

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Process discovery in project management: process and sequence mining approaches

José Alejandro Briones Romero



Mestrado em Engenharia de Software

Supervisor: João Mendes Moreira

Second Supervisor: Ricardo Teixeira Sousa

June 30, 2021

Process discovery in project management: process and sequence mining approaches

José Alejandro Briones Romero

Mestrado em Engenharia de Software

Abstract

Process mining and data mining practices have discovered knowledge that allows companies to manage resources efficiently. The current thesis work relies on process mining and sequential pattern mining implementations from event logs obtained of Strongstep certification projects. The objective is to extract, update and recommend a standard process for nine project phases of CMMI, ISO 27001, and GDPR projects. After collecting the data, a data understanding and data preparation phase occur to execute cleaning tasks and guarantee model robustness. The process discovery implementation of process mining uses the Inductive Miner infrequent (IMi) of PM4Py library to generate the correspondent model through Petri nets and Directly-Follow Graphs. Then, all the process models are assessed with four quality metrics: fitness, precision, simplicity, and generalization. As every phase process is understood as a set of activities without parallel occurrences, a sequential pattern mining approach is executed to support the results of the first technique. This implementation uses the CM-SPAM algorithm of the SPMF tool to get the most frequent pattern for each phase. The generated outcome defined six standard phase processes that Strongstep can use for new project instances. The data was highly variable for the other three phases, which led to the generation of inconsistent models and the lack of frequent pattern recommendations. Both process and sequence mining techniques showed their effectiveness in real scenarios. Nevertheless, there are deployment and optimization tasks that can be realized in future work.

Keywords: Process discovery, process mining, sequential pattern mining.

Resumo

Process mining e data mining são metodologias que permitem ajudar as empresas a gerir os recursos de forma eficiente. O trabalho desta tese baseia-se em implementações de process mining e sequential pattern mining a partir de registos de eventos obtidos de projetos de certificação da Strongstep. O objetivo é extrair, atualizar e recomendar um processo padrão para nove fases de projetos de CMMI, ISO 27001, e RGPD. Após a recolha dos dados, ocorre uma fase de compreensão e preparação dos dados para executar tarefas de limpeza e garantir a robustez do modelo. A implementação de process discovery do process mining utiliza o Inductive Miner infrequent (IMi) da biblioteca PM4Py para obter o modelo correspondente representado em Petri nets e Directly-Follow Graphs. Posteriormente, todos os modelos do processo são avaliados com quatro métricas: *fitness*, precisão, simplicidade e generalização. Como cada processo de fase é entendido como um conjunto de atividades sem ocorrências paralelas, é executada uma abordagem de sequential pattern mining para apoiar os resultados da primeira técnica. Esta implementação utiliza o algoritmo CM-SPAM da ferramenta SPMF para obter o padrão mais frequente para cada fase. O resultado gerado definiu seis processos padrão que Strongstep pode utilizar para novas instâncias do projeto. Os dados foram altamente variáveis para as outras três fases, o que levou à geração de modelos inconsistentes e à falta de recomendações de padrões frequentes. Tanto as técnicas de process mining como as de sequential pattern mining mostraram a sua eficácia em cenários reais. No entanto, há tarefas de implantação e otimização que podem ser realizadas em trabalhos futuros.

Keywords: Process discovery, process mining, sequential pattern mining.

Acknowledgements

I would like to begin by thanking Professor João Moreira and Ricardo Sousa for all their help along with this dissertation project. Ricardo was always aware of the project, helping me with any issue, looking for solutions that could add value to the project, and promoting interest in new knowledge. Professor João had always precise recommendations due to his vast knowledge in the area, solving any concerns and being the connection between Strongstep, INESC, and FEUP.

I also want to thank all PROMESSA's project team for their continuous support throughout the project and particularly to Duarte Gomes for all his availability regarding Strongstep's processes doubts.

A thank you as well to all my professors and classmates of MESW for making this experience a very valuable time here in Portugal.

Thanks to my friends, to all those I met in Porto, for helping me coping these pandemic moments, and to those who are still present from Ecuador despite the distance. Thanks to Anita for all her love and support through this Master's program. Gracias, gracias, gracias.

Finally, I want to thank my family for being my engine, and my mother for being the sponsor of all my adventures as well as my life teacher, you are the best.

José Briones Romero

“E Scientia Hominis Salus”
Science to the people.
Runakuna yachay.

Contents

1	Introduction	1
1.1	Context and Motivation	1
1.2	Problem Definition	2
1.3	Methodology	3
1.4	Dissertation Structure	3
2	State of The Art	5
2.1	Introduction	5
2.2	Software Certification Projects	5
2.2.1	CMMI	6
2.2.2	GDPR	6
2.2.3	ISO 27001	7
2.3	Process Mining	8
2.3.1	Process Discovery	9
2.3.2	Conformance Checking	10
2.3.3	Process Enhancement	11
2.3.4	Process Mining Evaluation	12
2.4	Sequential Pattern Mining	13
2.4.1	Definitions	13
2.4.2	Algorithms	14
2.5	Use Case: Process Mining in Lasagna Process	15
2.5.1	Methodology	16
2.5.2	Results	18
3	Methodology	19
3.1	Introduction	19
3.2	Business Understanding	20
3.2.1	Business Objective	20
3.2.2	Mining Goals	20
3.3	Data Understanding	21
3.3.1	Data Collection	22
3.3.2	Data Description	22
3.3.3	Data Exploration	24
3.3.4	Data Quality Verification	25
3.4	Data Preparation	25
3.4.1	Data Selection	25
3.4.2	Data Cleaning	26
3.5	Process Mining Modeling	27

