# Investigating Change and Incident Management in a Proprietary Software Ecosystem - Survival Mode On

**Luiz Alexandre Costa**[1], **Alexandre Sampaio**[2]
**Awdren Fontão**[3], **Rodrigo Pereira dos Santos**[1]

[1]Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Rio de Janeiro – RJ - Brasil

[2]Universidade de Aveiro (UA) - Aveiro - Portugal

[3]Universidade Federal de Mato Grosso do Sul (UFMS)
Campo Grande – MS - Brasil

luiz.costa@edu.unirio.br, alex.sampa@ua.pt

awdren@facom.ufms.br, rps@uniriotec.br

***Abstract.*** *Proprietary Software Ecosystems (PSECO) are those that concentrate knowledge on a closed software platform. The growing number of demands causes organizations to work at an accelerated pace to deliver results. Changes and incidents have a negative impact on the workload, compromising its stability. Based on a case study in a large international insurance organization, this study aims to verify the behavior of development teams from the point of view of changes and incidents in a PSECO. The following research methods were applied: i) robotic process automation; ii) data mining; and iii) social network analysis. As a result, this study serves as an alert for the organization's IT board regarding the governance of traditional software development processes to survive in a competitive market.*

***Resumo.*** *Ecossistemas de Software Proprietário (ECOSP) são aqueles que concentram o conhecimento em uma plataforma de software fechado. O crescente número de demandas faz com que as organizações trabalhem em ritmo acelerado para entregar resultados. Mudanças e incidentes impactam negativamente na carga de trabalho, comprometendo sua estabilidade. A partir de um estudo de caso em uma grande organização internacional de seguros, este trabalho tem como objetivo verificar o comportamento das equipes de desenvolvimento sob o ponto de vista das mudanças e incidentes no ECOSP. Foram utilizados os seguintes métodos de pesquisa: i) automação robótica de processos; ii) mineração de dados; e iii) análise de redes sociais. Como resultado, este estudo serve como um alerta para a diretoria de TI da organização quanto à governança dos processos tradicionais de desenvolvimento de software para que sobrevivam em um mercado competitivo.*

## 1. Introduction

Organizations have been working cooperatively and competitively to support new products, satisfy customer needs, and eventually incorporate the next round of innovations. Increasing attention is being paid to connectivity and dependency in relationships among

organizations involving several actors (e.g., suppliers, distributors, outsourcing companies, software providers, developers, and managers) [Dahesh et al. 2020].

From this perspective, researchers created a concept to be analyzed in the software industry called software ecosystems (SECO). According to Jansen et al. (2009), SECO is a set of actors functioning as a unit and interacting in a shared market for software and services, centered on a common technological platform. A proprietary SECO (PSECO) can be characterized by the contributions of several products, technologies, and architectures protected by intellectual property and confidence agreement (e.g., SAP) [Costa et al. 2021]. The central organization in PSECO is called a keystone. A keystone must establish governance policies as a critical strategy for ensuring a sustainable platform that aims to resist to natural changes, such as business evolution and technological obsolescence [Dhungana et al. 2010].

The challenge of maintaining a sustainable platform has become a priority for large organizations based on the survey performed by Gartner Group[1]. The software assets (products and applications) that make up the architecture of the PSECO technological platform are built using several technologies and different human resources skills, incurring in architectural complexities. Sustaining these complex systems in order to avoid disruptions or downtime (also known as incidents) is a concern that causes major image and financial upheavals for a keystone [Grieves and Vickers 2017].

A factor influencing the synergy between the business strategy alignment and the PSECO technological platform evolution is market pressure for a state-of-the-art solution for every business need. It causes organizations to work at a highly accelerated pace, passing this anxiety to the IT project team, which must deliver results in an increasingly short time [Kappelman et al. 2006]. In this scenario, changes in the production environment, whether infrastructure or new software releases, are always dangerous, as any change can lead to negative impacts on workload, compromising its stability. However, a successful change management process in the organization allows software releases to be deployed with less risk on end-users (internal or external) and a higher return on investment [Lahtela and Jäntti 2011].

