- ORIGINAL ARTICLE -

Refining a Software System Deployment Process Model through Empirical Studies

Refinamiento de un Modelo de Proceso de Despliegue de Sistemas de Software a través de Estudios Empíricos

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

[1] *School of Computer Science, Computer Science Research Institute LIDI (III-LIDI), Universidad Nacional de La Plata, La Plata, Argentina.*
marisapanizzi@outlook.com; pbertone@lidi.info.unlp.edu.ar
[2] *Department of Technologies and Information Systems, University of Castilla-La Mancha, Ciudad Real, Spain.*
marcela.genero@uclm.es
[3] *Department of Engineering Information Systems, Universidad Tecnológica Nacional – Facultad Regional Buenos Aires, Argentina.*

Abstract

Software system deployment describes the activities associated with ensuring that a software system is available for its end users. Every company, regardless of its size, requires an efficient and effective software system deployment process to ensure the customer will accept the system software successfully. Small and Medium Enterprises (SMEs) often operate on limited resources and with strict time constraints and need to improve their processes. For this reason, the existing proposals for deployment processes are not usually useful for SMEs. This fact led us to propose DepProMod (Deployment Process Model) to help SMEs to execute the deployment process of software systems in a systematized and controlled manner. The initial version of DepProMod has subprocesses, activities and tasks defined in addition to a capability-level architecture which allow its implementation in a step-by-step manner, thus adapting to SMEs with different needs and resources. This paper presents the results of two empirical studies carried out to obtain feedback on the model so as to refine it and complete it. Specifically, a case study from an Argentinian level 1 Medium-size software development SME (approximately 55 employees) and the interviews held with two professionals from the software industry are presented. In summary, both empirical studies allowed us to modify, unify and eliminate elements in the model.

Keywords: Software Processes, Software System Deployment Process Model, Case Study, Interviews.

Resumen

El despliegue de sistemas de software describe las actividades asociadas para asegurar que un sistema de software esté disponible para sus usuarios finales. Cada empresa, independientemente de su tamaño, requiere un proceso de despliegue de sistemas de software eficiente y efectivo para garantizar que el cliente acepte el sistema de software con éxito. Las pequeñas y medianas empresas (PyMES) a menudo operan con recursos limitados y con estrictas limitaciones de tiempo, y necesitan mejorar sus procesos. Por este motivo, las propuestas existentes de procesos de despliegue no suelen ser de utilidad para las PyMES. Este hecho nos llevó a proponer DepProMod (Modelo de Proceso de Despliegue) para ayudar a las PyMES a ejecutar el proceso de despliegue de sistemas de software de manera sistematizada y controlada. La versión preliminar de DepProMod cuenta con subprocesos, actividades y tareas definidas además de una arquitectura de niveles capacidad que permite su aplicación de manera escalonada, adaptándose así a pymes con diferentes necesidades y recursos. Este artículo presenta los resultados de dos estudios empíricos realizados para obtener retroalimentación del modelo con el fin de refinarlo y completarlo. Específicamente, se presenta un estudio de caso de una PYME desarrolladora de software, mediana de tramo 1 (aproximadamente 55 empleados) de Argentina y las entrevistas realizadas a dos profesionales de la industria del software. En resumen, ambos estudios empíricos nos permitieron modificar, unificar y eliminar elementos en el modelo.

Palabras claves: Procesos de software, Modelo de proceso de despliegue de sistemas de software, Estudio de caso, Entrevistas.

1. Introduction

Small and Medium Enterprises (SMEs) need efficient and effective software engineering solutions. The proper implementation of software engineering techniques is however a difficult task for SMEs as they often operate on limited resources and with strict time constraints [1]. For this type of organizations, it is crucial to improve their processes and work methods as they account for the highest worldwide percentage of software development companies [2].

Deployment is a crucial process of the software development life cycle. After successful deployment, the software system is finally operational so that the customer can benefit economically from its use. At the end of this deployment effort, the software development organization receives payment from the customer and the project is considered successful from both the developer's and the customer's viewpoints [3].

In [4], it is stated that the result of non-standardized or inadequate deployment practices is reflected on the information systems that prove difficult to maintain and operate. In [3], it is postulated that an inadequate deployment execution causes time losses and increases costs in the maintenance phase to solve problems that pertain to the deployment phase. In the deployment phase of software systems, problems may arise, which may lead to users reporting software system errors after installation. In addition, the technical requirements for the installation environment are sometimes not correctly interpreted by the client company [4, 5].

There are automation solutions to improve the last stages of the life cycle [6], among which we can mention new techniques/practices such as DevOps [7] and Continuous Deployment [8] in the context of agile methodologies. Google, Amazon, Netflix, LinkedIn, Facebook, and Spotify are some examples of successful companies whose DevOps practices have been reported and disclosed in IT books, blogs and events [9]. These emerging solutions are not viable for many SMEs due to the lack of human resources and infrastructure that would allow them to adopt such solutions.

The objective of the research conducted in this paper, firstly focused on establishing the state of the art with respect to the process of deploying software systems. Secondly, the studies carried out for the validation of the preliminary version of DeProMod were presented.

This article is an extension of an article presented at CACIC 2021 [10] which was selected among the best articles. The main differences between the current article and that of CACIC 2021 are:

1) The analysis of the state of the art carried out to build DepProMod is extended, and an experts review of suggested lectures is included.
2) The elements of the preview version of DepProMod are presented in more detail.
3) A new empirical study based on interviews with two software engineering experts is presented as a means of obtaining their views on the elements of the model with a view to refining it.

An exhaustive analysis of the state of the art in the deployment of software systems was conducted and presented in Section 2 and from this, a preliminary version of DepProMod was built.

The rest of the paper is organized as follows: Section 3 presents an overview of DepProMod. The case study design and results are presented in Section 4. Section 5 presents the results of the interviews conducted with professionals in the software industry and, finally, our conclusions and proposals for future work are set out in Section 6.

