem-solving tool for decision support for SE practitioners (Banks 1999).
2. Simulation as a means to conduct controlled experiments (Abdel-Hamid 1988) and as an alternative to industrial case study research (Müller and Pfahl 2008).

To scrutinize the above two claims, it is useful to assess the following two aspects of SBS: (a) the cost of conducting effective SBS and (b) the strength of evidence generated in SBS. Other limitations of SBS in SE are further discussed by Pfahl (2014).

3.1 Cost of Conducting SBS

SBS are certainly less risky than experimenting with a physical model of a system or the actual system (Fig. 1). Physical models in software engineering are very uncommon, so that is not considered as a feasible alternative unless embedded or robotics systems are into play. Similarly, the limitations of generalizability when experimenting with students as subjects and artifacts that are not representative of industrial-scale are well-documented (Feldt et al. 2018). However, the cost of conducting SBS for industrial cases should be accounted for, and will include, e.g.: (1) the cost of developing and calibrating a simulation model, (2) design and analysis of simulation-based investigations, (3) the cost of necessary data collection, or (4) setting up a measurement program that can feed the simulation model with sufficiently reliable data.

Pfahl and Lebsanft (2000) reported 18 months of calendar time for a simulation study. They reported an effort of one person-year in consulting and 0.25 person-years for the development part of the study. In another study, Pfahl et al. (2004) report the calendar time of 3 months for knowledge elicitation and modeling and four meetings with the client while conducting a simulation-based study. Shannon

(1986) predicted a high cost for simulation as well, “a practitioner is required to have about 720 h of formal classroom instruction plus 1440 h of outside study (more than one man-year of effort).”

3.2 *Quality of Evidence from SBS*

The role of SBS as a decision-support tool and an empirical method requires a consideration of the strength of evidence generated by SBS. Some limitations discussed below are general limitations of simulation, but they get aggravated in SBS because of the nature of software development.

Models Are Simplifications Simulation models are a simplification, abstraction, and approximation of the system of interest (Christie 1999). SBS in SE often require modeling the process or project dynamics. Fully capturing these complex dynamics of a real-world process or project in a simulation model is not possible, which raises questions about the validity of a simulation model.

Measurement Challenges SBS in SE deal with quantification of variables and their relationships. Often strict cause–effect relations with determined magnitudes of relations are not available in SE (Christie 1999). Therefore, the confidence in a simulation model depends on the verification and validation of both the structure and behavior of the model (Christie 1999). The lack of reliable measures and quantified relations (Jørgensen and Kitchenham 2012; Kitchenham 2010) also adds another degree of uncertainty in the simulation models used in SE.

Lack of Data Accurate data to use in an SBS is extremely important, as a model without supporting data cannot deliver adequate predictions, and such a model is essentially a visual metaphor (Olsen 1993). However, in SE, a lack of empirical data is a common challenge faced by researchers conducting SBS (Kellner et al. 1999). Launching a measurement program to feed a simulation model is often not feasible, and often existing SBS rely on the use of industrial averages, expert estimates, or values acquired from analytical models. The lack of accurate data also challenges the reliability of SBS results.

3.3 *Simulation as a Problem-Solving, Decision-Support Tool for SE Practitioners*

The vision to have practitioners using simulation as a decision-support tool in SE practice suggests a transfer of technology from academia to industry. As we are proposing a new tool or practice to the industry, it is important to have the following prerequisite information (Ali 2016):

- Cost of adoption.
- Effectiveness over the existing methods.

Apart from the cost of conducting an SBS, we should take into account the cost of required tool support and training and the effort required for the development and maintenance of a simulation model. Besides, frequent changes in technology, software development process, environment, and customer requirements require keeping the simulation model up to date. Given the high cost of simulation model development and maintenance, it is unlikely that practitioners will invest the required effort.

There is a need to identify the challenges when using simulation in a company. This includes the integration of simulation models with the existing decision-support systems used at the company (Balci 1990; Murphy and Perera 2001).

Contrary to the claims of impact on the software industry (Zhang et al. 2011), an extensive literature review on the industrial applications of simulation in SE found no evidence of adoption (Ali et al. 2014).

3.4 Simulation as an Alternative Empirical Method

In the past, some authors had suggested the use of simulation in SE research as an alternative to case study research. For example, "the usual way to analyze process behavior is to perform the actual process in a case study and observe the results. This is a very costly way to perform process analysis, because it involves the active participation of engineers. Furthermore, results from a particular case study cannot necessarily be generalized to other contexts. Another way of analyzing processes is to simulate them" (Müller and Pfahl 2008). Others have indicated its use as an alternative to expensive controlled experiments, e.g., "Simulation modeling provides a viable experimentation tool for such a task. In addition to permitting less costly and less time-consuming experimentation, simulation-type models make 'perfectly' controlled experimentation possible" (Abdel-Hamid 1988).

Given the strength of evidence generated in SBS (as we briefly summarized in Sect. 3.2), we do not see the use of simulation as an alternative to controlled experiments or case study research. Rather, we suggest the use of simulation to complement other empirical methods, which aligns with advice by Münch et al. (2003) and by Pfahl and Ruhe (2002).

The cost of conducting an SBS and the lack of the inclusion of simulation in CS/SE undergraduate or graduate curriculum are likely to prevent industry practitioners from designing and conducting SBS anytime soon. However, there is sufficient evidence (Ali et al. 2014) to show that researchers can use simulations to support empirical research.

