Computer Standards & Interfaces 70 (2020) 103425

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Software Process Simulation Modeling: Systematic literature review

J.A. García-García[a,*], J.G. Enríquez[a], M. Ruiz[b], C. Arévalo[a], A. Jiménez-Ramírez[a]

[a] Escuela Técnica Superior de Ingeniería Informática, Web Engineering and Early Testing (IWT2) Group, University of Seville, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain

[b] Escuela Superior de Ingeniería, Software Process Improvement and Formal Methods (SPI&FM) Group, University of Cádiz, Avda. de la Universidad de Cádiz, 10, 11519 Puerto Real (Cádiz), Spain

ARTICLE INFO

Keywords:
Software process
Software engineering
Systematic literature review
Software Process Simulation Modeling
SPSM

ABSTRACT

Changes and continuous progress in logistics and productive systems make the realization of improvements in decision making necessary. Simulation is a good support tool for this type of decisions because it allows reproducing processes virtually to study their behavior, to analyze the impact of possible changes or to compare different design alternatives without the high cost of scale experiments. Although process simulation is usually focused on industrial processes, over the last two decades, new proposals have emerged to bring simulation techniques into software engineering. This paper describes a Systematic Literature Review (SLR) which returned 8070 papers (published from 2013 to 2019) by a systematic search in 4 digital libraries. After conducting this SLR, 36 Software Process Simulation Modeling (SPSM) works were selected as primary studies and were documented following a specific characterization scheme. This scheme allows characterizing each proposal according to the paradigm used and its technology base as well as its future line of work. Our purpose is to identify trends and directions for future research on SPSM after identifying and studying which proposals in this topic have been defined and the relationships and dependencies between these proposals in the last five years. After finishing this review, it is possible to conclude that SPSM continues to be a topic that is very much addressed by the scientific community, but each contribution has been proposed with particular goals. This review also concludes that Agent-Based Simulation and System Dynamics paradigm is increasing and decreasing, respectively, its trend among SPSM proposals in the last five years. Regarding Discrete-Event Simulation paradigm, it seems that it is strengthening its position among research community in recent years to design new approaches.

1. Introduction

In recent years, the application of process engineering principles to different environments as a basis for increasing the quality and excellence in organizations is worldwide accepted. In this sense, general reference standards [1], international project management guidelines [2,3] and reference standards in the Information and Communications Technology (ICT) context [4,5] promote that formal process management allows improving effectiveness and efficiency. This enhancement allows reducing costs, improving quality, and increasing the productivity and competitiveness of any organization.

This approach is well known as Business Process Management (BPM) [6], and its application can provide many advantages (e.g. faster time-to-market, higher productivity and efficiency, better product quality and reduced product cost, among others) [7]. BPM can be and has been successfully applied to different companies and software organizations are not an exception.

However, some difficulties in the software engineering context have been identified. For instance, Canfora compares Software Processes (SPs) with industrial processes [8]. This author identifies a set of features of SPs (e.g., SPs frequently evolve incorporating new lifecycles, SPs are complex and strongly influenced by unpredictable circumstances such as human work, SPs often need integration between frameworks and different technologies, etc.) and argue that BPM is usually applied by software companies to formally model their processes using formal notations. However, other aspects of BPM, such as the automation of SPs, are not usually approached because many software companies consider that these aspects include highly-complex tasks which could not be effectively automated with reduced costs [9]. In this context, the lack of automation of the process implies that each

* Corresponding author.
E-mail addresses: juliangg@us.es (J.A. García-García), jgenriquez@us.es (J.G. Enríquez), mercedes.ruiz@uca.es (M. Ruiz), carlosarevalo@us.es (C. Arévalo), ajramirez@us.es (A. Jiménez-Ramírez).

https://doi.org/10.1016/j.csi.2020.103425
Received 18 August 2019; Received in revised form 23 November 2019; Accepted 9 January 2020
Available online 11 January 2020
0920-5489/ © 2020 Elsevier B.V. All rights reserved.

involved human role or work team perform the process execution manually and/or unilaterally [8]. This situation causes that monitoring, user-oriented execution, maintenance, and measurement of SPs could become difficult and expensive tasks.

Considering this complex scenario, it should be very interesting to check software process models and predict their behavior in different situations to reduce the risk of their deployments (into production environments) and subsequent costs as much as possible. Although different approaches exist to deal with complex processes [10], even pursuing multiple objectives beyond risk [11], using simulation techniques on processes is one of the well-known techniques to achieve that [12].

Simulation techniques help to understand and analyze processes for strategic management process improvement, forecasting or prediction, among others. The ultimate goal is being able of quantitatively estimate the impact of process design on its own performance. This research challenge has been traditionally addressed under the topic named Software Process Simulation Modeling (SPSM). This concept is originally introduced by Tarek Abdel-Hamid and Stuart Madnick who summarized their approach in their seminal book [13]. One of the most important factors to succeed in applying SPSM is to define a rigorous model that reliably represents the real system being modeled. For this reason, SPSM has been frequently perceived by companies as an “additional” cost to software projects slowing its wide adoption in the software business world. However, the number of publications related to this topic has been growing exponentially since [13], because simulation helps assess solutions to real situations in a virtual world, giving the opportunity to know the consequences within a risk-free environment.

As mentioned before, SPSM is a topic that has been studied by the community since 1980. In fact, over the last decades, multiple contributions have emerged in the research literature on SPSM and each of them has its own particular proposal. There are different reviews and surveys on SPSM [14–22], but it has not been possible to find current and formal reviews that provide a complete view on SPSM of the last five years. In addition, previous reviews do not categorize SPSM proposals under study according to their nature or Knowledge Area (KAs) within the field of Software Engineering (SE).

This paper aims to fill this gap by performing a systematic review to identify trends and directions for future research on SPSM after studying which proposals in this topic have been defined and the relationships and dependencies between these proposals in the last five years. Therefore, this Systematic Literature Reviews (SLR) contributes in four ways to ongoing research in the field of SPSM: (i) showing and reviewing the most widely common SPS paradigms that have been used from 2013 to 2019, as well as the methods or techniques that assist the SPSM; (ii) summarizing which purposes that have been addressed by SPSM; (iii) creating a paradigm-based taxonomy to classify software simulation proposals; and (iv) offering trends for future research. All these aspects will be addressed in this paper following meticulously Kitchenham's method [23], as it is described in the following sections.

Finally, this paper is structured as follows. Related works are briefly described in Section 2. Later, Section 3 and Section 4 describe the method used for the systematic review and its planning, respectively. Once decided how the SLR is going to be performed, the review protocol whose results are presented in Section 5 was conducted. Section 6 offer analysis and discussions on these results and Section 7 presents conclusions, open issues and future works.

2. Related works

Software process simulation is an active research topic over the last decades. This research topic was mentioned for the first time at 1998 in «International Workshop on Software Process Simulation Modeling» (ProSim workshop). These authors describe their conclusions of «why, what and how» software process has to be simulated in [20], where they also presented an overview of SPSM to identify the goals and scopes of simulation on software processes. Guidance in choosing an appropriate modeling proposal are also provided in.

During the years, several SLRs and reviews about SPSM have been published in the scientific literature. Their results and conclusions are briefly presented below.

Liu et al. [21] perform a Systematic Mapping Study (SMS) of management proposals based on SPSM to evaluate risks of software processes. These authors present mainly the scope of the proposals under study and what tools have been used in primary studies. However, this work does not present information about all the steps of their SMS. For example, authors do not provide information on quality criteria during the selection of primary studies.

Zhang et al. [14] discuss the evolution of SPSM research from 1998 to 2007. This paper evaluates approximately 200 papers related to SPSM in order to identify their scopes, used paradigm, application domains, and relevant research challenges and needs. After performing this evaluation, authors summarize their conclusions as follows: (i) many papers are emerged to improve the efficiency and effectiveness of SPSM proposals; (ii) hybrid models are beginning to be used as paradigm in process simulation models because this paradigm allows successfully modeling the complexity of software processes; (iii) the paradigm most widely used is System Dynamics (SD) with 49% of the works using such paradigm, followed by the Discrete-event Simulation (DES) one, used in 31% of the studies.

Zhang et al. [15,16] extend their previous paper [14] to suggest the trends in SPSM (from 1998 to 2007). These ones are mainly: (i) SD and DES are the main used paradigms; (ii) most of new proposals improve the simulation research at the process level, which is the one that has been attracting more simulation research; and (iii) hybrid modeling (by combining SD and DES paradigms in a single simulation model) continues to be the most used simulation approach. Some years later, Zhang et al. [17] present a new extension of their previous paper [14]. The main difference is the inclusion of more data sources in their search strategy of the SLR, but they do not introduce new conclusions.

Moreover, Chao et al. present a SLR [18] which is focused on papers published between 2008 and 2012, and it reuses the same research questions and search strategy that previous reviews such as [14–17]. The authors conclude that the volume of new SPSM studies published regarding the review presented in [14] is neither relevant nor significant. For instance, SD and DES paradigms still being the most frequently used. In addition, hybrid modeling continues to be the main trend, because this paradigm allows capturing complex real-world features of the software processes.

A characterization scheme to classify Software Process Simulation Models is the object of study of [19]. This scheme incorporates features such as model verification procedures, type, data structure, output techniques and results of simulation. This paper differs of previous ones [19] because authors pay a much closer attention to the validation and verification of simulation models.

In [22], Ali et al. identify and evaluate the empirical evidence on the utility of Software Process Simulation (SPS) techniques when these ones are used in real industrial environments. The authors justify the importance of carrying out this review on the contradictory statements in the scientific literature about the usefulness of simulating software processes, and its practice in real environment. Simulation modeling techniques could be very useful techniques in software contexts, it is not often applied in the industrial yet. In addition, this review identifies the current trends in the area. Many papers describe proof-of-concept about process estimation or improvement, among other aspects, but these papers do not establish quality criteria in their evaluations. In addition, most of these papers do not provide objective evidence. In this context, the authors conclude that it is necessary to provide reports of objective studies of process simulation using formal experimentation techniques and methodologies.

Janssen et al. analyze advantages and disadvantages of several

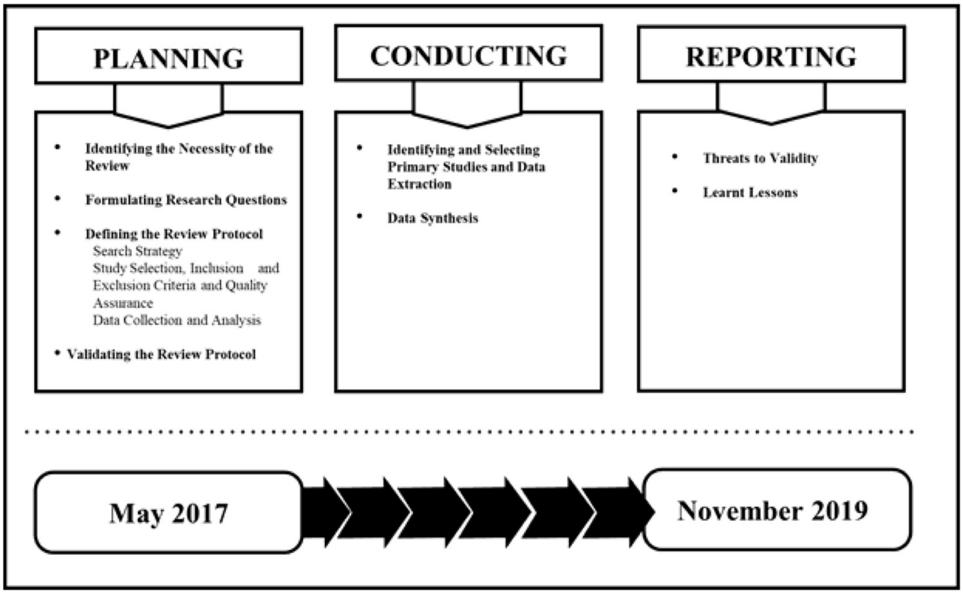


Fig. 1. Methodological review process and tasks.

process simulation tools [24]. The evaluated characteristics are related to process modeling, simulation technique, and analysis of results.

Finally, in this context and once the most relevant related works have been identified, we have identified two main research gaps: (i) lack of methodological rigor when the review is carried out because information on quality criteria, search protocol, etc., it is not provided in some reviews (this situation makes reproducibility of the study difficult); (ii) seniority of existing reviews (the most recent systematic review analyzes SPMS proposals that are published between 2008 and 2012). Both aspects are solved our systematic review. As we mentioned above, our SLR contributes in four ways: (i) showing and reviewing the most widely common SPS paradigms that have been used from 2013 to 2019 following meticulously the Kitchenham's method; (ii) summarizing which purposes that have been addressed by SPSM; (iii) creating a paradigm-based taxonomy; and (iv) offering trends for future research.

3. Review method

As mentioned above, we have followed Kitchenham's methodology [25] (which has been successfully applied in the software engineering context nuestra) to carry out our SLR. This methodology was updated over the years thanks to the fact that some authors have argued against it and/or presented improvements to Kitchenham's method. These aspects are mentioned below.