2. State of the art

In order to study the state of the art, an analysis of the existing literature on software system deployment processes is conducted by means of a SMS (*Systematic Mapping Study*), presented in Section 2.1. Then, Section 2.2 describes an exploratory study, based on a survey, about the current practice regarding software system deployment processes in SMEs in Argentina. Section 2.3 presents the literature review recommended by Software Engineering experts regarding deployment solutions.

2.1 Systematic Mapping Study

A SMS was performed to review the state of the art and to identify models, methodologies or methods which might serve as a guide for SMEs when deploying software systems [11]. The development of the SMS followed the guidelines proposed in [12, 13].

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

Table 1. Research questions.

RQ1	What contributions have been made with regard to the software system deployment process?
RQ2	In what activities or tasks of the software system deployment process are the contributions made?
RQ3	In what other technical processes and technical management processes related to the deployment process are the contributions made?
RQ4	What types of research are used?

The search string was built by choosing three major search terms: *"Deployment"*, *"Process"* and *"Software"*. We considered the firs0t major term from RUP [14] and alternative term employed in one internationally-recognized standard: the term 'transition'', from the ISO/IEC/IEEE 12207 standard [15]. 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 carrying 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 [14], [17] and earlier versions of ISO/IEC/IEEE 12207 standard [15].

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.

The study selection process consisted of the following steps: 1) carrying out a search in the three sources using the search string in the title, 2) removing duplicate papers, 3) applying the inclusion and exclusion criteria to the title, abstract and keywords 4) applying 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).

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/IEEE 12207 standard [15].
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 [15].
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 [18].

The SMS allowed us to study the state of the art in the deployment process of software systems. In this

study, 16 primary studies were selected from an initial set of 3483 articles.

After the analysis of the primary studies, it is concluded that:

- The tools proposed only address installation issues in an attempt to automate the process activity and therefore reduce costs and time.
- 38% (6) of the total number of studies found proposed methods, models and practices which only cover the automation of the installation activity.
- Only 25% (4) of the primary studies analyzed refer to activities and tasks of the deployment process of software systems, and only 19% (3 studies) consider technical management processes for the deployment of software systems.
- 69% (11) of the primary studies analyzed correspond to the "evaluation" type of research. Such evaluation was conducted in a real context by means of case studies, interviews, and experiments with prototypes. In the "personal experience" type of research, there were two significant points to mention. The first is the fact that the deployment process of software systems is one of the processes of the software development life cycle, which is not studied as frequently as the rest of the processes. The second point refers to the need of creating a model for the execution of the deployment process so that both the process and the specific roles can be systematized, given the diversity of roles participating in the processes of the companies studied.
- Two process models [5], [19] and one methodology [6] were found, which serve as a guide for software companies when conducting a deployment process. These models have the limitation that they delegate to the organizations the responsibility of making decisions on a number of deployment aspects. These aspects include artifacts, techniques, methods, tools, and the definition of roles because these models only include tasks. The delegation of such decisions makes these models more difficult to implement in SMEs since they require more detailed or descriptive processes to facilitate their implementation.

2.2 Survey

This section presents the results of the survey-based exploratory study performed to collect evidence on the current practice involving the deployment process of software systems in SMEs. In order to conduct the survey, the guidelines provided by Molléri *et al*. [20] were followed. It was carried out between July 22 and August 31, 2020, and it included 195 professionals from the software industry working in different SMEs in Argentina. The goal established for the survey was achieved, that is, the collection of evidence on the current practice regarding the deployment process of software systems.

The main findings obtained [21] for each RQ are detailed below:

RQ1: *What are the characteristics of SMEs considering the difference between those that have a deployment process established and those that do not?*

Of the total of 195 SMEs participating in the survey, 105 do not have an established deployment process, of which 51 are software development companies and 54 are in other types of business and have a systems area which develops software for their own use. The "micro" SMEs have a marked tendency not to have a deployment process in place.

Both SMEs that have an established deployment process and those that do not use the incremental or hybrid development models as life cycle models for their software projects.

Scrum is the most widely used framework for both SMEs with and without a deployment process. There is a tendency for SMEs without an implementation process not to use frameworks and methods in their software projects.

RQ2: What are the problems found in the deployment process?

SMEs without a systematized deployment process encounter more problems in the deployment activities compared to SMEs with a process in place. Among the difficulties encountered in the deployment activities, there is a first group of activities that are not deployment-specific but rather related to project management, the most common being time estimation, assignment of human resources, configuration management and deployment closing. In turn, the group of activities with deployment-specific difficulties include user training, migration, preparation of the installation environment, installation, acceptance tests, client documentation and non-compliance, technical requirements, installation tests, technical training, and data entry.

SMEs without a deployment process in place have more aspects affected by difficulties in the deployment activities. The most frequently affected aspects are delays and re-work, followed by the quality of the process and client unsatisfaction and, finally, aspects such as incomplete installation, lack of productivity, project closing, reassignment of human resources and management.

RQ3: What are the characteristics of the deployment process?

Firstly, the deployment process of the 90 SMEs that have such a process mainly has activities and tasks. Secondly, it includes the use of technological tools. Thirdly, the use of documentation and, finally, a small number of SMEs use practices, techniques, metrics, and methods.

In the case of SMEs that do not have a deployment process, the manual installation method is

approximately 50% more frequent than in those that do have one. The tools most frequently used by SMEs that have a deployment process are *Jenkins, Gitlab* and their own software.

The specific role most used by SMEs is that of "implementer" and in second place, the person in charge of "DevOps". The 129 SMEs that do not have specific roles for deployment assign roles such as developers, analysts, testers, software architects, and technicians, among others.

There are 115 SMEs using documentation in the deployment process and 80 SMEs that do not. And, to a greater extent, the documentation is used in the installation activity.

RQ4: Would it be desirable to have a well-defined deployment process?

