Software system deployment process: A systematic mapping study

Marisa Panizzi[1,2,3] *, Marcela Genero[4], Rodolfo Bertone[1]

[1] School of Computer Science, Computer Science Research Institute LIDI (III-LIDI), National University of La Plata, La Plata, Argentina
marisapanizzi@outlook.com
pbertone@lidi.info.unlp.edu.ar
[2] Department of Engineering Information Systems, Technological National University at Buenos Aires, Argentina
[3] Institute of Technology and Engineering, National University og Hurlingham, Hurlingham, Argentina
[4] Department of Technologies and Information Systems,
University of Castilla-La Mancha, Ciudad Real, Spain
marcela.genero@uclm.es

Abstract. Software system deployment is the process in the software development life cycle during which the software is transferred to the customer. Small and Medium Enterprises (SMEs) need to improve their processes and working methods, but they lack the knowledge and resources to do this. If their deployment process is not carried out properly, then the customer will not accept the software product; the costs and time in the maintenance phase will increase, thus causing the project to fail. We therefore decided to carry out a systematic mapping study in order to analyze the state of the art, aiming to discover whether there are models, methodologies or methods that would serve as a guide for SMEs when they set about deploying software systems, also looking at tools, practices, artifacts and techniques. The search was carried out in Scopus, IEEE Xplore and ACM from 2010 to October 2019. Of the 3,483 papers initially found, 16 primary studies were selected for analysis. 63% of the studies present solutions for the installation activity, of which half are tools that address the automation of these in the quest to reduce time and costs. Only two studies propose models for the deployment process.These models delegate decisions on aspects such as tasks, artefacts, techniques, and practices, to those who apply them; this makes the models more difficult to apply in SMEs This situation shows that it is necessary to have a holistic deployment process providing SMEs a detailed guidance on the phases, activities, tasks, templates and roles for systematizing the deployment of the enterprise software systems.

Keywords: Software processes, Software system deployment process, Systematic mapping study.

1 Introduction

Small and Medium Enterprises (SMEs) make up a large portion of the software industry in many countries around the world [1]. It has also been observed that in

* Corresponding author: Marisa Panizzi

recent years software SMEs have emerged very swiftly, and in most developing economies the sector is dominated by small, young enterprises [2]. These organizations have realized that it is crucial for their business to improve their processes and working methods, but they lack the knowledge and resources to do this. The only way to contribute to the success of projects, therefore, is to define, implement and stabilize the development processes [3]. Deployment is a crucial process of the software development life cycle, in which users frequently report errors while the software system is being deployed [4,5]. Some enterprise software may take months, if not years, to deploy completely, there is an absence of automatized activities, and the execution time for the process is lengthy in this type of enterprises [4]. The above problems present themselves once the deployment is completed, at the same time as the software system is being used, and the issues that arise must thus be resolved in the maintenance phase; this leads to unnecessary effort in terms of economic and human resources [4,5].

At the present time, the most important application of automation in software processes is to support the final phases of software development [6]. Although it is true that there are new techniques/practices such as DevOps [7,8] and Continuous Deployment [9] in the context of agile methodologies, which aim to make deployment more agile and automatised, these techniques are used mainly by large companies, such as Flickr, Netflix, Easy and Amazon, amongst others [6,7] which have both the human resources and the infrastructure to use them successfully. But there are a large number of SMEs which have developed enterprise systems that do not use agile methodologies, and they do not have the resources to use these emerging techniques. Yet the hard fact is that SMEs really need to improve their software development processe if they are to achieve a greater degree of competetiveness. We therefore believe that a model for software deployment process that guides SMEs in a holistic way would make it possible to correct problems that occur in the deployment.

All of the above considerations led us to define the objective of our long-term research, which is to propose a software system deployment process model to help SMEs execute the deploymemt process of enterprise systems. Before starting to define a model, we decided to analyze the existing literature on this topic by means of a systematic mapping study (SMS) [10,11].

The paper is organized as follows: Section 2 gives a description of the planning of the SMS, while Section 3 describes the execution. The results obtained are presented in Section 4. A discussion of the threats to validity is presented in Section 5, and finally, our conclusions and outlines of future work are set out in Section 6.

2 Planning of the SMS

In this section, we present the definition of the protocol of the SMS: research questions, search strategy, study selection criteria and process; data extraction form and data synthesis procedure.

