Refining a Software System Deployment Process Model: A Case Study

Marisa Panizzi[1,2,3 *], Marcela Genero[4], 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 Engineering Information Systems, Universidad Tecnológica Nacional – Facultad Regional Buenos Aires, Argentina.
[3] Institute of Technology and Engineering, Universidad Nacional de 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 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. This paper presents the results of a case study we carried out in order to examine the feasibility of the implementation of the initial version of DepProMod in a real environment with the purpose of refining it (if necessary) and completing it. We worked with the deployment process documentation of the "Company creation" module of a management system of advertising agencies for Latin America, in a software development SME in Argentina, to analyze the information requirements of the deployment procesgs and thus move towards the design of templates. In addition, a set of good practice recommendations was designed, not only for the deployment process but also for the rest of the company's software processes.

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

1 Introduction

Small an Medium Enterprises (SMEs) need efficient and effective software engineering solutions. But the proper implementation of software engineering techniques is 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 because they account for the highest percentage of software development companies all over the world [2].

* Corresponding author: Marisa Panizzi

Deployment is a crucial process of the software development life cycle because its result will determine whether the client successfully accepts, or not, the software system that has been delivered. There are automation solutions to improve the last stages of the life cycle [3], among which we can mention new techniques/practices such as DevOps [4] and Continuous Deployment [5] 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 [6]. These emerging solutions are not viable for a large number of SMEs due to the lack of human resources and infrastructure that would allow them to adopt such solutions.

Before starting the design of the model, a systematic mapping study of the literature (SMS) was perfomed in order to review the state of the art and identify models, methodologies or methods which might serve as a guide for SMEs when deploying software systems [7]. As a result of the SMS, two process models were identified which could guide SMEs during the deployment process. Such models have the limitation that they delegate the responsibility of making decisions on a series of deployment-related aspects to the organizations that apply them. These aspects include tasks, artifacts, techniques, methods, tools and role definitions. This delegated responsibility potentially hinders the application of these models in SMEs since this type of organizations require more detailed and descriptive processes and, therefore, more easily appliable. In order to supplement the SMS, with the purpose of gathering evidence on the current state of the software system deloployment process practice in SMEs in Argentina, a survey-based exploratory study was conducted [8]. The results of the survey confirmed the need for a software system deployment process model which helps SMEs to conduct deployments in a systematized manner by means of: a) the execution of well-defined activities and tasks, b) the use of guiding templates, c) the assignment of specific roles which possess the necessary competences to execute the deployment, and d) the use of tools to automate some of the process activities in order speed up the process.

All of the above considerations led us to define the objective of our long-term research, which is to propose a holistic software system deployment process model to help SMEs execute the deploymemt process of software systems in a systematized and controlled manner. Our preliminary version of the model was called Model of a Computer Systems Implantation Process (MoProIMP) [9], 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.

The preliminary version of DepProMod has subprocesses, activities and tasks. Our model differs from the existing proposals in its way of application; it allows SMEs to execute it in a step-by-step manner because its architecture is based on the capability levels of the CMMI-DEV standard [10]. The advantage of this application modality is the increased quality of the deployment process as well as the growth and improvement of the knowledge of the human resources of the SMEs. Since the DepProMod structure includes subprocesses, activities and management tasks for the

deployment process based on [11], it provides SMEs with more stabilized work methods. Another advantage is that it can be coupled with the development methodology used by the SME.

This paper presents the results of a case study we carried out in order to examine the feasibility of the implementation of the preliminary version of DepProMod in a real environment with the purpose of refining it (if necessary) and completing it. We worked with the deployment process documentation of the "Company creation" module of a management system of advertising agencies for Latin America, in a software development SME in Argentina to analyze the information requirements of the deployment process and thus move towards the design of templates.

The paper is organized as follows: Section 2 presents an overview of the DepProMod. The case study design and results are presented in Section 3 and, finally, our conclusions and proposals for future work are set out in Section 4.