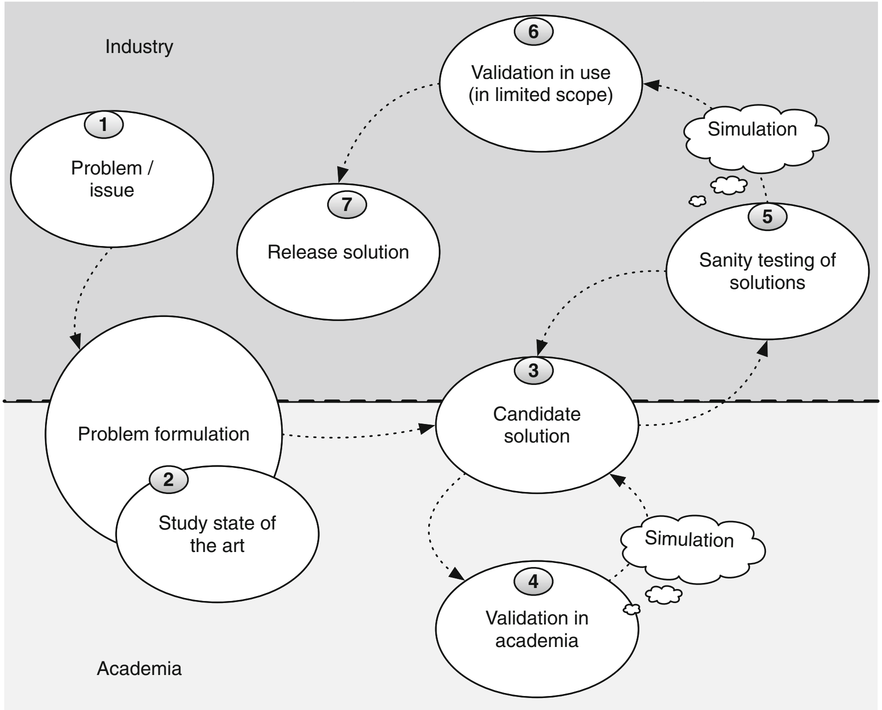

Fig. 2 The technology transfer model (adapted from Gorschek et al. 2006)

In our opinion, the use of simulation is most suitable as a means to *sanity test* solution proposals. Such SBS can be conducted both in academic and industrial settings, as shown in Fig. 2. Example 2 in Sect. 6 reports the use of simulation for supporting software process improvement decisions. Further use of simulation, with empirical evidence supporting its use, is for training and educational purposes (Pfahl 2014; Ali and Unterkalmsteiner 2014). Example 1 in Sect. 6 reports the use of simulation for training in an industrial setting.

Simulation Is Not a Silver Bullet

- Simulation is not free of cost, but there are contexts in which incurring this cost is worthwhile.
- Simulation results come with a certain degree of confidence. It is up to the experimenter to understand whether it has achieved the research goals.
- Simulation-based investigations do not mean to replace other empirical methods, but they can complement them in situations where the in loco observation is unfeasible.

4 The Need for Guidance

The conduction of SBS in the context of software engineering has several challenges, and here we present evidence on the need for guidance when adopting simulation to support research.

Two systematic literature reviews (De França and Travassos 2013; Ali et al. 2014) of SBS in software engineering found that these studies lack rigor. De França and Travassos (2013) identified lack of planning (study definition), V&V (before running simulation investigations) to assure minimal confidence in the simulation results, and output analysis procedures. Mainly, such activities are performed ad hoc, with particular studies using systematic procedures to perform one or another activity, but studies presenting a full systematic process or method are scarce. Ali et al. (2014) assessed the quality of software process simulation and modeling (SPSM) studies concerning rigor and practical relevance. They did not find reported cases of the successful transfer of SPSM to practitioners in the software industry. Furthermore, no studies were found reporting a long-term use of SPSM in practice and no evidence to back the claims of practical adoption and impact on industrial practice (Zhang 2012; Zhang et al. 2011). Finally, both reviews agree on the lack of information in simulation reports.

Regarding reporting of simulation studies in software engineering, De França and Travassos (2012) present general guidelines, and specifically for SPSM studies using continuous simulation, a good template is available in Madachy's book (Madachy 2008). Recently, Monks et al. (2019) proposed the STRESS checklist for reporting simulation studies.

Simulation can provide actual benefits when supporting software engineering research, and both simulation and SE literature have presented some guidance on conducting SBS. However, it is not an obvious decision for practitioners or researchers to select one process or a set of guidelines to follow, considering the little background in the area. For that, Ali and Petersen (2012) proposed a consolidated process for SPSM with supporting guidelines. Besides, De França and Travassos (2016) propose simulation guidelines for experimenting with dynamic models in SE. These contributions are the foundation for the life cycle presented in Sect. 5.

5 Simulation-Based Studies Life Cycle

When supporting a research process with simulations, researchers generally characterize the life cycle (process) for simulation studies as a knowledge-intensive and iterative process (Alexopoulos and Seila 1998; Balci 1990; Maria 1997; Banks 1999; Sargent 1999), including the following activities: study definition or phenomenon understanding (also called system observation); model design and development; verification and validation (V&V); performing simulation-based

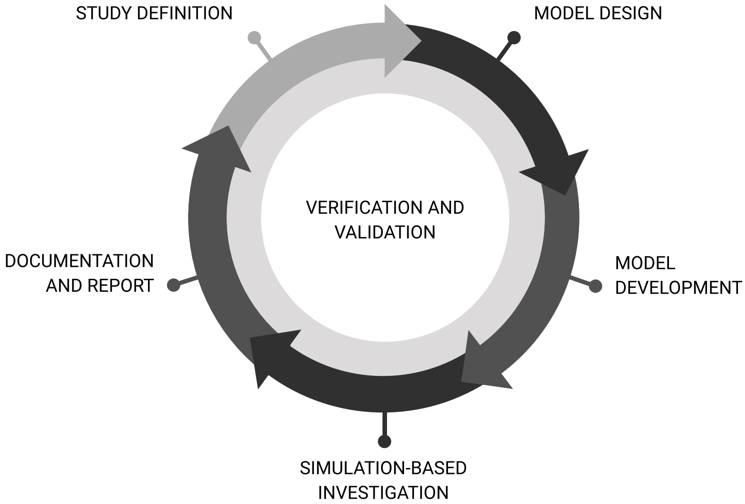

Fig. 3 Simulation-based studies life cycle

investigations, including experimental design and output analysis; and finally documentation and reporting.