3.5.1	Modeling Preparation	27
3.5.2	Algorithms Selection	28
3.5.3	Model Visualization	29
3.6	Sequential Pattern Mining Modeling	32
3.6.1	Modeling Preparation	33
3.6.2	Algorithms Selection	34
4	Discussion of Results	37
4.1	Introduction	37
4.2	Results of Project Phases	37
4.2.1	CMMI & ISO 27001 Projects	38
4.2.2	GDPR Projects	44
4.3	Analysis of Quality Metrics	45
4.3.1	Fitness	45
4.3.2	Precision	46
4.3.3	Simplicity	47
4.3.4	Generalization	47
4.4	Overall Evaluation	48
5	Conclusions and Future Work	51
5.1	Conclusions	51
5.2	Future Work	52
A	Inconsistent Directly-Follows Graphs	53
	References	55

List of Figures

1.1	Case distribution of process mining projects per year.[49].	2
2.1	CMMI maturity levels.	6
2.2	GDPR features.	7
2.3	ISO 27001 benefits.	7
2.4	Process mining workflow.	8
2.5	Example of a Petri net [19].	9
2.6	Process discovery approach proposed in [18].	10
2.7	Conformance checking process [54].	11
2.8	Process mining evaluation [16].	13
2.9	L* life-cycle model [54].	16
3.1	Experimentation methodology.	20
3.2	GDPR phases represented in SCRAIM.	21
3.3	Data frame entries and columns.	23
3.4	Variables type of the data frame.	23
3.5	Data frame visualization.	23
3.6	Statistical analysis of data frame.	24
3.7	Number of entries for project name.	24
3.8	Number of entries for project type.	25
3.9	Data selection for Kickoff phase.	26
3.10	Data selection for the Initial Gap Analysis phase.	26
3.11	Example of data before cleaning.	27
3.12	Example of data after cleaning.	27
3.13	Example of XES file for the Initial Gap Analysis phase.	28
3.14	Petri net model for Preparation and Planning phase.	30
3.15	Directly-Follows Graph model for Preparation and Planning phase.	31
3.16	SPMF graphical interface.	32
3.17	SPMF format in the input text file.	33
3.18	SPMF result with graphical interface for the Preparation and Planning phase. . .	34
3.19	SPMF result with Python wrapper for the Preparation and Planning phase. . . .	35
4.1	Directly-Follows Graph model of Kickoff phase.	38
4.2	Directly-Follows Graph model of Initial Gap Analysis phase.	40
4.3	Directly-Follows Graph model of Intermediate Gap Analysis phase.	42
4.4	Directly-Follows Graph model of Final Gap Analysis phase.	43
4.5	Directly-Follows Graph model of Project Closure phase.	43
4.6	Directly-Follows Graph model of Preparation and Planning phase.	44
4.7	Fitness for all process models.	46

4.8	Precision for all process models.	46
4.9	Simplicity for all process models.	47
4.10	Generalization for all process models.	48
A.1	Directly-Follows Graph model of Workshops phase.	53
A.2	Directly-Follows Graph model of Process Monitoring phase.	54
A.3	Directly-Follows Graph model of Execution and Implementation phase.	54

List of Tables

2.1	Example of a sequence database	14
3.1	Comparison of process discovery miners implemented in PM4Py	28
4.1	Petri nets properties and evaluations	48
4.2	Metrics of recommended sequential patterns	49

Abbreviations

AM	Alpha Miner
BAM	Business Activity Monitoring
BI	Business Intelligence
BPM	Business Process Management
BPMN	Business Process Model and Notation
CM-SPAM	Co-occurrence Map Sequential Pattern Mining
CMMI	Capability Maturity Model Integration
CMMN	Case Management Model and Notation
CRISP-DM	Cross-Industry Standard Process for Data Mining
CRM	Customer Relationship Management
CSV	Comma-Separated Value
DFG	Directly-Follows Graphs
EEA	European Economic Area
ERDF	European Regional Development Fund
ERP	Enterprise Resource Planning
ESI	European Software Institute
EU	European Union
FM	Fuzzy Miner
GDPR	General Data Protection Regulation
GM	Genetic Miner
HM	Heuristics Miner
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IM	Inductive Miner
IMi	Inductive Miner infrequent
IMd	Inductive Miner directly-follows
INESC TEC	Institute for Systems and Computer Engineering, Technology and Science
ISO	International Organization for Standardization
LIAAD	Laboratório de Inteligência Artificial e Apoio à Decisão
PM4Py	Process Mining for Python
ProM	Process Mining framework
PROMESSA	PRoject ManagEment intelligent aSSistAnt
SaaS	Software as a Service
SEI	Software Engineering Institute
SMEs	Small and medium-sized enterprises
SPMF	Sequential Pattern Mining Framework
SQL	Structured Query Language
TS	Transition Systems
UML	Unified Modeling Language
XES	eXtensible Event Stream

Chapter 1

Introduction

1.1 Context and Motivation

Although project management practices are used in projects planning, inefficiencies still appear within their processes. Some work has been done to deal with these issues, most of it based on data mining and business intelligence techniques. Furthermore, process optimization operations help to manage resources efficiently by finding workflow inefficiencies such as bottlenecks, longtime executions, and deadlocks.