In [26], authors conclude the total number of systematic reviews has significantly increased over last decade. However, the Kitchenham's method presented in [25] cannot be considered relevant in software engineering because it does not often provide guidelines for professionals to measure the quality of primary studies. Other authors offer conclusions in this sense, such as Da Silva et al. [27]. These authors also mention that many SLR does not often asses the quality of primary studies, which could reduce the possible impact of these reviews.

Other weaknesses of Kitchenham's method are published in [23]. These criticisms are focused on the search strategy because it is a crucial task to successfully perform systematic reviews. Zhang et al. conclude that the probability of occurrence of mistakes is high in this task, so this one should be adequately and carefully planned and carried out. Zhang's paper aims to mainly improve the search strategy of Kitchenham's method presented in [28]. For this purpose, the authors define the «quasi-gold standard (QGS)» and «quasi-sensitivity» concepts in the search strategy. The first one means that papers under study should be known, and the second one means that it is important and convenient to evaluate the performance of the search strategy to carry out improvements on it. In 2013, Zhang et al. publish other paper where they highlight the importance of conducting literature reviews using systematic methods [29].

Zhang et al. consider especially relevant this aspect from the empirical view, because it is important to balance the necessary effort and the rigorous application of the systematic method. In this context, the «snowballing» concept is presented by Wohlin et al. [30] to extend the search strategy. Wohlin et al. propose to analyze the references, citations, related works, etc., of each primary study obtained after applying Kitchenham's method. Those references related to the research subject under study, should be included and considered in the systematic review.

After receiving these criticisms and aspects of improvement, Kitchenham et al. analyze these comments and publish an update of their method with several conclusions [25]. The authors conclude that the construction of search queries from structured questions is a non-relevant aspect. However, they consider it is interesting to include the QGS concept in their method. QGS limits manual search to set up search queries and improve the evaluation of the search strategy. In addition, it is possible to improve inclusion and exclusion criteria using textual analysis tools. These tools could also allow the definition of more exact search queries. Anyway, the evaluation of the search queries have to be carried out in a thorough and rigorous way to guarantee that the systematic search is reproducible. Finally, the authors admit that, at present, there is still an important gap related to the evaluation of quality in studies that use empirical methods. This aspect is being addressed as future work by these authors.

Considering this context, this paper follows the last method presented by Kitchenham and described in [25]. Three phases are described in this method to perform a systematic review: (i) planning, which confirms the necessity of the research, defines the review protocol, research questions (the most important activity) and decides how researchers should carry out the review; (ii) conducting, which executes the defined protocol previously; and (iii) reporting, which presents and discuss results. Fig. 1 shows these phases and their tasks along the time. Next sections describe in detail how each phase to achieve the research goal that this paper presents have been conducted.

4. Planning

Next sections describe in detail the planning process that has been carried out in this SLR. During this process, it has been identified the need for performing this review, the research questions have been formulated, and the review protocol has been defined and validated.

4.1. Identifying the necessity of the review

Software process simulation appeared in the early 1980s and it has been growing from then on. This means it is not a new necessity. Since SPSM helps companies anticipate the effects of their decision-making within a risk-free environment, it has become an important tool to avoid risks and prevent the unnecessary waste of resources and budget.

The SPSM problem have been evaluated to identify the need for this review. The existing literature reviews about frameworks, methodologies or techniques on SPSM have been considered, but it has not been possible to find recent reviews related to this domain, so it could be relevant to update the information related to SPSM because it has become really important for the software industry. In this context, this study has been conducted with various purposes or objectives.

The main objective of our SLR is to identify recent SPSM approaches and challenges to be achieved in the future. In addition, our SLR identifies the nature (that is, application in the industry, an academic prototype, etc.) of each SPSM approach, as well as the Knowledge Areas (Kas) within Software Engineering (SE) area where each SPSM approach is framed. For this purpose, ISO/IEC 19759:2015 [31] was considered because this standard groups the SoftWare Engineering BOdy of Knowledge (SWEBOK). In addition, this standard establishes the subset of KAs internationally accepted by software engineers.

4.2. Formulating research questions

In recent years, different proposals have been published to find the best way to model the software process simulations and represent the appropriate parameters in these simulations. Consideration of Research Questions (RQ) is appropriate to clearly focus our research and improve the understanding of SPSM proposals under study. In this context, the general RQ that guide this review is: *«What is the current state-of-the-art of proposals for the modeling of software process simulation?»*. To answer this general research question, we ask several more specific questions. Table 1 presents the RQs proposed together with their motivations.

4.3. Defining the review protocol

After identifying the background of the SPSM and formulating the RQs, it is necessary to define the review protocol of this review. This protocol contains information on search and data extraction strategies, criteria for selection of studies and measurement of their quality, data synthesis, and dissemination strategy.

Table 2
Keywords giving main terms.

A	B	C
A1. Software process	B1. Simulation	C1. Model
A2. Software project	B2. Simulate	C2. Modeling
A3. Software product		C3. Metamodel
A4. Software development		C4. Language

4.3.1. Search strategy

This section aims to describe the followed procedure to find the most relevant studies related to SPSM. This strategy is going to be focused on the search of journal articles and conference papers into different digital research libraries. In addition, we have considered a search strategy in two different stages.

On the one hand, we select keywords for the search in the pre-search phase to confirm that most of the keywords are included into each research paper under study. In this context, the use of appropriate keywords is relevant for the quality of results. Table 2 shows the list of keywords. General terms and synonyms were used to guarantee that majority of papers will be included.

On the other hand, once the pre-search phase was defined, the systematic searches were carried out in different scientific databases using combinations of keywords. Eq. (1) formalizes mathematically the boolean expression of keywords (Table 2) that was used in the searches.

$$E_1 = (\vee_{i=1}^{4} A_i) \wedge (\vee_{j=1}^{3} B_j) \wedge (\vee_{k=1}^{4} C_k)$$

Eq. (1). Boolean expression of keywords

Regarding databases, the criteria presented by Ngai et al. [32] was considered. Authors propose as relevant scientific databases: Web of Knowledge, ACM Library, ABI/INFORM Database, Business Source Premier, Emerald Full text, IEEE Xplore, Academic Search Premier, Science Direct, Springer Link Journals, World Scientific Net, and ScienceDirect. However, after finishing the preliminary searches on these databases, it was detected that many papers returned by some databases are papers already found in other ones, so they did not add new value. Considering this, finally, it was decided to use the following databases to carry out the systematic review: ACM Library, IEEE Xplore Library ScienceDirect, and Springer Link. The management of references is done using a Microsoft Excel spreadsheet and the tool Jabref [33]. Both tools help to manage the papers under study and carry out the systematic search properly.

Moreover, it was applied the search expression shown in Eq. (1) in each scientific database. The application of this equation is performed on title–abstract–keyword metadata of each paper. Eq. (2) formalizes the application of the boolean expression of keywords (Eq. (1)) over the mentioned metadata.

Table 1
Research questions.

Research questions	Motivation
RQ1. What are the objectives and motivations studied in the SPSM proposals that have been published over the last five years?	The motivation of this RQ is to: (1) find the proposals that have been researched and published in the SPSM topic in last five years; (2) identify their objectives and motivations.
RQ2. What have been the most used simulation paradigms (or combination of paradigms) to define SPSM proposals over the last 5 years?	This RQ aims to identify the most popular simulation paradigms applied by researchers. In addition, the answer to this question allows establishing trends on the most used paradigms.
RQ3. What decisions does the simulation model help to make?	The motivation of this RQ is to determine what are the approaches that simulation helps to decide (decision-making).
RQ4. What is the KA of SE where each SPSM proposal is framed? What is its nature of each proposal?	This RQ aims to know what is the KA within SE where each SPSM approach is framed. We consider the areas included in SWEBOK to know this feature.[a] This RQ also aims identify the nature of each proposal.[b]
RQ5. What scientific validation methods have been applied in different proposals?	The motivation of this RQ is to identify the empirical validation method that has been used by the different primary studies.

[a] SWEBOK includes 15 KAs within SE: Software requirements, Software design, Software construction, Software testing, Software maintenance, Software configuration management, Software engineering management, Software engineering process, Software engineering models and methods, Software quality, Software engineering professional practice, Software engineering economics, Computing foundations, Mathematical foundations, and Engineering foundations.

[b] The following natures have been considered: Application in Industry (AI), Academic Prototype (AP) or Theoretical Model (TM).

Table 3
Search queries.

Digital library	Query
ACM	"query": {acmdlTitle:("Software Process" "software project" "software product" "software development") AND acmdlTitle:("Simulation" "Simulate") AND acmdlTitle:("Model" "Modeling" "Metamodel" "Language")} "query": {recordAbstract:("Software Process" "software project" "software product" "software development") AND recordAbstract:("Simulation" "Simulate") AND recordAbstract:("Model" "Modeling" "Metamodel" "Language")} "query": {keywords.author.keyword:("Software Process" "software project" "software product" "software development") AND keywords.author.keyword: ("Simulation" "Simulate") AND keywords.author.keyword:("Model" "Modeling" "Metamodel" "Language"}
IEEE explore	((("Document Title":"software process") OR ("Document Title":"software project") OR ("Document Title":"software product") OR ("Document Title":"software development")) AND (("Document Title":"simulation") OR ("Document Title":"simulate")) AND (("Document Title":"model") OR ("Document Title":"modeling") OR ("Document Title":"metamodel") OR ("Document Title":"language"))) ((("Abstract":"software process") OR ("Abstract":"software project") OR ("Abstract":"software product") OR ("Abstract":"software development")) AND (("Abstract":"simulation") OR ("Abstract":"simulate")) AND (("Abstract":"model") OR ("Abstract":"modeling") OR ("Abstract":"metamodel") OR ("Abstract":"language"))) ((("Author Keywords":"software process") OR ("Author Keywords":"software project") OR ("Author Keywords":"software product") OR ("Author Keywords":"software development")) AND (("Author Keywords":"simulation") OR ("Author Keywords":"simulate")) AND (("Author Keywords":"model") OR ("Author Keywords":"modeling") OR ("Author Keywords":"metamodel") OR ("Author Keywords":"language")))
Science direct	Title, abstract, keywords: ("software process" OR "software project" OR "software product" OR "software development") AND ("simulation" OR "simulate") AND ("model" OR "modeling" OR "metamodel" OR "language")
Springer link	"software process" AND ("simulation" OR "simulate") AND ("model" OR "modeling" OR "metamodel" OR "language") "software project" AND ("simulation" OR "simulate") AND ("modenl" OR "modeling" OR "metamodel" OR "language") "software product" AND ("simulation" OR "simulate") AND ("model" OR "modeling" OR "metamodel" OR "language") "software development" AND ("simulation" OR "simulate") AND ("model" OR "modeling" OR "metamodel" OR "language")

$$E_2 = title(E_1) \vee abstract(E_1) \vee keyword(E_1)$$

Eq. (2). Boolean expression on metadata of a paper

However, it is important to mention that each database has its own syntax to represent custom searches (based on logical expressions) on metadata of each paper. In addition, each database also has limitations on the number of logical clauses that can be applied in the same search. For this reason, the application of Eq. (2) has been divided into several queries.

Table 3 presents each search query that we have executed in all databases mentioned previously. For example, three queries in IEEE Xplore were executed, because the advanced search engine of this digital library does not allow indicating search expressions with more than 15 search terms. Regarding Springer Link, although this digital library allows executing the complete search query, it is not possible to export the results later because of limitations of this database. For this reason, the main query was divided into four secondary queries.

Moreover, it is relevant to mention that an important number of search queries were executed on each database and it was necessary to adapt each query to filters offered by each database. In addition, it was considered to apply other filters criteria (e.g., year of publication, scientific area, specific topic, etc.).

Finally, the systematic search was complemented applying the «snowball» technique [30], which proposes to expand the search process considering the reference lists of each paper under study as well as the citations of these papers. Section 5.1 describes in detail the execution of our search strategy and presents results of «snowball» technique.

4.3.2. Strategy for the selection of primary studies

The selection process of studies aims to define how relevant papers are identified, analyzed and considered in the review according to the objectives of our SLR. This process has been executed by two researchers: a senior researcher, who is the leader of the systematic review, and a junior researcher. Table 4 summarizes each phase that has been carried out in selection process of papers under study.

The first phase (P1) aims to automatically execute all search queries (Table 3) on each scientific database. Once this result is obtained, the first filter is applied in the second phase (P2). Specifically, this filter is based on excluding primary studies considering its online publication date (see Table 5). Also, the power of some scientific databases was harnessed to filter by research disciplines and kind of publication. Specifically, it was selected (when possible) «Computer Science», «Engineering and «Business and Management» disciplines because these group research areas that are appropriate for this review. In addition, it was filtered by «Software Engineering», «Management of Computing and Information Systems», «Simulation and Modeling», «IT in Business» and «Information Systems Applications» subdisciplines (in each discipline) into SpringerLink database and kind of publication (research articles, and book chapters) in ScienceDirect. These first phases are executed by the leading researcher of this paper.