The consolidated life cycle for SBS (Ali and Petersen 2012; Fig. 3) described in this section is based on processes identified both in general computer simulation and SE literature. For this, we adopt the following guiding principles:

- Start small and later on enhance the simulation model, looking for analogies to solve the problem rather than starting the model building from scratch and working over an extended period for developing the model in one go (Ahmed et al. 2008). It reinforces the iterative nature of the process.
- Involve and keep frequent contact with all stakeholders throughout the study (Murphy and Perera 2001; Ahmed and Robinson 2007; Ahmed et al. 2008). It is important to improve the utility and practical relevance for the simulation.
- Models are abstractions, and that imposes a trade-off between complexity and realism, so not pursuing a perfect model is recommended, but still critically analyzing the results (Madachy 2008).
- All models are incomplete, bounded by the behavior under study as much as it is required to answer the questions of interest (Madachy 2008). It highlights the need to constantly revisit the simulation goals and raises a question about the reuse of a model beyond its original intent.
- In industrial simulation studies, it is important to deliver results and recommendations quickly as the modeled system and its environment are likely to change (Ahmed and Robinson 2007; Murphy and Perera 2001).

- It is possible to model a phenomenon of interest in several ways. The differences may arise because of the perspective of stakeholders, level of detail, and the modeler's assumptions (Madachy 2008).
- Continually challenge the model to increase the credibility of the model through further verification and validation (V&V) (Madachy 2008; Ahmed and Robinson 2007). That is why V&V activities are placed as continuous and parallel.
- To facilitate effective communication, use simple diagrams to communicate with stakeholders until they seek more detail, such as equations, since it may not be necessary to present those details (Madachy 2008).

Next sections provide detailed discussion for each major activity in the simulation studies life cycle.

5.1 Study Definition

Simulation-based studies should present a definition including the research context, problem, goals, and questions. For the *context*, model users and audience should be taken into account, as well as organizational policies involved when conducting simulation studies in industry. There are two suggestions (Petersen and Wohlin 2009; Dybå et al. 2012) on how to identify and describe contextual information for software engineering research that we understand as useful for simulation studies. The motivating *problem* frames the SBS in a proper scope and can be used to define relevant usage scenarios. Ideally, simulation models answer questions based on a purpose that can be assessed. Therefore, research *goals* and derived *questions* should guide the whole life cycle so that the chances for practical acceptance are increased. Goal-Question-Metric (GQM) (Basili and Rombach 1988) goal template can support the study definition, also recommended in the IMMoS method (Pfahl 2001; Pfahl and Lebsanft 2000).

Kellner et al. (1999) mention common purposes for SPSM, such as strategic management; planning, control, and operational management; process improvement and technology adoption; understanding; and training and learning.

Before the simulation model development, technical feasibility should be checked (Pfahl and Ruhe 2002; Ahmed et al. 2005) considering prerequisites for model development and usages like adequacy of problem definition, availability of data, and process maturity (Pfahl 2001). For that, Balci (1990) proposed the following questions:

- Do the benefits of conducting a simulation-based study justify the cost?
- Is it possible to use simulation for the goal of the study?
- Is it possible to complete the study in the given time and resource constraints?
- Is the necessary information available, e.g., classified or not available?

Furthermore, we recommend questions to support this decision focusing on additional constraints regarding the model development and experimentation (De França and Travassos 2016):

- Are the risks of running the real system high, including loss or waste of money or time, reaching an irreversible state, or compromised safety?
- What are the available instruments and procedures for data collection?
- Is there enough data to support model calibration and validation, as well as statistical analysis?

5.2 *Model Design*

Simulation studies in SE span widely in terms of phenomena to simulate. Known behaviors like Brooks' law, process evaluation, and project estimation, as well as architectural issues such as performance and scalability assessment, are examples of objects of study. Mostly, software engineering simulations concentrate on software process and project issues (De França and Travassos 2013). In the following sections, we describe relevant aspects of simulation modeling and provide some specifics on software process and projects.

5.2.1 Input Parameters and Response Variables

Input parameters represent the independent variables for which users of a simulation model can define values that impact on the model state and, consequently, on the response variables. Inputs can be calibration or variable parameters. To identify these parameters, it is recommended to start designing the model early on (Kellner et al. 1999). Also, consider that it in case of unavailability of data it may not be practical to measure all relevant variables accurately.

In the case of stochastic simulation, select appropriate probability distributions for input parameters (Balci 1990). Law (2007) provides a detailed discussion of statistical methods to support this decision. Stochastic or not, the integrity of simulation data must be ensured by maintaining constant communication with stakeholders (Murphy and Perera 2001).

On the other side, to identify the response (output) response variables required for the model (Park et al. 2008; Kellner et al. 1999; Rus et al. 2003), one needs to address the problem statement by answering the key questions identified in the study definition. For that, it is recommended to use GQM to identify the response variables (metrics) that address the defined research goals (Madachy 2008; Rus et al. 2003). Additionally, to specify problematic or desired behaviors (called the reference behavior) (Müller and Pfahl 2008) can help to identify response variables.

Reference behaviors describe changes on variables by plotting their values over time (Madachy 2008), preferably using historical data (Pfahl 2001). However, it is also possible to consider behavioral patterns based on experience when actual data is not available (Madachy 2008; Pfahl 2001) and using relative measures instead of striving for absolute ones (Madachy 2008).

5.2.2 Conceptual Modeling

This activity requires both explicit and tacit knowledge about the phenomenon to be simulated. This way, modelers should identify constructs and behaviors (e.g., process elements, information flows, and decision rules) influencing the response variables and relevant to the simulation goal. In addition, it is essential to consult or interview domain experts as they have knowledge beyond the project or product documentation and can judge the relevance of the information to the problem under study (Müller and Pfahl 2008), avoiding missing important aspects and reducing the threat of misunderstanding (Pfahl and Lebsanft 1999).

At this stage, the creation of influence diagrams to describe the positive or negative influence of various parameters supports the identification of internal variables (Rus et al. 2003). In this sense, the researchers need to motivate the choice of cause–effect relationships with relevant data sources and evidence from the literature (Pfahl 2001). Besides, individual cause–effect relationships should be reviewed before introducing more combinations (Pfahl 2001).

In the case of SPSM, creating static process models helps to understand the flow of information and the transformation of artifacts in various activities (Rus et al. 2003).

The conceptual model should not capture the whole phenomenon at once, i.e., researchers should include only the constructs and relations necessary to generate the behaviors of interest, according to the goal and the scope of the model. For that, a top-down iterative approach provides additional details only when necessary (Madachy 2008).

