Journal of Information Systems and Technology Management – Jistem USP
Vol. 19, 2022, e202219015
ISSN online: 1807-1775
DOI: 10.4301/S1807-1775202219015 Original Article

RISK FACTORS IN THE SOFTWARE DEPLOYMENT PHASE: A CASE STUDY APPLIED IN TWO BRAZILIAN GOVERNMENT COMPANIES

Lidvaldo José dos Santos[1] https://orcid.org/0000-0002-6137-3359
Sildenir Alves Ribeiro[1,2] https://orcid.org/0000-0003-4808-1009
Eber Assis Schimitz[1] https://orcid.org/0000-0002-4839-4606
Mônica Ferreira da Silva[2] https://orcid.org/0000-0003-0951-6612
Antonio Juarez Sylvio Menezes de Alencar[1] https://orcid.org/0000-0003-4791-0581

[1]Universidade Federal do Rio de Janeiro. Rio de Janeiro, RJ, Brasil.

[2]Centro Federal de Educação Tecnológica do Rio de Janeiro. Rio de Janeiro, RJ, Brasil.

ABSTRACT

This work presents the results of a case study to identify the main risk factors in the software deployment phase involving two government Brazilian companies. The case study was developed through several on-site visits to monitor the deployment of the system adopted by companies and conduct interviews with team managers. The data were acquired mainly through questionnaires applied to the technical team (analysts and developers) involved with the software implementation. After acquiring the data, an empirical analysis was carried out, where the Risk Factors (RF) and the Containment Strategies (CS) identified in the literature were compared with the RF and CS found in the software deployment phase of the two companies. As a result, this work presents 11 risk factors and 14 Containment strategies found in the literature, in addition to a total of 9 RF and 9 CS recorded in the implementation of the software in companies A and B, which had not yet been cataloged in the literature.

Keywords. *Case Study, Containment Strategies, Deployment Phase, Risk Factors*

Manuscript first received: 2021-09-17. Manuscript accepted: 2022-07-01

Address for correspondence:
Lidvaldo José dos Santos, Centro Federal de Educação Tecnológica, CEFET/RJ. Rio de Janeiro, RJ, Brasil.
E-mail: lidiosantos@tic.ufrj.br
Sildenir Alves Ribeiro, Universidade Federal do Rio de Janeiro, PPGI/UFRJ, Rio de Janeiro, RJ, Brasil.
E-mail: sildenir.ribeiro@gmail.com
Eber Assis Schimitz, Universidade Federal do Rio de Janeiro, PPGI/UFRJ, Rio de Janeiro, RJ, Brasil.
E-mail: eber@nce.ufrj.br
Mônica Ferreira da Silva, Centro Federal de Educação Tecnológica, CEFET/RJ, Rio de Janeiro, RJ, Brasil.
E-mail: monica@nce.ufrj.br
Antonio Juarez Sylvio Menezes de Alencar, Universidade Federal do Rio de Janeiro, PPGI/UFRJ, Rio de Janeiro, RJ, Brasil.
E-mail: juarezalencar@nce.ufrj.br

INTRODUCTION

Deployment is the final phase of the software development process. In this phase, the transition is made, according to the structure of the Unified Process (UP), i.e., the delivery of the software product, implanted, tested and approved in production. This phase may involve even more configurations, to reflect the environment in which the software will be used, the transfer of data from existing software, the preparation of the final documentation and the training of users. The deployment phase can still present numerous obstacles during its execution. The user environment can be different from that predicted by the system developers and adapting the system to handle different user environments can be complex and expensive Sommerville (2016). The participation of end users has a great impact on the implementation of the software, which can lead to failures or success in the process. It is common to have resistance to changes during their execution, especially if they are imposed externally Sun-Jen & Wen-Ming, (2008), which can have a negative influence on the final result of the project.

If all possibilities of failure are not discovered and analyzed before the approval of the system, they will certainly be discovered later, with the software in production.

This can cause severe financial and administrative impacts for the organization, increasing the costs of the software, promoting losses and affecting the productive results in the organization.

According to Hijazi et al., (2014), the cost of correcting such failures will be high if they are not discovered and corrected in advance.

The main reason for software delays and cancellations is due to the large number of errors, the elimination of which can absorb more than 60% of the effort in large software projects. The North American average for removing software defects is 92.5%, referring to the 2017 period. In the implementation phase, the removal of defects in the beta and acceptance tests corresponds to approximately 20% Jones (2017).