The objective of this SMS was to answer the following main research question (RQ): *What is the state of the art as regards the software system deployment process?*. This question was simultaneously broken down into several other questions (RQ1-4) (see Table 1).

Table 1. Research questions.

	Question	Motivation
RQ1:	What contributions have been made with regard to the software system deployment process?	To find and understand what contributions there are with respect to software system deployment.
RQ2:	In what activities or tasks of the software system deployment process are the contributions made?	To discover what activities or tasks in the deployment process have been researched in relation to the tasks and activities defined in the ISO/IEC/IEEE 12207 standard [12].
RQ3:	In what other technical processes and technical management processes related to the deployment process are the contributions made?	To determine what other technical processes and technical management processes have been researched in relation to the technical management processes defined in the ISO/IEC/IEEE 12207 standard [12].
RQ4:	What types of research are used?	To classify the primary studies according to Wieringa´s classification [13] of types of research, as is recommended in [11],[14].

The search string was built by choosing three major search terms: *"Deployment"*, *"Process"* and *"Software"*. We considered the fisrt major term from RUP [15] and alternative term employed in one internationally-recognized standard: the term 'transition'', from the ISO/IEC/IEEE 12207 standard [12]. In the case of the second major term, we also considered the terms "model" "method", "guide" and "guidelines". Although we recognize that their reach is different, all of them help carry out the software system deployment process in software organizations. In the case of the third major term, we considered "computer system" and "application" from ISO/IEC/IEEE 24765 standard [16]. Finally, the search string was:

TITLE((transition OR deployment) AND (process OR model OR method OR guide OR guidelines) AND (software OR "computer system" OR application))

We decided to perform an automatic search in three digital libraries which are the ones most widely-used in Software Engineering research, namely Scopus, IEEE Xplore and ACM digital library, considering only journal and conference papers, from 2010 to October 19, 2019. We recognize that prior to 2010, standards and methodologies were proposed that consider the software deployment process [15], [17] and earlier versions of ISO/IEC/IEEE 12207 standard [12].

We decided that 2010 was a significant year to use as the starting date for our search, given that there were certain trends and challenges at that time that made an impact on software processes. Among these, we can mention Global Software Development, the Internet as a development environment, which implies a change in the processes of software construction and in the mode of operation of software systems, with the appearance of mobile software [6].

The inclusion and exclusion criteria used in the process of paper selection are shown in Table 2.

Table 2. Inclusion and exclusion criteria.

Inclusion criteria
I1.Papers that answer our research questions.
I2. Duplicate studies. When several papers are written by the same authors describing the same topic or a similar one, the most complete one will be considered.
I3. Papers published from 2010 to October 2019.
I4. Papers written in English.
Exclusion criteria
E1. The paper addresses software that is not considered to be a software system, such as embedded software, operating system, software middleware, services and web services, servers and data servers, software for optimizing communication networks, among others.
E2. Papers available only in the form of abstracts, PowerPoint presentations, PhD theses, books.
E3. The full paper is not written in English.

We decided to perform the search on the title, given that we carried out a pilot search considering title, abstract and keywords in Scopus and found a large number of papers (20,069) that were not relevant to our objective.

The study selection process consisted of the following steps: 1) carry out a search in the three sources using the search string in the title, 2) remove duplicate papers, 3) apply the inclusion and exclusion criteria to the title, abstract and keywords 4) apply the inclusion and exclusion criteria to the full text. This process allowed us to select the primary studies that will be analysed to provide answers to the RQs that were formulated**.**

The data extraction form (see Table 3) consists of two parts: the first concerning the metadata of each primary study, and the second related to each of the RQs. To help to answer each RQ we defined a classification scheme (see Table 3). Some of the categories of this scheme were defined in the planning of the SMS, but others were extracted from reading the full text. The description of the categories used in the classification scheme is presented in the Appendix [18].

Table 3. Data extraction form.