In order to improve the deployment process, the survey participants suggested, in the first place, the automation and use of technological tools. Secondly, they considered the proposal of trained and well-defined roles, as well as the need to have a process, guidelines, or a methodology. And, finally, they recommended the incorporation of the deployment process in the study plans of university degree programs in greater depth, the use of certain tools, the use of DevOps, among others. The participants with the greatest experience in the software industry also considered that the use of documentation, planning and control of the process, teamwork, communication, and synchronization of the participants in the deployment, the improvement of times and specific calendars for the deployments and test environments may improve the deployment process. Most of the participants find it useful to have a deployment process, as well as the existence of guiding templates for the execution of the process.

The results of this survey confirmed the need for a software system deployment process model that helps SMEs, particularly micro and small ones, to carry out the deployment in a systematic way through: a) the execution of well-defined activities and tasks, b) the use of guiding templates, c) the assignment of specific roles that have the necessary skills to execute the deployment, and d) the use of tools to automate some of the activities of the process with the purpose of speeding up and automating the process.

2.3 Literature suggested by experts

Moreover, we analyzed a set of software development methodologies and standards suggested by experts in Software Engineering taking into account the deployment process. Among these are the ISO / IEC / IEEE 12207 standard [15], RUP [14], AUP [22], Metrica v3 [23] and DSDM [17] methodologies. A comparison of a series of elements proposed by these solutions was made. The following elements of the deployment process were considered as features for analysis: “phases / activities / tasks”, “artifacts”, “roles”, “tools”, “techniques / practices” since our objective was to know the level of compliance with which these methodologies or standards support the deployment process.

From the comparative analysis, it is concluded that (see Table 4):

- Both the RUP, AUP, DSDM, Metrica v3 methodologies and the ISO / IEC / IEEE 12207 standard propose phases, activities, and tasks for the execution of the deployment process, as well as other processes that support its management.
- Regarding artifacts, the ISO / IEC / IEEE 12207 standard does not propose artifacts. In turn, although both RUP and AUP propose them and also make their design explicit, they require the use of the Rational suite. In the case of DSDM and Metrica v3, they propose artifacts and their objectives, but they do not provide templates to guide an SME.
- RUP, AUP, DSDM and Metrica v3 propose roles for the deployment process, while the ISO / IEC / IEEE 12207 standard does not have roles, since the standard has the limitation of delegating decisions to the stakeholders involved in the project, as well as the methods, techniques, and practices to be implemented.
- The only methodologies that propose automated tools are RUP and AUP. Although this is an advantage for companies that have sufficient financial resources to acquire them, for an SME it becomes a disadvantage due to the limited capacity of resources for their acquisition.
- Both RUP and AUP have good practices such as the visual modeling offered by UML through its diagrams, including the deployment diagram. Although DSDM proposes the “Facilitated Workshop” practice, which is used for project management activities, it does not explicitly mention techniques / practices for the deployment process. Metrica v3 has techniques / practices for the deployment process, as well as for its management activities.

To conclude, ISO / IEC / IEEE 12207 standard [15] does not have any artifacts, roles, techniques / practices, and tools because their selection is delegated to the stakeholders involved in the project. RUP [14] and AUP [22] have artifacts, roles, techniques / practices, and tools, but they are limited to using Rational. Metrica v3 [23] is the most complete but its robustness is not appropriate for its implementation in an SME. DSDM [17] has artifacts, roles, and techniques / practices but they focus on deployment management and not on technical aspects of the deployment.

All of the above considerations led us to define the objective of our long-term research, that is, to propose a holistic software system deployment process model to help SMEs execute the deployment process of software systems in a systematized and controlled manner.

Table 4. Comparison of methodologies and standards.

	ISO/IEC/IEEE 12207	Rational Unified Process	Agile Unified Process	Metrica v3	Dynamic System Deployment
Process to analyze	Transition	Deployment	Deployment	Implantation and Acceptance	Deployment
Phases/ Activities/ Tasks	x	x	x	x	x
Artifacts		x	x	x	x
Roles		x	x	x	x
Tools		x	x	x	
Techniques/ Practices		x	x	x	

Our preliminary version of the model was called Model of a Computer Systems Implantation Process (MoProIMP) [24], but since it was not compatible with the international terminology or with the methodologies that refer to this phase of the software development life cycle, we decided to rename it to DepProMod (Deployment Process Model) and this acronym will be used hereinafter for the entire paper.

DepProMod was developed to respond to the software system deployment process problem in SMEs in Argentina although the feasibility of extending it to the international context will be studied later.

3. Overview of DepProMod

The preliminary version of DepProMod has a life cycle model that adopts the 5 PMBOK process groups [25]. These groups are: Initiating, Planning, Executing, Monitoring and Controlling and Closing. The reason for this choice is that PMBOK is a globally recognized standard for use in the software industry. Each of these processes in DepProMod is called a "subprocess".

For the definition of the activities of DepProMod, a set of processes from the ISO / IEC / IEEE 12207 standard [15] were considered. The processes extracted from the standard are the technical management processes: risk management, configuration management, project management, and other technical processes: verification and validation. In our model, these processes are called "activities". At the "tasks" level, the model adopts a group of tasks proposed in the Metrica v3 [23] methodology as it is considered one of the most complete methodologies at the level of the tasks that are executed in the deployment process and those used in Spain and Latin America. In addition, a series of activities proposed in the "transition" technical process of the ISO / IEC / IEEE 12207 standard [15] were considered.

In order to implement the model in a step-by-step manner, three of the capability levels were adopted from the CMMI-DEV standard [26]. These levels are: level 1 = Done, level 2 = Managed and level 3 = Defined. Level 0 = Incomplete was not considered since it means the non-completion or partial completion of that process in the organization. These levels were analyzed and defined at a granularity level of the tasks considered in the model. The choice to consider capability levels rather than maturity levels is due to the fact that not all software development companies have reached maturity levels 4 and 5. This tiered architecture offers the advantage that software development companies can implement it in a step-by-step manner and, as they manage to stabilize the process at one level and achieve the necessary knowledge for its implementation, they can scale it to the next level.