After filtering for publication date, the two senior researchers analyzed each primary study to exclude the ones that were not related to the topic of this review. During this exclusion phase (P3), each researcher analyzed and considered the title, keywords and abstract of each initial primary study. However, some doubts may appear during the third phase of the selection method. For this reason, two face-to-face meetings between researchers were included in the study selection. These meetings allowed the researchers to jointly discuss and agree on those studies that were relevant to the systematic review. These face-to-face meetings allow us to minimize the bias of each researcher.

On the one hand, the first meeting (P4) received as input the result of third phase and its objective was to jointly select the semifinal primary studies. In addition, sometimes it was necessary to completely read a primary study when there were doubts after applying some inclusion/exclusion criteria. For this reason, researchers could decide (always jointly) to automatically exclude that primary study if it was not relevant to the review topic. It is important for these decisions to be agreed by more than one researcher to avoid subjective decisions.

On the other hand, the second meeting (P6) was also the last phase of the selection process. This meeting was carried out after performing the fifth phase (P5), where all the researchers applied the «snowball» technique on the semifinal primary studies. As mentioned above, these semifinal primary studies are the output of P4. The second meeting also enabled us to avoid subjective decisions when researchers apply the «snowball» technique.

Moreover, regarding inclusion/exclusion criteria, the papers under study had to meet some criteria such as: they had to be written in English, published from 2013 to 2018 in either well-reputed journals, such as the one indexed in JCR index (Journal Citation Reports) or in proceedings of conferences categorized in CORE Conference Ranking (in this case, consider only conferences with level A*, A and B within this ranking were considered). Discussion, surveys, reviews or opinion papers were excluded as well as abstract formats, duplicates and those whose main contribution is not related to SPSM. Table 5 summarizes

Table 4
Phases of the study selection.

Phase	Description	Participants
P1	Selection of primary studies based on automatic search in titles, abstracts and keywords.	Principal researcher
P2	Exclusion of primary studies based on publication date and research disciplines.	Principal researcher
P3	Exclusion of primary studies based on titles, abstracts and keywords as well as duplicated.	All researches
P4	Consensus meeting.	All researches
P5	Application of «snowball» technique to include primary studies based on full text.	All researches
P6	Consensus meeting.	All researches

the criteria that was defined in the review protocol. These criteria have been grouped according to the phase of the selection process in which the criterion is applied.

4.3.3. Quality questions

A questionnaire was proposed to objectively assess the quality of each paper under study. This questionnaire is going to be filled in for each selected paper in the conducting section. Each question has three possible answers to be selected: «Yes», «No» or «Partially». The quality questionnaire that was applied in this systematic review is shown in Table 6. The cumulative score for each criterion will make up the final quality score for the study in question. These scores are not used to exclude papers from the SLR, but they will make it possible to determine which studies are the most relevant and representative for consideration in future research.

4.3.4. Data schema

The information contained in each paper can be very varied and analyzing this information could be a tedious task. In this context, a data collection schema was defined (Table 7) to facilitate this task. At the beginning, this scheme is completed by all researchers after analyzing each paper under study. Once the initial version of the scheme has been completed, all researchers have discussed and analyzed each dubious issue. The result is the final data collection scheme.

4.3.5. Verifying the adequacy of our review protocol

As Kitchenham et al. recommends [25], the systematic review protocol should be reviewed to refine it and adequately achieve the objectives of the SLR. In this context, before formally executing the search strategy, random searches were carried out to refine and adjust the keywords; search chains and exclusion criteria. The protocol was also reviewed by a Full professor in Software Engineering at University of Seville (Spain) who is expert in SLR. Her observations led to some revisions to the protocol.

5. Conducting and quality results

Once the review protocol has been approved by all researchers, our review is carried out. For this purpose, firstly, the result of the selection process and data extraction are presented in Section 5.1. Also, a statistical study is performed. Later, Section 5.2 shows results after applying the quality questionnaire (Table 6) on each selected primary study.

5.1. Data extraction and selection of primary studies

As mentioned above, some scientific databases have limitations to apply complex search queries. This situation has made it necessary to design several specific search queries in each database. For instance, three queries in ACM digital library was executed (i.e., ACM1, ACM2 and ACM 3 as Table 3 shows) because the advanced search engine of this digital library does not allow indicating complex search expressions. Once executed these queries, their outcomes have been manipulated by the leader researcher to exclude duplicated primary studies. In addition, some phases of the selection protocol of candidate studies have been executed in several stages. This decision has been important for the execution of the revision protocol because it allows visualizing which is the result of the application of main exclusion criteria in each stage.

Fig. 2 summarizes the volume of primary studies that were obtained after applying each selection process and inclusion/exclusion criteria. In addition, Fig. 3 allows to visualize what the evolution of the search protocol has been along three main milestones.

On the one hand, 41,520 candidate primary studies in total have been found after completing the first phase (P1) of the selection process (Table 4) as Fig. 2 shows. These candidate studies have been obtained by executing the search queries shown in Table 3 on each digital library. Later, the second phase (P2) is conducted and 8070 papers are resulted from all digital libraries. In this phase, exclusion criteria such as publication date after 2013, papers framed into «Computer Science», «Engineering and «Business and Management» disciplines, etc., are applied by the leading researcher. The second phase (P2) is considered the first main milestone of the search protocol. This first milestone is related to the first series of data in Fig. 2. This series shows papers that are retrieved from all search engine after executing each query.

On the other hand, the third phase (P3) of the protocol has been executed in three stages (as Fig. 2 shows) due to the large volume of studies found and analyzed. As mentioned before, this choice allows

Table 5
Exclusion and inclusion criteria per phase.

Phase	Exclusion and inclusion criteria
P1	In this phase, the automatic search in each scientific database is carried out. Therefore, the inclusion criteria are that papers must contain the search query (as shown in Table 3).
P2	Only English; Full text obtained; Publication date after 2013; Papers framed into «Computer Science», «Engineering and «Business and Management» disciplines (as well as «Software Engineering», «Management of Computing and Information Systems», «Simulation and Modeling», «IT in Business» and «Information Systems Applications» subdisciplines) are included.
P3	In this phase, duplicated, survey, comparative study, review, discussion, tutorial, panel or opinion papers as well as abstract formats are excluded. In addition, papers that are not related to SPSM and papers with the following scope will also be excluded of this study: papers whose objective is to apply methods or procedures of SE to improve the definition or testing of simulation models, without providing specific applications in the context of SE.
P4	In this phase (1st meeting) no new exclusion/inclusion criteria are applied, but relevant papers to the SPSM problem are included.
P5	As mentioned above, this phase aims to apply the «snowball» technique. In this sense, it is necessary to re-apply criteria such as: Publication date after 2013; Only English; Full text obtained.
P6	In this phase (2nd meeting) no new exclusion/inclusion criteria are applied, but it is important to exclude duplicated papers.

Table 6
Quality questionnaire.

Code	Quality question	Answers
QA1	Does the proposal under study present related works?	**Yes, it does (+1).** The proposal compares previous related works in SPSM; **Partially (+ 0,5).** The proposal only mentions few previous related works and it does not establish a clear background for the topic; No, it doesn't (+0). Related works are not mentioned.
QA2	Is the proposal under study an evolution of another proposal and does it explain or justify this evolution?	**Yes, it does (+ 0,75).** The proposal is evolution of another proposal and it explains which its starting point is and why; **Partially (+ 0,5)**; The proposal under study mentions on its origin proposal, but it does not explain why it has been proposed; No, it doesn't (+1). The proposal under study is not an evolution of any other proposal, i.e., it is a new proposal.
QA3	Does the proposal under study propose concrete lines of future research?	**Yes, it does (+1).** It shows continuous research of its investigation; **Partially (+ 0,5).** The proposal proposes only possible future research on itself; **No, it doesn't (+0).** Future researches are not presented

visualizing results after applying the exclusion criteria step-by-step in P3. Firstly, 2807 duplicate candidate primary studies have been eliminated in stage P3.1. Indeed, P3.1 corresponds to the second major milestone in the search protocol. Fig. 3 represents this milestone as the second series of data and means the number of research papers that have been picked up from all search engine after deleting duplicated papers between digital libraries. Secondly, SLRs, opinion papers, comparative studies, SMSs, survey, etc., are also removed in P3.2. Later, the third stage (P3.3) aims to delete studies which are not related to SPSM. Then, 82 candidate primary studies are returned after executing the phase P3 of the selection protocol.

However, some researchers had doubts about the relationship of some of these candidate studies with the SPSM topic. In this sense, 49 studies are eliminated in the fourth phase (P4) after solving all doubts or discrepancies found by us. The result of this phase is 33 candidate studies as Fig. 2 shows. This phase is considered the third milestone of the search protocol and is represented in Fig. 3 as the papers that are finally included in our review after applying the inclusion criteria.

Once fourth phase is completed, the «snowball» technique is applied on semifinal primary studies during P5 and P6. After carrying out both phases, 3 new studies are included in the review. Finally, it resulted 36 primary studies to be analyzed.

Moreover, Fig. 4 presents two data series. The first one shows the distribution of primary studies (finally included in our SLR and returned from each digital library) divided by the number of primary studies selected from all digital libraries. It is possible to observe that SpringerLink provides more than 33% of primary studies. The second serie of Fig. 4 shows the distribution of primary studies (finally included in our analysis and returned from each digital library) divided by all different primary studies returned from the same digital library. This series of data shows that majority of digital libraries contain approximately 10% of primary studies. It is important to mention that we have considered (in Figs. 4 and 5) primary studies found is several digital libraries to decriminalize all digital libraries.

Finally, the selection process has been conducted across the six phases previously presented, whereby the number of primary studies decreased. Fig. 5 and Table 8 show the distribution of primary studies that have been excluded and included in each phase. It is important to remember that the fifth and sixth phases of the selection process study are related to the application of the «snowball» technique. Therefore, the result of these phases must be accumulated to the result of the fourth phase to obtain the number of studies that are finally included in this review. Finally, it could be relevant to mention that the primary studies included in the review are referenced in the last section of this paper but excluded studies have not been referenced due to space constraints.

5.2. Quality evaluation

The studies included in this SLR have been evaluated considering the questionnaire shows in Table 6 (Section 4.3.3). This questionnaire aims to objectively measure the degree of representativeness of each study what could help getting relevant conclusions. Once each quality questions are answered, the results will be discussed in order to identify the coverage degree of each question.

Fig. 6 shows the coverage of each quality question for the studies included. On the one hand, observing Fig. 6, most of research papers under study are not evolution of another proposal but are original proposals. This statement has been obtained after observing the coverage of more than 82 % of the answer «No» of QA2. On the other hand, it is also interesting to note that most articles studied only briefly mention some line of future work, but clear directions are not provided. This statement has been obtained after observing the coverage of more than 53 % of the answer «Partially» of QA3. Finally, a relevant handicap was detected in the description of the proposals under study. Specifically, 50% of the proposals are not compared with other related works (i.e., similar proposals) adequately or related works are not presented in the article. This conclusion is obtained after observing the percentage of «Partially» or «No» answers of QA1.

5.3. Threats in the validation

There are threats in the validation process of this work, because the search and processing tasks have been carried out by people. The selection of papers, as well as data extraction, may be inaccurate and errors in the process of classification may exist, due to the human factor. However, to minimize these risks, the process has been performed by subsequent reviews that confirm that the entire process has

Table 7
Data schema.

Characteristic	Description
Kind of publication	This information refers to research event (journal, conference or workshop) where the SPSM approach has been published.
Date publication	This information refers to the year of publication when the SPSM approach has been published.
Simulation paradigm	It refers to the simulation paradigm on which the SPSM approach is based. In this sense, variety of simulation paradigm in the literature are identified: Hybrid Simulation (HS), State-Based Simulation (SBS), Queuing models (QM), Discrete-Event Simulation (DES), System Dynamics (SD), Agent-Based Simulation (ABS), Petri-Net Models (PNM), Simulation-Based Teaching (SBT), and Monte Carlo Simulation (MCS) [34].
Description and motivation	This information refers to a brief description of each proposal under study and the motivation associated with its publication.
Simulation input parameters	It means the kind of input parameters which are going to be applied to evaluate and simulate the software process.
Nature and KA	This information refers to the nature and the KA within SE where each SPSM approach is framed. The possible values that these properties can take are defined in Table 1 (RQ4).