Metadata	Paper ID, year, title, authors, publication type (journal or conference), country, keywords.
RQ/Dimension	Categories
RQ1/Contribution	Tool, model, method, methodology, artifact, practices,

	techniques, no contribution.
RQ2/ Deployment - Activities and tasks	Install the software system, prepare the environment, migrate the data, initial data load, test procedures, training system (user and operator), prepare documentation, acceptance test, and others . We consider the activities and tasks in the transition process from the ISO/IEC/IEEE12207 standard [12].
RQ3/ Others; technical processes and technical management processes	Validation, verification, risk management, configuration management, planning, others. We consider the processes from the ISO/IEC/IEEE 12207 standard [12].
RQ4/ Types of research	Evaluation research, philosophical paper, solution proposal, validation research, experience report, opinion paper. We used Wieringa´s classification of types of research [13].

In our effort to answer each research question we planned to do a thematic synthesis based on the classification scheme that had been defined, showing the results by means of graphs and tables.

3 Conducting the SMS

By applying the search string to ACM DL, Scopus and IEEE Xplore we retrieved 3,483 papers. Some adaptations of the search string were made for each digital library, as is shown in the Appendix [18]. The first author then made the selection of the primary papers by applying the process of paper selection defined in the protocol. In parallel, the second author replicated the selection process and obtained a set of primary papers. The two sets of primary papers were checked by all authors. Discrepancies were discussed in order to determine whether it was appropriate to include a particular paper or not.

After the selection process, 16 primary studies were selected for analysis in the effort to answer the RQs that had been formulated. The complete list of the primary studies selected is shown in [18].

We should point out that the "snowballing" method was conducted, following the model described by Wohlin in [19]. That is, once a primary study had been identified in one of the data sources, the references in that primary study were explored recursively by following the same search criteria. Finally, the snowballing process did not produce any new paper which we had not already included.

4 Results

We present the results obtained by analyzing the data recorded in the data extraction form defined in Table 3, aiming to answer the RQs formulated previously after analyzing the selected primary studies, which are listed in [18]. A synthesis of the results is presented in Table 4.

Table 4. Results per research question.

Reference	Results for each RQ			
	Contribution (RQ1)	Deployment activities and tasks (RQ2)	Other technical processes and technical management processes (RQ3)	Types of research (RQ4)
[PS1]	Tool	Installation		Validation
[PS2]	Model	Installation Prepare environment Migrate data Initial data load Test procedures Training system Prepare documentation Acceptance test	Verification	Evaluation
[PS3]	Method	Installation		Validation
[PS4]	Practices	Installation		Validation
[PS5]	No contribution	Installation Prepare environment Migrate data Initial data load Test procedures Training system Prepare documentation Acceptance test		Personal experience
[PS6]	Methodology	Installation Prepare environment Migrate data Initial data load Test procedures Training system Prepare documentation Acceptance test Others	Validation Verification Risk Management Configuration management Planning Others	Evaluation
[PS7]	No contribution			Evaluation
[PS8]	Tool	Installation		Evaluation
[PS9]	Model	Installation		Evaluation
[PS10]	Model	Installation		Evaluation
[PS11]	Method	Installation		Validation
[PS12]	Tool	Prepare documentation		Evaluation

[PS13]	Model	Installation Prepare environment Migrate data Initial data load Test procedures Training system Prepare documentation Acceptance test	Validation Verification Risk Management Configuration management Planning Others	Evaluation
[PS14]	Tool	Installation		Evaluation
[PS15]	Method	Installation		Evaluation
[PS16]	Tool	Installation		Evaluation

RQ1: What contributions have been made with regard to the software system deployment process?

We included five papers that proposed tools. Asmaa et al., [PS1] proposed an automatic deployment tool which simplifies the deployment process for non-advanced users in the cloud. The purpose of the author's work is to put forward a deployment method and then implement it to automate the process of deploying applications in a cloud environment based on model-driven engineering. Song et al. [PS8] presented the CONSOLAS (CONstraint SOLving for Architecture Setup) tool for model-based, automatic configuration and deployment of cloud applications. CONSOLAS assists application operators in configuring and deploying the application automatically. Operators only need to provide simple hints on how they want to configure the application; the tool generates a complete and valid configuration and deploys it. Fein et al. [PS12], presented a CASE Tool called MATCON (Method and Tool for CONsultants) that guides consultants in structuring documents, templates, and work products into reusable objects; the tool enables them to be cataloged and indexed so that these objects can be easily found and reused on subsequent projects to adapt pre-packaged applications, such as Oracle or SAP, to an organization's needs. García-Galán et al. [PS14] presented a tool called ISA Packager, a generic tool for packaging and deploying SPLs (Software Product Line). The major contribution of ISA Packager is its support for product evolution by updating existing products, thanks to the links established between features and de/installation commands. The contribution made by Günalp et al. [PS16] is Rondo, a tool suite that enables continuous deployment for dynamic, service-oriented applications; as a deployment manager Rondo implements this process, and is capable of conducting deployment and of continuously adapting applications according to the changes in the current target platform.