In this work, we aim to investigate the practices and behaviors of the development team regarding change and incident management in the PSECO of a large international insurance organization and understand how these aspects affect the quality and stability of the software. We develop a "bot" using RPA (Robotic Process Automation) techniques to automate the changes and incidents data extraction from the enterprise ALM (Application Lifecycle Management) system. Next, we run a data mining process to explore the data set results to find behavior patterns. Finally, we map the network formed by the interactions established among change data, incidents and software development cost centers. The implications of our work take a holistic view of some governance mechanisms. It is an opportunity for the academic community to investigate the value of new metrics in increasing end-user satisfaction and attracting new business. On the other hand, practitioners in the software industry should be aware that organizations with mature software development processes have a competitive advantage over their competitors.

[1]Gartner is the world's leading research and advisory company (https://www.gartner.com/en/ documents/4006716).

## 2. Background

### 2.1. Software Ecosystems

According to Bosch (2009), software ecosystems (SECO) is a set of software solutions that enable, support, and automate the transactions among actors and the organizations that provide these solutions in an associated social or business ecosystem. It consists of a software platform, a set of internal and external developers, and a community of experts serving the needs of a community of users aiming to build valuable solutions.

A SECO classification from a value creation perspective can be: **i) proprietary**: in which the source code and other artifacts produced are protected by confidentiality agreements, as they are the products that would yield revenues to the ecosystem, e.g., e-commerce ecosystems; **ii) open**: in which the actors do not participate to obtain direct revenues from their activity in the ecosystem, e.g., Eclipse and Apache Foundations; and **iii) hybrid**: in which supports proprietary and open source contributions, e.g., iOS SECO may use proprietary strategies as app store and the source code repository to drive policies in the technological platform, and open source strategies for community engagement as tools, submission, and publication contributions [Manikas and Hansen 2013]. The focus of this work is on proprietary SECO (PSECO).

### 2.2. Change Management and Incident Management

SECO are complex environments composed by diverse actors interacting in a distributed software development environment based on a platform, where these actors give technological support to the established environment [Santos and Werner 2011]. SECO governance requires a careful balance of control and autonomy and becomes a managerial aspect for proprietary platform owners and open source communities [Alves et al. 2017]. In this context, the Information Technology Infrastructure Library (ITIL) emerges as a framework that can be used to implement best practices for governance and move towards more effective operations on the PSECO technological platform.

The ITIL library comprises five distinct volumes: Service Strategy; Service Design; Service Transition; Service Operation; and Continual Service Improvement. The theoretical lens of this study is focused in two core process in the context of PSECO: incident management and change management. Change management is the process responsible for controlling the lifecycle of all changes in a standardized and systematic manner with minimum disruption to IT services. On the other hand, incident management is the process responsible for solving the incident as quickly as possible [Addy 2007]. According to ITIL, a change is the addition, modification, or removal of anything that could have an effect on services, and an incident is an unplanned interruption of an IT service.

## 3. Related Work

Costa et al. (2021) propose an approach to measuring the health of a PSECO and discuss governance strategies that can be implemented based on metrics. Jafari et al. (2020) addressed the effective communication between several project actors to control delays and costs in change management. The authors applied social network analysis (SNA) methods to the change process, extracting and mining huge volumes of data. This study described how individuals are engaged in the communication network and emphasized the communication in process enhancement.

Paszkiewicz and Picard (2013) provided real-life datasets in the form of event logs used across a PSECO to support incident and problem handling. The authors used process mining and SNA techniques to detect the existence of hidden events that degraded the overall performance of incident handling and pointed out non-compliance processes. Thus, the relevant difference to our work is that we have a more concrete approach that designs and promotes a service management strategy that works for a specific PSECO use case towards resilience, availability, and reliability objectives. We discuss governance of the software development team, in which software releases, changes, testing, and deployment are not effectively managed, resulting in incident tickets.