Contextualization: Risk Factors in Software Deployment

Risk factors are uncertain conditions that can negatively affect the cost, duration and quality of a project, and if ignored or not reduced, they can present serious threats to the software project (Hijazi et al., 2014).

According to (Lehtinen et al., 2014), the main causes of risk factors found in the software deployment phase are associated with a low priority in the execution of tests, which can result in: (1) delays in software implementation; (2) problems related to productivity; (3) resistance to change; and (4) rejection of the software.

Problems discovered during the acceptance test can also cause a project to be suspended or restarted (Lehtinen et al., 2014).

The study carried out by (De Wet, Visser, 2013) shows that the management of risk factors, regardless of the environment used, produces a better result for the software project. According to Bannerman (2008), managing risk factors can lead to a series of organizational and project benefits, including: (1) better alternatives for action; (2) greater confidence in achieving the project's goals; (3) better chances of success; and (4) decreased team efforts.

Related Works

The literature has some works that highlight the importance of identifying, analyzing and measuring the risk factors and their impacts on the software process, which includes the implementation phase.

- In the work of Hijazi *et al*., (2014), each phase of the software development lifecycle is vulnerable to several types of risk factors. It presents a comprehensive theoretical study of the risk factors that threaten each phase of the software life cycle.
- The work of Shrivastava & Rathod, (2015), proposes the creation of a broad set of risk factors that affect the performance of projects in agile development, and identifies the risk management methods that are normally used in practice to control risks.
- In the research published by De Wet & Visser, (2013), failure analyzes of software projects were carried out in four companies. The proposal aims at a better understanding of the causes, failures and their risk factors.
- In his study, Gondal *et al*., (2018) makes a comprehensive analysis of the risk factors that may occur during each phase of the software development lifecycle. These factors are validated through questionnaires conducted by employees and employers of several software companies.
- In the proposal of Menezes et al., (2019), a study is presented aiming at the identification and mapping of risk factors in environments of software development projects. It carried out a systematic review of the literature, where he conducted a study that extracted and classified 41 works related to risk factors, according to the taxonomy proposed by the Institute of Software Engineering (ISE).
- Other studies that also contextualize the risk factors in the implementation of software and also involve the identification of strategies to contain the risk factors.
- To Shahzad et al., (2010), strategies for preventing and mitigating risk factors are seen based on the frequency of their occurrence in the software process.
- Already Khdour & Hijazi., (2012), proposes a model that integrates risk management with the software development process using a technique using the statistical method of variance.
- The proposal of Shahzad et al., 2010, presents an analysis of risk mitigation strategies in terms of effectiveness for the reduction of time and cost of software projects

Literature Review

This work started with a literature review presented by Santos *et al*., (2020), planned and executed from five essential steps, as suggested by Petersen *et al*., (2008): (1) the definition of research questions for the LR, (2) the mapping of relevant primary studies, (3) the sorting of documents, (4) keywording of abstracts, and (5) the extraction of data from primary studies. The results collected in the secondary study of Santos *et al*., (2020) is the benchmark for the comparative analysis of the risk factors and containment strategies identified in this work.

GENERAL CONCEPTS

Pressman and Maxim (2016) define the software process as a group of activities, actions and tasks orientated to creating of a software product. Where, (1) activity: is related to the achievement of broader objectives; (2) actions: they are seen as a set of tasks that result in a software artifact; and (3) task: it is related to specific objectives and the production of tangible results.

These three fronts, in general, are allocated in methodologies and models that determine an interactive cycle in the software process.

Software Development Life Cycle (SDLC)

The Software Development Life Cycle (SDLC) is one of the oldest development models and still the most applied in software projects (Begum, et al, 2010).

According to Hirama (2012), among the models that constitute the SDLC, stands out the waterfall model, which is a process focused on documents and artifacts, and that defines a linear and sequential software development. The software deployment phase is the last phase of these model.

Software Deployment Phase

Rezende (2013) states that the management process of deployment software is complex and dynamic, and that it generates significant changes in the structure, management and planning of organizations.

According to Mäntylä & Vanhanen (2011), software deployment is a set of critical activities for all software vendors, from an order for a new software requirement to the necessary measures for a new version available to the customer. This deployment is composed of activities that are essential to make a product available, such as: installation of dependencies, configuration files and installation of the application itself.

