A Case Study to Validate Feasibility of Risk Proposal in the Deployment Process of Software Systems

Chapter · May 2021

DOI: 10.1007/978-3-030-75836-3_9

3 authors:

Felipe Ortiz

2 PUBLICATIONS **2** CITATIONS

SEE PROFILE

Marisa Panizzi

Universidad de Morón

10 PUBLICATIONS **4** CITATIONS

SEE PROFILE

Rodolfo Bertone

Universidad Nacional de La Plata

32 PUBLICATIONS **87** CITATIONS

SEE PROFILE

A case study to validate feasibility of risk proposal in the deployment process of software systems.

Felipe Ortiz [1, * [0000-0003-2291-1924]], Marisa Panizzi[1,2[0000-0002-9699-1544]] and Rodolfo Bertone[2[0000-0003-0609-0310]]

[1] Master's Program in Information Systems Engineering. Technological National University at Buenos Aires, Argentina.
[2] Department of Information Systems Engineering. Technological National University at Buenos Aires, Argentina-
[3] School of Information Systems – Computer Science Research Institute LIDI (III-LIDI). National University of La Plata, La Plata, Argentina.
ortizfd@gmail.com; marisapanizzi@outlook.com; pbertone@lidi.unlp.edu.ar

Abstract. Deployment is the process by which a software system is transferred to a business client. A risk is defined as the likelihood for a loss to occur. In a software project, a risk might imply decreased quality of the software product, increased costs, a delay in project completion or a flaw, among others. A case study is developed with the aim to refine the set of risks. Furthermore, procedures are proposed for their prevention, mitigation and/or transfer for the software system deployment process. This article presents the results of a case study which analyzed the documentation related to deployment of functionalities in a bank's Human Resources Portal conducted by an Argentina based software Small and Medium Enterprise (SME[1]).

Keywords: software system deployment, risk management, case study.

1 Introduction

There are various factors that can affect software projects, such as modifications in priorities and inadequate planning [1]. One of the most important factors might be unmanaged risks. A risk is the probability for a loss to occur. In a software project, such loss might take the form of decreased quality of the software product, increased development costs, a delay in project completion or a flaw [2]. A prevailing condition for the growth of the software industry is that companies offer higher quality products that satisfy customer demands and requirements, but above all that generates confidence at

[1] Presidencia de la Nación. (2018). https://www.argentina.gob.ar/noticias/nuevas-categorias-para-ser-pyme. Last updated on 09/05/2018.

* Corresponding author: Felipe Ortiz

the time of use [3]. This is achieved through the application of internationally recognized risk management models and methodologies. However, according to the 2019 annual report published by the Permanent Observatory of the Software and Information Services Industry (OPSSI) [4], in Argentina, the Software Industry is mainly made up of small and medium-sized companies (SMEs), which represent almost 80% of the sector (this constitutes them as a fundamental link in the country's economy), but in this kind of companies it's difficult to implement this type of models and methodologies because it involves a large investment in money, time and resources. At the international level, the same reality is reflected regarding SMEs make up a large portion of the software industry [5]. 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 [6].

A large number of projects lack formal approaches for risk management. The identification thereof usually depends, at an informal level, on the abilities and level of experience of software managers [7]. Although software risk management plays a key role in successful project management, it is usually not properly implemented in real world software projects, particularly in SMEs in Argentina [8].

Software system deployment is the phase of the development life cycle in which the software product is transferred to the client. The deployment process entails practices which tend to pose problems, such as the lack of components (generally external), incomplete downloads and faulty installations [9].

Software deployment is usually conducted in distributed and heterogeneous environments, which add complexity, thus causing time consumption and additional costs [10]. Deployment entails a series of changes at several levels: processes, working methods, technology and organizational structure [11].

According to Reascos Paredes et al. [12], the main causes of technological risks include heterogeneous and incompatible infrastructure, SMEs' poor technological capabilities and competences, the complexity of these systems, and bad data quality and safety. Forbes et al. [13] argue that the results of non-standardized and inadequate deployment practices are reflected in the information systems, which are difficult to maintain and operate.

This work presents the results of a case study aimed at refining (if necessary) the set of risks, as well as the procedures for their prevention, mitigation and/or transfer defined for the deployment process of software systems SMEs' in Argentina. Through this case study, the set of risks proposed for each of the tasks for the deployment process of software systems presented in CACIC 2020 [14] has been validated.

This article is organized as follows: related works are described in section 2; section 3 presents the set of risks for the deployment process; section 4 addresses the case study; and finally, section 5 presents the conclusions and future works.