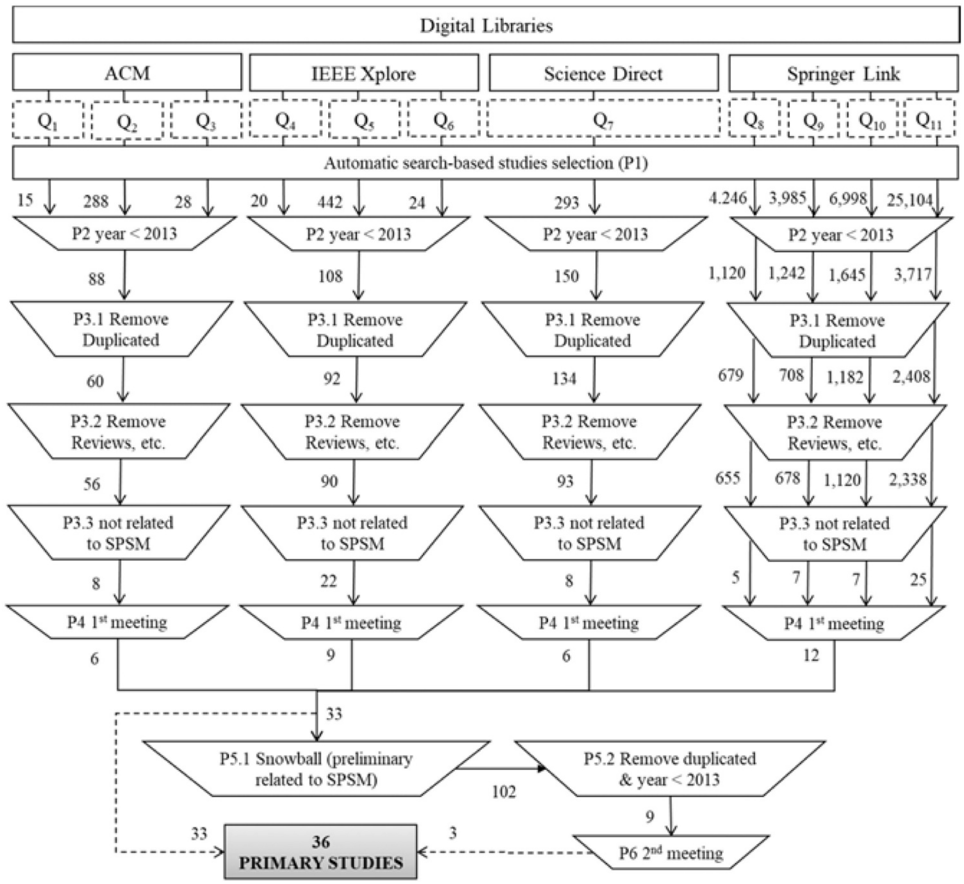


Fig. 2. Results during selection process.

been developed under the established criteria.

On the one hand, despite being an exhaustive process, it is impossible to guarantee a full coverage of every work published about a particular topic, and there may be papers not indexed or grey literature not included in this research. On the other hand, a study by Schmucker et al. [35] states that in very few occasions this type of publications has relevance in the final results of the systematic reviews. For this purpose, the search terms were used in 4 online libraries databases for this research. The scope of these journals databases covers a wide range of topics. These databases provide coverage large enough to achieve reasonable reliability for the field on which this study has been focused.

To achieve an objective miscellany process, the research questions have been defined before the choice, and the primary studies selection has been lined in a series of stages involving all authors applying rules defined in the review protocol.

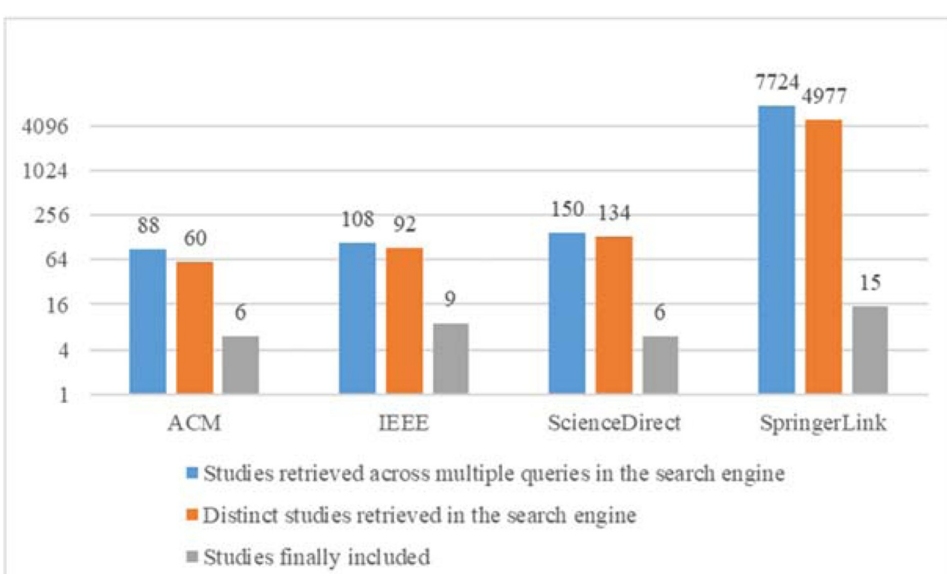


Fig. 3. Studies retrieved through search engines.

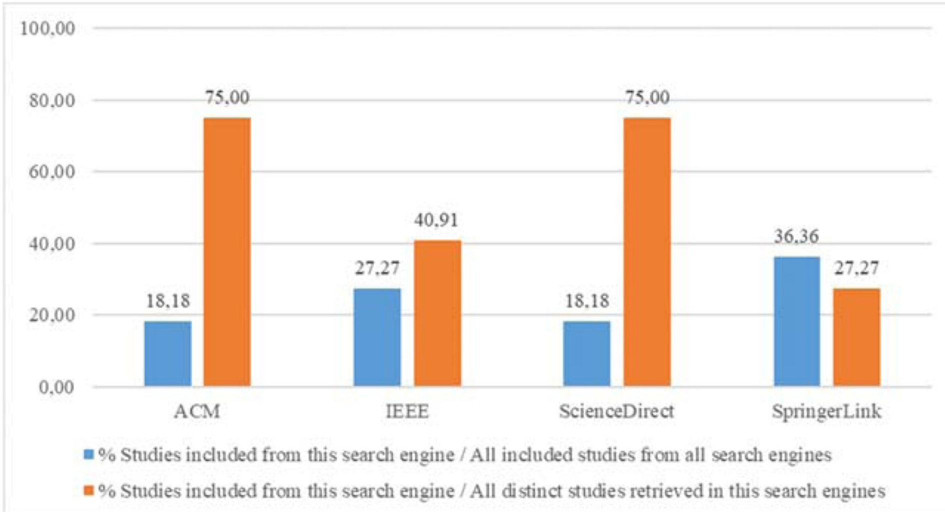


Fig. 4. Distribution of primary studies from digital libraries regarding the total of primary studies.

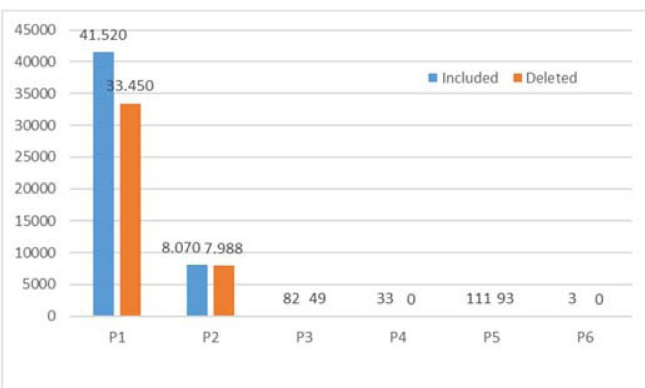


Fig. 5. Distribution of deleted and included primary studies per phase.

Table 8
Distribution of excluded and included primary studies per phase.

Phase	Included primary studies	Excluded primary studies
P1	41,520	33,450
P2	8070	7988
P3	82	49
P4	33	0
P5	111	93
P6	3	0

6. Analysis, open issues and discussion

In this section, each research question of our systematic review is answered and discussed in detail to detect the weaknesses of each primary study.

6.1. RQ1. «What are the objectives and motivations studied in the SPSM proposals that have been published over the last five years?»

Table 9 presents a structured description of each proposal under study (Primary studies). This description is based on the characterization scheme presented in Table 9, which also shows the quality measurement scores for each primary study after applying of all quality criteria on to each paper (as mentioned in Section 4.3.3). It is important to mention that this quality measurement has not been used to reduce the number of papers, but rather to identify the most important ones for

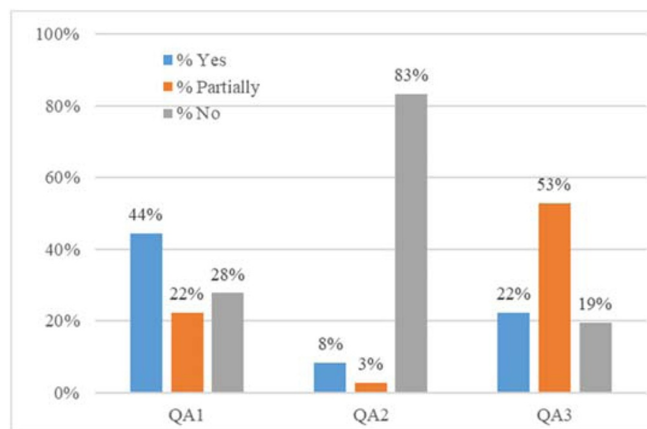


Fig. 6. Distribution of primary studies according to results of the quality questionnaire.

future research.

6.2. RQ2. «What have been the most used simulation paradigms (or combination of paradigms) as basis to define SPSM proposals on the last 5 years?»

Regarding this research question and considering Table 9, it was noticed that Agent-Based Simulation (ABS) and Discrete-Event Simulation (DES) are most used simulation paradigms. Fig. 7 shows most primary studies of the last five years are based on the DES paradigm (28 %) followed by the ABS paradigm (25 %), what means that both are equally the most widely used simulation paradigms in SPSM. On the opposite side, some simulation paradigms have been little used (QM, SBS, MCS; 3%) to define SPSM proposals in the last five years.

After analyzing these results, it could be interesting to mention a fact that was obtained. The Hybrid Simulation (HS) paradigm has been a paradigm widely used in past years [17,20] (the combination of SD and DES was the most common in HS). However, after executing this SLR, it has been detected that the HS paradigm has suffered a decrement. As Fig. 7 shows, only 2 studies based on HS and combining «DES, ABS, SD» paradigms [36] and «ABS, SD» paradigms [39], respectively, have been found. These two studies represent 6% of the total papers analyzed.

Finally, another interesting conclusion could be obtained from the study. Fig. 8 shows how many times the most used paradigms (DES, HS,

Table 9
Description of SPSM proposals.

Year	SPSM [ref.]	Scores (max.3)	Sim. approach	Type of publ.	KA;nature	Description and motivation
2018	[36]	1,50	HS (DES + ABS + SD)	Workshop	Software engine. management; TM	Author presents a hybrid simulation model to address the risk of task failure in software development process relate to competitive crowdsourcing platforms. This process is based on the development of mini-tasks that are developed by online volunteer workers as suppliers, which could raise uncertainty, failure and possible inefficiency about the service received.
2017	[37]	3,00	DES	Workshop	Software quality; AP	To cover the conceptual leap when going from the system model to the simulation implementation, authors present a tool for providing a visual representation of the scheduling and execution of events in a discrete event simulation. This proposal allows to improve the quality of the simulation models and their implementation since it provides visual tools to observe the execution and planning of events during simulation executions.
2016	[38]	2,00	ABS	Workshop	Software quality; TM	Authors present an agent-based model to simulate software processes with different types of developers. This proposal is motivated to offer a quality trend of software projects that are similar in size and duration using the simulation of parameters such as effort and size (number of tasks) of the project.
2015	[39]	2,00	HS (ABS + SD)	Conference	Software engine. management; TM	Authors propose combining agent-based and system dynamics models to simulate software development process. This simulation model evaluates the relationship between workload, management approach, individual performance, stress, attitude, and team performance, among other factors.
2013	[40]	2,00	SD	Conference	Software quality; TM	Authors present a SD model to analyze 5 techniques (Ad Hoc, Checklist, Perspective, Scenario, N-fold) that can be applied to analyze the inspection phase in software projects. The model allows estimating the total number of defects from early stages of software development, as well as the effectiveness and cost of inspection activity. The parameters evaluated are related to number of pages, average number of defects per page, average number of defects detected by inspector, number of pages per inspection, number of inspections, process maturity level CMM level, number of parallel teams, number of inspectors per team.
2017	[41]	3,00	DES	Workshop	Software engine. management; TM	Authors present a process simulation model to analyze different parameters (throughput, total time, project size, and team size, among others) to obtain the most appropriate planning during software development in Cloud.
2014	[42]	2,25	ABS	Conference	Software engine. management; TM	Authors present a method to develop academical simulation games of software project management to give the right task to the right developer at the right time. In this proposal, project members are modeled as self-acting agents and artifacts are modeled as resources used by these agents. Authors also define two kinds of simulation rules: (1) how agent parameters and the current state of the product influence the new state of the product while an agent performs an activity; and (2) how the agents choose which activity to perform.
2014	[43]	1,50	MCS	Conference	Software quality; TM	Author presents a simulation model (based on the Monte Carlo technique) to apply Lean Six Sigma on agile software development processes. This model simulates future activities based on historical data. The purpose is to improve Software quality control and quality assurance of software development.
2013	[44]	2,00	PNM	Conference	Software construction; AP	Authors present a simulation model (using Petri nets) to optimize variables (in terms of skill level of employees, software complexity and PC computing capacity) that allow to reduce carbon footprint during the execution of software development processes.
2018	[45]	1,50	SD	Conference	Software engine. management; TM	Authors presented a Kanban development process based on system dynamic models to minimize the number of developers and testers in software development projects. The simulation is performed in terms of task complexity, number of pending and completed development tasks, and completed and pending test tasks.
2013	[46]	1,75	SBT	Workshop	Software engine. models and methods; AP	Authors present a method to define simulation models and facilitate the development of simulation games that allow students to acquire skills in software development projects.
2015	[47]	1,50	SD	Workshop	Software engine. management; TM	Authors propose a simulation model to evaluate the performance of software developers in crowdsourcing software projects. This simulation considers more than 20 variables (registered worker, active worker, winner worker, company's reputation, tasks, taken tasks, submitted tasks, associated tasks, drop taken, task duration, etc.) which are associated with Software worker behavior in crowdsourcing platform, task uploading behavior in platform and influence among software workers and uploaded tasks.
2014	[48]	2,00	QM	Journal	Software testing; AP	Authors propose a queueing simulation framework to analyze reliability assessment for debugging activities of Open Source Software (OSS) projects. The simulation is divided into conditions between four steps (Allocating Step, Reporting Step, Judging Step, and Fixing Step) which are repeated until the end of simulation.
2015	[49]	2,00	ABS	Workshop	Software quality; AP	This paper describes a simulation model to evaluate the interplay of different possible future scenarios to improve the quality of software projects. This simulation is based on mining software repositories and focuses on system growth, bugs lifetime and developer activity.
2015	[50]	1,50	ABS	Conference	Software quality; TM	Author presents an agent-based simulation tool to predict what is the behavior of the software project (with respect to the temporal dimension). This simulation is based on the evaluation of ongoing process and possible development trends at several points in time.