## 4. Research Method

The goal is to understand the behavior of development teams from the perspective of changes and incidents in the PSECO of a large international insurance organization, based on the study of Costa et al. (2021) that the development teams present homogeneous behavior in relation to the proportion among changes and incidents, since these teams should be under the same governance guidelines in PSECO. The research question for our study is: *"How do PSECO software development teams behave in relation to their interactions (e.g., incidents and changes) in the productive environment?"*

To answer the research question, we ground our work on methods from Empirical Software Engineering guidelines [Wohlin et al. 2012]. Our methodology is divided in the following steps, as shown in Figure 1: ***(1) Ad hoc Literature Review:*** the method proposes an initial literature review as a way of collecting and synthesizing previous research to build knowledge for the foundation of other activities [Tranfield et al. 2003]. In our context, we studied the literature on change and incident management based on ITIL guidelines, data mining processes, and SNA as a methodological approach to explore the interactions of development teams in PSECO. Ad hoc Literature Review supported us to reinforce the need for studies covering changes and incidents in PSECO; ***(2) Robotic Process Automation (RPA):*** this method allowed us to gather changes and incident unstructured data in an automated way from the enterprise ALM of the organization; ***(3) Data Mining Process:*** this method was used by the researchers to turn raw data into useful information; and ***(4) Social Network Analysis (SNA):*** this method served to better understand the relationships inherent in PSECO through innovative methods of analysis. SNA was used to depict, diagnose, and evaluate the relationships and networks therein.

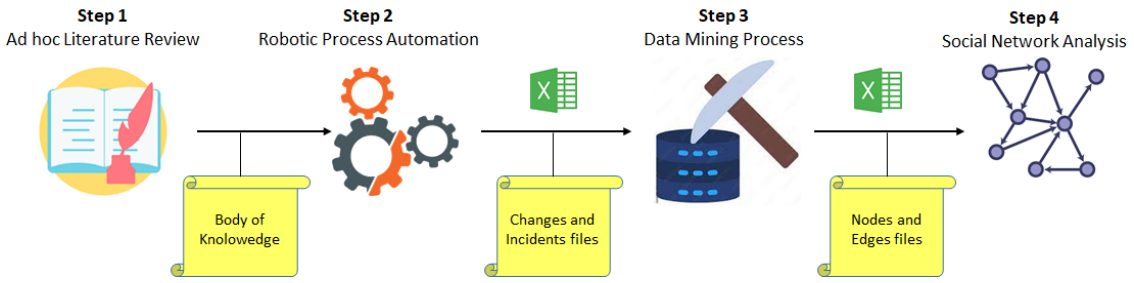


**Figure 1. Research methodology.**

## 5. Case Study

Case study is an adequate research method for situations in which it is difficult to establish a clear link between the studied phenomenon and its context, in such a way that it is not possible to investigate the phenomenon outside of the practical environment [Yin 2005] (i.e., when studying the technological and social scalability of a system within an organization, it is not possible to separate the actors from the processes).

### 5.1. Organization Characterization

Due to privacy reasons, the organization's name was omitted, and we will refer to it as Organization "X" from now on. Founded more than 80 years ago, "X" is currently one of the largest insurance groups in Latin America, operating internationally in several insurance segments: auto, property, health, capitalization, and open supplementary pension. The organization has more than 200 branches (service centers, offices and customers service) across the country and a partnership with more than 40,000 insurance brokers.

### 5.2. Organization Diagnosis

In a business competitive environment, IT plays an important role in the performance of the organization, especially when it provides a flow of information that adds value without weakening organizational efficiency [Brown 2003]. Organization "X" decided to change its vision, focusing on client services and generating value for the company's business. This mindset is also called Digital Transformation (DT).

DT is the process in which companies use digital technologies to solve traditional problems, such as drops in performance, productivity, agility, and effectiveness. This transformation must start with a structural change in organizations. For that reason, the organization "X" faces a major challenge: the rules that drove business progress in the pre-digital era (before the rise of the Internet) no longer apply. An example of the challenge is the modernization of the software development process through DEVOPS culture. DEVOPS (acronym of software DEVelopers and OPerationS) is the combination of practices and tools designed to increase an organization's ability to deliver applications and services faster than traditional software development processes [Davis and Daniels 2016].

Increasingly, the market pressure for a state-of-the-art solution causes organization "X" to work at a highly accelerated pace, passing the anxiety to IT project team, which must deliver results in an increasingly short time. Some problems emerged as a consequence of the growing number of demands added to the lack of flexible and modern processes, such as people's healthcare problems, software developers looking for new job opportunities, late projects, over budget, and the deployment of software releases (managed by the change management process) may cause unplanned interruptions (managed by the incident management process).