As researchers gain more knowledge on model variables (explicitly described), the simulation feasibility should be assessed again, as described in Sect. 5.1.

5.3 Model Implementation

To develop a simulation model, it is required to understand the selected 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 regarding any of these aspects may impose different types of threats to validity (De França and Travassos 2015).

In addition, the simulation model should be developed with high modularity (Ahmed et al. 2008), separating data from the model to support modification and experimentation (Murphy and Perera 2001), and always kept in a state that it could be simulated (or tested) (Madachy 2008).

5.3.1 Simulation Approach

An executable model is implemented from the conceptual model using a simulation language, which should be based on a simulation approach like system dynamics (SD), discrete-event simulation (DES), or agent-based simulation (ABS). The simulation approach abstracts the essential characteristics and behaviors the model has to fit in.

The choice of a proper simulation approach depends on the particular goal of the study (Madachy 2002; Kellner et al. 1999). One consequence of concentrating on software process and project issues is the focus on continuous and discrete simulation (De França and Travassos 2013). Madachy (2002) considers continuous simulation (e.g., SD) more suitable for strategic analysis, high-level perspectives, and long-term trends, while discrete simulations can make detailed process analysis, resource utilization, and relatively short-term analysis more convenient.

5.3.2 Simulation Environment and Tools

The simulation environment consists of all instruments needed to perform the study, encompassing the simulation model itself, data sets, data analysis tools (including statistical packages), and simulation tools/packages. This way, planning and reporting studies considering those aspects are very important (see chapter "Open Science in Software Engineering" for a broader discussion on making the experimental package available). Simulation packages often differ in how they implement the simulation engine mechanism. Therefore, it is possible to get different results depending on the engine implementation.

Also, a simulation package should support not only the underlying simulation approach, but also the experimental design and output data analysis. Several tools are available to facilitate this analysis, providing a graphical interface to support output visualization, walk-through, interactive simulation, sensitivity analysis, and integration with third-party applications.

Raw input data requires extra effort to understand its properties (e.g., data distribution and shape, trends, and descriptive stats) and perform the transformations (e.g., 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.

Madachy (2008) and Ahmed and Robinson (2007) list the price, ease of use, training, documentation and maintenance support, computer platform and user familiarity, and performance requirements as the criteria for choosing an appropriate simulation tool. This choice also depends on the fit of the research questions, assumptions, and the theoretical logic of the conceptual model with those of the simulation approach (Houston et al. 2001).

Another important perspective concerns the computational infrastructure. The simulation needs to be settled up and reported so that one can understand the details for replicating the study. This way, processor capacity, operating system, amount of data, and execution time interval are relevant characteristics to estimate schedule and costs for the study.

Finally, simulations involving multiple trials/runs often need to summarize information from each intermediate trial for the final output analysis. Mean and standard deviation are common measures for this purpose and determine confidence intervals, for instance. This way, the individual measures are stored in a database or external files.

5.3.3 Model Calibration

It is recommended to use actual data for model calibration (Madachy 2008; Kellner et al. 1999). Such data is used for the generation of equations and parameters and to determine the distribution of random variables. However, the lack of data for calibration and validation in real-world settings (Kellner et al. 1999) imposes some threats as the desired data is often poorly defined, inaccurate or missing altogether, considering it was not planned and collected to support a particular simulation-based study.

An alternative is to work with synthetic data (Ören 1981). However, it requires evidence on data validity, i.e., provide indications that the simulated data represents the real phenomenon. For that, statistical tests can be applied to verify how close both real and synthetic samples are. Additionally, modelers may consult domain experts to deduce the accuracy and relevance of data (Murphy and Perera 2001; Raffo and Kellner 2000).

Planning the data collection avoids measurement mistakes, promoting the collection of data as soon as they are made available for the target variables and tracking contextual information (including qualitative data), which provides better and accurate reasoning when performing output analysis by supporting the explanations.

After the collection, quality assurance procedures ought to take place to verify their consistency and accuracy, avoiding the inclusion of outliers or incomplete data. For instance, to avoid biased observations and exposure to risks (i.e., undetected seasonal periods in time series), the data collection period should represent both transient and steady states.

Data used for model calibration and setting model parameters in the investigations need to share the same context. Therefore, the values used for investigation have to be consistent, avoiding attempts to generalize behaviors to different contexts inappropriately. The use of cross-company data is an example of how it can impose a threat to internal validity on the simulation results.

5.4 Verification and Validation

As the phenomenon under investigation is essentially observed through the execution of a simulation model, validity aspects concentrate on both model and data validity. Besides, model validity is a relative matter (Madachy 2008), depending on the purpose of the study. This way, evidence regarding model (conceptual and implemented) validity means the researcher should be aware of all the initiatives of submitting the simulation model to V&V procedures and their results.

The list provided in Table 1 supports the identification of attempts to verify or validate a simulation model in existing reports. These procedures were identified both in the context of SE (De França and Travassos 2013) and general discrete simulation (Sargent 1999). Barlas (1989) presents general procedures for validating SD models. None of these V&V procedures can avoid all the potential threats to validity alone. However, properly combining some of them can increase the confidence in simulation results.

Face validity is a *white-box* procedure for reviewing both simulation model and I/O matching from the perspective of experts and it is a relevant indication that, in the face of the model representation and its generated behavior, it gives the impression of being valid. It enables the review of internal properties and behaviors of a simulation model like model variables, equations, and relationships, rather than dealing with it as black box, i.e., observing just the I/O matching. This way, domain experts may identify threats to construct validity in advance. Face validity sessions may happen on workshops, group or individual interviews. The main idea is to present the model by following a walk-through approach to show how the input values generate outcomes, exemplifying with real scenarios so that experts can realize the model behavior and validate the simulation results for a given set of inputs.

To have the causal relationships and assumptions supported by empirical evidence improves external validity (Davis et al. 2007). Besides, it reduces the modeler's bias, not relying exclusively on experts' opinions or ad hoc observations. This way, secondary studies may be performed to search for evidence. However, models with many causal relationships may impose a great effort into using this approach.