(*continued on next page*)

Table 9 (*continued*)

Year	SPSM [ref.]	Scores (max.3)	Sim. approach	Type of publ.	KA;nature	Description and motivation
2014	[51]	2,25	SBT	Journal	Software engine. management; AP	Authors present a simulation-based model to train skills in software project management. This model evaluates different parameters (cost, resources, etc.) in different dimensions of the project (initiation, conceptual design, planning and execution).
2013	[52]	1,50	DES	Journal	Software engine. management; TM	Author presents a simulation model to evaluate and establish different scenarios on software development process management. This model allows simulating aspects related to risk management in terms of costs, planning, and resources.
2018	[53]	3,00	ABS	Journal	Software maintenance; AP	Authors define an agent-based simulation model that allows quantitively to represent of real-world aspects to measure the evolution of software during maintenance phases.
2013	[54]	2,50	DES	Conference	Software engine. models and methods; TM	Authors present a formal method to define simulation models based on machine learning. Authors aim to improve the configuration of software processes in companies based on gathered knowledge considering input parameters of simulation such as number of tasks, number of developers, minimum/maximum estimated hours and scrum iteration duration.
2018	[55]	1,00	DES	Journal	Software construction; AI	Authors present a simulation model to reproduce the behavior of software development in a distributed manner. (specifically, when throughput, total time, project size, team size and work-in-progress tasks may vary). This simulation model is raised on Software Factory projects.
2015	[56]	2,50	SD	Journal	Software quality; TM	Authors present a SD simulation model to help decision-making in the scope of usability evaluation and quality on software user interfaces. This model improves the configuration of software tester who must execute software testing processes in terms of usability evaluation.
2014	[57]	2,50	DES	Conference	Software engine. process; AP	Authors propose a DES metamodel to integrate the definition of business processes with DES models. Authors also define different transformation rules that systematically generate simulation model from BPMN (Business Process Model and Notation) model.
2015	[58]	2,00	DES	Conference	Software engine. management; TM	Authors propose simulation model to estimate the KSS (Kanban-based scheduling system) performance and improve the management of incidents that cause blockages on software projects. It allows to coordinate work queues to address software systems engineering processes for evolving systems of systems.
2010	[59]	2,00	PNM	Conference	Software requirements; TM	Authors presents a simulation model to assess the quality of use cases. The proposal is based on the evaluation of the case-based descriptions provided by non-technical stakeholder.
2017	[60]	3,00	ABS	Conference	Software engine. management; AP	Authors present a scalable agent-based simulation platform which allows to predict the behavior of a software development project in terms of costs, resources, activities, etc.
2014	[61]	1,50	SD	Journal	Software engine. management; AP	Authors present a SD simulation model of a software factory product line. Authors aims to analyze the changes in behavior when selecting either one of the above-training alternatives.
2017	[62]	2,00	ABS	Conference	Software engine. management; AP	Authors evaluate the application of modeling and simulation techniques to improve the management of projects that are managed using the eXtreme Programming (XP) methodology. This evaluation is based on levels of expertise, number of team members, team member expertise, salary, number of tasks, estimated duration, etc., of XP projects.
2016	[63]	2,00	DES	Journal	Software engine. management; AP	Authors present a simulation method to identify competencies during the process of resource management in software projects. This simulation model considers aspects of capability and productivity of software developers.
2013	[64]	1,50	DES	Conference	Software testing; AP	Authors present a discrete-event simulation model to test the behavior of third-party software components. This model ensures that these software components do not compromise the integrity of the host system.
2017	[65]	1,00	SD	Conference	Software engine. management TM	Authors propose a dynamic simulation modeling to compare agile methodologies (mainly XP or Scrum). This evaluation allows to decide which is the most appropriate methodology for a collaborative software project that has specific requirements in terms of resources, planning, costs, etc.
2016	[66]	1,00	ABS	Conference	Software quality; AP	Authors present a Multi-Agent System that implements an auction mechanism for simulating task allocation in OSS. Authors aim to optimize the task allocation and reduce rework during project execution.
2016	[67]	3,00	PNM	Conference	Software quality; TM	Authors propose a simulation model to evaluate the dynamic and static aspects of software process before these ones are deployed to workflow system for automated execution. Authors aim to maximize efficiency and reduce errors in tasks of software process that are deployed on workflow systems.
2018	[68]	2,50	SD	Journal	Software engine. management; TM	Authors present a decision-making framework and simulation models built to improve decision-making within the ITIL context. Authors aim to systematically build simulation models and solve real-world organization problems applying ITIL recommendations.

(*continued on next page*)

Table 9 (*continued*)

Year	SPSM [ref.]	Scores (max.3)	Sim. approach	Type of publ.	KA;nature	Description and motivation
2019	[69]	2,00	ABS	Conference	Software requirements; TM	Authors propose a scrum-based methodology using multi-agent simulation system to meet the requirements of stakeholders. Authors have also implemented multi-agent modeling to extract meta-data; Nouns and Verbs to structure a team for task distribution.
2019	[70]	3,00	DES + PNM	Journal	Software construction; TM	This article proposes the steps to specify a PetriNets-based simulation engine and based on discrete event system using a distributed cloud architecture.
2019	[71]	2,00	SBS	Journal	Software quality; TM	This paper describes a methodology to analyze security protocols using scenario-based simulations as a method to assess quality. Authors use state machines as a method to define these simulation scenarios.

Acronyms and Abbreviations - HS: Hybrid Simulation, SBS: State-Based Simulation, QM: Queuing models, DES: Discrete-Event Simulation, SD: System Dynamics, ABS: Agent-Based Simulation, PNM: Petri-Net Models, SBT: Simulation-Based Teaching, MCS: Monte Carlo Simulation; AI: Application in Industry, AP: Academic Prototype, TM: Theoretical Model.

SD, ABS) have been applied to define SPSM proposals since 1998. The present review is focused on studying papers published between 2013 and 2019, so the previous data (between 1988 and 2012) have been obtained from the review published by Ali, Petersen and Wohlin [22]. SD and DES paradigms were the most used paradigm in SPSM proposals published between 1988 and 2012, according to conclusions obtained by Ali et al. However, an interesting finding of this study is that ABS and SD paradigm is increasing and decreasing, respectively, its trend among SPSM proposals in the last five years. Regarding DES paradigm, it seems that it is strengthening its position among research community in recent years.

6.3. RQ3. «What decisions does the simulation model help to make?»

After considering all primary studies, it has been possible to identify an important variety of reasons to carry out simulations on software process models. The main reasons are related to the improvement in decision-making during the operational management of these processes. In fact, an important conclusion we have obtained is that the application of simulation techniques facilitates project management. This conclusion has been reached after studying each primary study. Most of these studies are focused on monitoring process performance and comparing it with planned values during simulation. Table 10 presents a summary of all decisions that simulation usually helps to make.

6.4. RQ4. «What is the KA of SE where each SPSM proposal is framed? What is its nature of each proposal?»

Regarding this research question and considering the information showed in Table 9, very interesting information for this systematic literature review has been able to extract.

On the one hand, it was noticed that «Software quality» and «Software engineering management» are the most KAs in Software Engineering where SPSM proposals are framed. Fig. 9 shows the distribution of primary studies per knowledge area. It is important to remember that these KAs are standardized by ISO/IEC 19759:2015 as mentioned in Section 4.1 and Table 1. In addition, most primary studies of last five years are framed on «Software engineering management» (44 %; i.e., 16 primary studies) and «Software quality» (25 %, i.e., 9 primary studies), what means that both are equally the most KAs where SPSM are applied. On the opposite side, some KAs have been little (e.g., «Software maintenance» and «Software requirements» with 3% (i.e., 1 primary study) or «Software engineering process» with 3% (1 primary studies), among others, or not considered (e.g., «Engineering foundations», «Software engineering professional practice», among others) to define SPSM proposals in the last five years.

Once each SPMS proposal is categorized, it is possible to establish a discussion to identify limitations and open issues after analyzing each primary study:

(1) *Regarding «software engineering management» area*, SPSM proposals are focused on *team performance* [39,58], *project management* [42,51,60,62,65,66,68], *resource management* [45,47,61,63], *planning management* [41] or *risk management* [36,52].

On the one hand, *team performance* is evaluated to reduce blocking on software projects. For this purpose, the Alshammri's simulation model [39] evaluates mainly human factors (such as, individual performance, stress, attitude, etc.), but does not relate those factors to organization process, job tasks, team interaction and individual behavior; whereas Tregubov and Lane [58] propose a simulation model to estimate the performance of their Kanban-based scheduling system. It allows to coordinate work queues and know work in progress, limited work in progress, and identification of issues causing blocked work. However, this model is presented in academic environments and

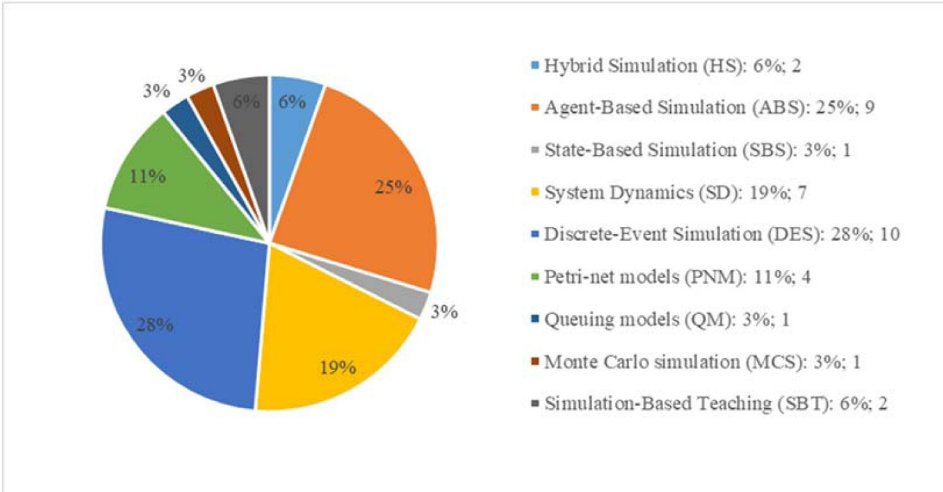


Fig. 7. Distribution of simulation paradigms applied in the studies included.

requires calibration using data from real software projects.