According to Sommerville (2016), during the deployment, errors in system configuration can generate new vulnerabilities, that can lead to system operations errors. When making changes to the system, some considerations made during the original purchase can be forgotten and again, vulnerabilities can be introduced into the system.

Therefore, identifying the risk factors of the software deployment phase is an activity that can facilitate the management of the deployment process.

Risk Factors in the Software Deployment Phase

According to Hijazi (2014), risk factors are uncertain conditions that can negatively affect the cost, duration and quality of a project and, if ignored or not reduced, can present serious threats to the software project.

The main causes of risk factors found in the software deployment phase are associated with low priority in test execution, which can result in problems related to software testing. External changes can also result in user resistance to changes and delays in software deployment.

In research carried out by De Wet & Visser, (2013), the management of risk factors produces improvements to the software project, regardless of the environment used.

There are several risks related to the deployment phase of a software product. However, few strategies and actions are established to avoid or address risks during the software development process, including deployment. These are probably because project risks have been discussed only in terms of cost, schedule, and technical aspects. (Tao, 2008).

Therefore, the software deployment stage can be problematic concerning the environment that will be deployed, since it is subject to several risk factors, ranging from the need to train users and cultural and regional aspects. Also involving organizational policies and planning, that if not considered, can cause damage to the software product and consequently lead the project to failure.

CASE STUDY: METHODOLOGY AND PLANNING

The methodology used in this Case Study, as well as its protocol, was based on the texts of (Wohlin et al., 2012), Yin (2015) and PMI (2017) for the definition of an action plan. In this way, the research was structured and executed according to the topics presented below.

Case Study Protocol

The protocol allows to improve the reliability of the case study, guiding the researcher in conducting the data collection. In this research, the protocol developed was guided by the proposal of PMI (2017), mainly on the following items: (1) maintaining a target on the topic of the case study; and (2) force the anticipation of problems, including the way in which results are packaged.

Environment Selection

This research was carried out in two government Brazilian companies, called Company A and Company B. Company A operates in the development of geosciences and has national coverage with approximately 2.140 employees - including government servants and third parties, working in offices based in the main Brazilian capitals. Company B operates in the metrology segment, performing technical laboratory tests, standardization and standardization with a focus on industrial quality. Company B also has offices in the main Brazilian capitals and has the support of around 2.000 employees, including civil servants and third parties.

Selection of Participants

The choice of participants took place through interviews carried out with the support of the managers of the technical teams responsible for implementing the SEI (Electronic Information's System). The criteria used were: (1) involvement with the project; (2) position or technical function; and (3) technical training. Figure 1 (a) and (b) shows the profile of participants from Company A, while Figure 1 (c) and (d) shows the profile of participants from Company B.

Selection of the Study Object

The SEI (Sistema Eletrônico de Informações), is a tool for managing electronic documents and processes, developed by the Federal Regional Court of the 4th Region (TRF4) and which has been adopted by several government companies and federal government agencies.