The process pattern used for the representation model is the one proposed in the Competisoft model [27] since it is a process improvement model for Ibero-American SMEs in the software industry with some adaptations to the needs of the DepProMod definition.

We believe this model could be coupled to the software development methodologies used by these SMEs, as long as they apply them gradually until the knowledge of human resources and the maturity of the process are achieved. So far, however, we have only applied it to SMEs that follow traditional software development methodologies. Therefore, in the future, we plan to apply it to SMEs using modern/contemporary software development process, such agile methodologies.

DepProMod has a total of 16 activities distributed in 5 subprocesses. In Fig. 1 the relationships between the DepProMod subprocesses are presented and the activities of each subprocess are detailed

DepProMod has a total of 45 tasks grouped by subprocesses and activities. Table 5 presents the Subprocess: INI – Initiating with its activities and tasks. The purpose of this subprocess is to learn about the software project and the organization's processes to define the deployment of a software system.

Fig. 1. Relationship between DepProMod subprocesses

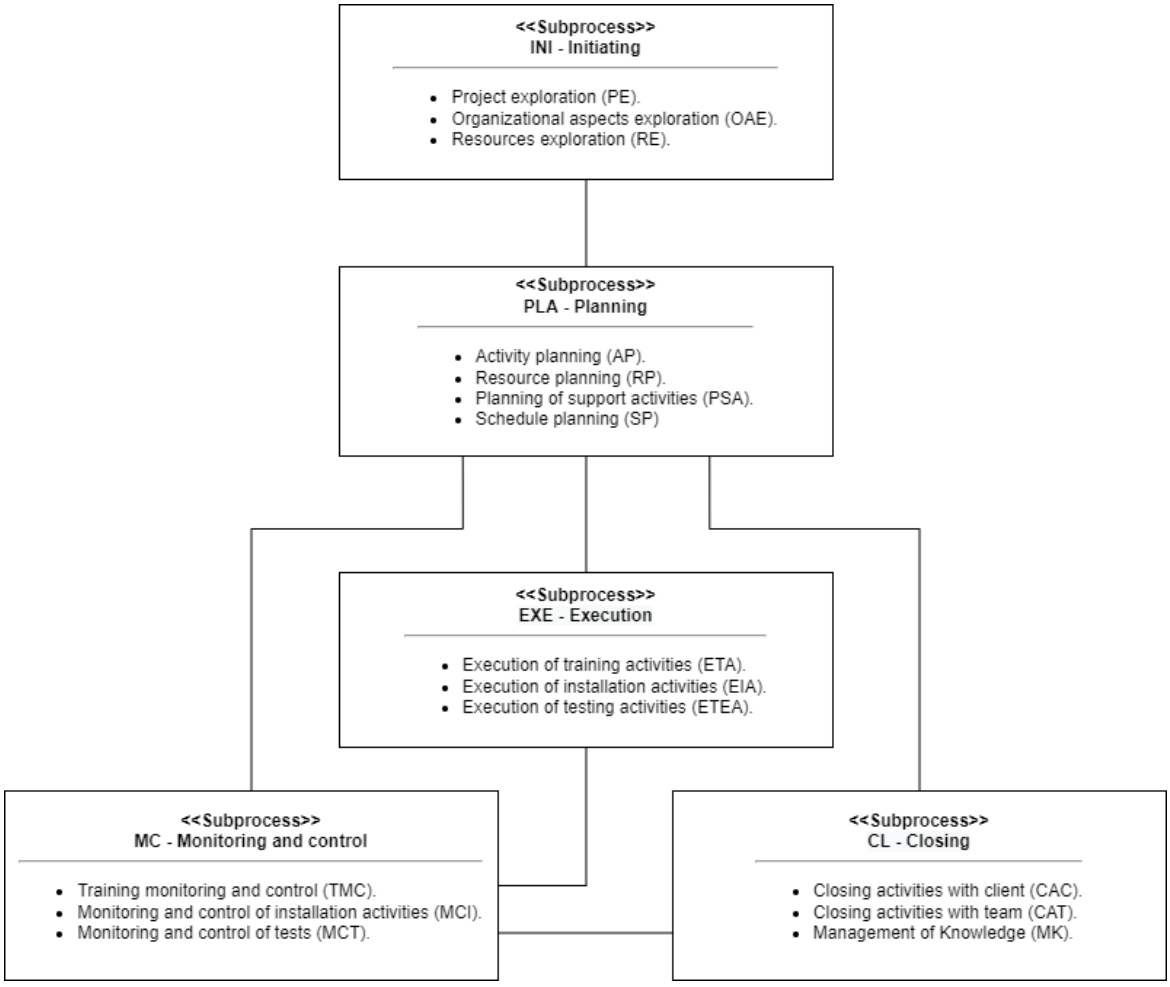

Table 5. Subprocess: INI – Initiating, activities and tasks.

Tasks by activity
PE. Project Exploration
PE.1. Understanding the general characteristics of the project. PE.2. Understanding the requirements specification. PE.3. Understanding the system specification. PE.4. Understanding the operation of the software product. PE.5. Understanding the characteristics of the deployment.
OAE. Organizational Aspects Exploration.
OAE.1. Learning about the configuration management. OAE.2. Learning about the documentation policies. OAE.3. Learning about the communication policies. OAE.4. Defining the configuration management, and the communication and documentation protocol.
RE. Resources exploration
RE.1. Identifying the existing resources.

Table 6 presents the Subprocess: PLA – Planning, whose purpose is to establish the full scope of the deployment process effort, define objectives, and develop the strategies required to achieve those objectives. In this subprocess, the deployment management plan is developed.