2 Related works

A Systematic Mapping Study (SMS) was performed to build the state of the art on risk management for the deployment process of software systems [15]. After analyzing 100 primary studies, it was found that the most commonly used methodologies, methods and standards addressing risk management are CMMI [16], PMBOK [17] and SOFTWARE RISK EVALUATION [18].

To complement the SMS, a comparative analysis of the previously mentioned methodologies, methods and standards was conducted based on the DESMET method characteristics [19]. MAGERIT [20] was added to the comparison since it is one of the pioneering risk management methodologies.

The comparative analysis for the deployment addressed three dimensions: "Process", "Person" and "Product" [21]. After this comparative analysis, it was concluded that in the "Process" dimension all the methodologies, methods and standards analyzed address the risks for the deployment process. In the "Product" dimension, SOFTWARE RISK EVALUATION as well as PMBOK and MAGERIT include the risks of the deployment process while CMMI does not. Finally, in the "Person" dimension, none of the methodologies, methods or standards evaluated address the risks of the deployment process.

3 Risks of the deployment process

The activities and tasks considered for the definition of the risks of the deployment process are those stated in the technical process called "Transition" of the ISO/IEC/IEEE 12207:2017 standard [22]. This standard was chosen because it is internationally recognized.

For a better structuring of the solution, as well as so that its application in the industry can be carried out in a systematic way in different deployment projects, it is decided to define a coding for it. The proposal of Runeson et al. [23] that proposes guidelines for the design of a coding scheme for the analysis and interpretation of the data in the case studies. These guidelines are detailed below:

- Code as much as possible.
- Codes must be prioritized as follows:
  - High-level codes, based on research questions.
  - Mid-level codes, based on code groupings: code categories.
  - The low-level code is your interpretation of the text (in the Comments field).

The resulting coding for the activities and tasks of the deployment process are detailed in Table 1.

Table 1. Activities and tasks of the technical process "Transition" [22].

Activities	Tasks
A1 Preparation for deployment	T1 Identify technological restrictions.
	T2 Obtain access to the environments, systems or services enabled.
	T3 Analyze existing policies and standards.

	T4 Define unit testing policies.
	T5 Define deployment priorities in order to support the data and software migration and transition.
	T6 Identify restrictions in the deployment strategy.
A2 Deployment execution	T7 Develop or adapt software elements according to the deployment strategy.
	T8 Record requirement compliance evidence.
	T9 Adapt hardware elements and software services.
	T10 Staff training.
A3 Deployment results management	T11 Record the results and anomalies found.
	T12 Keep traceability.
	T13 Provide key artifacts.
	T14 Document fulfillment of expectations and capabilities.
	T15 Evaluate the need or opportunity for improvement.

The risk classification used is the one proposed in [7], with adjustments made considering the evolution of software engineering in the last few decades and the deployment process of software systems. For risk weighting, the proposal established in the ISO/IEC 31010:2009 standard [24] is adopted, since it is one of the main international references in terms of risk management for the software industry.

The definition of risks was established considering a three-dimensional approach, given by the "Process" dimension, the "Person" dimension and the "Product" dimension [21].

4 Description of the case study

This section presents a detailed account of the case study following the guidelines proposed in [23].

4.1 Case study design

The main objective is to examine the feasibility of the application of a set of risks, as well as the procedures for their prevention, mitigation and/or transfer in the deployment process of software systems in a real environment with the aim to refine them (if necessary). According to Robson's classification [25], case studies fall under the scope of exploratory studies. We worked with documentation related to the deployment of capability deliverables for a bank's Human Resources Portal performed by an Argentina-based software SME.

4.2 Research questions

In order to address the objective of this study, the following research questions (RQ) are posed:

RQ1: How were risks managed during the activities of the software system deployment process (identification, analysis and severity)?

This question is intended to provide information about the risks encountered during the execution of the deployment process and the treatment provided by the consulting company in order to compare them with the proposal made.

RQ2: How can the software system deployment process be strengthened in this company?

This question is intended to determine the way in which the consulting company can enhance its deployment process. For this purpose, the identification of a set of risks is proposed, along with the procedures for their prevention, mitigation and/or transfer.

4.3 Case and unit of analysis.

This section describes the context, the case and the unit of analysis of the case study. According to Yin's classification [26], it is a holistic single-case study.

Context: the case study was conducted in a software SME located in the Autonomous City of Buenos Aires, with a total of 430 employees. This company develops customized information systems for clients of different industry sectors, including finance, automotive, pharmaceutical and banking. Its software projects combine agile practices with iterative life cycle development methodologies. Access was granted to the documentation of the project subject to an agreement not to disclose the name of the company and a commitment to inform about any findings and recommendations to be considered for deployment process risk management.