On the other hand, *project management* is the topic most considered by the research community in this knowledge area. Nassal [42] focuses on games and simulation models to teach how to manage software projects, but this agent-based simulation model does not consider agent behavior, emotions, cognition, and social status, among other aspects. The Cohen's simulation model [51] and Ahlbrecht et al. [60] allows training in management skills and predicting behavior of software development projects considering adjustable parameters (costs, resources, activities, etc.), but theses model are prototypes that need to be improved its support to improve decision making in real software projects. In addition, it has been possible to identify a couple of primary studies related to the application of agile methodologies in the management of software projects. Firstly, Hurtado et al. [62] proposes a simulation model (on software projects that use XP) to evaluate parameters of work teams (expertise, number of team members, salary, number of tasks, estimated duration, etc.), but this proposal does not consider other aspects such as costs, overload of tasks, skills, etc. This handicap is partially covered by Alexandros' proposal [65] which focuses on the evaluation of XP and Scrum to decide which is the most appropriate methodology to be applied in collaborative software project in terms of resources, planning and costs. Moreover, we have located a primary study related to service management and project management. Specifically, Orta et al. [68] define a decision-making framework to systematically built and improve simulation models that solve real-world organization problems. This proposal focuses on ITIL recommendations, but it would be interesting to explore other ITSM (Information Technology Service Management) processes such as CMMI, ISO 9000 or ISO/IEC 27000. Finally, it is also possible to find some work that relates simulation models to the optimization of tasks in software projects. Specifically, Rúbio et al. [66] proposes a simulation model (which is implemented on their MAESTROS tool) to optimize the cost of software projects and reduce rework on open source software development project. This proposal has open issues related to risk management and resource management.

Likewise, related to *resource management*, firstly, Gong's simulation model [45] evaluates parameters to minimize the number of developers and testers in software projects, but this proposal is a *work-in-progress* proposal that needs to consider aspects such as transition between developers and testers, customer participation, pair programming and rework. Saremi and Yang [47] propose a simulation model to evaluate the performance of software developers in crowdsourcing software projects. This simulation considers more than 20 variables (which are associated with Software worker behavior in crowdsourcing platform, task uploading behavior in platform and influence among software workers and uploaded tasks), but it does not consider aspects such as skills of workers or complexity of tasks. Kuchař and Vondrák [63] presents a simulation model to evaluate the capacity and productivity of software developers according to their skills, but these authors do not consider aspects such as more capable workers make fewer errors and their capability should influence the error rate of the activity. Matalonga et al. [61] proposes a simulation model to analyze the changes in

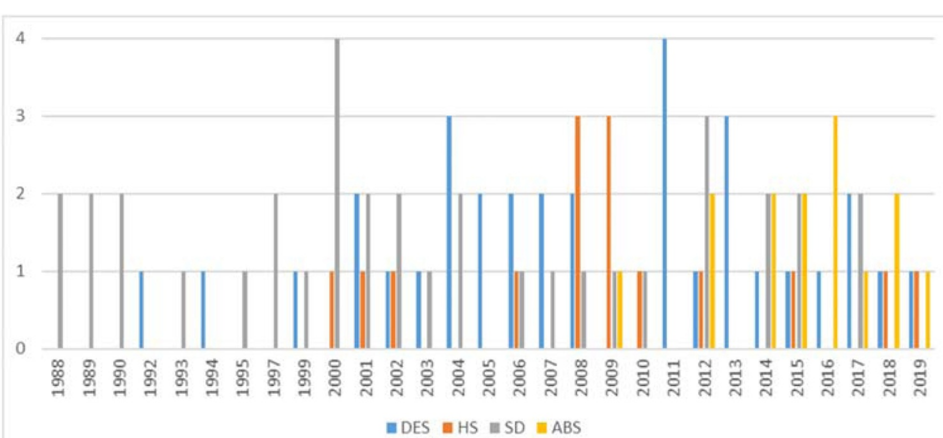


Fig. 8. Temporary trend of the most used SPSM paradigms.

Table 10
Description of decisions that simulation usually helps to make.

Decisions that simulation usually helps to make	Refs.
Risk of task failure in Software Development Process (SDP).	[36]
Evaluate the execution and planning of events associated with SDPs.	[37]
Commit behavior for different software developer types.	[38]
Behavior of team members into software development process.	[39]
Reduction of the risk of software projects with the identification of defects in SDP using, or not, software tests.	[40,43,49,64,71]
Agile techniques to meet the requirements of stakeholders during the execution of software requirements engineering process	[69]
Application of simulation models as support for software construction	[70]
Distributed software development processes by the project and team size, number of artifacts, and project duration.	[41]
Ergonomic and psychological factors to improve project management and the creation of software teams in software projects.	[42]
Selection of policies for SDP which allow reducing carbon footprint.	[44]
The impact of ratio between total of software developers and total of software testers on Kanban Software Development (KSD) process.	[45]
Apply techniques to improve the training of engineers in software processes or reduce risk associated with decisions on software projects.	[46,51,52,60]
Creation of crowdsourced software developer teams to reduce time-to-market of software production and improve successfully completion tasks.	[47]
Minimizing rework and maximizing reliability in OSS project to determine the optimal version-updating moment.	[48,66]
Identification of issues that affect the maintenance and evolution of software. The purpose is to predict the factors that cause these issues.	[50,53]
Factors to improve decision making in the technology service management.	[68]
Reduce the risk of carrying out modifications of software processes in knowledge-based companies.	[54]
Identification the risk of implementing software processes in a distributed manner to reduce the cost.	[55]
Identification of factors associated with usability evaluation team during software processes.	[56]
Application of BPMN as source model to define DES model.	[57]
Evaluation of the difficulty to capture current and relative status of software processes which evolves in Systems of Systems (SoS).	[58]
Identifying the risk of low-quality descriptions in the definition of use cases to reduce defects and problems in later stages of SDP.	[59]
Improving the productivity of software developers through the improvement of technology skills, human behavior and creativity.	[61,63]
Identifying the most appropriate software development agile methodology.	[62,65]
Efficiency of software process deployment in workflow systems.	[67]

behavior when selecting either one of the above-training alternatives. The purpose is to improve the resource management (work teams) of software development projects and reduce the occurrence of defects in software through the learning improvement on new technologies. This model evaluates variables such as skills training and process training, software defects, project non-conformances, and product size, among others. However, it is possible to identify some limitations on how they impact outdated technology skills impact quality during project execution.

Moreover, on the topic of *planning management* within «software engineering management» area, it has been possible to find Lunesu's simulation model [41] which evaluates different input variables in distributed software development projects. This model aims to help project managers select the most suitable planning alternative considering throughput, total time, project size, and team size, among other variables. However, this model only partially addresses the (quantitative) relationship of these variables without considering aspects such as human factors. In addition, this model is only applicable to cloud development projects.

Finally, our review has identified several proposals focused on *risk management* from different points of view. Firstly, Uzzafer's simulation model [52] evaluates parameters (such as cost, planning and resources) to manage risks in software projects. However, this proposal does not consider quality aspects and specifications in project management and its risks. Secondly, Saremi [36] presents a simulation model to assess (in terms of task, agent decisions, different available crowdsourced markets, quality and similarity of tasks, worker profile and worker skill) the failure rate in crowdsourcing software development process, but this proposal is a *work-in-progress* proposal without theoretical evidence of this simulation model or validation in real projects.

(1) *Regarding «software quality» area*, SPSM proposals are focused on *quality of simulation models* [37,50], *quality of software projects* [38,43,56], *security software* [71] and *error reduction* [40,49,67].

On the one hand, several primary studies have been found that address *quality during the definition of simulation models*. Firstly, Collins' proposal [37] describes a tool that provides a visual representation of the scheduling and execution of events in a discrete-event simulation. This tool allows to improve the quality of simulation models and their

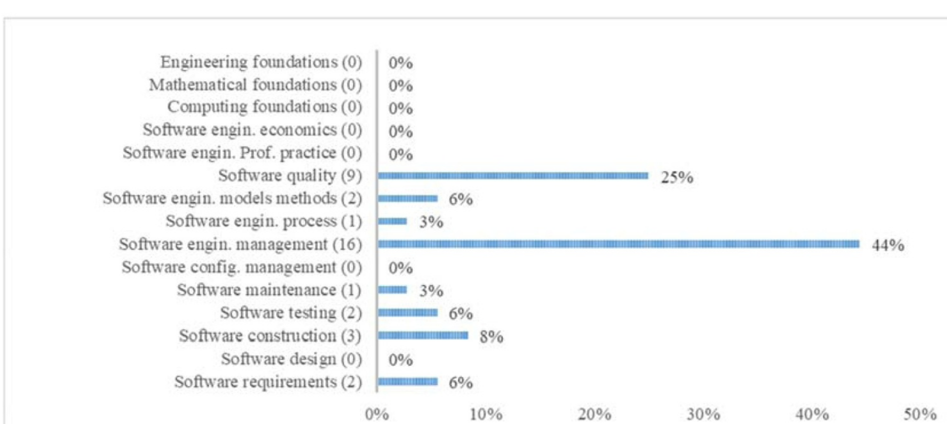


Fig. 9. Distribution of KAs in software engineering where SPSM proposals are framed.

implementation, but currently it is an academic prototype that needs to be integrated with simulation software tools and present reverse engineering mechanisms to obtain the event model from an existing simulation. Secondly, Honsel's proposal [50] tries to evaluate and predict the quality of simulation parameters on software processes in terms of system growth, bug trends and developer activity. Honsel also proposes to use software techniques mining for fitting her agent-based simulation model. However, this primary study is in early stages and provides initial results of this research without theoretical evidence of this simulation model or validation in real projects.

On the other hand, *quality of software projects* is the topic most considered by the research community in this knowledge area. Firstly, some authors propose a simulation model to monitor software quality through different software project variables (effort, number of tasks, errors found, developers) [38]. This simulation model has been defined considering specific case studies in controlled environments, but it is convenient to calibrate the model with well-maintained software projects and improve the introduction of bug in the model. Secondly, Ghane [43] presents a simulation model (based on the Monte Carlo technique) to apply Lean Six Sigma on agile software development processes and improve software quality control and quality assurance simulating future activities. For this purpose, the model evaluates input variables such as time, resource, cost and scope, and uses such historical data to model the error rate of activity estimated. However, it does not consider other quality attributes to improve the prediction of the performance of activities, such as, human factor, defect density per usage time, defect handling efficiency, reported defects, customer satisfaction and backlog performance, among others. Thirdly, Hurtado et al. [56] propose the application of modeling and simulation techniques to help decision-making in the usability evaluation process and evaluate the quality on software user interfaces. This model evaluates effects of different compositions of software tester teams would have on the outcome of the evaluation (in terms of time, cost and number of problems found). A possible limitation of this proposal is its focus on the evaluation process and could be applied to other processes of the usability engineering lifecycle.

Moreover, *security software* is a subject valued through simulation techniques. Specifically, Al-Shareefi et al. [71] describes a methodology to analyze security software protocols using scenario-based simulations as a method to assess quality. Authors use state machines as a method to define these simulation scenarios. However, authors do not propose any method to evaluate this methodology in practice, so it is not possible to verify the feasibility of the proposal.

Finally, *error reduction* is another relevant topic in «software quality» area. On the one hand, De Sousa Coelho et al. [40] proposes a simulation model to evaluate scenarios in the inspection of software projects in terms of 10 input variables (such as technique, number of inspectors, parallel equipment, etc.). However, the simulation model establishes very little relationships between variables, so it is convenient to improve this aspect to obtain a more robust and complete model. On the other hand, Honsel et al. [49] and Czopik et al. [67] respectively propose a simulation model to reduce errors in tasks of software process and analyze the evolution of software projects. The first one is also based on analysis of bugs lifetime and developer activity, mainly. The objective is to improve the quality of software development projects, but it does not analyze the evolution of defects during the software lifecycle nor consider other types of agents and their skills.

(1) *Regarding «software engineering models and methods» area*, two primary studies are found [54,46]. On the one hand, Košinár and Štrba [54] propose a method to define simulation models on software processes, but it only considers a few input parameters (number of tasks, developers, minimum/maximum estimated hours and typical iteration duration considering Scrum) and does not consider external factors (such as, software skills, dependencies between tasks, work overload, etc.) in simulations that could influence the software product development. On the other hand, Peixoto et al. [46] present a method to define simulation models and facilitate the development of simulation games that allow students to acquire skills in software development projects.