2 Overview of DepProMod

The preliminary version of DepProMod has a life cycle model that adopts the 5 PMIBOK process groups [11]. These groups are: Initiating, Planning, Executing, Monitoring and Controlling and Closing. The reason for this choice is that PMIBOK 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 of the ISO / IEC / IEEE 12207 standard [12] 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 [13] 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 [12] 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 [10]. 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 [14], since it is a process improvement model for Ibero American software industry SMEs with some adaptations to the needs of the DepProMod definition.

3 Case Study Description

In this section, we present the detailed description of the case study, following the guidelines proposed in [15, 16].

3.1 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 [16] 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 for the software system deployment process and thus move forward towards the design of the templates necessary for our model. We believe the case study is suitable because we wish 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.

This a single embedded case study [16] according the classification of Yin (2002) with the following characteristics:

- Context: although our model arose in response to the need for SMEs to improve and stabilize their software system deployment process, the case study that we had available involved an SME (55 employees), 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 features in detail 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, set up holidays, set up working days, create departments, create hierarchies, create work groups, assign modules to groups, create and assign administrative functions to people, assign people to work groups, assign menu options to the employees, assign clients to work groups, fill the parameters module with information, enter the accounts plan with its respective additional information.

- Unit of analysis: deployment documentation of the "Company creation" module of the advertising agency management system called “T&C”.

3.2 Preparation for data collection

The third grade collection technique was used according to the classification proposed in [16]. 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 [16]. 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 Table 1, an extract of the coding scheme is presented. The rest of the coding scheme used is presented in the Appendix [17].

Table 1. Extract of the coding scheme for data collection.

Group Code	Category Name	Description
S1 – Initiating subprocess	PRO: Project	The project plan, software requirements and software architecture are explored.
	ORG: Organization	The organization’s communication aspects, documentation protocols and configuration management handling are explored.
	RES: Resources	The organization’s human resources, the users and technological resources are identified.

3.3 Analysis and interpretation of results

Since this is an exploratory study, the "Hypothesis Generation" technique was used to analyze the data [16]. In this case study, we consider that the research could be verifying 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,
- It is possible that from the analyzed documentation, the company is provided with a set of recommendations of good practices to improve its deployment process.

In a first instance, the drawing of conclusions from the collected data was carried out by the first author as part of the research process of the doctoral thesis and then agreed with the other authors, the thesis supervisors.

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 was called "recommendations" and was used to record recommendations for the deployment process analyzed (of the case). The information collected and analyzed is presented in the Appendix [17].

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 developed 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 suprocess defined in our model and each category corresponded to an aspect to consider 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 reflect on good practices to recommend to the consulting company.

3. 4 Results

Table 2 shows the traceability of the documents reviewed for each DepProMod subprocess.

Table 2. Traceability of the documents reviewed for each DepProMod subprocess.

Documents/ Subprocesses	Subprocess 1: Inttiating	Subprocess 2: Planning	Subprocess 3: Executing	Subprocess 4: Monitoring and Controlling	Subprocess 5: Closing
T&C Project	x				
General documentation	x	x			
User requirements	x				

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		
Acceptance test instructions			x		
New company application form			x		
Installation script			x		
Progress report			x		
Meeting memo				x	
Smoke test results			x		
Acceptance test results			x		
Installation completion report					x

The results related to the research questions formulates 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 the information to be documented to be unified, 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 will 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" is updated.

Subprocess 5: Closing. The two reviewed documents containing information related to this subprocess, one of them is the updated deployment plan and the other contains information of 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 a knowledge base.

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.

3.5 Threats to validity

To analyze the validity of the study, the factors proposed in [16] 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 group 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.