Case: deployment of deliverables for a Human Resources Portal conducted at a bank based in Argentina. It consisted in adding new capabilities, using a modular strategy. These were: integration with a new data source, publication of Application Programming Interfaces (APIs), integration with a distance learning portal, modification of the final user interface, new employee management alerts and notifications, appearance modifications to the application organigram, and modification to approval flows.

Unit of analysis: documentation related to the deployment of deliverables for a Human Resources Portal.

4.4 Preparation for data collection

A third-degree technique was used combined with an independent method according to the classification proposed in [27]. A template with a coding scheme made up of 3 groups was used. Each group coincides with the 3 activities of the technical process called "Transition" of the ISO / IEC / IEEE 12207: 2017 Standard [22] (A1 Preparation for deployment, A2 Deployment Execution and A3 Deployment Results Management).

Table 2 shows the traceability of the documents analyzed and the risks associated with each of the dimensions.

Table 2. Traceability of the documents analyzed for the case study.

Documents/ Activities	A1	A2	A3
Risk monitoring spreadsheet	RProc6, RPers3 and RProd1	RProc10	RProd15

Progress Report	RPers4	RProc7 and RProd9	RPers13
Deliverable 1 - Closing report	RProd4	RProc8 and RPers9	RProc14, RPers15 and RProd13
Deliverable 1 - Deployment report		RProd8	RProc15 and RPers12
Deliverable 1 - Deployment Summary		RPers8	RProc11 and RProd12
Deliverable 1 - Deployment Tests Guide	RProc4, RPers2 and RProd5	RProd10	
Deliverable 1 - Deployment Test cases	RProc4, RPers2 and RProd3		
Deliverable 1 – installation scripts	RPers1 and RProd2		RProc12
Deliverable 1 – Work Plan	RProc5 and RPers5	RProd7 and RPers10	
Deliverable 1 – Installation Requirements	RProc1 and RProd6	RProc9 and RPers7	
Deliverable 1 - Deployment Completion report	RProc2 and RPers6	RProd9	RProc13, RPers14 and RProd14
Deliverable 2 - Closing Report	RProd4	RProc8 and RPers9	RProc14, RPers15 and RProd13
Deliverable 2 - Deployment Report		RProd8	RPers12
Deliverable 2 – Deployment Summary		RPers8	RProc11, RProc15 and RProd12
Deliverable 2 - Deployment Tests Guide	RProc4, RPers2 and RProd5	RProd10	
Deliverable 2 - Deployment Test cases	RProc4, RPers2 and RProd3		
Deliverable 2 – installation scripts	RPers1 and RProd2		RProc12
Deliverable 2 – Work Plan	RProc5 and RPers5	RPers10 and RProd7	
Deliverable 2 – Installation Requirements	RProc1 and RProd6	RProc9 and RPers7	
Delivery 2 - Deployment Completion Report	RProc2 and RPers6	RProd9	RProc13, RPers14 and RProd14
General Documentation	RProc3		RPers11 and RProd11

Table 3 shows the resulting risk weighting for the “Process” dimension in the case study.

Table 3. Risk weighting of the "Process" dimension for the case study.

Activity	Risk	Weight	Result
A1	RProc1	[Probability (H) * Impact (H)] =	VH
	RProc2	[Probability (M) * Impact (H)] =	H
	RProc3	[Probability (L) * Impact (L)] =	L
	RProc4	[Probability (H) * Impact (VH)] =	VH
	RProc5	[Probability (VH) * Impact (VH)] =	VH
	RProc6	[Probability (L) * Impact (H)] =	H
A2	RProc7	[Probability (L) * Impact (VH)] =	VH
	RProc8	[Probability (L) * Impact (M)] =	M
	RProc9	[Probability (M) * Impact (H)] =	H
	RProc10	[Probability (L) * Impact (H)] =	H
A3	RProc11	[Probability (L) * Impact (M)] =	M
	RProc12	[Probability (H) * Impact (H)] =	VH
	RProc13	[Probability (H) * Impact (VH)] =	VH
	RProc14	[Probability (M) * Impact (H)] =	H
	RProc15	[Probability (L) * Impact (L)] =	L

The risk weighting of the "Person" dimension is presented in Table 4.