We also found several models and methods. Subramanian [PS2] proposed a model deployment process that can be broken down into eight steps: verification, negotiation, procurement, installation, initialization, configuration, user acceptance testing (UAT), and production use.

Tian et al. [PS3] contributed with a new method with which to deploy software automatically in large-scale clusters in an effort to improve the traditional deployment method that is based on an installation package.

Tyndall [PS4] presented best practices for building a phased installation software deployment paradigm. When combined with thick image techniques, software installations in both academic and administrative machines become more organized, flexible, and discreet.

Toufaili et al. [PS5] explored the criteria concerning the departments, basic activities and roles involved in the transition process, and researched the ways in which they are implemented in four large software organizations. This paper was, however, included in the no contribution category because, taking into account the classification shown in Table 3, nothing was built.

Carrizo et al. [PS7] presented a methodological guideline for the deployment of any software system; its foundations lie in an adaptation of an ASAP-based methodological guideline.

Paredes et al. [PS6] presented a conceptual framework for understanding the problems that affect the deployment of Enterprise applications in small and medium enterprises (SMEs); this was achieved by conducting a literature review. In addition, by carrying out interviews with companies that used ERP (Enterprise Resource Planning), it was possible to build a summary of influence factors for the deployment of IT enterprise applications; three contexts were considered: Technological, Organizational and Environmental. This paper was, however, included in the no contribution category because, according to the classification shown in Table 3, nothing was built.

Dubois et al. [PS9], presented a model for optimizing the costs of running cloud-based applications. The authors define an approach for refactoring a cloud application in such a way that, when it is deployed, it requires less computational capacity and therefore fewer resources.

Falazi et al. [PS10] proposed a decentralized deployment modeling approach that achieves accountability by using public blockchains and a decentralized storage system to store intermediate states of the collaborative deployment model. This research work focuses on the automation of application deployment.

Saatkamp et al. [PS11] presented a method for automating the deployment of applications in cloud environments. The goal of this method is to reduce the information and APIs which have to be exposed to the outside.

Reascos et al. [PS13] proposed a model that serves as a reference for the deployment of IT applications in public institutions; this model is carried out in 5 stages: initiating, planning, executing, monitoring & controlling, and closing, in compliance with the PMI guidelines. In addition, the model identifies four cross-cutting areas of concern: leadership, communication, change management and project management; these make it possible to balance and control the process of deployment of the IT application.

Deshmukh et al. [PS15], for their part, put forward an innovative method for installing and configuring the application in different environments. It uses the SaltStack tool to deploy the application on different servers. A pillar method is employed to ensure data security and easy data flow.

Of all the contributions, only two [PS2,PS13] propose an activity model that could serve as a guideline for software companies when carrying out the entire deployment process in a systemized manner. In [PS6] we are introduced to a method that serves as a guide for the deployment of any computer system. [PS1,PS8,PS16] propose tools for the automization of the installation activity, and in [PS12] a tool is presented that prepares documentation for the deployment process. In [PS14], ISA Packager provides a way to package SPL assets into an installer, as well as to deploy and update products.

While [PS4] puts forward a set of practices that allow the categorization of software packages into baselines, thus enabling system administrators to organize installers, the authors of [PS5,PS7] do not, according to our proposed classification, make a specific contribution. In [PS3,PS11,PS15] the researchers present a method that enables software to be deployed automatically. The authors of [PS9] define a model for refactoring a cloud application in such a way that, when it is deployed, it requires less computational capacity and therefore fewer resources. Only one contribution [PS10] proposes a decentralized deployment modeling approach.

RQ2: In what activities or tasks of the software system deployment process are the contributions made?

Asmaa et al. [PS1] proposed a tool to automate the installation of applications, in an attempt to reduce the developer's work as regards the cost of installation at the "time and effort" level.