It is noteworthy that in such a complex scenario, the organization "X" decided to implement change and incident management practices using ITIL methodology in the first semester of 2019, aiming to minimize those problems. The company is divided into organizational units called cost centers. Each cost center is made up of several families of software assets based on the development teams. As such, it established governance policies and guidelines as a critical strategy for maintaining a sustainable PSECO platform. A sustainable approach refers to how the platform can resist natural changes, e.g., business evolution, technology obsolescence, and community changes [Dhungana et al. 2010].

### 5.3. Robotic Process Automation

The enterprise ALM tool was designed to support the critical tasks of managing the change process throughout the lifetime of an application. It touches all parts of the organization "X", including development, operations, and governance, providing strong collaboration between all stakeholders. As a consequence, the tool is also used to manage the software change process and ensure effective end-to-end incident tracking and reporting.

However, the enterprise ALM tool has a limitation and does not provide an API (Application Programming Interface) to provide a comprehensive knowledge base for end-user self-service and optimize technical support efficiency. As an alternative solution, we developed an RPA "bot" using Python programming language and *Selenium* library to automate mouse clicks and keystrokes (i.e., simulate what a human would do). We fill the input date interval filter on the ALM search screen to download all incident and change data from the last two years.

### 5.4. Data Mining

The *incident* and *change* data files are inputs to the process of data mining. Data mining is the process of structuring, analyzing, and formulating massive amounts of raw data in order to find patterns and anomalies through mathematical and computational algorithms [Lee and Siau 2001]. Through learning the techniques of data mining, we can use this knowledge to generate new insights and find new trends. The process of mining data can be divided into three main parts: gathering, collecting, and cleaning the data. Next, we apply a data mining technique to the data and validate the results.

There are several techniques (i.e., MapReduce and Clustering) that we could use to perform data mining. However, we focus on the Link Analysis (LA). LA is useful in cases where relationships between different data points can be observed more easily [Lee and Siau 2001]. LA is a data mining technique based on a branch of mathematics called graph theory. Graph theory represents different actors (nodes) and the relationships between them (edges) as a graph [Gross et al. 2018]. LA was performed in four steps using Python programming language and the *Pandas* library:

1. Data Processing: we collect and manipulate the incident data and change data files using sorting and classification algorithms;
2. Transforming: we convert data from one structure into another suitable for the analysis we choose to perform;
3. Analysis: once the data has been transformed, we applied SNA process to extract the desirable information; and
4. Visualization: we published information using graphs for visual representation.

### 5.5. Social Network Analysis

The *change* data source contains data from March 2017 to November 2022, and the *incident* data source contains data from October 2021 to December 2022. The differences between periods are due to the log persistence rules for each category of data source. To ensure consistency, we filter the data in both sources using the shortest available period criteria in common (October 2021 to November 2022). We found three event types:

- EV1: CC(i) $\rightarrow$ RFC(j) as Cost Center (i) performs Request For Change (j);

- EV2: INC(ni) $\rightarrow$ CC(i) as Incident (ni) is associated to Cost Center (i); and
- EV3: CC(i) $\rightarrow$ CC(k) as Cost Centers (i) and (k) collaborate with one another.

Organization "X" operates by distributing its several families of software assets to different cost centers based on the development teams. It is worth noticing that there are no data in the available data sources of any direct associations of the INC $\rightarrow$ RFC type, which would allow for specific inferences. The need to understand the behavior of cost centers in EV1 and EV2 types and verify possible indirect connections among the same cost centers in EV3 type led us to choose SNA methods and graph theory as the mathematical support. We used the Gephi[2] tool to render graphs and calculate metrics.

## 6. Results and Discussion

From the perspective of SNA, we found three distinct types of network entities: the cost center (CCi), the incident (INCin), and the change (RFCj). These networks can be represented through a multipartite directed graph where the nodes are the entities and the edges are events of types EV1 and EV2.