Table 4. Risk weighting of the "Person" dimension

Activity	Risk	Weight	Result
A1	RPers1	[Probability (H) * Impact (H] =	VH
	RPers2	[Probability (L) * Impact (H)] =	H
	RPers3	[Probability (M) * Impact (H)] =	H
	RPers4	[Probability (L) * Impact (H)] =	H
	RPers5	[Probability (H) * Impact (VH)] =	VH
	RPers6	[Probability (H) * Impact (H)] =	VH
A2	RPers7	[Probability (H) * Impact (H)] =	VH
	RPers8	[Probability (ML) * Impact (H)] =	M
	RPers9	[Probability (L) * Impact (H)] =	H
	RPers10	[Probability (M) * Impact (H)] =	H
A3	RPers11	[Probability (L) * Impact (M)] =	M
	RPers12	[Probability (H) * Impact (H)] =	VH
	RPers13	[Probability (L) * Impact (H)] =	H
	RPers14	[Probability (H) * Impact (H)] =	VH
	RPers15	[Probability (H) * Impact (VH)] =	VH

The risks weight of the "Product" dimension is presented in Table 5.

Table 5. Risk weighting of the "Product" dimension

Activity	Risk	Weight	Result
A1	RProd1	[Probability (L) * Impact (H)] =	H
	RProd2	[Probability (M) * Impact (H)] =	H
	RProd3	[Probability (M) * Impact (H)] =	H
	RProd4	[Probability (L) * Impact (VH)] =	H
	RProd5	[Probability(H) * Impact (H)] =	VH

	RProd6	[Probability (MH) * Impact (H)] =	VH
A2	RProd7	[Probability (M) * Impact (M)] =	M
	RProd8	[Probability (H) * Impact (M)] =	H
	RProd9	[Probability (M) * Impact (H)] =	H
	RProd10	[Probability (H) * Impact (H)] =	VH
A3	RProd11	[Probability(H) * Impact (M)] =	H
	RProd12	[Probability(H) * Impact (H)] =	VH
	RProd13	[Probability (L) * Impact (H)] =	H
	RProd14	[Probability (M) * Impact (H)] =	H
	RProd15	[Probability (H) * Impact (VH)] =	VH

4.5 Analysis and Interpretation of Results

The results of the research questions defined for the case study are presented below:
RQ1: How were risks managed during the activities of the software system deployment process (identification, analysis and severity)?

Based on the documentation analyzed, it was possible to find flaws in the risk management proposed for the activities of the deployment process:

- Activity 1 (A1) – Preparation for Deployment: The deployment progress reports showed that, due to the few investments in technology made in recent years, the resources (hardware and basic software) assigned to the production environment did not comply with the minimum requirements requested by the consulting company to carry out the deployment in accordance with the established work plan. According to the deployment reports analyzed, the technicians (bank employees) did not have the knowledge and skills necessary for the correct deployment of scripts and monitoring of the guides sent by the consulting company. This is because the technicians who participated in the original deployment left the organization and were replaced by personnel with little technical or functional experience. The general documentation of the project shows that the bank does not have an adequate personnel retention policy, which generates frequent rotation.
- Activity 2 (A2) – Deployment Execution: according to the progress reports of the deployment project, the technical flaws mentioned in the previous stage (separation of technical personnel with experience in the technologies involved and greater complexity of the product) generated friction between the consulting company and the managers of the bank. This was due to non-compliance with the deadlines established in the work plan, which ended up activating a penalty clause against the consulting company.

  During the documentary analysis, incomplete test plans and inadequate deployment metrics were found. According to the deployment completion reports, the consulting company had to face cost overruns for not having the document management procedures required by the bank in the contract and in corporate policy. In addition, it was necessary to add technical resources from the consulting company to address the lack of technical expertise of the bank's employees, who had to be trained to carry out future deployments.

These technical drawbacks, added to a very demanding work schedule for internal reasons and needs of the bank (shown in the closing reports), were some of the causes that produced very important delays and friction between different sectors of the organization that even considered the cancellation of the deployment project on several occasions.

- Activity 3 (A3) - Deployment Results Management: problems with the software repositories (lack of necessary permissions, previous versions, lack of components, etc.), in addition to the low commitment and inexperience of the bank's technicians, generated multiple drawbacks during the deployment. These technical drawbacks strongly impacted on the quality of the final product and the satisfaction of the users who saw their productivity affected due to failures in the application's capabilities once the deployment was complete.

  In the deployment completion reports, it was also evidenced that there was a wrong dimensioning of the deliverables and that the necessary security tests were not carried out. This gave end users access to sensitive human resource information.

RQ2: How can the software systems deployment process be strengthened in this company?