4 Conclusions and Future work

This paper presented the results of a case study we carried out in order to examine the feasibility of the application of the initial version of DepProMod in a real environment with the purpose of refining it (if necessary) and completing it. After carrying out the case study, we can conclude that:

- RQ1 allowed us to identify the information (stated in section 3.6.) required in the deployment process in a real context and, given the diversity of the documentation structure, we consider it is necessary to create templates to complete our model. This will allow the information to be documented to be unified, with a clear objective of use, distribution and the definition of a person responsible for its creation, modification and approval.
- RQ2 allowed us to create a set of recommendations (presented in section 3.6.) for the company to improve its deployment process as well as to introduce good practices for the rest of the software project processes.

Our future work will consist of refining and completing the DepProMod in order to allow SMEs to systematize the deployment process of their software systems and provide detailed guidance on the subprocesses, activities, tasks, templates, roles, techniques/practices and tools and a definition of levels so that it can be implemented in stages. Our model can be coupled to the software development methodologies used by these SMEs. Next, we plan to carry out case studies in Argentine software development companies to test the usefulness of DepProMod, especially in SMEs.

Acknowledgements. The research work presented in this paper is framed within the following projects: GEMA (JCCM, SBPLY/17/180501/000293) and AETHER-UCLM (MICINN, PID2020-112540RB-C42).

References

1. Felderer M., Ramler R. Risk orientation in software testing processes of small and medium enterprises: an exploratory and comparative study. Software Quality, 24, pp. 519–548 (2016).
2. Mishra D., Mishra A. Software Process Improvement in SMEs: A Computer Science and Information Systems, 6, pp. 111 – 140 (2009).
3. 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).
4. Bass L., Weber I., Zhu L. DevOps: A Software Architect's Perspective (2015).
5. Scaled Agile. https://www.scaledagileframework.com/continuous-deployment. Last accessed 2020/03/03.
6. Díaz J., Pérez J., Yague A., Villegas García A., de Antona A. DevOps in Practice - A preliminary Analysis of two Multinational Companies. In: Proceeding of the 20 th Product-Focused Software Process Improvement. (PROFES 2019), pp. 323-330 (2019).
7. Panizzi M., Genero M., Bertone R. Software system deployment process: A systematic mapping study In: Proceedings of the XXIII Iberoamerican Conference on Software Engineering, CIbSE 2020, Curitiba, Paraná, Brazil, November 9-13, pp. 138-151 (2020).
8. Panizzi M., Genero M., Bertone R. Encuesta para analizar las necesidades con respecto al proceso de despliegue de las PyMES en Argentina. In: Proceedings of the XXIII Iberoamerican Conference on Software Engineering, CIbSE 2021, San José, Costa Rica, Agosto 30 – Setiembre 3. (2021).
9. Panizzi M., Bertone R., Hossian A. (2019) Proposal for a Model of a Computer Systems Implantation Process (MoProIMP). In: Pesado P., Aciti C. (eds) Computer Science – CACIC 2018. CACIC 2018. Communications in Computer and Information Science, vol 995. Pages 157-170. Springer, Cham. https://doi.org/10.1007/978-3-030-20787-8-11.
10. CMMI® Institute. CMMI Development V1.3 (2010).
11. A Guide to the Project Management Body of Knowledge. (PMIBOK® Guide) – Fifth Edition. Project Management Institute (2013).
12. IEEE ISO/IEC/IEEE 12207:2017(E). Systems and software engineering - Software life cycle processes (2017).
13. Portal de Administración Electrónica. Gobierno de España. Métrica versión.3. https://administracionelectronica.gob.es/pae_Home. Last accessed 2020/03/07 (2001).
14. Competisoft. Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica. Last accessed 2021/04/03. https://alarcos.esi.uclm.es/competisoft/web/completo/index.htm
15. Runeson P., Höst M. Guidelines for conducting and reporting case study research in software engineering. Empir Softw Eng 14:131–164 (2009).
16. Runeson P., Höst M., Rainer A., Regnell B. Case study research in software engineering: guidelines and examples. Wiley Publishing, Hoboken (2012).
17. Panizzi M., Genero M., Bertone. Appendix - Refining a Software System Deployment Process Model: A Case Study. https://doi.org/10.6084/m9.figshare.15000642.v1