Usually, flaws within organization processes are not visible, and concerns about data integrity and data quality appear. Some processes are generated through empirical experimentation and draw subjective conclusions, which leads to missing some activities that should be part of the pathway. For this reason, it is necessary to perform a process update operation to obtain the complete workflow. All of these misunderstandings would cause a decrease in collaboration among the team, leading to inefficient workflows that would be a peril to the achievement of the process's goal[1].

To prevent these issues, the company would check and improve operational processes. If the organization takes time to continuously assess internal processes, efficiency and overall performance would increase. This can be reflected in the improvement of communication within the organization, better management of data, optimized processes that would increase the compliance of the company's goals, and a decrease in manual work through automation.

This project was born as a joint effort between Strongstep company, LIAAD department of INESC TEC, and FEUP. The LIAAD (Laboratório de Inteligência Artificial e Apoio à Decisão) center is responsible for managing different research and dissertation projects that are part of the PROMESSA (PRoject ManagEment intelligent aSSistAnt) consortium. This consortium is co-funded by the European Regional Development Fund (ERDF) of the European Union (EU) through NORTE 2020 (Programa Operacional Regional do Norte) in Portugal. The organizations responsible for PROMESSA are Strongstep, the University of Porto, the Fraunhofer Association, and INESC TEC.

PROMESSA projects intend to use artificial intelligence and data mining techniques to deal with inefficiencies in project management. The main goal is to develop solutions analyzing all the information available and providing recommendations to project managers. Thus, they can make better decisions in future project instances.

1.2 Problem Definition

Based on the importance of improving processes in organizations, the current project intends to solve process inefficiencies using process mining and sequential pattern mining techniques to extract, update and recommend enhanced Strongstep processes within their certification projects.

This proposal leads to the following research question: What benefits can provide process mining and sequential pattern mining techniques to the extraction, update, and recommendation of certification processes, and how can this be measured?

Process mining is a relatively novel family of techniques that are increasingly used as part of Business Intelligence solutions[49]. Process mining techniques include process discovery, conformance checking, and process enhancement, where data science and process management helps to get insights and take actions based on event data. Event data needs to have a timestamp property for every entry, this timestamp property will allow ordering the process described. Company software like project management tools would provide event data in form of logs, this information will be the input of a process discovery operation where the output will be a process model.

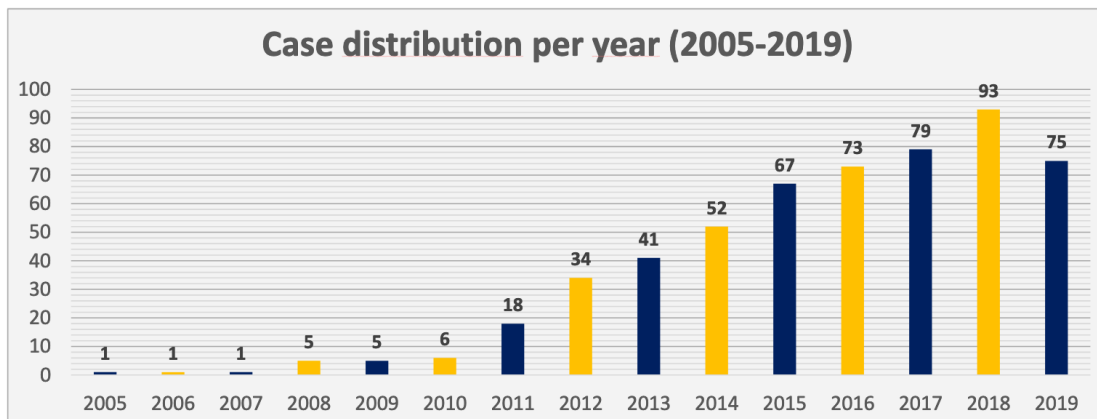


Figure 1.1: Case distribution of process mining projects per year.[49].

Sequential pattern mining is a subject of data mining that finds relevant patterns among different sequences. As the processes of Strongstep projects are not intended to work in a parallel way, this pattern mining approach is also considered in this project. Its objective is to find the most frequent processes in all project phases. Sequential pattern mining is an older research field that has been developed several algorithms to optimize resources and work with big data. Hence, it is an important field with some relevant contributions.

Even there is some research for process mining and a bit more for sequential pattern mining, there are no surveys where both approaches work on IT certification projects. In this case, projects are part of audit processes for companies that develop software and IT solutions.

1.3 Methodology

All data came from General Data Protection Regulation (GDPR), Capability Maturity Model Integration (CMMI), and ISO 27001 projects developed by Strongstep. Strongstep was established in 2009 and is a software engineering consulting company focused on process improvement, certification, auditing, training, and business strategies with national and international reach. It operates in a wide range of industries, such as healthcare, retail, telecommunications, energy, IT, education, and public administration. Some partners are CMMI Institute, Software Engineering Institute (SEI), and European Software Institute (ESI).

The event data came from SCRAIM. SCRAIM is a SaaS solution developed by Strongstep, based on project and process management methodologies that standardize best practices within organizations. It is focused on small and medium software companies (SMEs) that lack efficient project management practices or do not have their software development processes well-defined. SCRAIM provides a better cost-benefit experience among traditional project management tools.