Proper risk management minimizes drawbacks in the deployment process. The following recommended procedures has been presented to the software consulting company in order to prevent, mitigate and / or transfer each of the risks associated with the "Process", "Person" and "Product" dimensions. Table 6 shows the procedures associated with the risks of the “Process” dimension.

Table 6. Procedures associated with the risks of the “Process” dimension.

Risks (RProc)	Procedures (PProc)
RProc1 Few investments in technology.	PProc1 Accurate software measurements are the best prevention method for this type of risk. The methodology is based on adequately managing costs, deadlines, and other quantitative and qualitative factors associated with deployment projects.
RProc2 Friction between the software management and senior executives.	PProc2 Once friction is generated between the top executives and the software management, it is not easy to continue the project properly. Some of the approaches to control introduce radical changes, such as outsourcing software management and reducing the size of deliverables during deployment.
RProc3 Void or nonexistent corporate regulations.	PProc3 Having an adequate corporate policy allows for clear and unambiguous objectives during the deployment project, forcing the use of rules and procedures.
RProc4 Poorly drawn test plans.	PProc4 One of the methods to prevent this type of risk is to prepare the deployment test plan during the analysis and design phase, thus anticipating the necessary requirements. The software testing methodology will depend on the one used for the project management.
RProc5 Reduced schedule or work plan.	PProc5 There are several estimation methods for deployment projects with the aim of mitigating these types of risks, such as expert opinion, the use of estimation models, the decomposition of the work plan and the comparison by analogy with other similar projects. among others.
RProc6 Cancellation of the deployment process.	PProc6 The most effective prevention method is planning and estimating the deployment project. That is, well-defined goals and appropriately assigned tasks. Fluid communication must also be maintained between all participants.

RProc7 Friction between the client and the software company.	PProc7 In order to minimize the likelihood of friction between clients and contractors and the consequences that this may bring to the deployment project, it is advisable to have legal personnel trained in the software domain, so that they can comply with the contractual terms if necessary.
RProc8 Inadequate metrics.	PProc8 Analogies with metrics use in other projects is one of the most effective methods of preventing inadequate metrics during deployment. The larger the number of analog projects (not less than 25), the more effective the result will be.
RProc9 Cost overruns.	PProc9 as the project progresses, it is more difficult to control the associated costs. Cost overruns can occur for various reasons. The best form of mitigation is detailed monitoring of the deployment project. Any excess of time or resources used can generate cost overruns. In particular, the use of overtime for staff may be a factor that triggers the risk.
RProc10 Inadequate training plans.	PProc10 Each and every one of the necessary aspects of education and training for all the members of the deployment project, including technicians and end users, must be covered sufficiently in advance. Each one of the trainings carried out must be registered and its level of compliance must be evaluated according to the needs of the project.
RProc11 Inadequate deployment management tools and methods.	PProc11 The most effective approach to preventing the use of inappropriate software engineering tools during the deployment project is to conduct surveys and generate metrics for the tools most used by the software industry.
RProc12 Inadequate repositories.	PProc12 One of the most effective preventive steps for unsuitable configuration control of the repositories to be used during the deployment of the software product is to carry out a complete analysis of all the types of components that were produced, how they are connected and how often they are updated.
RProc13 Inaccurate estimation of deliverables.	PProc13 The most effective prevention methodology for estimation errors is the accurate measurement of the sizes of all deliverables and the resources required to produce them. The use of metrics during their deployment is also recommended
RProc14 Low user satisfaction.	PProc14 User satisfaction is a complex and multifaceted issue. Some of the seemingly effective preventive steps include user experience specialists. In addition, user satisfaction surveys are the basic monitoring mechanism to guarantee it during deployment.
RProc15 Ambiguous improvement goals.	PProc15 Establishing a formal software measurement program and adopting functional metrics are effective preventative measures to eliminate ambiguous goals during software deployment.

Table 7 shows the procedures associated with the risks of the “Person” dimension.

Table 7. Procedures associated with the risks of the “Person” dimension.