The verification of model assumptions increases the reliability of simulation results. Face validity can be combined with other procedures to compare empirical data/behavior (Sargent 1999) to assess explicit model assumptions. For instance, to use comparison with reference behaviors, historical validation, or predictive validation to understand if the model is capable of reproducing an empirical behavior in terms of internal variables and outcomes. The modelers make, even implicitly, some assumptions regarding the phenomenon. For instance, the increase in a response variable directly caused by the presence of a given parameter. If these assumptions are embedded in the model, it may represent a threat to internal validity, since this behavior should not be coded directly in the model, rather it should be treated as an effect of a chain of actions, events, and conditions generating such behavior in the output variable.

When performing sensitivity analysis, it is important to consider constraints from the real world to make sure that the model reflects real-world behavior (Madachy 2008).

Table 1 Verification and validation procedures (De França and Travassos 2016)

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 with reference behaviors	Compare the simulation output results against trends or expected results captured from historical data or reported in the literature
Comparison with other models	Compare the results (outputs) of the simulation model being validated to the results of other valid (simulation or analytic) models. Controlled experiments can be used to arrange such comparisons
Event validity	Compare the events triggered during the simulation runs to those of the real phenomenon to determine if they are similar. This technique is applicable to event-driven models
Historical data validation	If historical data exists, 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	Use logic deductions from model assumptions to develop the correct (valid) model, by assuming that everyone knows whether the stated underlying assumptions are true
Predictive validation	Use the model to forecast the phenomenon behavior, and then compare this behavior to the model forecast to determine if they are similar (or equal). The data may be obtained by observing the real phenomenon or conducting experiments, e.g., field tests for provoking its occurrence. Also, data from the literature may be used when there is no complete data in hands
Internal validity	It is likely used for measuring model accuracy. Several runs of a stochastic model are made to determine the amount of (internal) stochastic variation. A large amount of variation (lack of consistency) may threat the model confidence, even if it is typical of the problem under investigation
Sensitivity analysis	Consists of systematically changing the values of the input and internal parameters of a model to determine the effect upon the model output. The same effects should occur in the model as in the real phenomenon. This technique can be used semi-qualitatively—trends only—and quantitatively—both directions and (precise) magnitudes of outputs
Testing model structure and behavior	Submit the simulation model to test 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 behavior can be tested by appropriate selection of values of parameters
Based on empirical evidence	Collect evidence from the literature (empirical studies reports) to develop the model causal relationships (mechanisms)
Turing tests	Individuals knowledgeable about the phenomenon are asked if they can distinguish between real and model outputs

Performance measures such as bias, accuracy, coverage, and confidence intervals can be used as criteria to benchmark more accurate simulation models (Burton et al. 2006). For instance, if the outcomes have low accuracy or are in a wide confidence interval, these results may be distant from reality. This information also brings credibility to the simulation study.

Finally, modelers should avoid false expectations (e.g., perfect predictions on the first model run) by reviewing patterns for qualitative similarity (Madachy 2008).

5.5 Simulation-Based Investigation

The experimental design defines a causal model establishing relationships between parameters and response variables, based on research goals and questions. Balci (1990) mentions that different parameters (input variables), behavioral relationships, and auxiliary variables may represent model variants. Thus, during simulation execution, the model variables may be held constant or allowed to vary according to conditions established a priori.

To fully describe the experimental design, we suggest using a design matrix, in which every row is a design point or scenario, which is a combination of different alternatives for each factor (column). However, several designs can be generated for the same set of factors. Kleijnen et al. (2005) claim that the design of experiments for simulation is different since we are not limited by real-world constraints.

Factorial designs are the most common one for simulation as they include all possible combinations (scenarios) for a set of factors. For instance, a full factorial design for k factors using two alternatives per factor is denoted as 2^k design, meaning the number of scenarios required to determine effects from k factors and their interactions. In addition, there are variants for situations in which the simulation runs are time-consuming, as execution time grows exponentially with the number of factors and alternatives. Therefore, it is possible to reduce the number of scenarios by executing just a fraction of the scenarios (fractional factorial designs), but still having an effective estimator. In the following, we list (not exhaustively) important aspects to select an adequate design as suggested by De França and Travassos (2016):

- *Simulation goals*: designs for understanding or characterization are not the same for comparison or optimization;
- *Experimental frame*: whether the area of interest is local or global, and its impacts in the range of levels;
- *Number of factors and levels*: they exponentially increase the number of scenarios in full factorial designs;
- *Domain of admissible scenarios*: full factorial designs may generate inadmissible (unrealistic) scenarios;

- *Deterministic and stochastic components of the model*: they affect how to deal with variation in the experimental design;
- *Terminating conditions*: if it is a steady-state or a terminating simulation, with an event to specify the end of the experiment.

Sensitivity analysis is a useful technique to select interest factors and a range of alternatives. Furthermore, such a systematic approach reduces the bias and avoids the "fishing" for positive results. For characterization purposes, it is recommended to keep a low number of levels per factor, but covering a high region of interest (Montgomery 2017).

When designing a simulation-based investigation, researchers should consider as factors (and levels) not only the input parameters, but also internal parameters, different data sets and versions of the model, implementing alternative strategies to be evaluated (De França and Travassos 2016). For instance, Garousi et al. (2009) use a design with two distinct data sets as alternatives for calibration based on data from the technical literature to derive the scenarios and the simulation model remaining constant. This way, different calibrations representing the different simulation scenarios can be compared.

Ad hoc designs explore the use of scenarios. In this case, the modeler plans the scenarios of interest (Barros et al. 2000) and then derives the design. By adopting this strategy, the relevance and adequacy of each chosen scenario should be explained and tied to the study goals. Furthermore, the description of scenarios needs to be as precise as possible, clarifying all the relevant contextual information, as well as input parameters for them.

The main drawback of ad hoc designs is the possibility of introducing bias, with no opportunity to investigate side effects such as interactions between factors.

Selected scenarios and the nature of the model (deterministic or stochastic) drive the number of simulation runs. Each scenario consists of an arrangement of experimental conditions where possible alternatives are assigned to specific factors. The more scenarios involved, the more simulation runs are required.