Consequently, there is an impulse to pursue an outcome that will benefit real stakeholders. The results are focused not on projects but project phases. For each project phase, process mining techniques generate a process model, and in the same way, a frequent pattern is obtained using sequential pattern mining algorithms. Each of the CMMI, ISO 27001, and GDPR projects has several phases, and for every phase, the processes are not well-known. Therefore, these processes are identified as semi-structured. This means that exists a description of the activities executed but real processes do not meet completely with their respective specification.

The workflow followed a methodology based on CRISP-DM with the following phases: business understanding, data understanding, data preparation, modeling, and evaluation. Jupyter Lab Framework and Python programming language were the technologies used. Some Python libraries were used: PM4Py for process mining modeling, pandas for data management and preprocessing, and a Python wrapper of the SPMF (Sequential Pattern Mining Framework) tool for pattern mining. All the information about the experimentation phase is explained in Chapter 3.

1.4 Dissertation Structure

This dissertation work is organized into five chapters. Chapter 2 presents the State of The Art of the used techniques and some important concepts. Chapter 3 includes the experimentation work and its methodology. The discussion of the results is described in Chapter 4. Finally, Chapter 5 sums up the conclusions of the thesis project and suggests future work to be done afterward.

Chapter 2

State of The Art

2.1 Introduction

Most process mining challenges depend on workflow event logs. An event log consists of groups of activities with timestamps and case identification. Both process mining and data mining operations rely on a statistical foundation, which allows to compare several approaches between these two fields and find a complete solution to the current case.

To fetch valuable information some keywords were used to search for conference papers and articles from databases such as Engineering Village, IEEE Xplore, and Scopus. Some of the keywords used were: "process discovery", "process mining", "sequential pattern mining" and "process enhancement". The results showed that the approach of the present work is a novel contribution and that process mining is a relatively new field with plenty of scope for research. A second search was focused on sequential pattern mining solutions related to process mining techniques. It was not found results comparing both techniques or using them in the same survey.

2.2 Software Certification Projects

Project design is a significant step in realizing successful projects. For this matter, a project plan is created and focuses on the objectives defined between the stakeholders and the organization or company. This planning activity gathers information about the people responsible at every stage of the project, the time for reaching the outcomes, and the features that will be developed, among others. It also helps to estimate the budget and find risk areas to monitor and correct workflows. Project managers stress the importance of good planning to avoid pitfalls and ease the maintenance of the project, bearing in mind all the time and scope aspects so that if any risk appears it can be attended at once.

As mentioned in Chapter 1 all processes were part of three IT certification projects: CMMI, GDPR, and ISO 27001.

2.2.1 CMMI

Capability Maturity Model Integration is a standard model that assesses and improves processes in a software company. The last version of CMMI is version 2.0 and was published in 2018. The CMMI process areas include Process management, project management, engineering, and support. CMMI maturity levels are Initial, Managed, Defined, Quantitatively Managed, and Optimizing.

The Initial level represents an unstable environment where the processes are abandoned. The second level, Managed, gets requirements, processes, and products managed according to a generic goal. On the next level, Defined, the management of goals is not only generic but also specific of every process area. On level four, Quantitatively Managed, the contribution comparing with previous levels rely on subprocesses. Finally, the Optimizing level focuses on continuous improving actions [32].

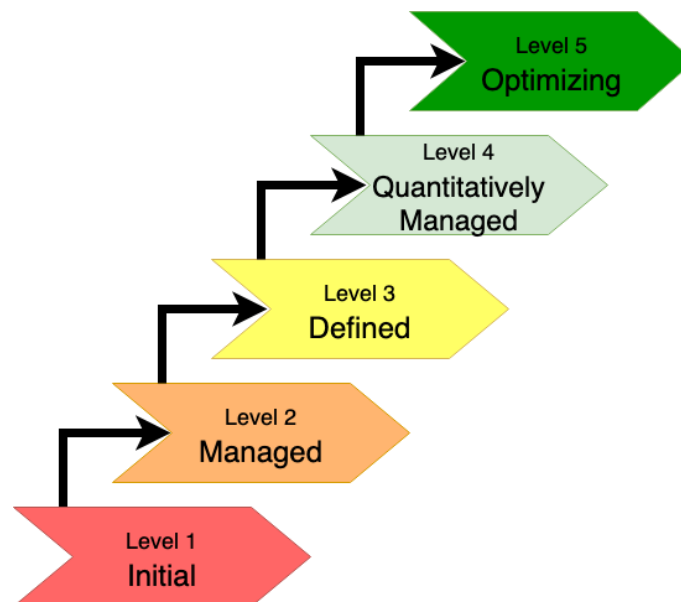


Figure 2.1: CMMI maturity levels.

2.2.2 GDPR

General Data Protection Regulation (GDPR) is a data protection and privacy regulation that has been implemented since 2018 in Europe, but its impact has transcended continental borders. This regulation specifies rules for managing, storing, and processing data from users within the European Union (EU) and European Economic Area (EEA).

This regulation gives consumers the right to control their information used by organizations including the withdrawing of consent and the elimination of their data. Data controllers and processors are part of this schema that focuses on user's privacy along with all the activities within a process activity[37].

Consulting companies are helping to implement GDPR policies in software organizations, event logs related to these consulting processes are included in the present work.