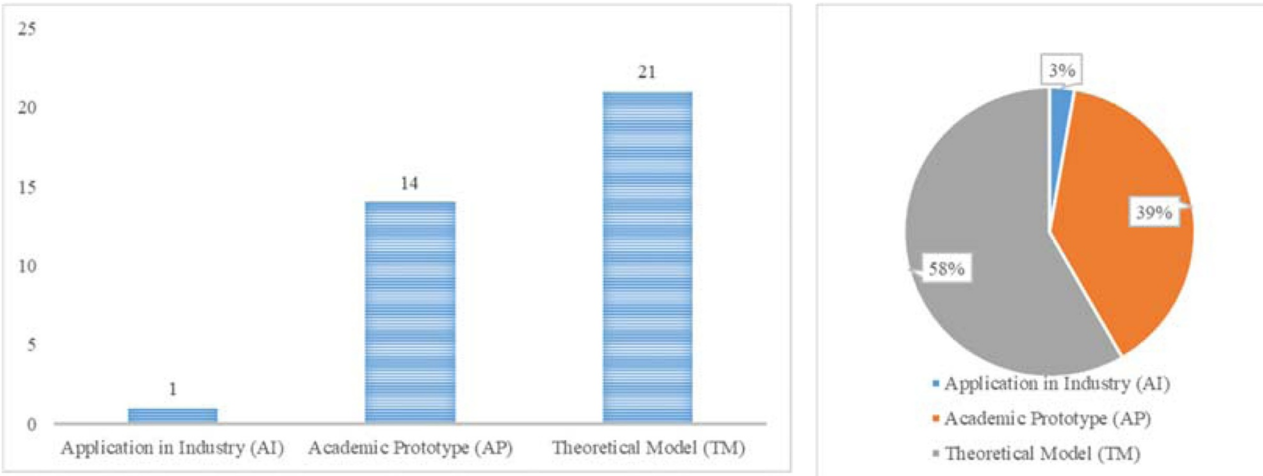
(2) *Regarding «software engineering process» area*, after executing our review protocol, it has only been possible to locate the proposal of García et al. [57]. This proposal defines a simulation metamodel and transformation rules to obtain this metamodel from BPMN models, but, at present, this proposal does not systematically configure simulation parameters nor does it provide integration with Business Process Management Suites.

(3) *Regarding «software maintenance»*, the proposal of Mohammed Ali et al. [53] has been found. These authors define a simulation model to evaluate quantitively real-world aspects of software during maintenance phases. However, it has some limitations. The main one is that the attitude value of each participant in the model is based on a generalization from expert opinion without refereeing a particular development project.

(4) *Regarding «software testing» area*, two primary studies are found related to this knowledge area [48,64]. On the one hand, Lin and Li [48] propose a rate-based queueing simulation model to analyze testing and debugging processes of lifecycle software (specifically, open source software projects). The main limitation of this proposal is that it is dependent on the configuration of the input parameters. It would be convenient to adapt the conditions to improve the predictions of the authors' model. On other hand, Lipka et al. [64] describes a simulation model that allows testing of third-party software components of component-based software applications to ensure the integrity of the host software system. Currently, this test is carried out in isolation for each software component and it would be convenient to improve the situation in an orchestrated way (time handling, network communication, etc.), between components.

(5) *Regarding «software construction» area*, several primary studies are found. Firstly, Karunakaran and Rao [44] defines a simulation model based on the evaluation of some variables (skill level of employees, software complexity and PC computing capacity) to reduce carbon footprint during the execution of software development processes. The application of these variables is not quantitative and could be improved through metrics based on function points, software architecture, CPU cache management, etc. Secondly, The Lunesu's simulation model [55] is oriented to the distributed software development. This model analyzes different input parameters (throughputs, total time, project size, team size and work-in-progress tasks) and their impact on projects performed in the Cloud. However, this model is only applicable on "pure" Scrum and Kanban processes. It would also be interesting to include other quantitative parameters such as number and list of issues, estimated time and time spent for resolving issues, priority of issues, etc.). Finally, Bañares and Colom [70] proposes the process to specify a simulation engine based on discrete event system. The process is also based on Transitions of Petri Nets and shown that in a centralized environment Transitions is better than other simulation techniques when the number of processes grows above a threshold.

(6) *Regarding «software requirements»*, two primary studies are found related to this knowledge area which could also be framed in «software quality» area. Firstly, authors present a simulation model to evaluate the quality of the definition of use cases [59]. Currently, this proposal simulates use case based on descriptions in isolation in a controlled environment. A promising future line could be related to the integration of this simulation model with some requirement specification methodology and other requirement types. Secondly, Wysocki and Orłowski [69] propose a scrum-based methodology using multi-agent simulation system to meet the requirements of stakeholders. Authors have also implemented multi-agent modeling to extract meta-data, but these models have only been tested in



(A) (B)

Fig. 10. Statistics associated with the classification by nature of each SPSM proposal.

academic settings and in small projects, so it is not possible to verify the feasibility of the proposal.

On the other hand, it was noticed that the nature of most of the SPSM proposals is Theoretical Model (TM) (58 %) followed by Academic Prototype (AP) (39 %). On the opposite side, the applications of SPSM in the industrial domain during the last 5 years have been scarce (Application in Industry (AI); 3%). Fig. 10 shows the distribution of studies per nature.

6.5. RQ5. «What scientific validation methods have been used in different proposals?»

After analyzing the primary studies shown in Table 9, it is possible to observe more than half of the papers (i.e., 63,89 %) use some type of scientific validation. Table 11 shows in detail the distribution of primary studies and evaluation methods (experiment, case studies, proof-of-concept) used to validate each proposal. It is possible to observe that 11, 8 and 4 primary studies carry out their evaluation by means of experiments, case studies or proof of concept, respectively.

The most commonly used methods are experiments and case studies. However, something similar happens when we have analyzed the types of proposals. Specifically, it is seen that the term case study is used when it is really being referred to as application experiences or applications in practice in projects (although this is something that also happens in other fields of software engineering). Anyway, we have observed that no papers have a comprehensive validation plan or protocol that allows the systematic execution of the proposal in practice. This situation hinders the reproducibility of these validations, but it should be solved if a rigorous and correct evaluation is explained and carried out to verify the contribution and the suitability of each primary study.

Table 11
Distribution of the primary studies according to their scientific validation method.

Scientific Validation Method	Primary Studies	Total	%
Experiment	[40,48,53,55,56,58,61,63,64,66,68]	11	30,56
Case studies	[38,41,49,50,51,52,57,62]	8	22,22
Proof-of-concept	[44,47,54,59]	4	11,11
No evaluation	[36,37,39,42,43,45,46,60,65,67,69,70,71]	13	36,11
	Total	36	

7. Conclusion and future works

Defining software process models is a mechanism that allows abstract representations of procedures, methodologies, or set of steps in any topic of software engineering. However, the implementation of these process models in real environments can become costly and complex tasks due to characteristics of software processes [8]. In this context, it is very convenient to have mechanisms or techniques to verify and validate software process models before these ones are deployed into production environments. The goal is to minimize costs later and improve decision-making. One of the well-known techniques for achieving this goal is to use simulation techniques on software processes. This topic has been very studied and reviewed over last decades in the research literature. However, regarding systematic reviews, although different papers have been published in the last decade, they are not current nor categorize each SPSM proposals according to their nature or knowledge area within software engineering.

This paper aims to fill this gap by carrying out a systematic review to know the *state-of-the-art* in this area since 2013 to 2019 and identify opportunities for new research works. It identifies 36 primary studies considering objective quality criteria to measure the quality of each primary study and determinate evolution and representativeness levels of each study. In this context, it was concluded that most of research papers are original proposals (i.e., they are not evolution of another).

After analyzing these primary studies, it has been possible to identify that DES and ABS are the most used simulation paradigms. On the opposite side, some simulation paradigms have been nothing (e.g., SBS) or little used (e.g., QM, MCS) to define SPSM proposals in the last five years. In addition, we have also identified a trend change. The HS paradigm has been a paradigm widely used in past years according to [20,17], but after executing our SLR, we have detected that HS paradigm has suffered a decrement. This conclusion can be confirmed if our results are compared with other reviews. Especially, SD and DES paradigms were the most used paradigm in SPSM proposals published between 1988 and 2012, according to conclusions obtained in [22]. However, an interesting finding of this study is that ABS and SD paradigm is increasing and decreasing, respectively, its trend among SPSM proposals in the last five years. Regarding DES paradigm, it seems that it is strengthening its position among research community in recent years.

Moreover, after executing our SLR, some conclusions have been also obtained related to the nature and knowledge area. Regarding nature, it has been noticed that most of SPSM proposals are theoretical models or academic prototypes. Some of these ones describe validations or applications on controlled case studies or proof-of-concepts (see Table 11), but applications of SPSM in software industrial contexts are missing today [35]. It has not been possible to locate research papers where simulation models have been applied in real business contexts related to software engineering. Regarding knowledge area where each primary study is framed, it is noticed that «software engineering management» and «software quality» are most knowledge area in software engineering where SPSM proposals are mainly framed [72,73] whereas knowledge areas related to «software maintenance», «software requirements» and «software engineering models and methods» are the least ones researched. The previous section categorizes and discusses each SPMS proposal according to its area of knowledge.

In addition, we have observed more than half of papers use some type of scientific validation methods (mainly, experiments and case studies). However, we have identified that the «case study» term is erroneously used because it is really being referred to application experiences or applications in practice in projects. Comprehensive validation plan or protocol are missing. This situation makes the reproducibility of these validations difficult. Therefore, it is necessary to promote greater compliance with formal guidelines by the scientific community when its members publish contributions and results on a specific topic in an area of research. This compliance should be a commitment and inherent requirement of any quality scientific production.

Finally, once completed this review, we plan to explore a research line that could be interesting. We refer to investigate the possibilities offered by model-driven engineering paradigm to facilitate the integration, definition and management of simulation models in the BPM context in practical environments. The use of this paradigm has obtained satisfactory results in other areas [72–75] and it could be interesting its application is our future objective.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research has been supported by the POLOLAS (TIN2016-76956-C3-2-R) and BadgePeople (TIN2016-76956-C3-2-R) projects of the Spanish Ministry of Economy and Competitiveness under ERD funds.