Table 6. Sub-process: PLA – Planning, activities and tasks

Tasks by activity
AP. Activity Planning
AP.1. Defining the activities of the deployment plan. AP.2. Validating the deployment proposal. AP.3. Defining testing activities. AP.4. Validating the activity plan.
RP. Resource Planning
RP.1. Assigning human resources. RP.2. Determining technological resources. RP.3. Validating resources. RP.4. Determining training activities. RP.5. Validating training.

Tasks by activity
PSA. Planning of support activities
PSA.1. Defining metrics. PSA.2. Defining risks. PSA.3. Defining knowledge base. PSA.4. Validating support activities.
SP. Schedule Planning
SP.1. Estimating the schedule of the activities. SP.2. Validating schedule.

Table 7 presents, the Sub-process: EJE- Execution whose purpose is to complete the work defined for the deployment process defined in the planning phase. This sub-process implies the coordination of people and resources, the management of the expectations of stakeholders, as well as the integration and execution of the activities of the plan defined for the management of the deployment process.

Table 7. Subprocess: EJE – Execution, activities and tasks.

Tasks by Activity
ETA. Execution of Training Activities.
ETA .1. Training users. ETA .2. Training technicians. ETA .3. Managing training progress.
EIA. Execution of installation activities.
EIA.1. Preparing the installation environment. EIA.2. Installing the software. EIA.3. Performing data upload and/or migration. EIA .4. Managing installation progress.
ETEA. Execution of Testing Activities
ETEA.1. Performing installation tests ETEA.2. Performing user acceptance tests. ETEA.3. Managing testing progress.

Table 8 presents the Sub-process: MC - Monitoring and control, whose purpose is to review and organize the progress and performance of the deployment process, identify any areas where changes to the plan are required, and initiate changes accordingly.

Table 8. Subprocess MC: monitoring and control, activities and tasks.

Tasks by Activity
TMC. Training monitoring and control.
TMC.1. Evaluating training activities. TMC 2. Managing support activities.
MCI. Monitoring and control of installation activities.
MCI.1. Evaluating installation activities MCI 2. Managing support activities.
MCT. Monitoring and control of tests.
MCT.1. Evaluating testing activities. MCT.2. Managing support activities.

Table 9 presents the Sub-process: CD – Closing, whose purpose is to formalize the completion of the deployment process.

Table 9. Subprocess CD: closing, activities and tasks.

Tasks by activity
CAC. Closing activities with client.
CAC.1. Formalizing deployment closing with client.
CAT. Closing of team activities.
CAT.1. Formalizing deployment closing with team. CAT.2. Holding meeting with team.
MK. Management of Knowledge.
MK.1. Managing learned lessons.

Table 10 presents the roles defined for DepProMod along with their acronym, name, description and competencies.

Table 10. DepProMod roles.

Role Name	Description and competencies
Project Leader (PL)	The person responsible for the project on behalf of the client that requires the software system deployment process. It can be an external or internal customer. They have the knowledge to request changes.
Key User (KUS)	The person or persons who know the business or processes, validate the tests, validate the training plan and accept the closure of the deployment.
User (US)	The person or persons who operate or interact directly with the Software System.
Deployment Leader (DL)	The person in charge of carrying out the deployment and complying with the objectives. They have knowledge and experience in project management, decision making, review techniques, software development, and estimating and costing techniques.
Installer (INS)	The person responsible for carrying out the activities related to the preparation of the installation site,

Role Name	Description and competencies
	installation, loading and/or migration of data and tests. They have knowledge and experience in infrastructure, software development and configuration management.
Trainer (TR)	It is the person responsible for carrying out training activities, as well as preparing training schemes (platforms, strategies and modalities). They have knowledge and experience in developing training activities and configuration management.

In addition, DeproMod defines the tools to be used in the tasks, which include configuration management, installation, documentation, deployment plan management, training, and design tools. The tools are proposed based on their usefulness, and the SMEs are free to use the tools they have.

4. Case Study Description

In this section, we present a detailed description of the case study following the guidelines proposed in [28, 29].

4.1 Case study design and research questions

The main goal of our case study is to examine the feasibility of the implementation of the DepProMod preliminary version in a real environment with the purpose of refining it (if necessary) and completing it. This case study is of an exploratory type [30] because it makes it possible to find out what is happening in the deployment process, seeking new points of view and generating ideas and hypotheses for our research. We worked with the documentation of the deployment process of the "Company creation" module of a management system for advertising agencies for Latin America to analyze the information requirements of the software system deployment process and thus move towards the design of the necessary templates for our model. We believe this case study is suitable to find the information requirements of the software system deployment process.

To achieve our goal, we posed the following research questions (RQ):

RQ1: Is it necessary to refine the model to adapt it to the existing needs in the industrial context?

Through this question, we sought to obtain the information needs for the execution of the tasks carried out by the consulting company in the deployment process to compare them with our model in order to refine it and complete it.

RQ2: Was the implementation of the model useful for the company?

With this question we tried to determine how the consulting company can strengthen its software system deployment process. For this purpose, we will provide a set of specific recommendations for the process as well as Software Engineering practices in general.

The context, the case and the unit of analysis of the case study are described below. It is a holistic single-case study [31] and follows the classification by Yin (2002):

- Context: Although our model arose as a response to the need of SMEs to improve and stabilize their software system deployment process, the case study available to us involved an SME (55 employees) according to the classification proposed in [32], located in Argentina, which offers consulting products and services. This company uses a development methodology with an iterative-incremental life cycle model, with the conventional stages: Analysis and Design, Construction, Testing and Implementation. In each stage, product/s-artifact/s are built to continue with the next stage. They also incorporate some practices of agile methodologies such as extreme programming (XP), pair programming. The first author of this work had access to the company's facilities and project documentation subject to an agreement not to disclose the company's name as well as a commitment to inform about the findings and recommendations to be considered.
- Case: Deployment of the "Company creation" module (in a new country) of the management system of advertising agencies for Latin America. This module corresponds to a management system called "T&C" that has the following modules: customers, suppliers, accounting, treasury, administration and parameters (module where master entities are created and the system is configured), expense reports and security. The module "Company creation" of the T&C management system that was implemented contains the following global features: creation of the company in the system and preparation of initial information and parameters to operate it. The specific features are: upload the general data of the company, enter the provinces or states, create the divisions, upload the people master file, upload the suppliers master file, upload the clients / brands / products / projects master file, among others.
- Unit of analysis: deployment documentation of the "Company creation" module of the advertising agency management system called "T&C".