Stochastic simulation models produce an inherent variation in the output due to the pseudorandom number generation. Therefore, running each scenario only once is not enough to reveal the amount of variation. On the other side, the higher the number of runs (replications), the closer one gets of the desired accuracy level. Replication is achieved by using different pseudorandom numbers to simulate the same scenario. In this case, each output is derived from auto-correlated observations (Kleijnen et al. 2005) that cannot be aggregated as they are not independent. Thus, given the required accuracy and a sample estimate from a few model runs, it is possible to determine the number of required runs and avoid threats to conclusion validity. Such a procedure for the calculation can be found in Law and Kelton (2000).

5.6 *Threats to Simulation-Based Studies Validity*

The SE community has discussed threats to validity, and most of the reported threats concerned with in vitro experimentation are described by Wohlin et al. (2012) and categorized under a positivist perspective as threats to construct, internal, external, and conclusion validity (Petersen and Gencel 2013). We consider this perspective is more suitable for considering and addressing threats to validity for SBS than other world-views. Therefore, most of the known threats to controlled experiments have to be considered when conducting simulation studies, especially considering in vitro experiments, in which the human subjects drive the simulations, which introduces additional risks to the validity of a study. Moreover, new situations emerge from in silico experiments, in which common types of experimental validity are closely related to the simulation model and data validity (De França and Travassos 2015). A list of categorized threats to simulation studies validity can be found in De França and Travassos (2015).

Garousi et al. (2009) and Raffo (2005), for instance, consider model validity in several perspectives, such as model structure, supporting data, input parameters and scenarios, and simulation output. We understand that these aspects are relevant, but researchers should not be limited to them (De França and Travassos 2015), also considering the simulation investigation design as well. This way, researchers should consider checking for threats to the simulation study validity before running the experiments and analyzing output data to avoid bias. Additionally, non-mitigated threats, limitations, and non-verified assumptions must be reported (De França and Travassos 2016).

The use of simulation promotes both the construct and internal validity as it demands accurately specifying and measuring constructs (and their relationships) and the theoretical logic that is enforced through the discipline of algorithmic representation in software, respectively (Davis et al. 2007). However, De França and Travassos (2015) identified threats to construct validity, such as inappropriate cause–effect relationship definitions, inappropriate real-world representation by model parameters and calibration data and procedure, hidden or invalid underlying assumptions regarding model concepts, and the simulation model not capturing the corresponding real-world building blocks and elements.

External and conclusion validity can be accomplished by reproducing empirical behaviors and applying adequate statistical analysis over the model outputs, respectively. However, conclusion validity also relates to design issues like sample size, the number of simulation runs, model coverage, and the degree of representation of scenarios for all possible situations.

Simulation Life Cycle Considerations

- A simulation-based study is about gaining knowledge and dynamically analyzing it. So, an iterative approach is fundamental to allow explication and reflection about the phenomenon under study.
- Keeping involved stakeholders when modeling and analyzing simulation results is as important as in software development.
- Verification and validation procedures play a fundamental role in the validity of a study.
- Mostly, this consolidated process is based on solid simulation literature, empirical evidence, and the authors' experience. This way, some recommendations may require a level of adaptation.

6 Practical Examples

To illustrate some potential uses of SBS, in the following sections, we briefly describe two practical scenarios where we have conducted SBS in industrial settings.

6.1 Example 1: Use of Simulation to Encourage Behavioral Change in a Company

We, together with the representatives from a case company, identified a need to illustrate the benefits of early integration. The intention was to highlight the consequences of missing a test iteration in the current way of working. Given the scale and complexity of the test process, it was not possible to demonstrate this with static process diagrams. The aim was to develop a simulation model that adequately represents the test process at the case company so that it is relevant and realistic for the developers. This realism made the results more relatable for them. Also, it provided the ability to show the consequences of various what-if scenarios in terms of the flow of requirements through the various stages of development, testing, and release.

We used interviews, process documentation, and guided walk-through of the testing process to develop a realistic process understanding. This understanding was modeled in a simulation model using system dynamics. The model was calibrated using data collected from various repositories in the company. For variables and relations where data was not available, we relied on expert opinion to estimate the missing values. For details, please see (Ali and Petersen 2012).

6.2 Example 2: Use of Simulation to Sanity-Check Process Improvement Ideas

We were supporting an organization in a process assessment and improvement initiative. Two main alternatives being considered were (a) changing the sprint length (particularly for testing) or (b) improving the flow. We conducted an SBS to assess the likely impact of the two improvement proposals. The simulation model represented both scenarios and was calibrated using the company data.

The use of a simulation model was received positively by the practitioners, and the results of the simulation study influenced the choice of improvement actions pursued by the company. For details, please see Ali et al. (2015).

7 Recommended Further Reading

7.1 Simulation Modeling and Approaches

The two most common approaches for simulation modeling are continuous and discrete-event simulation. Madachy's book (Madachy 2008) is an excellent resource when using system dynamics for continuous simulation of SE phenomena. For discrete-event simulation, Law and Kelton (2000) provide an excellent resource covering all technical steps in developing a simulation model. However, their focus is not on the modeling of SE phenomena.

7.2 Verification and Validation in SBS

For additional discussion on verification and validation in simulation-based studies, readers are encouraged to refer to the software engineering (De França and Travassos 2015) and general computer simulation literature (Sargent 1999; Babuska and Oden 2004).

7.3 Simulation-Based Investigations

For detailed instructions on designing simulation-based investigations or experiments using a simulation model, please see the works in De França and Travassos (2016), Kleijnen et al. (2005), and Houston et al. (2001).

7.4 Software Process and Project Simulation

For guidelines to investigate the software process and project using simulation, please see the guidelines from Ali et al. (2014), as well as the ones presented by Pfahl (2014).

8 Conclusion

This chapter summarizes a wide span of knowledge on computer simulations in the context of SE. We discussed, under several perspectives, the role of SBS in software engineering research and methodological aspects relevant for conducting this sort of study to support SE research.