Subramanian's model [PS2] considers all the activities that we have defined for this dimension, and takes into account the transition process activities from the ISO/IEC/IEEE 12207 standard [14].

The proposals of Tian et al. [PS3], Tyndall [PS4], Song et al. [PS8], Dubois et al. [PS19], Falazi et al. [PS10], Saatkamp et al. [PS11], García-Galán et al.[PS14], Deshmukh et al.[PS15] and Günalp et al. [PS16], cover only the activity of installing the software sytem deployment process.

In their study on benchmarking, Toufaili et al. [PS5] consider the following activities: installation, preparing the environment, test procedures, migrate data, initial data load, training system, preparing the documentation, and an acceptance test.

The proposal put forward by Carrizo et al. [PS6] considers all activities that we have defined for this dimension in Table 3.

Paredes et al. [PS7], do not cover any of the activities of the software system deployment process in their work.

Fein et al. [PS12] consider the activity of preparation of the documentation for the deployment process.

In Reascos et al. [PS13], the authors present a model that takes into account all the activities defined in the classification scheme shown in Table 3.

Despite the fact that there are few primary studies, ten of them focus on providing solutions to the installation activity in the software system deployment process.

RQ3: In what other technical processes and technical management processes related to the deployment process are the contributions made?

Only two papers mentioned technical management processes. Subramanian [PS2] considered a verification process in his model, to assist in understanding software requirements and in obtaining sizing estimates. He also identified risks to the organization's information security that may be caused by the deployment of the software. Reascos et al. [PS13] address the whole management process with reference to the processes defined in the ISO/IEC/IEEE 12207 standard [12]. Carrizo et al.´s methodological guide [PS6] considers the entire management process in relation to the activities defined in the classification scheme shown in Table 3.

RQ4: What types of research are used?

Of the primary studies analyzed, eleven of them correspond to "evaluation research" [PS2,PS6,PS7,PS8,PS9,PS10,PS12,PS13,PS14,PS16], four to the "validation type research" [PS1,PS3,PS4,PS11] and one corresponds to a "personal experience" [PS5].

In [PS2] applied the generic deployment model to the deployment of three common types of software: web applications, SaaS, and mobile applications, to attempt to understand the extent of this model's applicability in the real context.

In [PS6] to justify the adoption of their methodology, the authors conducted two types of *ex post* evaluations of deployments of systems in the University of Atacama: a) a validation list, the purpose of which was to compare the deployment carried out with three software systems against their methodology, and b) a survey carried out on the deployers of another two software systems. .

In [PS7] a case study was carried out at a vehicle sales company in northern Ecuador, aiming to understand how small and medium enterprises implement new software applications in their organizations. In the case study only a set of three interviews was codified and processed, the purpose being to fine-tune future interviews. The authors considered that for their study to be complete, it would be necessary to conduct interviews with other actors in the company. In this research in progress, the authors were able to study in depth only the problem of the deployment of IT enterprise applications.

[PS8] proposes an automatic tool for the configuration and deployment of software applications in the cloud. A case study is presented in which a cloud application is used to show how developers use the tool to specify their applications, as well as to demonstrate how operators use the application to configure and deploy the applications automatically.

In [PS9] the authors' model is evaluated under different scenarios inspired by a real system; the results show that their model-driven application refactoring reduces deployment costs by up to 60% when compared to the results produced by the same approach but without their model's being employed.

In [PS10] deployment models were introduced and the prototype was assessed in real-world use cases by evaluating the costs and additional execution times incurred when using it.

The tool presented in [PS12] demostrates a significant saving in training costs, a 20–30% improvement in productivity, and positive results in large Oracle and SAP implementations.

The model proposed in [PS13] is grounded on the findings of a case study carried out in a medium-sized town that had recently gone through the deployment process of an ERP IT application. The study allowed critical features to be identified in different phases of the deployment process. The study mainly involved interviews with key participants in the process. The model proposed emerged during this study, accounting for the particular issues and concerns identified.

García-Galán et al. [PS14] present a real-world experience through ISA Packager, a generic tool to package and deploy SPLs (Software Product Lines); the case study presented was a local software company which specializes in building Supervisory Control And Data Acquisition (SCADA) system.

In [PS15] a case study is used to demostrate the validity of the installation method presented.

Günalp et al. [PS16] present their approach, validated in multiple projects. Results show the ability of their model to handle the initial provisioning, as well as the continuous adaptations of applications coming from different repositories such as application stores.