4.2 Preparation for data collection

The third-grade collection technique was used according to the classification proposed in [33]. Qualitative data were collected from the documentation used in the deployment of the "Company creation" module of the T&C management system, which was obtained from different sources and / or repositories of the project.

In order to facilitate the preparation of the documentation to be collected, a data collection template was defined with a coding scheme according to the template approach mentioned in [1]. The template coding scheme is made up of a set of 5 groups, each of which coincide with the 5 subprocesses of DepProMod (Initiating, Planning, Executing, Monitoring and Controlling, and Closing).

For each group, a series of categories and their description were defined in [1]. Additional information of the coding scheme used is presented in [34].

4.3 Analysis and interpretation of results

Since this is an exploratory study, the "Hypothesis Generation" technique was used to analyze the data [29]. In this case study, we consider that the research verifies the following:

- Based on knowing the information required in the software system deployment process in a real context, DepProMod is refined and completed with the definition of templates for its tasks and,
- From the analyzed documentation, the company is provided with a set of recommendations of good practices to improve its deployment process.

Two columns were added to the template designed to collect the study data. The first, called "comments", was used to record additional information in the analyzed document. The second column, "recommendations", was used to record proposals for the deployment process analyzed (of the case). The information collected and analyzed is presented in Appendix [34].

Within the reviewed documentation, the content of the emails found in the Incident Follow-up System (IFS) was also analyzed, since this allowed the acquisition of information on relevant milestones of the project.

In total, twenty-one documents were analyzed. The review was conducted in a systematic way, each document was associated with the defined coding, seeking traceability of its use in the different groups defined in the coding. Each group corresponded to the subprocess defined in our model and each category corresponded to an aspect to be considered in its subprocesses, such as aspects of the project, the organization, etc. This method of analysis allowed us to contrast the information needs of a real case with our model and, simultaneously, to reflect on good practices recommendations to the consulting company.

4. 4 Results

Table 11 shows a summary of the documents reviewed for each DepProMod subprocess, which is represented by a cross at the intersection between the name of the document and the name of the DepProMod subprocess.

The results related to the research questions formulated for this case study are as follows:

RQ1: Is it necessary to refine the model to adapt it to the existing needs in the industrial context?

Based on the documentation analyzed, a series of requirements were obtained to complete the definition of DepProMod, which are presented below according to the subprocess structure:

Subprocess 1: Initiating. Five documents were reviewed. There was incomplete or inaccurate information which made it impossible to associate it with the deployment tasks. From this analysis, we consider that, in our model, it is necessary to design templates that allow unification of the information to be documented, with a clear objective of use, distribution and the definition of a person responsible for its creation, modification and approval.

Subprocess 2: Planning. Six documents with the information related to this subprocess were reviewed. There was information that could not be analyzed either because it was not found or was incomplete. In the documentation reviewed, only the use of two metrics, time and effort, was found. These are considered in our model along with others, such as productivity and error rate of installation tests. In contrast to our model, it was not possible to obtain new information because DepProMod would contemplate more specific metrics.

Subprocess 3: Executing. Eleven documents with information related to this subprocess were reviewed. There was no information related to data migration because it was the deployment of a new system module. For this subprocess, the model is enhanced by building the following templates in the previous subprocess (planning) which will be used in this subprocess: "deployment strategy", "guide for site preparation", "installation guide", "data migration", "data upload", "test specifications", "user acceptance testing", "required human resources", "required technological resources", "competencies of the technical team", "users to be trained", "metrics", "measurement report", "deployment risks" and "contingency plan". Within this process, the following templates will be designed: "end user assistance report", "technical team assistance report" and "activity report".

Subprocess 4: Monitoring and Controlling. Two documents with information related to this subprocess were reviewed. There was information

that was not found or was insufficient to contrast with our model. There was documentation that reflected the monitoring of activities (work plan) and the meeting memo was also reviewed, which includes the decisions made by the project participants. There was no information regarding those who participated in the training activities (users, trainers and technicians).

DepProMod will incorporate two templates that allow registering of the activities carried out as part of the "activity report" deployment that will be shared with the client and the information from the "report of risks occurred" and the "measurement report" will be updated.

Subprocess 5: Closing. Two documents containing information related to this subprocess were reviewed: one of them is the updated deployment plan and the other contains information on the installation activities.

There was no evidence of the closing of the training activities, the closing of the deployment team or the learned lessons. DepProMod proposes “acceptance document”, “closing report” and to register lessons learned in a knowledge base.

Table 11. Summary of reviewed documents.

Documents/ Subprocesses	**Subprocess 1: Initiating**	**Subprocess 2: Planning**	**Subprocess 3: Executing**	**Subprocess 4: Monitoring and Controlling**	**Subprocess 5: Closing**
Requirements procedure	x				
Project standards	x				
Work plan		x	x	x	x
Requirements for the installation site		x	x		
Instructions to structure the submission		x			
Installation test procedure		x			
Acceptance test procedure		x			
User’s manual			x		
Smoke test instructions			x		
Data entry instructions			x		
Installation script			x		
Acceptance test instructions			x		
New company application form			x		
Progress report			x		
Meeting memo				x	
Smoke test results			x		
Acceptance test results			x		
Installation completion report					x
T&C Project	x				
General documentation	x	x			
User requirements	x				

RQ2: Was the implementation of the model useful for the company?