Although we do not intend to discuss in detail each potential application of simulation, it is notable the need for systematic observation of the system or phenomenon to be simulated before developing and performing investigations based on simulation models. Moreover, the experimental basis is fundamental to benefit from simulation studies. Therefore, SBS should be considered as part of a greater research methodology, i.e., it should not be seen as an "end" or a sole research method, but a tool to achieve complementary evidence and to reduce risk before intervening on the target context or moving towards more focused observation.

Acknowledgements Breno has been supported by CNPq-Brazil (Grant 141152/2010-9) during the development of part of this work. Nauman has been supported by a research grant for the VITS project (reference number 20180127) by the Knowledge Foundation in Sweden and by ELLIIT, a Strategic Area within IT and Mobile Communications, funded by the Swedish Government. We would also like to thank the reviewers for their feedback that has helped us to improve the presentation and the contents of the chapter significantly.

References

Abdel-Hamid TK (1988) Understanding the "90% syndrome" in software project management: a simulation-based case study. J Syst Softw 8(4):319–330

Ahmed R, Robinson S (2007) Simulation in business and industry: how simulation context can affect simulation practice? In: Proceedings of the 2007 spring simulation multiconference, vol 3. Society for Computer Simulation International, pp 152–159

Ahmed R, Hall T, Wernick P, Robinson S (2005) Evaluating a rapid simulation modelling process (RSMP) through controlled experiments. In: 2005 International symposium on empirical software engineering. IEEE, Piscataway, p 10

Ahmed R, Hall T, Wernick P, Robinson S, Shah M (2008) Software process simulation modelling: a survey of practice. J Simul 2(2):91–102

Alexopoulos C, Seila AF (1998) Output data analysis. In: Banks J (ed) Handbook of simulation. Wiley, New York, pp 225–272

Ali NB (2016) Is effectiveness sufficient to choose an intervention?: considering resource use in empirical software engineering. In: Proceedings of the 10th ACM/IEEE international symposium on empirical software engineering and measurement, ESEM 2016, Ciudad Real, September 8–9, 2016, pp 54:1–54:6
Ali NB, Petersen K (2012) A consolidated process for software process simulation: state of the art and industry experience. In: 2012 38th Euromicro conference on software engineering and advanced applications. IEEE, Piscataway, pp 327–336
Ali NB, Unterkalmsteiner M (2014) Use and evaluation of simulation for software process education: a case study. In: European conference software engineering education (ECSEE), Shaker Verlag, Herzogenrath, pp 59–73
Ali NB, Petersen K, Wohlin C (2014) A systematic literature review on the industrial use of software process simulation. J Syst Softw 97:65–85
Ali NB, Petersen K, de França BBN (2015) Evaluation of simulation-assisted value stream mapping for software product development: two industrial cases. Inf Softw Technol 68:45–61
Babuska I, Oden JT (2004) Verification and validation in computational engineering and science: basic concepts. Comput Methods Appl Mech Eng 193(36):4057–4066
Balci O (1990) Guidelines for successful simulation studies. In: Proceedings of the winter simulation conference. IEEE Press, New Jersey, pp 25–32
Banks J (1999) Introduction to simulation. In: Proceedings of the 31st conference on Winter simulation: simulation – a bridge to the future, WSC 1999, Phoenix, December 05–08, 1999, vol 1, pp 7–13
Barlas Y (1989) Multiple tests for validation of system dynamics type of simulation models. Eur J Oper Res 42(1):59–87. https://doi.org/10.1016/0377-2217(89)90059-3
Barros M, Werner C, Travassos G (2000) Applying system dynamics to scenario based software project management. In: Proceedings of the 2000 international system dynamics conference
Basili VR, Rombach HD (1988) The TAME project: towards improvement-oriented software environments. IEEE Trans Softw Eng 14(6):758–773
Burton A, Altman DG, Royston P, Holder RL (2006) The design of simulation studies in medical statistics. Stat Med 25(24):4279–4292
Christie AM (1999) Simulation: an enabling technology in software engineering. CROSSTALK J Def Softw Eng 12(4):25–30
Davis JP, Eisenhardt KM, Bingham CB (2007) Developing theory through simulation methods. Acad Manag Rev 32(2):480–499
De França BBN, Travassos GH (2012) Reporting guidelines for simulation-based studies in software engineering. In: Proceedings of the 16th international conference on evaluation assessment in software engineering (EASE), pp 156–160
De França BBN, Travassos GH (2013) Are we prepared for simulation based studies in software engineering yet? CLEI Electron J 16(1):9
De França BBN, Travassos GH (2015) Simulation based studies in software engineering: a matter of validity. CLEI Electron J 18:5
De França BBN, Travassos GH (2016) Experimentation with dynamic simulation models in software engineering: planning and reporting guidelines. Empir Softw Eng 21(3):1302–1345
Dybå T, Sjøberg DI, Cruzes DS (2012) What works for whom, where, when, and why?: on the role of context in empirical software engineering. In: Proceedings of the ACM-IEEE international symposium on empirical software engineering and measurement. ACM, New York, pp 19–28
Eck JE, Liu L (2008) Contrasting simulated and empirical experiments in crime prevention. J Exp Criminol 4(3):195–213
Feldt R, Zimmermann T, Bergersen GR, Falessi D, Jedlitschka A, Juristo N, Münch J, Oivo M, Runeson P, Shepperd M, Sjøberg DIK, Turhan B (2018) Four commentaries on the use of students and professionals in empirical software engineering experiments. Empir Softw Eng 23(6):3801–3820. https://doi.org/10.1007/s10664-018-9655-0
Garousi V, Khosrovian K, Pfahl D (2009) A customizable pattern-based software process simulation model: design, calibration and application. Softw Process Improv Pract 14(3):165–180. https://doi.org/10.1002/spip.411

Gorschek T, Wohlin C, Garre P, Larsson S (2006) A model for technology transfer in practice. IEEE Softw 23(6):88–95

Houston DX, Ferreira S, Collofello JS, Montgomery DC, Mackulak GT, Shunk DL (2001) Behavioral characterization: finding and using the influential factors in software process simulation models. J Syst Softw 59(3):259–270. https://doi.org/10.1016/S0164-1212(01)00067-X

Jørgensen M, Kitchenham B (2012) Interpretation problems related to the use of regression models to decide on economy of scale in software development. J Syst Softw 85(11):2494–2503