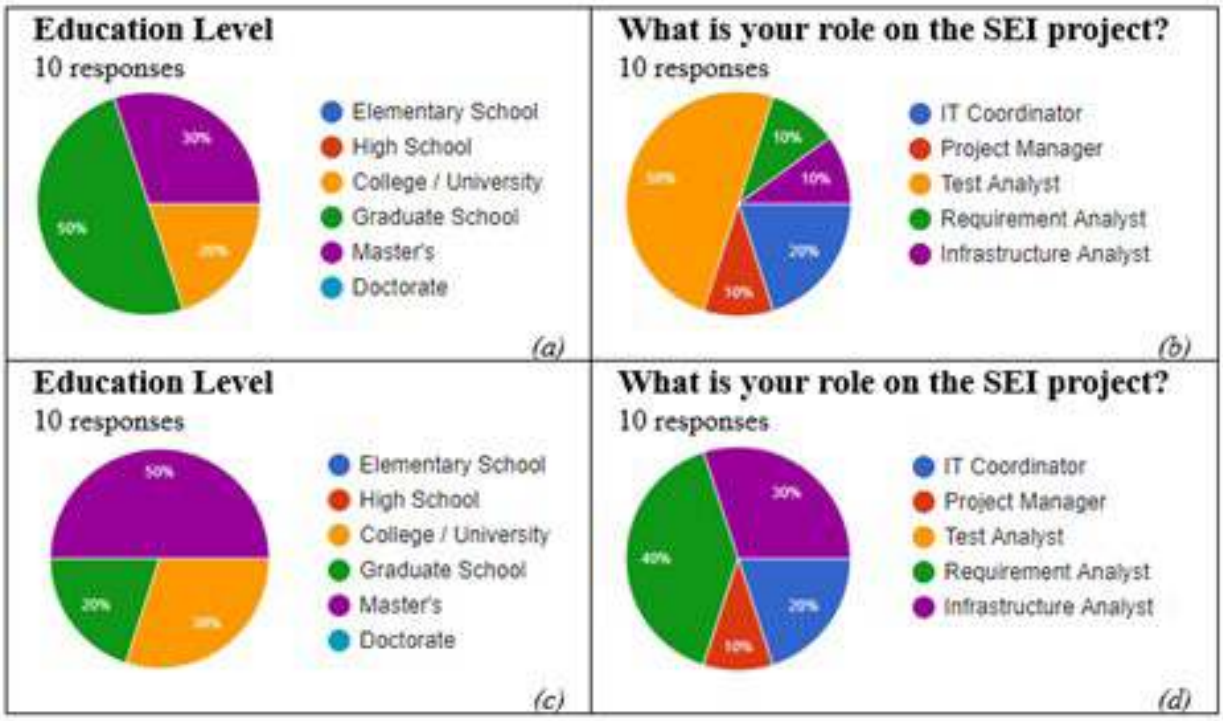

Figure 1. (a), (b), (c) e (d): Profile of participants

The SEI aims to promote administrative efficiency, in a joint initiative with bodies and entities from different spheres of government administration, with the aim of building a government infrastructure of electronic administrative processes and documents. SEI uses the benefits of electronic media to ensure efficiency, transparency and security. One of its remarkable innovations is the sharing of knowledge by electronic means, improving communication in real time. The system also provides the virtualization of the work process in the administrative area and allows the simultaneous performance of several units in the same process (Amaral, Uchôa, 2014).

Data Collection Method

To collect data from this research, the following mechanisms were adopted: (1) Technical visits for follow-up and on-site notes; (2) Meetings and interviews with the managers of the technical teams; (3) Application of questionnaires to members of technical teams, and (4) Application of questionnaires to other employees of the company that use the SEI.

The questionnaires were developed based on the Likert scale and were the main source of data collection to identify new risk factors and new containment strategies. The application of the questionnaires as well as the collection of their answers was given through electronic forms sent to the participants.

Research Forms

In addition to the interviews conducted at the companies, a research form was structured for application to the group of participants in the SEI implementation process. In total, 10 employees from Company A and 10 employees from Company B were selected.

The interview script was made up of topics related to the perception about the probability and impact of the risk factors existing in the implantation of the SEI.

EMPIRICAL ANALYSIS OF THE RESULTS

Data analysis showed an important contribution to the management of risk factors in the software deployment phase. Once risk factors and containment strategies were identified that had not yet been cataloged in the literature.

With the analysis of the data, it was also possible to quantify the users' perception and the impacts caused in the implantation of the SEI in the two studied companies.

Risk Factors Identified in the Literature

The secondary study presented by (Santos et al., 2020), resulted in the investigation of 23 primary studies that listed 11 risk factors and 13 containment strategies. This study was important because it directed the research and allowed the identification of new risk factors and new containment strategies during the software deployment phase.

Table 1 shows the risk factors and containment strategies identified in the literature. This is important because it allows you to view and compare with the data extracted from the two companies.

Table 1. Risk Factors and Containment Strategies identified in the literature

Risk Factors	Containment Strategies
Change in environment	Testing in a production environment.
Difficulties in using the system	Conduct user training.
Insufficient data handling	Adoption of a new methodology for analysis and control of existing data
Missing capabilities	Assign a cost to each identified risk based on the probability of occurrence and its level of impact
New requirements emerge	Perform actions that facilitate group discussions, making demonstrations to the client. Perform acceptance tests, evaluating the system against its original requirements and the current needs of the customer.
Problems in installation	Establish new management techniques for proper software integration.
Suspension and Resumption problems	Establish release dates for component integration and software deployment.
Testers do not perform well	Automate the management of test data using appropriate tools. Conduct a complete acceptance test with the user.
The effect on the environment	Adopt new hardware and software tools based on the organization's management guidelines.
Too many software faults	Prioritize the correction of software failures. Hiring experienced programmers.
User resistance to change	Conduct user training in order to solve the problem of acceptance of the software.