Table 1 shows the refinement process carried out and the variations in the network: i) from the initial graph, we filtered the nodes with zero degree centrality and removed them (incidents without an associated cost center). This action resulted in an intermediate graph, with a significant reduction in disconnected components; and ii) from the intermediate graph, after verifying the representativeness of each of the remaining 79 components, we found that the giant component encompassed 92.6% of the nodes and 93.5% of the connections. That is the reason why we chose it as the focus of the analysis. In SNA, giant component is the one that has the largest number of connected nodes. This graph is referred to as GMulti.

**Table 1. Results of the refinement process.**

| | Inicial Graph | Intermediate Graph | Giant Component |
|---|---|---|---|
| Nodes | 11,452 | 11,175 | 10,353 |
| Edges | 11,791 | 11,791 | 11,028 |
| Components | 356 | 79 | 1 |
| Cost Centers | 169 | 169 | 83 |
| Changes | 3,892 | 3,892 | 3,558 |
| Incidents | 7,391 | 7,114 | 6,712 |

Each INC(ni) node is always directed to a CC(i) node, thus containing only out-degree values. On the other hand, RFC(j) nodes will only contain in-degree values, with connections provided by CC(i) nodes. After calculating the centrality metrics, we found that 17% of changes RFC(j) had an in-degree greater than one, which characterized a phenomenon of sharing change events (EV1) between different cost centers. This finding motivated us to generate a monopartite graph of EV3 interactions to measure the behavior. This graph is referred to as GMono. Figure 2 shows the GMulti and GMono graphs rendered by the Force Atlas algorithm[3]. CC(i) nodes were anonymized for privacy reasons.

[2]Gephi 0.9.7 (https://gephi.org/)

[3]Algorithm that allows the representation of a graph taking into account the individual characteristics of the nodes and their connections.

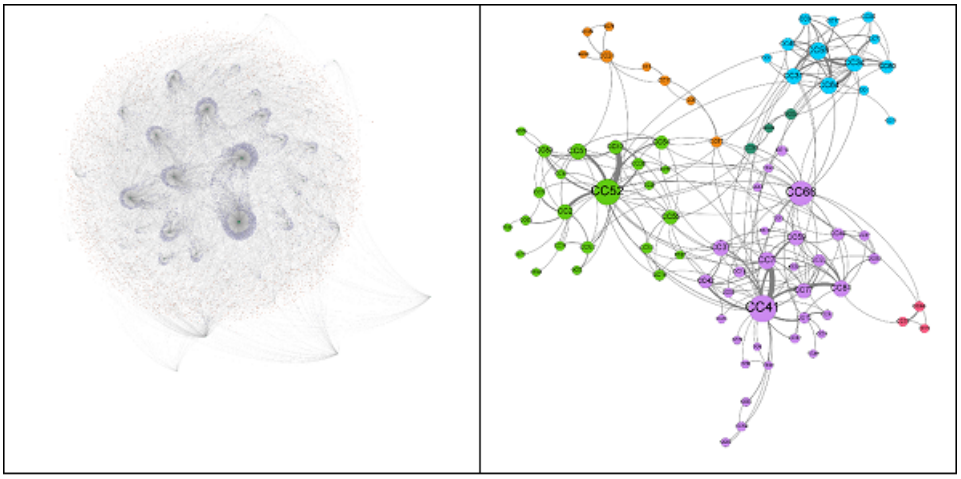


**Figure 2. GMulti graph on the left side and GMono on the right side.**

The GMono graph provides 83 cost centers (CCi) in total, interconnected by 233 connections, representing the EV3 events. Node dimensions vary according to their weighted degree centrality, which represents how much a cost center shares changes with others. The thickness of connections indicates the strength of sharing for a specific pair of cost centers. The colors of the nodes highlight six different sharing communities in total, identified by the detection algorithm of Blondel et al. (2008). In SNA, communities characterize groups of nodes that establish common patterns of connection.