Kellner MI, Madachy RJ, Raffo DM (1999) Software process simulation modeling: Why? What? How? J Syst Softw 46:91–105

Kitchenham B (2010) What's up with software metrics? – a preliminary mapping study. J Syst Softw 83(1):37–51

Kleijnen JPC, Sanchez SM, Lucas TW, Cioppa TM (2005) State-of-the-art review: a user's guide to the brave new world of designing simulation experiments. INFORMS J Comput 17(3):263–289. https://doi.org/10.1287/ijoc.1050.0136

Law AM (2007) Simulation modeling and analysis, vol 4. McGraw-Hill, New York

Law AM, Kelton WD (2000) Simulation modeling and analysis, vol 3. McGraw-Hill, New York

Madachy RJ (2002) Simulation. Wiley, New York

Madachy RJ (2008) Software process dynamics. Wiley-IEEE Press, New York

Maria A (1997) Introduction to modeling and simulation. In: Proceedings of the 29th conference on Winter simulation, WSC, pp 7–13

McCall JA, Wong G, Stone A (1979) A simulation modeling approach to understanding the software development process. In: Fourth annual software engineering workshop. Goddard Space Flight Center, Greenbelt

Melis M, Turnu I, Cau A, Concas G (2006) Evaluating the impact of test-first programming and pair programming through software process simulation. Softw Process Improv Pract 11(4):345–360

Monks T, Currie CS, Onggo BS, Robinson S, Kunc M, Taylor SJ (2019) Strengthening the reporting of empirical simulation studies: introducing the stress guidelines. J Simul 13(1):55–67

Montgomery DC (2017) Design and analysis of experiments. Wiley, Hoboken

Müller M, Pfahl D (2008) Simulation methods. In: Shull F, Singer J, Sjøberg DIK (eds) Guide to advanced empirical software engineering. Springer, London, pp 117–152

Münch J, Rombach D, Rus I (2003) Creating an advanced software engineering laboratory by combining empirical studies with process simulation. In: Proceedings of the international workshop on software process simulation and modeling (ProSim), pp 3–4

Murphy S, Perera T (2001) Key enablers in the development of simulation. In: Proceedings of the Winter simulation conference, pp 1429–1437

Olsen NC (1993) The software rush hour (software engineering). IEEE Softw 10(5):29–37

Ören TI (1981) Concepts and criteria to assess acceptability of simulation studies: a frame of reference. Commun ACM 24(4):180–189

Park S, Kim H, Kang D, Bae DH (2008) Developing a software process simulation model using SPEM and analytical models. Int J Simul Process Model 4(3–4):223–236. https://doi.org/10.1504/IJSPM.2008.023684

Petersen K, Gencel C (2013) Worldviews, research methods, and their relationship to validity in empirical software engineering research. In: 2013 Joint conference of the 23rd international workshop on software measurement and the 8th international conference on software process and product measurement. IEEE, Piscataway, pp 81–89

Petersen K, Wohlin C (2009) Context in industrial software engineering research. In: Proceedings of the 3rd international symposium on empirical software engineering and measurement (ESEM), pp 401–404

Pfahl D (2001) An integrated approach to simulation based learning in support of strategic and project management in software organisations. Fraunhofer-IRB-Verlag, Stuttgart

Pfahl D (2014) Process simulation: a tool for software project managers? In: Software project management in a changing world. Springer, Berlin, pp 425–446

Pfahl D, Lebsanft K (1999) Integration of system dynamics modelling with descriptive process modelling and goal-oriented measurement. J Syst Softw 46(2):135–150. https://doi.org/10.1016/S0164-1212(99)00007-2

Pfahl D, Lebsanft K (2000) Knowledge acquisition and process guidance for building system dynamics simulation models: an experience report from software industry. Int J Softw Eng Knowl Eng 10(04):487–510

Pfahl D, Ruhe G (2002) IMMoS: a methodology for integrated measurement, modelling and simulation. Softw Process Improv Practice 7(3–4):189–210

Pfahl D, Stupperich M, Krivobokova T (2004) PL-SIM: a generic simulation model for studying strategic SPI in the automotive industry. In: IET conference proceedings. Citeseer, pp 149–158. https://doi.org/10.1049/ic:20040454

Raffo DM (2005) Software project management using prompt: a hybrid metrics, modeling and utility framework. Inf Softw Technol 47(15):1009–1017

Raffo DM, Kellner MI (2000) Empirical analysis in software process simulation modeling. J Syst Softw 53(1):31–41. https://doi.org/10.1016/S0164-1212(00)00006-6

Rus I, Neu H, Münch J (2003) A systematic methodology for developing discrete event simulation models of software development processes. In: Proceedings of the 4th international workshop on software process simulation and modeling (ProSim 2003), Portland, May 3–4

Sargent R (1999) Validation and verification of simulation models. In: WSC'99. 1999 Winter simulation conference proceedings. 'Simulation – a bridge to the future' (Cat. No. 99CH37038), vol 1. IEEE, Piscataway, pp 39–48

Shannon RE (1986) Intelligent simulation environments. In: Proceedings of the conference on intelligent simulation environments, pp 150–156

Travassos GH, Barros MO (2003) Contributions of in virtuo and in silico experiments for the future of empirical studies in software engineering. In: 2nd Workshop on empirical software engineering the future of empirical studies in software engineering, pp 117–130

Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2012) Experimentation in software engineering. Springer, New York

Wu M, Yan H (2009) Simulation in software engineering with system dynamics: a case study. J Softw 4(10):1127–1135

Zhang H (2012) Special panel: software process simulation – at a crossroads? In: Proceedings of the international conference on software and system process (ICSSP). IEEE, Piscataway, pp 215–216

Zhang H, Kitchenham B, Pfahl D (2010) Software process simulation modeling: an extended systematic review. In: Proceedings of the international conference on software process (ICSP). Springer, Berlin, pp 309–320

Zhang H, Jeffery R, Houston D, Huang L, Zhu L (2011) Impact of process simulation on software practice: an initial report. In: Proceedings of the 33rd international conference on software engineering (ICSE), pp 1046–1056