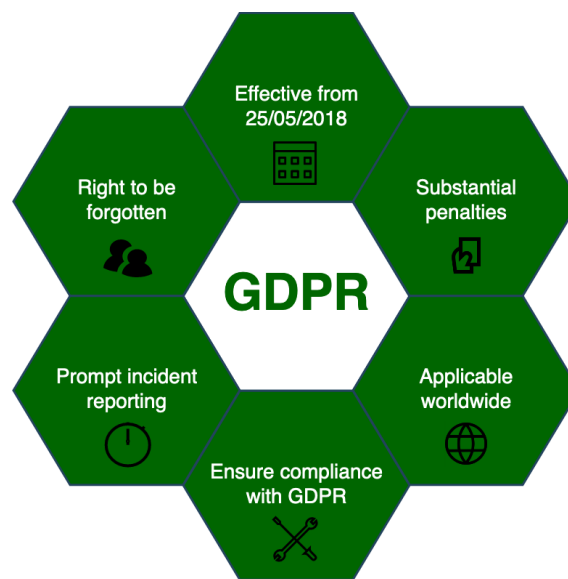


Figure 2.2: GDPR features.

2.2.3 ISO 27001

ISO/IEC 27001, better known as ISO 27001, is an international standard that guarantees the proper management of information security. ISO is the International Organization for Standardization, and IEC is the International Electrotechnical Commission. Both organizations are worldwide standard associations.

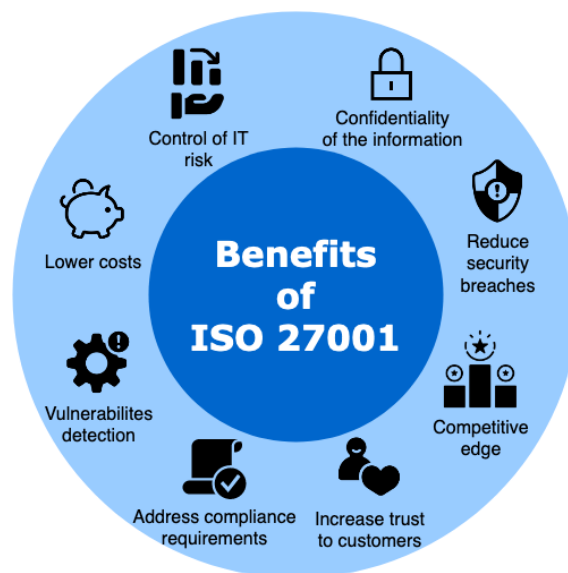


Figure 2.3: ISO 27001 benefits.

By implementing ISO 27001 certification, organizations not only meet client's security requirements following a set of recognized best practices but also increase brand value in the market.

[44]. An ISO 27001 process can be summed up with the following steps: Find out the objective of the organization, get organization support, characterize scope, define risk assessment methodology, conduct risk treatment plan, implement security program, documentation of adopted controls, reviewing of compliance and internal audit[44]. Companies should be prepared for ongoing evaluations as part of the three-stage external audit: preliminary, formal, and periodic evaluation. Consulting companies help on these processes developing projects related to this specification as it will be shown in Chapter 3.

2.3 Process Mining

Process mining consists of a group of techniques that are intended to extract information from event logs to model a process. The challenge is to create a process model given historical data such as information logs composed by events, the created model should be consistent with the real process. Once this model is created, it can also be improved by applying enhancement operations. Process mining is considered part of the field of Business Process Management (BPM) but is also related to Business Activity Monitoring (BAM) and Business Intelligence (BI). Cook and Wolf found that process mining is related to data mining and reverse engineering, discovering knowledge analyzing data is also a core activity in process mining. Thiede, Fuerstenau, and Bezerra emphasize that data mining techniques are used for “the extraction of knowledge from data produced by a set of systems and processes”[52]. Regarding reverse engineering, the main goal is to redesign a system from its implementation, starting from a low-level specification like the execution of the process and getting a high-level specification such as a process design. Nevertheless, process mining gets more out of the data mining field than reverse engineering.

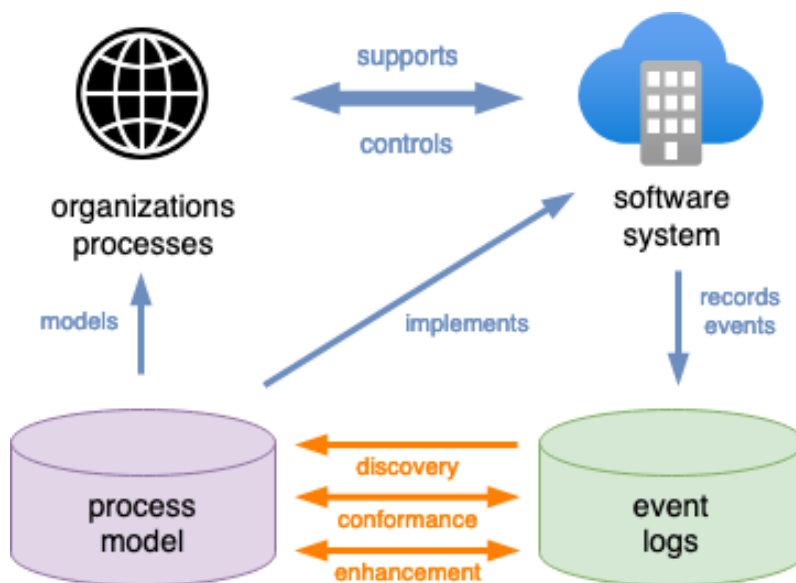


Figure 2.4: Process mining workflow.