Table 2 presents a comparison of common risk factors, found in companies A and B and identified in the literature.

Table 3 compares the containment strategies adopted by the two companies in relation to the risk factors identified in the literature and which occurred in the implementation of the SEI.

Table 2. Comparative - Risk Factors Identified in Companies A & B Present in the Literature

Ref.	Risk Factors Identified in the Literature	Risk Factors (RF)	
		Company A	Company B
FR1	Difficulties in using the system	Insufficient or inadequate training of the project team or new users of the system.	Difficulty of employees in using the system.
FR2	User resistance to change	User resistance to new system routines.	User resistance to new technologies.
FR3	Problems in installation	Inadequate or insufficient parameterization.	Difficulties in installing system modules
FR4	Missing Capabilities	Absence of room for training users of the system	Lack of space / equipment for storing system information

Table 3. Comparative - Containment Strategies Adopted in A & B Companies

Ref.	Risk Factors identified in the literature	Containment Strategies (CS)	
		Company A	Company B
CS1	FR1	Carry out training plan: diagnose difficulties and promote actions to solve them.	Training Plan: conduct continuous training.
CS2	FR2	Maintain actions related to motivation.	Support from senior management and communication plan.
CS3	FR3	Strategy adopted in real time.	Lack of technical training of Information Communication Technology (ICT) members to treat incidents and implement the solution.
CS4	FR4	Training Plan: review planning and adapt it to the new context.	Update data storage solution(storage) Initially scan only the processes being processed.

Where: *CS: Containment Strategy; RF: Risk Factor*

Table 4 shows that the strategies adopted to reduce risk factors are related to political and operational issues of the companies. However, in Company A, more attention is focused on the strategic management of risk factors, unlike Company B, which prioritizes its strategies based on technical and operational solutions.

Table 4. Containment Strategies of Company A and B not identified in the literature.

Company A			Company B		
Ref.	**Risk Factors**	**Containment Strategies**	**Ref.**	**Risk Factors**	**Containment Strategies**
ECEA1	Delay in concluding the technical cooperation agreement with the Ministry of Planning.	Promote contacts with the Ministry of Planning to streamline procedures.	ECEB1	Insufficient capacity of scanners printers in the areas	Make compatible the distribution of printers from the Outsourcing of printing with the implementation Project of the system.
ECEA2	Designation of project members for other activities, not related to the project.	Intensify actions aimed at raising awareness among managers.	ECEB2	Unit authentication integration problems.	Maintain distributed authentication infrastructures.
ECEA3	Government servant's departure from the Project.	Check the possibility of replacement.	ECEB3	Maintenance of old processes in the system without need.	Do not initially scan the processes already closed. It will be carried out according to demand.
ECEA4	Lack of alignment of expectations with the Executive Board.	Carry out a communication plan in line with the Executive Board.		Not identified.	Not identified.
ECEA5	Delay in the acquisition of equipment.	Promote meetings with the responsible sectors and check the possibility of prioritizing demands.		Not identified.	Not identified.
ECNE6	Idle employees (secretaries, messengers) due to the change in work routines.	Resize and / or redistribute employees across sectors.		Not identified.	Not identified.

Where: ECEA: Containment Strategy of Company A; ECEB: Containment Strategy of Company B

Users Perception - Company A

Table 5 presents a list of risk factors and responses of the respective employees, based on the probability of their occurrence.

Table 5. Probability of Risk Factors - Company A

Risk Factor (Probability)	Yes	No
Problems during deployment	10	-
User's Resistance to Change	9	1
New requirements emerging	9	1
Lack of Resources	8	2
Failure in the SEI	8	2
Changes in the Environment	7	3
Changes to the Operating Environment	6	4
Difficulties in using the system	6	4
Data manipulation failed	6	4
Test Failures	6	4
Process Suspension and Restart	5	5

All respondents from company A flagged the Problems during deployment as the main risk factor. According to (Hijazi et al., 2014), if developers do not have enough experience of the nature of the system and how it works, deployment problems may occur, and with this, key functionality may be installed incorrectly.