From the centrality metrics provided for the GMulti graph, we verify in Figure 3: i) 80% of changes come from 26% of the total cost centers (step 1, 22 of 83); ii) 80% of incidents are attributed to only 16% of the total cost centers (step 2, 13 of 83); and iii) 12% belongs to both groups previously noticed (step 3, 10 of 83). From the centrality metrics of the GMono graph, we also verify in Figure 3: i) 80% of the sharing are attributed to 28% of the total cost centers (step 1, 23 of 83); and ii) 8 cost centers are responsible for 65% of all incidents (step 2, 8 of 13). The imbalance in the distribution of network connections between cost centers could also be observed in the distribution of degrees in both graphs through a power function, as shown in Figure 4.

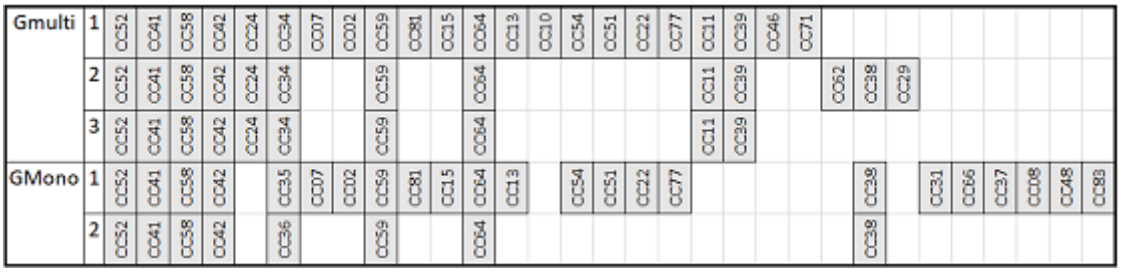


**Figure 3. Highlights for the cost centers verified in the analysis steps out of a total of 83.**

It is important to give attention to the fact that the concentration of event types EV1, EV2, and EV3 in certain cost centers reveals much of the strategy of the organization "X" in the software process development. We also found significant Pearson's correlation values among the betweenness centrality metrics of the GMono graph (EV3 events) and

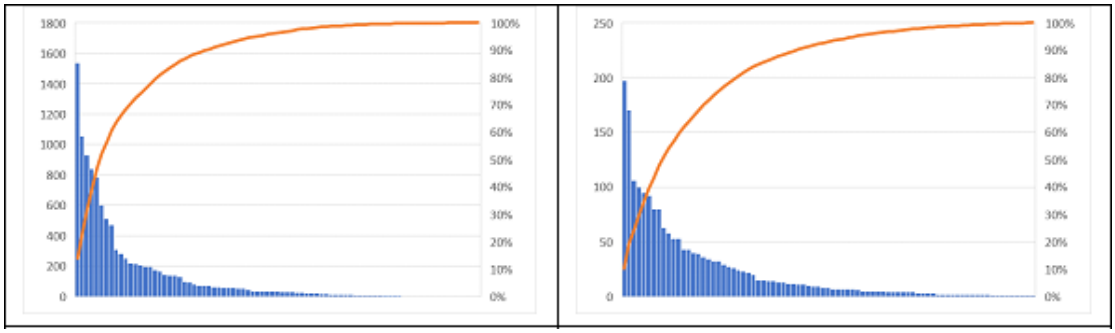


**Figure 4. GMulti graph degree centrality (in-degree + out-degree) on the left side and GMono graph degree centrality (in-degree) on the right side.**

the out-degree centrality metrics of the GMulti graph (EV1 events), with a value of 0.859. It is suggested that cost centers, despite dealing with different product families, share common software assets, which demands orchestration management at deployment.

Although the persistence of the cost centers discussed above are both in the group that requests the most changes and in those that cause the most incidents, we verified that the general distribution of in-degrees and out-degrees among the cost centers did not seem to follow similar behavior patterns, as initially expected. We ratify this finding using Pearson's correlation, where we obtain a value of 0.489 (weak correlation) for event types EV1 and EV2. The same calculation was performed for the cost centers of each of the six sharing communities. In this context, we found different realities, with values ranging from 0.590 to 0.967 (moderate to strong).

Such variation implies at least two hypotheses regarding the software development teams in organization "X": i) they present different behaviors, even being influenced by the same group of systemic factors; and ii) due to the influence of different groups of systemic factors, they reveal different behaviors. In the first case, we can assume a differentiation for reasons inherent to the teams, with systemic factors regulated through a horizontally driven governance in relation to the different families of products. In the second case, we can expect differentiation due to the total or partial absence of regulatory mechanisms through less imposing governance and, sometimes, only in an advisory scope.