The first step in a process mining operation is to gather information about the process involved. This information should be the outcome of events recorded in the form of logs, this set of events should be ordered for every case. The order of the events will be set by a timestamp that is unique for every event. It is quite common to find this kind of information in business process management systems such as Enterprise Resource Planning (ERP) or Customer Relationship Management (CRM) systems and even in proprietary process management software that exports data as an event log.

In most cases the process models are represented as Petri nets [40]. Van Der Aalst in [3] explains the use of Petri nets in workflow management. A classical Petri net is a type of graph that has two types of nodes called places, represented as circles, and transitions represented as rectangles. These nodes are connected with arcs where every arc associates a place with a transition but never a node of the same type as it can be seen in Figure 2.5. This figure presents a process model that can start from two different tasks, task A and task E. After executing task A, tasks B and C are executed in parallel, this will add more complexity to the specification of the process.

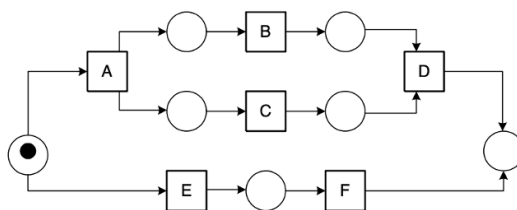


Figure 2.5: Example of a Petri net [19].

This example considers a process log with few tasks, but a larger process would present a lot more cases where not all the possible paths could be detected in real-time executions. Hence, the model would help to find these steps that could not be considered. Another factor that can make difficult the process model elaboration is the outbreak of errors in the log, also known as noisy data.

Noise is not the only challenge that process mining would face, Van der Aalst and Weijters in [19] described hidden tasks, duplicate tasks, loops, incompleteness, among others. It is important to deal with these concerns to find the simplest process model considering generated process events.

2.3.1 Process Discovery

Process mining concepts were first related as a matter of a process discovery operation [18]. Cook and Wolf called process discovery to the activity of capture process events of real executions to generate a model that describes the behavior of that process.

In [18] they present three methods: the first one, purely statistical (neural networks), another one, purely algorithmic, and the third one, a hybrid of statistical and algorithmic approach, based on Markov models that help to deal with noise.

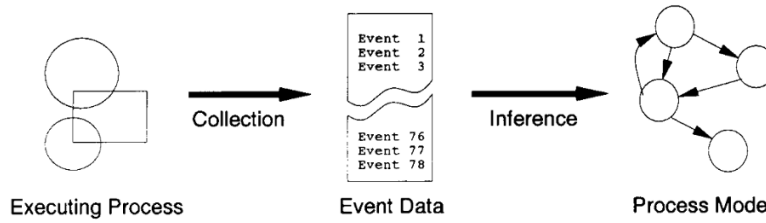


Figure 2.6: Process discovery approach proposed in [18].

In [46], Rivas and Bayona-Oré present a comparison of multiple algorithms for process discovery. They conclude that noise still is a challenge that needs special attention to increase the use of process mining. Noise is described as the result of out-of-order events, strange behaviors, or recording errors according to [51].

Another challenge in process discovery is the considerable amount of time that, in some cases, takes to generate the model. As algorithms would use all the event data, the standard hardware would take a lot of time to apply data mining techniques. In [47], Sani, Van Zelst, and Van der Aalst proposed an approach to overcome performance problems based on just working with selected process instances. The strategies used for selecting specific event data, instead of using random data, consider some process features such as frequency, length, similarity, and structure. The results showed that this particular selection speeds up process discovery.

2.3.2 Conformance Checking

Conformance checking is the second process mining technique that needs to be implemented to guarantee that a process model responds to the actual behavior of process instances. As this technique can be considered a process model evaluation some organizations could be interested as they need to perform audits to ensure process compliance.

Dunzer, Stierle, Matzner, and Baier reviewed some conformance checking techniques described in 37 research publications [21]. The techniques are classified by modeling language, perspective, algorithm type, and quality metric. This survey allows us to have a sight of which techniques are commonly used and which have some issues yet to solve.

Modeling languages are classified into two types by researchers. Declarative and procedural. Procedural modeling languages are more popular such as Unified Modeling Language (UML), Business Process Model and Notation (BPMN) and Petri nets [45]. Some declarative modeling languages are Declare, Case Management Model and Notation (CMMN), and Dynamic Condition Response[48].

The modeling languages identified in the survey [21] were Petri nets, BPMN and Declare. Petri nets were highly used, 28 of 37 publications used that model technique. Regarding the perspective dimension, all the publications had a control-flow perspective and just a few consider multi-perspective approaches like time or roles concerns. Something similar happens with the

algorithm type used, most of the cases use a trace alignment approach while a few use a log replay technique.

With respect to quality metrics, Replay fitness was the most used technique to evaluate process execution. Simplicity, precision, and generalization were also used but in a smaller amount.

An important future work that Dunzer et. al pointed out is a multi-perspective approach that includes measures like time or roles in the activities, this could bring interesting outcomes of the analysis of complex business processes.