The company found the DepProMod implementation useful since we provided a report with a set of recommendations to improve its deployment process for future projects as well as suggestions for good Software Engineering practices in general. These recommendations can be listed as follows:

- Use appropriate tools for the administration of the project plan since the project plan was managed with Excel.
- Analyze the deployment process strategy through a feasibility study.
- Expand the definition of metrics for the deployment process as well as for the rest of the software development processes since the only metrics they use are time and effort.
- Define risk management and its mitigation procedure.

- Effectively delegate the activities to be carried out by the client, since the preparation of the installation site was carried out by the client without adequate supervision by the consulting company.
- Create an institutional space to share knowledge not only regarding the deployment process, but also the rest of the processes of the software development life cycle.

4.5 Threats to validity

To analyze the validity of the case study, the factors proposed in [29] were taken into account:

- Construct validity. Results were obtained in relation to the information needs of a deployment process in a real context, which allowed us to answer the defined research questions, determining their pertinence and suitability for the case.
- Internal validity. The documentation used belongs to a real case, a deployment of a module of an advertising agency management system (T&C). To achieve greater precision and validity of the studied process, the need to combine the data source (project documentation) with another type of source, such as interviews and / or focus groups to ensure a "Data (Source) Triangulation", is recognized. Furthermore, the collected and analyzed qualitative data could be combined with quantitative data resulting from the project thus ensuring a "Methodological Triangulation".
- External validity. The use of a single case study may limit the generalization of the results. However, reporting on these first findings is considered necessary, as it serves as an incentive for other researchers to replicate our study in different case studies.
- Reliability. The study data were collected by a single researcher. Although they were analyzed with the thesis supervisors, this can be considered a threat to the research. To add a higher degree of reliability, it would be advisable for another researcher to apply the template with the coding created here in another case study.

5. Interviews with experts

This section presents a summary of the different interview sessions held with two professionals from the software industry aimed at presenting DepProMod to the experts, learning about their opinion on the definition of each element of the model and its usefulness, and refining it if necessary. At the time the experts were contacted, the model already included the templates. The interviews held with the experts have been semi-structured. First, the objectives for each of the sessions were set, the experts gave their consent to record the interviews and then they were transcribed and validated with the experts [29].

The experts who participated in this study are:

- Expert 1: Systems Auditor and Consultant at a real estate investment company in Argentina. Professor of Information Systems Auditing and Design in undergraduate and graduate courses. 25 years of experience in the software industry.
- Expert 2: Systems Coordinator of the Digital Lab Head Office (Web and Mobile) of the Systems Management department for a private medical company in Argentina. Professor of Supervised Professional Practice in undergraduate courses. 17 years of experience in the software industry.

Five sessions of approximately one hour and a half were held with Expert 1 and four sessions were held with Expert 2, three of which lasted approximately one hour and a half, and the fourth one lasted two hours. The meetings with the Experts were held virtually through the Google Meet platform.

The main findings obtained from the interviews with each expert and the changes made in DepProMod are detailed below:

- Expert 1: Roles were reassigned. Based on the expert's feedback regarding the names of some of the inputs and outputs of Sub-Process INI - Initiation, some of them were modified, one activity was eliminated, and a change was also introduced in a template of the same Sub-Process.
  In relation to Subprocess PLA – Planning, two tasks were merged, an action was added to a task and the deployment cost estimation was removed within the scope of the model.
  In relation to the Sub-Process EXE – Execution, templates were unified and renamed, a template related to the training of end users was incorporated and the scope of some tasks related to tests was also reconsidered.
  Regarding the Sub-process MC – Monitoring and control, templates related to deployment risks and monitoring were unified and simplified.
  Finally, a task was modified in Subprocess CL – Closing.
- Expert 2: One element of the model, originally named "infrastructure", was renamed as "resources". A modification was introduced regarding the scope of a training-related task in the Subprocess EXE – Execution.
  In addition, a change was made to two templates related to training.

6. Conclusions and Future Work

This paper presents the results of two empirical studies carried out to obtain feedback about the model and thus refine it and complete it. These empirical studies have been satisfactory.

Specifically, a case study from an Argentinian level 1 Medium-size software development SME was

presented. We worked with the deployment documentation of the "company creation" module from a management system for advertising agencies for Latin America which allowed us to identify the information requirements needed to design the DepProMod templates.

The second study, the interviews held with two professionals from the software industry, provided their opinion on elements defined in the preview version of the DepProMod, which allowed us to redefine activities, tasks and roles.

Our future work will consist in conducting studies to validate DepProMod in the real context. Specifically, we plan to focus on evaluating users' perception of DepProMod in several SMEs in Argentina after using DepProMod in real projects. We will gather information on the users' perceptions by measuring variables such as "*Perceived Ease of Use*", "*Perceived Usefulness*" and "*Intention to Use*", taken from the *Technology Acceptance Model* (TAM) proposed by Davis [35].

Furthermore, there are plans to extend the evaluation of DepProMod to an international context, given that the characteristics of the model are applicable to any country considering the drawbacks that are evident in the implementation processes.

Authors' contribution

MP designed DepProMod and conducted the empirical studies to refine the preliminary version of the model. MP designed with MG the empirical studies. The results were analyzed by MP and reviewed by MG and RB. All authors read and approved the final manuscript.

Acknowledgements

The research work presented in this paper is framed within the following projects: ADAGIO (Consejería de Educación, Cultura y Deportes de la JCCM, SBLPY/21/180501/000061) and AETHER-UCLM (MICINN, Spain, PID2020-112540RB-C42).

References

[1] M. Felderer., R. Ramler, "Risk orientation in software testing processes of small and medium enterprises: an exploratory and comparative study," *Software Quality*, vol. 24, pp. 519–548, 2016.

[2] D. Mishra, A. Mishra, "Software Process Improvement in SMEs: A comparative view," *A Computer Science and Information Systems*, vol. 6, no. 1, pp. 111 – 140, 2009.