The respondents also mentioned the risk factors related to the user's resistance to change and the emergence of new requirements. The results showed that the risk factor that obtained the lowest probability index was related to Suspension and Process Restart (Table 6).

Table 6. Impact of Risk Factors – Company A

Risk Factor (Impact)	Too Low	Low	Medium	High	Too High
Changes in the Environment	1	4	4	1	-
Difficulties in using the system	1	2	2	4	1
Data manipulation failed	2	2	3	3	-
Lack of Resources	2	2	3	3	-
New requirements emerging	1	1	5	3	-
Problems during deployment	-	2	4	3	1
Process Suspension and Restart	3	2	2	2	1
Test Failures	2	2	2	3	1
Changes to the Operating Environment	-	5	2	3	-
Failures in the SEI	1	3	1	4	1
User's Resistance to Change	-	1	1	6	2

Table 7 shows that all respondents mentioned that the main containment strategy in the SEI implementation process is related to the Training of Software Users, which corresponds to one of the strategies identified in the literature.

According to (Sharma & Yetton, 2007), when organizations invest in new systems or software, end-user training is a critical factor for system success.

Table 8 provides a list of the identified risk factors and the responses of the respective employees, based on the probability of their occurrence.

Table 8 shows that the risk factors mentioned by most respondents are related to: (1) Problems during implantation with a 90% index; (2) Lack of Resources and Test Failures with an index of 80%, and (3) Difficulty in using the System and User Resistance to Changes with an index of 70%.

Table 9 shows the result of the impact of risks on the view of the respondents of company B.

Table 10 shows the containment strategies according to the perception of company B respondents.

Table 7. Perception of Users Interviewed - Company A

Containment Strategies	Percentage
Training of software users.	100%
Establish a software release date, maintaining the infrastructure for component integration.	90%
Support from senior management as well as organizational changes.	80%
Allocate the experienced team for critical tasks to ensure that no delay is expected.	70%
Empower the team to predict new software requirements and adhere to the dynamic circumstances of the company.	70%
Perform actions that facilitate group discussions, performing demonstrations to the client.	70%
Deployment of new resources and allocations.	60%
Promote actions for the greatest degree of involvement and commitment of users.	60%
Adequately manage the integration of the software with the existing business.	40%
Perform a full acceptance test with the user.	40%
Selection of new implementation methodology.	20%
Apply the management of risk factors in the phases prior to software deployment.	-
Automate test data management using appropriate tools.	-
Hiring experienced programmers.	-
Conducting automated tests for the management of risk factors.	-
Other	-

Table 8. Probability of Risk Factors – Company B

Risk Factor (Probability)	Yes	No
Problems during deployment	9	1
Lack of Resources	8	2
Test Failures	8	2
Difficulties in using the system	7	3
User's Resistance to Change	7	3
Failures in the SEI	7	3
New requirements emerging	5	5
Changes to the Operating Environment	5	5
Changes in the Environment	5	5
Data manipulation failed	4	6
Process Suspension and Restart	4	6

Table 9. Impact of Risk Factors – Company B

Risk Factor (Impact)	Too Low	Low	Medium	High	Too High
Changes in the Environment	3	3	1	1	2
Difficulties in using the system	1	1	5	1	2
Data manipulation failed	-	3	3	-	4
Lack of Resources	-	-	2	5	3
New requirements emerging	2	2	4	1	1
Problems during deployment	-	1	4	2	3
Process Suspension and Restart	2	3	-	4	1
Test Failures	-	3	2	4	1
Changes to the Operating Environment	2	4	-	2	2
Failures in the SEI	-	3	2	2	3
User's Resistance to Change	1	2	2	4	1

Table 10. Perception of Users Interviewed - Company B

Containment Strategies	Percentage
Training of software users.	90%
Establish a software release date, maintaining the infrastructure for component integration.	90%
Perform actions that facilitate group discussions, performing demonstrations to the client.	80%
Promote actions for the greatest degree of involvement and commitment of users.	80%
Support from senior management as well as organizational changes.	70%
Adequately manage the integration of the software with the existing business.	60%
Allocate the experienced team for critical tasks to ensure that no delay is expected.	50%
Deployment of new resources and allocations.	50%
Perform a full acceptance test with the user.	40%
Apply the management of risk factors in the phases prior to software deployment.	30%
Empower the team to predict new software requirements and adhere to the dynamic circumstances of the company.	20%
Conducting automated tests for the management of risk factors.	20%
Selection of new implementation methodology.	20%
Automate test data management using appropriate tools.	10%
Hiring experienced programmers.	10%
Other	-

It is possible to observe in Table 10 that 90% of the respondents in company B indicated the training of users of the software and the establishment of a release date as the main containment strategies.