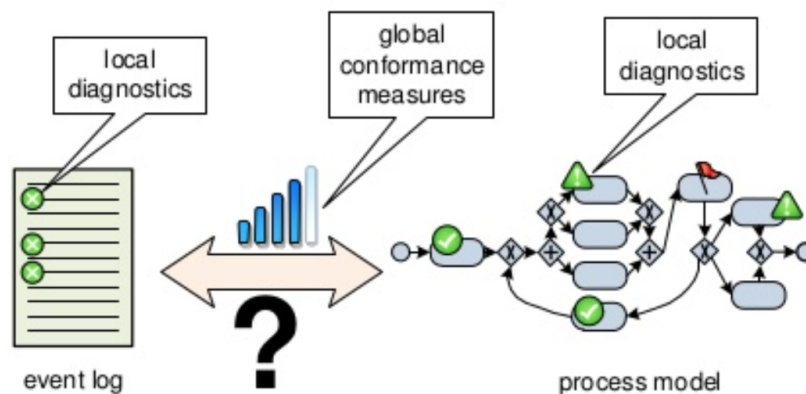


Figure 2.7: Conformance checking process [54].

2.3.3 Process Enhancement

The third technique of process mining is related to improve an existing model, Yasmin, Bukhsh, and Silva in their process enhancement review [59] found that the most cited definition is: “the extension or improvement of an existing process model using information about the actual process recorded in some event log”. This concept was first described by Van der Aalst in [6]. Process extension was another concept found in this survey; this refers to a type of process enhancement related to new perspectives in apriori models.

According to [59], after conformance checking is realized and the process model does not conform to the real process, two operations can be done: repair or extension. These two kinds of process enhancement activities are intended to allow the model to rerun most of the events described in the event log.

Repair techniques consider control-flow while extension techniques are divided into three perspectives: organizational, time, and case perspective. The organizational perspective targets resources hidden in the log, the time perspective focuses on the time and frequency of the events, and the case perspective concentrates on the properties of the cases. Another conclusion that the authors reached out is that the most used applications of process enhancement are related to medical, governmental, and financial cases, in that order.

This survey also pointed out an important variety of frameworks and approaches with different solutions. Process enhancement approaches can also be classified into objective-based and constraints-based. The first one refers to the goals that the approach aims for, this includes repairing the business model, finding bottlenecks, identifying resource community, among others. The constraint-based variant focuses on the boundaries of applicability in a process model.

Finally, Yasmin et al.[59] found that the most used tool in process enhancement practices was ProM, followed by Disco Fluxicon and SQL.

2.3.4 Process Mining Evaluation

There are different quality measures to assess the process models that represent the results of a process mining execution. The most used are: replay fitness, precision, simplicity, and generalization.

2.3.4.1 Replay fitness

Replay fitness refers to the comparison of the behavior in the log and the process model. Some methods for calculating this measure are based on token replay and alignments. In [12], Berti and Van Der Aalst propose a faster and scalable token replay approach that deals with concerns like token flooding, this condition exists when transitions that should not be activated are enabled. Nevertheless, alignments have become the most used technique because of their ability to find an optimal match between the model and the execution of the event log. The disadvantage of the alignments approach is that for complex process models and event logs the performance is not good. For these cases, the token replay method can be an interesting alternative to use.

2.3.4.2 Precision

Precision measure the extent of the behavior of the process model observed in the event log. Different log prefixes are replayed on the process model, this means that a set of transitions enabled in the model is compared with a set of activities of the prefix. The proportion of similarity of both would define the precision. In this case, several methods can be divided between token-based replay and alignments. The difference between these two approaches is that the token-replay method is faster and the alignments method is more accurate[41][8].

2.3.4.3 Simplicity

This quality dimension wants to evaluate how simple is a model for human sight. It follows Occam's razor: "one should not increase, beyond what is necessary, the number of entities required to explain anything". It takes into consideration only the Petri net model. In [14], Rojas uses the inverse arc degree to evaluate simplicity. This type of evaluation takes into account the average degree for a place and transition which is defined as the sum of input arcs and output arcs.

2.3.4.4 Generalization

The Generalization measure deals with the problem of overfitting a model. This occurs when a model has a good precision for a particular case but when new cases are evaluated it starts to give poor results. An approach can be seen in [16], the survey shows that a model is considered general just if its elements are visited enough times during a replay operation using an event log which means that a token-based technique is performed.

The following figure gives a summary of the four quality dimensions for process mining models.

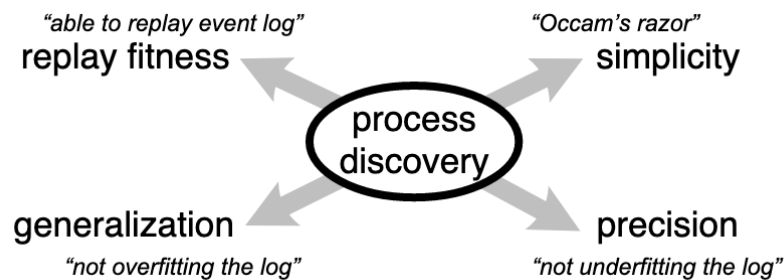


Figure 2.8: Process mining evaluation [16].

2.4 Sequential Pattern Mining

The present work intends to model an optimized process and find the most repeated activities considering the existence of sequential processes, i.e., these processes do not have parallel activities. The main goal is to find frequent patterns for sequential information comparing the similarity between processes and recovering missing sequence data. Process mining is not the only approach that can be used to improve process management, there are also pattern mining algorithms that can fit as well. In the end, both procedures can be compared as well as their results.

Sequential pattern mining is an older technique described in [10] as the following: “Given a database of transactions, mining sequential patterns is to find the maximal sequences among all sequences that have a certain user-specified minimum support. Each such maximal sequence represents a sequential pattern.”