Risks (RPers)	Procedures (PPers)
RPers1 Lack of specialization in the technologies and processes involved.	PPers1 A method of prevention and / or mitigation of this type of risk is to create an inventory of the skills of employees within the company and to establish specialization criteria and training study plans according to the deployment project.
RPers2 Users without adequate access permissions.	PPers2 In order to avoid delays and drawbacks during the deployment, it is recommended that all necessary access permissions for all the members of the deployment project, including technicians and end users, be analyzed and requested in advance according to the methodologies established by the Organization.
RPers3 Inadequate staff retention policy.	PPers3 The most effective approach to prevent the loss of resources during the deployment project is for the organization to have an adequate human resources policy that includes extra incentives for meeting project milestones not only economic but also based on other professional aspects.
RPers4 Functional or business inexperience on the part	PPers4 In order to minimize this risk, the members of the deployment project should be carefully selected according to their role within the organi-

of the users in charge of the tests.	zation, their commitment and functional knowledge of the business. Another important aspect is to achieve proper communication between the members of the project.
RPers5 Constant changes in priorities.	PPers5 Having adequate management and monitoring of the deployment project is the best way to prevent this risk. Frequent follow-up meetings must be held in which the feasibility of the proposed changes and the impact they have on the project times are analyzed.
RPers6 Additional efforts and / or resources.	PPers6 Having an agile and structured induction plan during deployment reduces the drawbacks associated with this risk. New staff must be able to join in and assume their responsibilities transparently.
RPers7 Little experience in present systems.	PPers7 All platforms and interfaces that will be part of the deployment project must be analyzed and documented, and expert technicians must be selected in order to mitigate this risk. If training on any of them is necessary, it must be carried out in advance.
RPers8 Lack of expertise.	PPers8 It is recommended that the human resources assigned to the deployment project have technical and business expertise in order to ensure the correct operation of each of the deliverables. Each of the tasks must be properly documented.
RPers9 Bad professional practice.	PPers9 One of the most effective preventive steps to mitigate this risk is to establish a deployment plan based on expert opinion in each of the components involved (hardware and software services) in order to adapt them for proper deployment.
RPers10 Significant drop in resources assigned to the project.	PPers10 Clear policies must be established within the Organization so that the resources assigned to the deployment project are not reallocated to other tasks. Similarly, at a legal level, it is recommended that the need to maintain the quantity and technical level of the assigned resources be established with the contractors.
RPers11 Document management inexperience.	PPers11 One of the best practices to reduce or mitigate this risk is to train the resources assigned to the deployment project on the best practices of the document management methodology selected for the deployment project. Sometimes it is advisable to outsource this task to specialized personnel.
RPers12 Various criteria or interpretations.	PPers12 It is recommended that a standard traceability model be defined for the entire deployment process that includes the project participants, the sources (documents and models) and the objects or artifacts to be traced. These elements and their evolution must be explicitly identified in each flow of the deployment project.
RPers13 Low productivity.	PPers13 Proper monitoring of the tasks assigned to each of the members of the deployment project is the best way to minimize low productivity. To carry out adequate documentation and periodic follow-up meetings is recommended in order to resolve deviations or delays.
RPers14 Lack of collaboration from end users.	PPers14 To mitigate this risk, it is important that the top management of the organization embrace and disseminate the importance of carrying out the functional tests of the software product to the personnel affected by the deployment project so as to avoid operational drawbacks.
RPers15 Low commitment.	PPerso15 The most effective methodology is to work from different perspectives (technical, human, etc.) so that all the members of the deployment project (client and contractors) feel the project as their own and challenging.

Table 8 shows the procedures associated with the risks of the “Product” dimension.

Table 8. Procedures associated with the risks of the “Product” dimension.

Risks (RProd)	Procedures (PProd)