However, only 10% of respondents flagged as relevant the automation of test data management and the hiring of experienced programmers.

Comparison of Risk Factors identified in Companies A and B

Based on the data collected with the application of questionnaires / research forms, Table 11 demonstrates a comparison of the risk factors identified in Companies A and B:

Table 11. Comparison of Risk Factors – Companies A and B

Risk Factors	Company A (%)	Company B (%)
Problems during deployment	100%	90%
User's Resistance to Change	90%	70%
New requirements emerging	90%	50%
Lack of Resources	80%	80%
Failures in the SEI	80%	70%
Changes in the Environment	70%	50%
Changes to the Operating Environment	60%	50%
Difficulties in using the system	60%	70%
Data manipulation failed	60%	40%
Test Failures	60%	80%
Process Suspension and Restart	50%	40%

Table 11 shows that risk factors related to problems during implantation and user resistance to change were between 70% and 100% of the probability of occurrence. However, according to the respondents, the risk factor related to the suspension and restart of the process obtained a low index, ranging from 40% to 50% probability of occurrence. This implies that, in both cases, a good action plan was executed, following the planning of project management.

Comparison of Containment Strategies Identified in the Research

Table 12 presents a comparison of the main containment strategies adopted in Companies A and B and which had their results closer to the percentage of respondents of the survey:

Table 12. Comparison of Containment Strategies - A and B Companies

Containment Strategies	Company A (%)	Company B (%)
Training of software users.	100%	90%
Establish a software release date, maintaining the infrastructure for component integration.	90%	90%
Support from senior management as well as organizational changes.	80%	70%
Perform actions that facilitate group discussions, performing demonstrations to the client.	70%	80%
Deployment of new resources and allocations.	60%	50%
Perform a full acceptance test with the user.	40%	40%
Selection of new implementation methodology.	20%	20%

The results presented in Table 11 indicate that the companies had their highest index of containment strategies in actions aimed at training software users and determining a release date of these trainings. On the other hand, the company's respondents indicated a 20% index for the containment strategy related to the selection of a new implementation methodology, suggesting that the methodology adopted for the implementation of the SEI was appropriate.

CONCLUSION

This work presented a case study carried out in two Brazilian government companies using a "*quali-quantitative*" approach on risk factors and their containment strategies. The data were collected from an empirical investigation involving the case study environment (companies A and B) and the object of study (SEI), with the application of a survey to the employees of the companies.

As a result, this research presented 9 new risk factors and 9 containment strategies that had not been cataloged to date, thus expanding the list of risk factors and containment strategies already identified in the software deployment phase.

Limitations and Further Works

First, this research sought the risk factors and containment strategies already identified in the literature through an RL. In the second phase, it was investigated whether these same factors also occurred in the studied environment. Finally, the study sought to identify new risk factors and containment strategies in both companies. Therefore, this study was limited to investigating a specific scenario, being restricted to only two companies. The work was also limited to investigating only in the software deployment phase, *i.e.,* the results presented do not apply to other phases of the software process.

However, the gaps left in this study emerge as potential fronts for future research and serve as a guideline to start new studies in this area, such as: (1) Repeat the study in a private corporate environment and compare the results with those presented in this work; (2) Repeat the study in government companies using other information systems and verify that the risk factors and containment strategies are the same; (3) Apply this study in other phases of the software development process; (4) Produce a technical report based on risk factors and containment strategies related to the software deployment process; and (5) Develop an impact assessment model of identified and untreated risk factors in the software deployment process.

REFERENCES

Amaral, V. L.; Uchôa, C. E. (2014) National electronic process: its collaborative construction and its perspectives. In: VII Consad Congress of Government Management, Brasília - DF. Available in: http://consadnacional.org.br/vi-congresso-consad-trabalhos-apresentados/.

Bannerman, P. L., (2008), 'Risk and risk management in software projects: A reassessment', The Journal of Systems & Software, 81, Best papers from the 2007 Australian Software Engineering Conference (ASWEC 2007), Melbourne, Australia, April 10-13, 2007, pp. 2118-2133, ScienceDirect, EBSCOhost.