[3] N. Subramanian, "The software deployment process and automation," *CrossTalk*, vol. 30, no. 2, pp. 28-34, 2017.

[4] H. Abushama, "PAM-SMEs: process assessment method for small to medium enterprises," *Journal of Software: Evolution and Process*, vol. 28, pp. 689-711, 2016.

[5] D. Carrizo y L. Sanchez, "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 de Economic development, Innovation Management, and Global Growth,* Madrid, 2017.

[6] A. Fuggetta, E. Di Nitto, "Software Process," *in FOSE'14 Proceedings of the 36th International Conference on Software Engineering - Future of Software Engineering*, pp. 1-12, 2014.

[7] L. Bass, I. Weber, L. Zhu, *DevOps: A Software Architect's Perspective*." *S*EI series in software engineering, 2015.

[8] Scaled Agile. Available at: `https://www.scaledagileframework.com/continuous-deployment`. Accessed on 2020-03-03.

[9] J. Díaz, J. Pérez, A. Yague, A. Villegas García, A. de Antona, "DevOps in Practice - A preliminary Analysis of two Multinational Companies," *in PROFES '19 Proceedings of the 20th Product-Focused Software Process Improvement*, pp. 323-330, 2019.

[10] M. Panizzi, M. Genero, R. Bertone, "Refining a Software System Deployment Process Model: A Case Study," *in CACIC '21 Proceedings of the 27th Argentine Congress of Computer Science,* pp. 439-448, 2021.

[11] M. Panizzi, M. Genero, R. Bertone, "Software system deployment process: A systematic mapping study," *in CIbSE '20 Proceedings of the 23th Iberoamerican Conference on Software Engineering*, pp. 138-151, 2020.

[12] B. Kitchenham, D. Budgen, P. Brereton, *Evidence-Based Software Engineering and Systematic Reviews*, Chapman and Hall 1 st. Editon. Chapman and Hall/CRC. New York, 2015.

[13] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, "Systematic mapping studies in software engineering," *in EASE' 2008 Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, pp. 68–77, 2008.

[14] I. Jacobson, G. Booch, J. Rumbaugh, *The Unified Software Development Process*. 1st Edition. Addison Wesley, USA, 1999.

[15] IEEE ISO/IEC/IEEE 12207:2017(E). Systems and software engineering - Software life cycle processes, 2017.

[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). Available at: `https://www.agilebusiness.org/resources/dsdm-handbooks`. Accessed on 2019-05-29.

[18] R. Wieringa, N. Maiden, N. Mead., C. Rolland, "Requirements engineering paper classification and evaluation criteria: A proposal and a discussion," *Requirements Engineering*, vol. 11, pp. 102–107, 2005.

[19] I. Reascos, J. Carvalho, S. Bossano, "Implanting IT Applications in Government Institutions: A Process Model Emerging from a Case Study in a Medium-Sized Municipality," *in Proceedings of the 12th International Conference on Theory and Practice of Electronic Governance, Melbourne*, 2019.

[20] J. Molléri, K. Petersen, E. Mendes, "An empirically evaluated checklist for surveys in software engineering," *Information and Software Technology*. vol. 119, pp. 1-33, 2020.
[21] M. Panizzi, M. Genero, R. Bertone, "Encuesta para analizar las necesidades con respecto al proceso de despliegue de las PyMES en Argentina," *in CIbSE '21 Proceedings of the 24th Iberoamerican Conference on Software Engineering*, 2021.
[22] S. Ambler, The Agile Unified Process (AUP). Available at: http://www.ambysoft.com/unifiedprocess. Accessed on 2020-03-03.
[23] Portal de Administración Electrónica. Gobierno de España. "Métrica versión.3". Available: https://administracionelectronica.gob.es/pae_Home. Accessed on 2020-03-07.
[24] M. Panizzi, R. Bertone, A. Hossian, "Proposal for a Model of a Computer Systems Implantation Process (MoProIMP)," *Communications in Computer and Information Science*, vol. 995, pp.157-170, 2019.
[25] A Guide to the Project Management Body of Knowledge. (PMBOK® Guide) – Fifth Edition. Project Management Institute, 2013.
[26] CMMI® Institute. CMMI Development V1.3, 2010.
[27] Competisoft. Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica". Available at: https://alarcos.esi.uclm.es/competisoft/web/completo/index.htm. Accessed on 2021-04-03.
[28] P. Runeson, M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empir Softw Eng* vol. 14, pp. 31–164, 2009.
[29] P. Runeson, M. Höst, A. Rainer, B. Regnell, *Case study research in software engineering: guidelines and examples*. Wiley Publishing, Hoboken, 2012.
[30] C. Robson, *Real world research*, 2nd edition. Blackwell, 2002.
[31] R. Yin. *Case study research: design and methods*. 5th Edition. Sage Publications, 2014.
[32] Ministerio de Desarrollo Productivo, Nuevas categorías para ser PyMES, 2018. Available at: www.argentina.gob.ar/noticias/nuevas-categorias-para-ser-pyme.
[33] T. Lethbridge, S. Sim, J. Singer, "Studying software engineers: data collection techniques for software field studies," *Empir Softw Eng* vol. 10, no. 3, pp. 311–341, 2005.
[34] M. Panizzi, M. Genero, R. Bertone, Appendix – "Refining a Software System Deployment Process Model: A Case Study." Available at: https://doi.org/10.6084/m9.figshare.15000642.v1.
[35] F. Davis, "Perceived usefulness, perceived ease of use and user acceptance of information technology," *MIS Quarterly,* vol. 13, pp. 319-340, 1989.

Citation: M. Panizzi, M. Genero and R. Bertone. *Refining a Software System Deployment Process Model through Empirical Studies.* Journal of Computer Science & Technology, vol. 23, no. 1, pp. 71-84, 2023.
DOI: 10.24215/16666038.23.e06
Received: April 16, 2022 **Accepted:** October 24, 2022.