In [PS1] with the tool proposed for the automation of the installation of an application, the researchers have experimented with only two basic web applications, since this domain is easily manageable. Knowledge is still lacking as regards how to extend the use of the tool to other types of applications (IoT applications, mobile applications, etc.).

In [PS3] the authors proposed a new method to implement software automatically on a large scale, developing a prototype SDS software: Software Deployment System, to verify and test the method in a laboratory context.

In [PS4] best practices for software installation activity are presented; the examples are implemented using IBM Tivoli Endpoint Manager and Active Directory/Group Policy in the Pennsylvania State University.

The researchers in [PS11] presented a method to facilitate the redistribution of application components and thus reflect strategic decisions on the technical deployment layer, while also validating the practical feasibility by means of a prototype.

In [PS5] four companies are selected for the benchmarking study; one of them manufactures innovative and complicated systems that serve as a backbone for the IT sector, the second one provides the routing infrastructure, the third one manufactures software for automobiles and finally, the last one was a company developing a range of telecom products.

5 Threats to validity

In this section, we analyze the potential threats to the validity that could affect our study, with respect to the four categories suggested by Wohlin et al.[20].

Construct validity. In this SMS, in order to mitigate these threats, we described the meaning that we have given to the software system deployment process, based on internationally-recognized standards and methodologies [12],[15],[17].

Internal validity. To mitigate the internal validity concerns, a review protocol was created by the first author as part of the research of her doctoral thesis, and this was reviewed by the other two authors.

External validity. We decided to use only three search engines in our search of the journals and conference proceedings that are relevant and recommended for the software engineering field. We did not consider grey literature, such as papers available only in the form of abstracts, PowerPoint presentations, PhD theses, or books, because including these might have affected the validity of our results.

Reliability. We attempted to mitigate the publication bias by carefully defining (a) our inclusion and exclusion criteria so as to be able to select primary studies and (b) our exclusion criteria, endeavouring to select rules in this work, based on our predefined research questions. To increase reliability, the first author was responsible for applying these criteria, with the help of the other authors whenever there was any disagreement regarding the inclusion or exclusion of a rule. The entire systematic procedure was performed by the first author. In parallel, the second author replicated the selection process and obtained a set of primary papers. The two sets of primary papers were checked by all authors; discrepancies between them were discussed, the purpose being to determine whether it was appropriate to include a particular paper or not, and the final list was thereby obtained. Moreover, we created a data spread sheet and mapped research questions with the data extraction properties in order to comply with the objectives of this study. We believe that the potential effect of this bias has less importance in mapping studies than in Systematic Literature Reviews. To further substantiate the search process, the snowballing method was applied, but unfortunately it did not allow us to find more primary studies than those found in the initial search.

6 Conclusions and future work

This paper presents the results of an SMS we carried out to discover the state of the art of software system deployment processes. We selected 16 relevant primary studies from an initial set of 3,483 papers, searching in Scopus, IEEE Xplore and the ACM digital library from 2010 until October 2019. After analyzing these primary studies, we can conclude that:

- The tools proposed deal only with aspects of the installation, in an attempt to automate this activity in the process and thereby reduce costs and time.
- 38% (6 studies) of the studies found propose methods, models and practices that cover only the automation of the installation activity.
- Only 25% (4 studies) of the primary studies found contemplate activities and tasks for the deployment of the software system, and only 19% (3 studies) consider technical management processes for the deployment of the software system.
- Of the primary studies analyzed, 69% (11 studies) correspond to the type of research known as "evaluation research"; this evaluation was carried out in the real context through case studies, surveys, and experiments with prototypes. In the "personal experience" research type, two significant points are highlighted. The

first refers to the fact that the software system deployment process is a life cycle process that is not studied as often as the others, while the second refers to the need to create a model for the execution of the process so that both the process and the roles that are specific to it can be systematized, given the diversity of the roles that participate in the processes of the companies studied.

- We discovered two process models that could serve as a guideline for software companies when carrying out the entire deployment process. These models have a certain limitation, which is that they delegate to the organisations that apply them the responsability for making decisions on a series of aspects linked to their deployment. These include tasks, artefacts, techniques, methods, tools and the definition of roles.This delegated responsability makes the application of these models in SMEs potentially more difficult, as these kinds of organisations need processes that are more detailed or more descriptive, and thus easier-to-apply.