RProd1 Novel technology or with little use.	PProd1 It is recommended that new technology be used, but with enough maturity and local support to avoid problems during the deployment project. If possible, an analysis of similar projects should be carried out to verify its adaptability to the necessary capabilities.
RProd2 Incompatibility with existing infrastructure	PProd2 to mitigate this risk, it is essential that a thorough analysis of compliance with the basic hardware or software requirements needed for the deployment be carried out in advance. All tasks must be properly documented and validated.
RProd3 Lack of adaptation to new technologies.	PProd3 Given the emergence of new technologies, such as DevOps and / or continuous deployment, it is necessary to adapt the organization's policies and / or procedures to the one selected for the deployment project. If necessary, it is recommended that external consulting be added to carry out this task adequately.
RProd4 Lack of components.	PProd4 It must be ensured that all components linked to the deliverables are available at the time of deployment. The best way to do this is through the proper traceability thereof.
RProd5 Incompatible data format.	PProd5 in order to prevent this risk, data sets must be selected for each of the technologies affected by the deployment process in order to validate their compatibility during their import into the new technology. These tests must be documented and supervised.
RProd6 Little flexibility.	PProd6 Defining the deployment strategy clearly and concretely will make it possible to choose the best technology for the software project so that it has the capacity to adapt to the changes that may arise during the deployment.
RProd7 Greater complexity.	PProd7 It is recommended that the capabilities and scope to be fulfilled by the software product be defined and properly documented in order to avoid increasing costs and time during the deployment project.
RProd8 Flaws or Errors in operation.	PProd8 A methodology should be used to record the fulfillment of all the capabilities of the product during the deployment, establishing reviews with the objective of guaranteeing that all technical and functional aspects were covered.
RProd9 Loss of characteristics and / or functions.	PProd9 to comply with all the necessary hardware requirements to avoid adapting the product due to technical incompatibility during deployment and to carry out a check-up in advance together with specialists is recommended.
RProd10 Lack of knowledge of the capabilities of the product.	PProd10 Fully documenting all the capabilities of the software product in an end user manual makes it possible to take full advantage of its features and ensure its correct deployment.
RProd11 Scarce documentation	PProd11 Having a knowledge base allows recording of the results and anomalies found during deployment. They serve to detect recurring problems and improve the process continuously.
RProd12 Inconsistencies in product versions.	PProd12 Controlling the different versions of all analysis and design documentation, disseminating the latest versions as soon as possible and alerting the entire team is one of the best ways to prevent and / or mitigate this risk during deployment.
RProd13 Incomplete capabilities.	PProd13 the configuration parameters of key artifact components should be determined in advance (for example: Libraries, Shell Scripts parameters, among others). Completeness of all components within software re-

	positories must be guaranteed during deployment. Additionally, the requirements on the workstations (plugins, active X components, etc.) must be identified.
RProd14 Low quality	PProd14 Compliance with all capabilities of the software product should be thoroughly reviewed and recorded to ensure quality during deployment. It is recommended that a list previously defined in collaboration with the key users of the project be used.
RProd15 Security tests not performed.	Prod15 It is recommended that cybersecurity methodologies be applied during the deployment project in accordance with the best market practices and procedures defined by the Organization.

4.6 Threats to validity

To analyze the validity of the study, the factors proposed in [27] were considered:

Construct validity. The results were obtained based on the documentary analysis of a set of risks for the process of deployment of software systems in a real context. This allowed us to answer the defined research questions, determining their relevance and suitability for the case.

Internal validity. The documentation used refers to a real case, a deployment of new deliverables for a Human Resources Portal performed in a bank in Argentina. In order to achieve greater precision and validity of the studied process, the need to combine the data source (project documentation) with other types of sources, such as interviews and / or focus groups to guarantee "data triangulation (source)", is recognized. Furthermore, the qualitative data collected and analyzed could be combined with quantitative data resulting from the project, thus ensuring a "Methodological Triangulation".

External validity. Carrying out a single case study may limit the generalizability of the results. However, a preliminary case study was conducted in [21]. These two experiences allow us to present results, which can be used by other researchers to carry out more studies with the same principles.

Reliability. The study data was collected and analyzed by the research group.

4.7 Lessons learned

- Method selection: a validation of a set of risks, as well as the procedures for their prevention, mitigation and / or transfer, for the process of deployment of software systems, was needed in a real environment, in order to refine them (if required). The results obtained allowed us to analyze the application of the set of risks defined in a real environment. Therefore, the method used is considered to have yielded the expected results.
- Data collection: although the documentation of the software system deployment process has been reviewed in order to analyze how the risks were managed, it is considered that the case could be strengthened if the data collected were complemented by another source or by quantitative data.
- Selected coding. The coding scheme selected for the design of the data collection and analysis template was adequate and allowed the systematic recording of risk information.

- Results report: Although the case is made up of two research questions, it is considered that the work carried out took into account an adequate level of detail for understanding the phenomenon under study.

5 Conclusions and future work

The results of a case study were presented to determine the feasibility of applying a set of risks, as well as the procedures for their prevention, mitigation and / or transfer for the process of deploying software systems in a real environment. It consisted of the risk analysis of the deployment of new deliverables for a Human Resources Portal carried out by a software SME in a bank in Argentina. After conducting the case study, it is concluded that:

• The first question allowed us to identify shortcomings in risk management through documentary analysis. These shortcomings include the lack of specialization of project personnel, mixed interests between the intervening areas and non-compliance with requirements of the installation environment.

• The second question allowed us to design a set of recommended procedures (presented in section 4.5) for the company to improve its deployment process and to introduce good risk management practices for future software system deployments.

The lessons learned from the case showed that the research method was adequate to validate the proposal.

The following are identified as future works: (a) to validate the risk proposal for the software deployment process in different case studies in order to refine it. (b) To propose the use of the risks defined for the deployment of software systems, as well as the procedures for the prevention, mitigation and / or transfer thereof, by other professionals in the industry.