Begum, Z., Khan, M. S. A., Hafiz, Z., Islam, S.;. Software development standard and software engineering practice: A case study of Bangladesh. arXiv: preprint1008.3321, 2010.

De Wet, B.; Visser, J. (2013). An Evaluation of Software Project Risk Management in South Africa. The South African Journal of Industrial Engineering, Vol 24, Iss 1, Pp 14-28 (2013), 1, p. 14, Directory of Open Access Journals, EBSCOhost.

Gondal, H. A. H., Din, S. M. U., Fayyaz, S., Zeb, M. D., & Nadeem, B. (2018, February). Preeminent risk factor affecting software development. In 2018 International Conference on Advancements in Computational Sciences (ICACS) (pp. 1-7). IEEE.

Hijazi, H, Alqrainy, S Muaidi, H. (2014). Risk factors in software development phases. European Scientific Journal, January, edition, vol.10, Nº.3.

Hirama, K.; Engenharia de software: qualidade e produtividade com tecnologia. Rio de Janeiro: Elservier, 2012. 209 p.

Jones, C., (2017). Exceeding 99% in Defect Removal Efficiency (DRE) for Software., Technical Report, Software Estimation Journal; Namcook Analytics; Saint John – AB.

Khdour, T., Hijazi, H. (2012, September). A step towards preventive risk management in software projects. In Proceeding of the 2012 4th International Conference on Software Technology and Engineering. Phuket, Thailand, September (pp. 1-2).

Lehtinen, T., Mäntylä, M. V., J., Itkonen, J., Lassenius, C., (2014). Perceived causes of software project failures – An analysis of their relationships. Information and Software Technology 56,6 (2014),623–643.

Mäntylä, M.; Vanhanen, J. Software deployment activities and challenges - a case study of four software product companies. In: *Software* Maintenance and Reengineering (CSMR), 2011 15th European Conference on., 2011. p. 131–140. ISSN1534-5351

Menezes, J., Gusmão, C., & Moura, H. (2019). Risk factors in software development projects: a systematic literature review. Software Quality Journal, 27(3), 1149-1174.

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008, June). Systematic mapping studies in software engineering. In Ease (Vol. 8, pp. 68-77).

PMI - Project Management Institute. (2017). A Guide to the Project Management Body of Knowledge (PMBOK® Guide) 5th Edition.

Pressman, Roger S.; Maxim, Bruce R. (2016), Engenharia de *Software*: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016.

Rezende, D. A.; (2007), Planejamento de Sistemas de Informação e Informática: Guia Prático para Planejar a Tecnologia da Informação Integrada ao Planejamento Estratégico das Organizações, Editora Atlas; São Paulo, SP, 2007.

Santos, L., Ribeiro, S., Schmitz, E., Silva, M., & Alencar, A. (2020). Risk Factors in the Software Deployment Phase: A Literature Review., Holos, 1, 1-14. http://doi.org/10.15628/holos.2020.8640.

Santos, L., (2020). Risk Factors in the Software Implementation Phase: A Case Study Applied to Brazilian Government Companies., Master's Thesis. Tércio Pacitti Institute; PPGI/UFRJ; Rio de Janeiro-RJ.

Sharma, R., Yetton, P. (2007). The contingent effects of training, technical complexity, and task interdependence on successful information systems implementation. MIS Ouarterly, 31(2), 219-238.

Shahzad, B., Al-Mudimigh, A. S., Ullah, Z. (2010). Risk identification and preemptive scheduling in software development life cycle. Global Journal of Computer Science and Technology.

Shahzad, B., Safvi, S. A. (2010). Risk mitigation and management scheme based on risk priority. Global journal of computer science and technology.

Shrivastava, S. V., Rathod, U. (2015). Categorization of risk factors for distributed agile projects. Information and Software Technology, 58, 373-387.

Sommerville, I. (2016). Software Engineering, Addison Wesley.

Tao, Y. (2008), A study of *software* development project risk management. Proceedings of the International Seminar on Future Information Technology and Management Engineering, p.309-312, 2008.

Sun-Jen, H.; Wen-Ming, H.; (2008). Exploring the relationship between software.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). Experimentation in software engineering. Springer Science & Business Media.

Yin, R. K. (2015). Case study: planning and method. Translation: Cristhian Matheus Herrera. Edition-Porto Alegre: Bookman.