Having analysed the existing literature, our long-term research work will consist in defining a software system deployment process model for deploying enterprise systems. This model can be coupled to the software development methodologies that these SMEs use, and will provide detailed guidance on the phases, activities, tasks, templates and roles for systematizing the deployment of the enterprise's software systems. In addition, we plan to validate the proposed model through case studies in Argentine software development companies.

Acknowledgements. The research work presented in this paper has been developed within the following projects: GEMA ("*Consejería de Educación, Cultura y Deporte de la Dirección General de Universidades, Investigación e Innovación de la JCCM"*, SBPLY/17/180501/000293, Spain) ECLIPSE ("*Ministerio de Ciencia, Innovación y Universidades, y Fondo Europeo de Desarrollo Regional FEDER*", RTI2018-094283-B-C31, Spain).

References

1. Hisham M. Abushama. PAM-SMEs: process assessment method for small to mediumenterprises. Software: Evolution and Process, 28, pp. 689 –711 (2016).
2. Sharma P., Sangal A.L. Building a hierachical structure model of enablers that affect software process improvement in software SMEs-A mixed method approach. Computer Standards & Interfaces, 66, pp. 1–23 (2019).
3. Ianzen A., Mauda E.C., Paludo M.A., Reinehr S., Malucelli A. Software process improvement in a financial organization: an action research approach. Computer Standard & Interfaces, 36, pp 54–65 (2013).
4. Subramanian N. The software deployment process and automation. CrossTalk, 30(2), pp. 28-34 (2017).
5. Carrizo D., Sanchez L. Benchmarking to Adopt an Asap-Based Methodological. Guideline for Software Systems Deployment. In: Proceedings of the 30th International Business Information Management Association Conference, (IBIMA '17): Sustainable Economic development, Innovation Management, and Global Growth, pp. 183-193 (2017).
6. Fuggetta A., Di Nitto, E. Software Process. In: Proceedings of the 36th International Conference on Software Engineering - Future of Software Engineering (FOSE'14), pp. 1-12 (2014).

7. Erich F.M.A., Amrit C., Daneva M. “A Qualitative Study of DevOps Usage in Practice”. Software: Evolution and Process, 29, pp. 1-20(2017).
8. Bass L., Weber I., Zhu L. DevOps: A Software Architect's Perspective (2015).
9. Agile Alliance. Agile Glosarry. https://www.agilealliance.org. Last accessed 2020/02/25.
10. Kitchenham B., Budgen D., Brereton P. Evidence-Based Software Engineering and Systematic Reviews, Chapman and Hall 1 st.Editon. Chapman and Hall/CRC. USA (2015).
11. Petersen K., Feldt R., Mujtaba S., Mattsson M. Systematic mapping studies in software engineering, In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, pp. 68–77 (2008).
12. IEEE ISO/IEC/IEEE 12207:2017(E). Systems and software engineering — Software life cycle processes (2017).
13. Wieringa R., Maiden N., Mead N., Rolland C. Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. Requirements Engineering, 11, pp. 102–107 (2005).
14. Petersen K., Vakkalanka S., Kuzniarz L. Guidelines for Conducting Systematic Mapping Studies in Software Engineering: An Update. Information and Software Technology, 64, pp. 1-18 (2015).
15. Jacobson I, Booch G., Rumbaugh J.The Unified Software Development Process. 1 st Edition. Addison Wesley, USA (1999).
16. IEEE ISO/IEC/IEEE 2475:2010(E). Systems and software engineering - Vocabulary (2010).
17. Agile Business Consortium. The DSDM Agile Project Framework (2014 Onwards). https://www.agilebusiness.org/resources/dsdm-handbooks, last accessed 2019/05/29.
18. Panizzi M., Genero M., Bertone R. Appendix - Software system deployment process: A systematic mapping study. https://doi.org/10.6084/m9.figshare.11922165.v1
19. Wohlin C. Guidelines for snowballing in systematic literature studies and a replication in software engineering, In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, pp. 1–10 (2014).
20. Wohlin C., Runeson P., Hst M., Ohlsson M.C., Regnell B., Wessln A. Experimentation in Software Engineering. Springer Publishing Company (2012).