The organization "X" can be classified in the latter case due to the presence of PSECO characteristics whose systemic interference factors are not properly regulated, such as: i) predominance of tacit knowledge over explicit knowledge; ii) software development teams with different seniority profiles; iii) absence of guidelines for the software development process, allowing IT managers autonomy to adopt (or not) specific practices; iv) absence of an integrated, automated, and dedicated environment for regression testing to be used by developers during the deployment process; and v) uneven attribution of goals and performance indicators among development teams.

The culture of knowledge management may not be noticeable in organization "X". The resistance to sharing knowledge allows for the false notion that information belongs to a single owner. The organization becomes hostage to people and software consultants, as knowledge transfer can mean loss of influence, reputation, respect, and job security.

Such a level of dependency prevents proper management of processes and quality indicators. To address this situation, organization "X" could implement new governance strategies, such as investing in learning processes that allow for the continuous improvement of its intellectual capital through techniques such as coaching, mentoring, and training.

The autonomy of IT managers who act without clear governance guidelines may lead to inconsistencies and errors in estimates, resulting in inadequate budgets, unsuccessful projects, and poor-quality software assets. To mitigate such risks, a governance sector should take into account the best practices in the software industry, such as the use of agile methodologies, cost-benefit analysis, and treatment of risks and uncertainties. As for the maturity of the software development process at organization "X", the products should undergo continuous validation to reduce the number of defects.

Finally, the success of the PSECO technology platform requires motivators across the network of actors (e.g., developers, IT managers, and businesses) to ensure a common commitment to the final quality of product. In organization "X", an inadequate performance indicator is associated with the IT project manager, who is only evaluated by the agreed deadlines. This type of problem causes unexpected workloads, which have centralized sustaining teams whose key indicator is the incident resolution time.

## 7. Threats to Validity

The validity of the study is closely related to the reliability of the results. Every study involves risks that should be handled and taken into account alongside the findings, according to the classification described in [Runeson et al. 2012]. Regarding the *internal validity*, our concern is that the cause-and-effect relationship established in the work cannot be explained by other factors. To mitigate the risk that the results might be affected by a biased researcher, when there was doubt about the protocol, the matter was discussed with other researchers involved in this work and a common understanding was reached.

Regarding the *external validity*, which refers to how to generalize the findings to apply to other settings from a case-specific perspective, the study involved only one organization with a particular context and culture. Therefore, it is not possible to generalize the results from these cases to organizations not similar to a BFSI (Banking, Financial Services, and Insurance industry) organization.

## 8. Conclusion and Future Work

We report a case study that looked into the dynamics of development teams from the perspective of changes and incident tickets in the PSECO of a large international insurance company. We develop a "bot" using RPA techniques, perform data mining to explore the data set results, and map the network formed by the connections established between change data, incidents, and software development cost centers using SNA concepts.

This study's results reveal a series of behaviors of development teams that serve as an alert for the organization's IT board because it highlights the importance of adapting to the most modern and flexible software development processes to remain competitive in the market. The survival mode must be on. In order to allow an optimal and stable behavior of the development teams and ensure the quality of the final product, the governance guidelines must not treat the specificities of each of the development teams in a homoge-

neous way. Some reasons why this alert is important include: i) market changes - the technology market is evolving constantly and the organization should adapt to these changes; ii) deployment speed - organizations that can launch new release software products and features faster have a competitive advantage; iii) customer satisfaction - customers expect high quality products that meet their needs and desires; and iv) competition - the intensity of the technology market means that organizations need to stand out to succeed.

Relating to the contributions to the software industry practitioners, IT managers try to implement more than they can to meet the market pressure for state-of-the-art solutions. Thus, new software releases bring new production defects, poor quality deliveries, project delays, stress, and burnout. Considering the new strategies to implement governance mechanisms related to PSECO, our study takes the position that a lack of control in the development environment, in which software changes, testing, and other processes are not effectively orchestrated and will provide risk into the organization. For the academic community, wasted effort and increased downtime are some outcomes of failed software deployments, often as a result of poorly documented release processes. The actions to improve the governance strategies depend on three pillars: people, process, and technology. There is no point in privileging investments in just one of these pillars.