References

1. Charette R. Why software fails [software failure]. IEEE spectrum, 42(9), 42-49. (2005).
2. Dhlamini, J. & Nhamu, I. y Kaihepa, A. Intelligent risk management tools for software development. 33-40 (2009)
3. Cámara de Software y Servicios Informáticos - CESSI. Anuario de la Industria Argentina de TI 2007/2008. Last updated on 02/12/2008.
4. Reporte anual 2018 sobre el Sector de Software y Servicios Informáticos de la República Argentina. OPSSI, 2018. Disponible en https://www.cessi.org.ar/opssi. Last updated on 04/2019
5. Hisham M. Abushama. PAM-SMEs: process assessment method for small to mediumenterprises. Software: Evolution and Process, 28, pp. 689 –711 (2016).
6. 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).
7. Jones C., Assessment and control of software risk. Yourdon Press (1994).
8. Liu D., Wang Q., Xiao J. The role of software process simulation modeling in software risk management: A systematic review. In Proceedings of the 3rd International Symposium on

Empirical Software Engineering and Measurement. Empirical Software Engineering and Measurement, pp. 302-311 (2009).
9. Jansen S., Brinkkemper S. Definition and validation of the key process of release, delivery and deployment for product software vendors: Turning the ugly duckling into a swan IEEE International Conference on Software Maintenance, ICSM, art. no. 4021334, pp. 166-175. (2006).
10. Subramanian, N. The software deployment process and automation. CrossTalk, 30 (2), pp. 28-34 (2017).
11. Tyndall J. Building an effective software deployment process. In Proceedings of the 40th annual ACM SIGUCCS conference on User services, pp. 109-114 (2012).
12. Reascos I., Carvalho J., Bossano S. 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, pp. 80-85 (2019).
13. Forbes J., Baker E. Improving Hardware, Software, and Training Deployment Processes. In: Proceedings of 19th International Conference on Software Maintenance, pp. 377-380. IEEE, The Netherlands. (2003).
14. Ortiz F., Panizzi M., y Bertone R. Risk refinement in the deployment process of software systems: a case study. En las Actas del XXVI Congreso Argentino de Ciencias de la Computación - CACIC 2020. Universidad Nacional de La Matanza, 5 al 9 de octubre. ISBN 978-987-4417-90-9.
15. Ortiz F, Davila M., Panizzi M. y Bertone R. State of the art determination of risk management in the implantation process of computing systems. En las Actas del I Congreso Internacional sobre Avances en Nuevas Tendencias y Tecnologías (ICAETT 2019). Ecuador, Guayaquil Ecuador, 29 al 31 de mayo, pp- 23-32 (2019). ISNB 978-3-030-32022-5.
16. CMMI Institute, «Capability Maturity Model Integration». https://cmmiinstitute.com. Last updated on 24/06/2020.
17. Project Management Institute. https://www.pmi.org/pmbok-guide-standards. Last updated on 24/06/2020.
18. Software Engineering Institute, «Software Risk Evaluation Method» (1999). https://resources.sei.cmu.edu/asset_files/TechnicalReport/1999_005_001_16799.pdf
19. Kitchenham B., Linkman S., Law D.T. DESMET: A method for evaluating software engineering methods and tools. Keele University (1996).
20. Portal de administración electrónica, «MAGERIT v.3: Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información» 2012. Last updated on 24/06/2020.
21. Ortiz F., Panizzi M., y Bertone R., Risk determination for the implantation process of software systems. En las Actas del XXV Congreso Argentino de Ciencias de la Computación - CACIC 2019. Universidad Nacional de Río Cuarto, 14 al 18 de octubre, pp- 817- 825 (2019). ISBN 978-987-688-377-1
22. ISO/IEC/IEEE 12207:2017. Systems and software engineering — Software life cycle processes (2017).
23. Runeson P, Höst M, Rainer A, Regnell B. Case study research in software engineering: guidelines and examples. Wiley Publishing, Hoboken (2012).
24. International Organization for Standardization, «ISO/IEC 31010:2009». https://www.iso.org/standard/51073.html. Last updated on 24/06/2020.
25. C. Robson. Real world research 2nd edition. Blackwell (2002)
26. Yin, R., Case study research: design and methods. 5th Edition. Sage Publications. (2014).
27. Lethbridge T., Sim S., Singer J., Studying software engineers: data collection techniques for software field studies. Empir Softw Eng 10(3):311–341 (2005).