References

[1] International Organization for Standardization (ISO), BS en iso 9001:2008, 2008. Last accessed: January 2020.
[2] K.H. Rose, A guide to the project management body of knowledge (PMBOK® guide)-fifth edition, Proj. Manag. J (2013), https://doi.org/10.1002/pmj.21345.
[3] Office of Government Commerce Prince2, Managing Successful Projects with Prince2, Prince2 Official Editorial, 2009.
[4] Software Engineering Institute, CMMI for Development, Version 1.3, Carnegie Mellon University, 2010, p. 482 CMU/SEI-2010-TR-033 ESC-TR-2010-033.
[5] Cartlidge, A., Hanna, A., Rudd, C., Macfarlane, I., Windebank, J., & Rance, S. (2007). An Introductory Overview of ITIL v3: A high-level overview of the IT Infrastructure Library. version 1.0. *EUA: itSM*.
[6] W.M.P. Van Der Aalst, Business process management demystified: a tutorial on models, systems and standards for workflow management, Advances in Petri Nets, Lecture Notes in Computer Science, 2003, pp. 1–65.
[7] J. Stark, Product Lifecycle Management: 21st Century Paradigm for Product Realisation, Springer, 2011.
[8] G. Canfora, Software process: characteristics, technology and environments, UPGRADE 5 (2004) 6–10.
[9] M.P. Papazoglou, E-business: Organizational and Technical Foundations (2006).
[10] J.-R. I. Barba, J. Fernandez-Olivares, C. Del Valle, B. Weber, Time prediction on multi-perspective declarative business processes, Knowl. Inf. Syst. (2018), https://doi.org/10.1007/s10115-018-1180-3.
[11] I.B. Jiménez-Ram'irez, C. Del Valle, B. Weber, Generating multi-objective optimized configurable business process models, Proceedings of the Sixth International Conference 2012 Res. Challenges Inf. Sci. (RCIS), 2012, pp. 1–2.
[12] J. Swanson, Business dynamics—systems thinking and modeling for a complex world, J. Oper. Res. Soc. (2002), https://doi.org/10.1057/palgrave.jors.2601336.
[13] T.K. Abdel-Hamid, S.E. Madnick, Software project dynamics: An Integrated Approach, Prentice-Hall, Inc., 1991.
[14] H. Zhang, B. Kitchenham, D. Pfahl, Reflections on 10 years of software process simulation modeling: a systematic review, International Conference on Software Process, Springer, Berlin, Heidelberg, 2008, pp. 345–356.
[15] Kitchenham, Software process simulation over the past decade: trends discovery from a systematic review, Integr. Vlsi. J. (2008), https://doi.org/10.1145/1414004.1414077.
[16] H. Zhang, B. Kitchenham, D Pfahl, Software process simulation modeling: facts, trends and directions, in: neonatal, Paediatr. Child Heal. Nurs. (2008), https://doi.org/10.1109/APSEC.2008.50.
[17] H. Zhang, B. Kitchenham, D. Pfahl, Software Process Simulation Modeling: An Extended Systematic Review, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), SpringerLink, 2010.
[18] C. Gao, S. Jiang, G Rong, Software process simulation modeling: preliminary results from an updated systematic review, Proceedings of the 2014 International Confernce Software System Process – ICSSP 2014, 2014, https://doi.org/10.1145/2600821.2600844.
[19] B. França, G. Travassos, Are we prepared for simulation based studies in software engineering yet? CLEI Electronic Journal 16 (2013) 9.
[20] M.I. Kellner, R.J. Madachy, D.M. Raffo, Software process simulation modeling: why? what? how?, J. Syst. Softw. (1999), https://doi.org/10.1016/S0164-1212(99)00003-5.
[21] D. Liu, Q Wang, J. Xiao, The role of software process simulation modeling in software risk management: a systematic review, Proceedings of the 2009 Third International Symposium on Empirical Software Engineering and Measurement ESEM 2009, 2009, https://doi.org/10.1109/ESEM.2009.5315982.
[22] N. Bin Ali, K. Petersen, C. Wohlin, A systematic literature review on the industrial use of software process simulation, J. Syst. Softw. (2014), https://doi.org/10.1016/j.jss.2014.06.059.
[23] H. Zhang, M.A. Babar, P. Tell, Identifying relevant studies in software engineering, Inf. Softw. Technol. 53 (2011) 625–637, https://doi.org/10.1016/j.infsof.2010.12.010.
[24] Monique Jansen-Vullers, Mariska Netjes, Business process simulation–a tool survey, Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, Aarhus, Denmark, Vol. 38 2006, pp. 77–96.
[25] B. Kitchenham, P. Brereton, A systematic review of systematic review process research in software engineering, Inf. Softw. Technol. 55 (2013) 2049–2075, https://doi.org/10.1016/j.infsof.2013.07.010.
[26] B. Kitchenham, R. Pretorius, D. Budgen, O. Pearl Brereton, M. Turner, M. Niazi, S. Linkman, Systematic literature reviews in software engineering – a tertiary study, Inf. Softw. Technol. 52 (2010) 792–805, https://doi.org/10.1016/j.infsof.2010.03.006.
[27] F.Q.B. Da Silva, A.L.M. Santos, S. Soares, A.C.C. Frana, C.V.F. Monteiro, F.F. MacIel, Six years of systematic literature reviews in software engineering: an updated tertiary study, Inf. Softw. Technol. 53 (2011) 899–913, https://doi.org/10.1016/j.infsof.2011.04.004.
[28] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, Engineering 2 (2007) 1051, https://doi.org/10.1145/1134285.1134500.
[29] H. Zhang, M.A. Babar, An empirical investigation of systematic reviews in software engineering, Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement, 2011, pp. 87–96., , https://doi.org/10.1109/ESEM.2011.17.
[30] C. Wohlin, R. Prikladniki, Systematic literature reviews in software engineering, Inf. Softw. Technol. 55 (2013) 919–920, https://doi.org/10.1016/j.infsof.2013.02.002.
[31] P. Bourque, R.E. Fairley, SWEBOK v.3 – Guide to the Software Engineering – Body of Knowledge, IEEE Computer Society Press, 2014.
[32] E.W.T. Ngai, Y. Hu, Y.H. Wong, Y. Chen, X. Sun, The application of data mining techniques in financial fraud detection: a classification framework and an academic review of literature, Decis. Support Syst. 50 (2011) 559–569, https://doi.org/10.1016/j.dss.2010.08.006.
[33] S. Feyer, S. Siebert, B. Gipp, A. Aizawa, J. Beel, Integration of the Scientific Recommender System Mr. DLib Into the Reference Manager JABREF, Integration of the Scientific Recommender System Mr. DLib Into the Reference Manager JABREF, SpringerLink, 2017.
[34] D.J. Power, Concepts and resources for managers, Decis. Support Syst. (2002), https://doi.org/10.1111/1471-0528.12173.
[35] C.M. Schmucker, A. Blümle, L.K. Schell, G. Schwarzer, P. Oeller, L. Cabrera, E. Von Elm, M. Briel, J.J. Meerpohl, Systematic review finds that study data not published in full text articles have unclear impact on meta-analyses results in medical research, PLoS ONE (2017), https://doi.org/10.1371/journal.pone.0176210.
[36] R. Saremi, A hybrid simulation model for crowdsourced software development, Proceedings of the Fifth International Workshop on Crowd Sourcing in Software Engineering, 2018, pp. 28–29.
[37] S.C. Collins, N.D. Gonda, L.C. Dumaliang, J.F. Leathrum Jr, R.R. Mielke, Visualization of event execution in a discrete event system, Proceedings of the Fiftieth Annual Simulation Symposium. ANSS 2017, Part 2017 Spring Simulation Multi-Conference, SpringSim 2017, 2017, https://doi.org/10.1007/978-3-319-19276-5_7.
[38] V.H. Honsel, M. Welter, S. Waack, J. Grabowski, Monitoring software quality by means of simulation methods, Proceedings of the Tenth ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2016, https://

doi.org/10.1145/2961111.2962617.
[39] M. Alshammri, Simulation Modeling of human aspects in software project environment, Proceedings of the ASWEC 2015 Twenty-forth Australasian Software Engineering Conference – ASWEC ' 15 Vol. II, 2015, https://doi.org/10.1145/2811681.2824995.
[40] J.J. De Sousa Coelho, J.L. Braga, B.G. Ambrósio, System dynamics model for simulation of the software inspection process, ACM SIGSOFT Software Engineering Notes, ACM SIGSOFT, 2013.
[41] M.M. Lunesu, J. Münch, M. Kuhrmann, Using measurement and simulation for understanding distributed development processes in the cloud, Proceedings of the Twenty-seventh International Workshop on Software Measurement (Twelfth International Conference on Software Processing Products and Processes, 2017, pp. 1–11.
[42] Nassal, A general framework for software project management simulation games, Proceedings of the Conference on Information Systems and Technology, 2014, https://doi.org/10.1109/CISTI.2014.6877074.
[43] K. Ghane, A model and system for applying lean six sigma to agile software development using hybrid simulation, 2014 Proceedings of the IEEE International Technology Management Conference ITMC 2014, 2014, https://doi.org/10.1109/ITMC.2014.6918594.
[44] D. Karunakaran, G.S.V. Rao, A petri net simulation of software development lifecycle towards green IT, 2013 Proceedings of IEEE Conference on Open System ICOS 2013, 2013, https://doi.org/10.1109/ICOS.2013.6735048.
[45] H. Gong, B. Liu, D. Shao, A simulation model of kanban software process, Proceedings of the Asia Pacific Software Engineering Conference APSEC, 2018, https://doi.org/10.1109/APSEC.2017.96.
[46] D.C.C. Peixoto, R.F. Resende, C.I.P.S. Pádua, An educational simulation model derived from academic and industrial experiences, Proceedings of the Frontiers in Education Conference FIE, 2013, https://doi.org/10.1109/FIE.2013.6684914.
[47] R.L. Saremi, Y. Yang, Dynamic simulation of software workers and task completion, Proceedings of the Second International Workshop on Crowd Sourcing in Software Engineering – SE 2015, 2015, https://doi.org/10.1109/CSI-SE.2015.11.
[48] C.T. Lin, Y.F. Li, Rate-based queueing simulation model of open source software debugging activities, IEEE Trans. Softw. Eng. (2014), https://doi.org/10.1109/TSE.2014.2354032.
[49] V. Honsel, D. Honsel, J. Grabowski, Software process simulation based on mining software repositories, Proceedings of the IEEE International Conference on Data Mining Workshops. ICDMW, 2015, https://doi.org/10.1109/ICDMW.2014.35.
[50] V. Honsel, Statistical learning and software mining for agent based simulation of software evolution, Proceedings of the International Conference on Software Engineering, 2015, https://doi.org/10.1109/ICSE.2015.279.
[51] I. Cohen, M. Iluz, A. Shtub, A simulation-based approach in support of project management training for systems engineers, Syst. Eng. (2014), https://doi.org/10.1002/sys.21248.
[52] M. Uzzafer, A simulation model for strategic management process of software projects, J. Syst. Softw. (2013), https://doi.org/10.1016/j.jss.2012.06.042.
[53] S. Mohammed Ali, M. Doolan, P. Wernick, E. Wakelam, Developing an agent-based simulation model of software evolution, Inf. Softw. Technol. (2018), https://doi.org/10.1016/j.infsof.2017.11.013.
[54] M. Košinár, R. Štrba, Simulations of agile software processes for healthcare information systems development based on machine learning methods, IFAC Proceedings Volumes, 2013, https://doi.org/10.3182/20130925-3-CZ-3023.00028.
[55] M.I. Lunesu, J. Münch, M. Marchesi, M. Kuhrmann, Using simulation for understanding and reproducing distributed software development processes in the cloud, Inf. Softw. Technol. (2018), https://doi.org/10.1016/j.infsof.2018.07.004.
[56] N. Hurtado, M. Ruiz, E. Orta, J. Torres, Using simulation to aid decision making in managing the usability evaluation process, Inf. Softw. Technol. (2015), https://doi.org/10.1016/j.infsof.2014.06.001.
[57] M.T. García, M.A. Barcelona, M. Ruiz, L. García-Borgoñón, I. Ramos, A discrete-event simulation metamodel for obtaining simulation models from business process models, Inf. Syst. Dev. (2014), https://doi.org/10.1007/978-3-319-07215-9_25.
[58] A. Tregubov, J.A. Lane, Simulation of kanban-based scheduling for systems of systems: initial results, Procedia Comput. Sci. (2015), https://doi.org/10.1016/j.procs.2015.03.004.
[59] V. Hoffmann, H. Lichter, A model-based simulation environment for structured textual use cases, Proceedings of the International Conference on Software and Data Technologies, 2010, pp. 109–124.
[60] T. Ahlbrecht, J. Dix, N. Fiekas, J. Grabowski, V. Herbold, D. Honsel, S. Waack, M. Welter, Agent-based Simulation for Software Development Processes, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), SpringerLink, 2017.
[61] S. Matalonga, M. Solari, T.S. Feliu, An empirically validated simulation for understanding the relationship between process conformance and technology skills, Softw. Qual. J. (2014), https://doi.org/10.1007/s11219-013-9214-2.
[62] N. Hurtado, M. Ruiz, C. Capitas, E. Orta, Applying agent-based simulation to the improvement of agile software management, Proceedings of the International Conference on Software Process Improvement and Capability Determination, 2017, pp. 173–186.
[63] Š. Kuchař, I. Vondrák, Automatic allocation of resources in software process simulations using their capability and productivity, J. Simul. (2016), https://doi.org/10.1057/jos.2015.8.
[64] R. Lipka, T. Potužák, P. Brada, P. Herout, SimCo – Hybrid Simulator for Testing of Component-Based Applications, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), SpringerLink, 2013, , https://doi.org/10.1007/978-3-642-35843-2_36.
[65] N.K. Alexandros, D.P. Sakas, D.S. Vlachos, N.K. Dimitrios, Comparing scrum and XP agile methodologies using dynamic simulation modeling, Strategic Innovative Marketing, Springer, 2017, pp. 391–397.
[66] T.R.P.M. Rúbio, H.L. Cardoso, E da Costa Oliveira, MAESTROS: multi-agent simulation of rework in open source software, Stud. Comput. Intell. (2016), https://doi.org/10.1007/978-3-319-25017-5_6.
[67] J. Czopik, J. Stolfa, S. Stolfa, M.A. Košinár, I. Vondrák, Software process resource utilization simulation using CPN, Proceedings of the Advances in Intelligent Systems and Computing, 2016, https://doi.org/10.1007/978-3-319-29504-6_5.
[68] E. Orta, M. Ruiz, N. Hurtado, D. Gawn, Decision-making in it service management: a simulation-based approach, Decis. Support Syst. (2014), https://doi.org/10.1016/j.dss.2014.06.002.
[69] W. Wysocki, C. Orłowski, A multi-agent model for planning hybrid software processes, Procedia Comput. Sci. 159 (2019) 1688–1697.
[70] J.Á. Bañares, J.M. Colom, Model and simulation engines for distributed simulation of discrete event systems, Proceedings of the International Conference on the Economics of Grids, Clouds, Systems, and Services, Springer, Cham, 2019, pp. 77–91.
[71] F. Al-Shareefi, A. Lisitsa, C. Dixon, Analysing security protocols using scenario based simulation, Proceedings of the International Conference on Verification and Evaluation of Computer and Communication Systems, Springer, Cham, 2019, pp. 47–62.
[72] F.J. Dominguez-Mayo, M.J. Escalona, M. Mejias, H. Torres, A quality model in a quality evaluation framework for MDWE methodologies, Proceedings of the 2010 International Conference on Research Challenges in Information Science (RCIS), 2010, https://doi.org/10.1109/RCIS.2010.5507323.
[73] F.J. Domínguez-Mayo, M.J. Escalona, M. Mejías, M. Ross, G. Staples, Quality evaluation for model-driven web engineering methodologies, Inf. Softw. Technol. 54 (2012) 1265–1282, https://doi.org/10.1016/j.infsof.2012.06.007.
[74] J.G. Enríquez, R. Blanco, F.J. Domínguez-Mayo, J. Tuya, M.J. Escalona, Towards an MDE-based approach to test entity reconciliation applications, Proceedings of the 7th International Workshop on Automating Test Case Design, Selection, and Evaluation, ACM, 2016, pp. 74–77.
[75] R. Blanco, J.G. Enríquez, F.J. Domínguez-Mayo, M.J. Escalona, J. Tuya, Early integration testing for entity reconciliation in the context of heterogeneous data sources, IEEE Trans. Reliab. 67 (2) (2018) 538–556.