As future work, we intend to conduct a qualitative field study through semi-structured interviews with IT managers of the organization "X", responsible for six different sharing communities and the eight cost centers representing 65% from all incidents. The goal is to confirm the behavior of the development teams using the governance guidelines established in the PSECO software development process.

## Acknowledgments

This study was partially funded by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - Finance Code 001, FAPERJ (Proc. 211.583/2019 & Proc. 210.688/2019) and UNIRIO (DPq/PPQ 2022).

## References

Addy, R. (2007). *Effective IT service management: to ITIL and beyond!* Springer-Verlag.

Alves, C., Oliveira, J., and Jansen, S. (2017). Understanding governance mechanisms and health in software ecosystems: A systematic literature review. In *International Conference on Enterprise Information Systems*, pages 517–542. Springer.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: theory and experiment*, 2008(10):P10008.

Bosch, J. (2009). From software product lines to software ecosystems. In *SPLC*, volume 9, pages 111–119.

Brown, C. V. (2003). The it organization of the future. *Competing in the Information Age: Align in the Sand (Ed, Luftman, JN). Oxford University Press, New York, NY*, pages 191–207.

Costa, L. A., Fontão, A., and Santos, R. (2021). Toward proprietary software ecosystem governance strategies based on health metrics. *IEEE Transactions on Engineering Management*, 69(6):3589–3603.

Dahesh, M. B., Tabarsa, G., Zandieh, M., and Hamidizadeh, M. (2020). Reviewing the intellectual structure and evolution of the innovation systems approach: A social network analysis. *Technology in Society*, 63:101399.

Davis, J. and Daniels, R. (2016). *Effective DevOps: building a culture of collaboration, affinity, and tooling at scale*. "O'Reilly Media, Inc.".

Dhungana, D., Groher, I., Schludermann, E., and Biffl, S. (2010). Software ecosystems vs. natural ecosystems: learning from the ingenious mind of nature. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, pages 96–102.

Grieves, M. and Vickers, J. (2017). Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. *Transdisciplinary Perspectives on Complex Systems: New findings and approaches*, pages 85–113.

Gross, J. L., Yellen, J., and Anderson, M. (2018). *Graph theory and its applications*. Chapman and Hall/CRC.

Jafari, P., Mohamed, E., Lee, S., and Abourizk, S. (2020). Social network analysis of change management processes for communication assessment. *Automation in Construction*, 118:103292.

Jansen, S., Brinkkemper, S., and Finkelstein, A. (2009). Business network management as a survival strategy: A tale of two software ecosystems. *IWSECO@ ICSR*, 2009.

Kappelman, L. A., McKeeman, R., and Zhang, L. (2006). Early warning signs of it project failure: The dominant dozen. *Information Systems Management*, 23(4):31–36.

Lahtela, A. and Jäntti, M. (2011). Challenges and problems in release management process: A case study. In *2011 IEEE 2nd International Conference on Software Engineering and Service Science*, pages 10–13. IEEE.

Lee, S. J. and Siau, K. (2001). A review of data mining techniques. *Industrial Management & Data Systems*.

Manikas, K. and Hansen, K. M. (2013). Software ecosystems–a systematic literature review. *Journal of Systems and Software*, 86(5):1294–1306.

Paszkiewicz, Z. and Picard, W. (2013). Analysis of the Volvo IT incident and problem handling processes using process mining and social network analysis. In *BPIC@ BPM*.

Runeson, P., Host, M., Rainer, A., and Regnell, B. (2012). *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons.

Santos, R. P. and Werner, C. M. L. (2011). A proposal for software ecosystems engineering. In *IWSECO@ ICSOB*, pages 40–51.

Tranfield, D., Denyer, D., and Smart, P. (2003). Towards a methodology for developing evidence-informed management knowledge by means of systematic review. *British Journal of Management*, 14(3):207–222.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in Software Engineering*. Springer Science & Business Media.

Yin, R. K. (2005). *Introducing the world of education: A case study reader*. Sage.