2.4.1 Definitions

Fournier-Viger, Chun-Wei, Uday, and Sing give a straightforward explanation of the basis of sequential pattern mining in [22].

An itemset is defined as a set of items. A sequence can contain items and itemsets. A sequence database SDB is a list of sequences $s_1, s_2, \dots, s_N$, having sequence identifiers (SIDs) $1, 2, \dots, N$. In Table 2.1 the sequence database contains three sequences with the SIDs: 1, 2, and 3. A sequence can be contained in another sequence, when this happens it is said that one sequence is a subsequence of the other sequence.

SID	Sequence
1	$(\{a,b\}, \{c,d\}, \{e\}, \{f,g\})$
2	$(\{a,c\}, \{d\}, \{e\}, \{f,g\})$
3	$(\{a,b\}, \{c\}, \{d\})$

Table 2.1: Example of a sequence database

The support of a subsequence in a sequence database is defined as the number of sequences that contain that subsequence. For instance, the support of the subsequence $(\{e\}, \{f,g\})$ is 2 because it appears in two sequences (SID 1 and SID 2), as it is shown in Table 2.1.

The support is also defined as a ratio or relative support, and it is interpreted as the number of appearances of the subsequence divided by the number of total sequences in the database. In Table 2.1 the relative support of the subsequence $(\{e\}, \{f,g\})$ is 0.66 as it appears in 2/3 of the database.

2.4.2 Algorithms

In [22], Fournier-Viger et al. present a survey of Sequential Pattern Mining algorithms. The study shows a complete description of this type of pattern mining algorithms.

AprioriAll was the first sequential pattern mining algorithm [10]. Then, looking for an improvement of this algorithm GSP was created [50]. These two algorithms are part of the Apriori-based type of algorithms inspired by the Apriori algorithm of frequent itemset mining [9]. Sequential Pattern Mining algorithms can be classified into Apriori-based and pattern-growth.

The Apriori-based algorithms are classified into horizontal database format and vertical database format. The horizontal formatting means that the data is represented with an ID and a sequence of itemsets, the order of the sequence is set by the timestamps of the events. On the other hand, in vertical database format, the rows of the database contain the timestamps associated with events. GSP is an example of an Apriori-based algorithm that uses a horizontal database representation. Some algorithms that use a vertical database representation are SPADE [61] and SPAM [11].

Pattern Growth algorithms are usually complex to develop in comparison with Apriori algorithms, although, these algorithms should perform in a faster way when large volumes of data are used. The most popular is PrefixSpan [31] which inspires in the FPGrowth algorithm of itemset mining [42].

Another classification divides sequential pattern mining algorithms into depth-first search and breadth-first search algorithms. SPADE algorithm improves the GSP approach by using depth-first search. Breadth-first search algorithms, like GSP, are understood as level-wise algorithms since patterns are scanned in ascending length order. Depth-first search starts the search method from the sequences containing single items and then generates larger sequences recursively. Some examples of depth-first algorithms are SPADE, PrefixSpan, and SPAM. More recent research has developed CM-SPAM and CM-SPADE algorithms [23], these new algorithms use co-occurrence pruning to solve the candidate patterns overgeneration of SPAM and SPADE.

Some variations of sequential pattern mining are described below.

- Closed sequential patterns: Set of sequential patterns that represent the largest subsequences that are not included in other sequential patterns with the same support. Some examples are CloSpan [58] and Bide [55], which extend PrefixSpan, and CloFast [28] and CM-Clasp [23] which use co-occurrence pruning.
- Maximal sequential patterns: The smallest set of sequential patterns that are not included in other sequential patterns. However, maximal sequential patterns are not lossless, unlike closed sequential patterns. Some examples are MaxSP [27] that uses pattern-growth, and VMSP [26] that uses vertical algorithms.
- Generator sequential patterns: The smallest set of sequences that characterize a sequence database. Some examples are FEAT [29] and FSGP [60] using pattern-growth approach. Another example is VGEN [25] which is a recent algorithm that extends the CM-SPAM algorithm.

2.5 Use Case: Process Mining in Lasagna Process

Process mining applications are growing every day and have been performed in several fields. Some case studies were conducted at hospitals, as can be seen in [38], and in industrial processes of governmental agencies [53], but in the following review, an application of Process Mining is carried out in the non-alcoholic beverage industry in Chile.

Chamorro and Maturana [17] present an application in the bottling industry knowing that improving processes would lead to increase the quality of service and satisfy customers. The research aims to guide the knowledge areas that are related to process mining. The study follows the L* life-cycle model to develop the process mining method.

Companies are always pursuing process improvement to increase efficiency and reduce resource usages such as cost and time. Process mining can be used for that purpose, can help to transparent processes, and support decision-making. This investigation is focused on the distribution process of the non-alcoholic beverage industry. Distribution processes have short cycles with many daily cases because of the high demand for this kind of beverage.

The L* life-cycle model was created to mine Lasagna processes. Lasagna processes are called like that because, in contrast to Spaghetti processes, they have a relatively clear structure, and the cases are handled in a quite controlled way. The definition given in [54] maintains that a process is a Lasagna process when is possible to create a process model with a fitness of at least 0.8, in other words, 80% of events would occur as initially planned.

The L* life-cycle model described in [54] divides a Process Mining project into five stages, as shown in Figure 4.

Stage 0: Plan and justify

Stage 1: Extract information

Stage 2: Create a control-flow model and connect event log

Stage 3: Create integrated process model

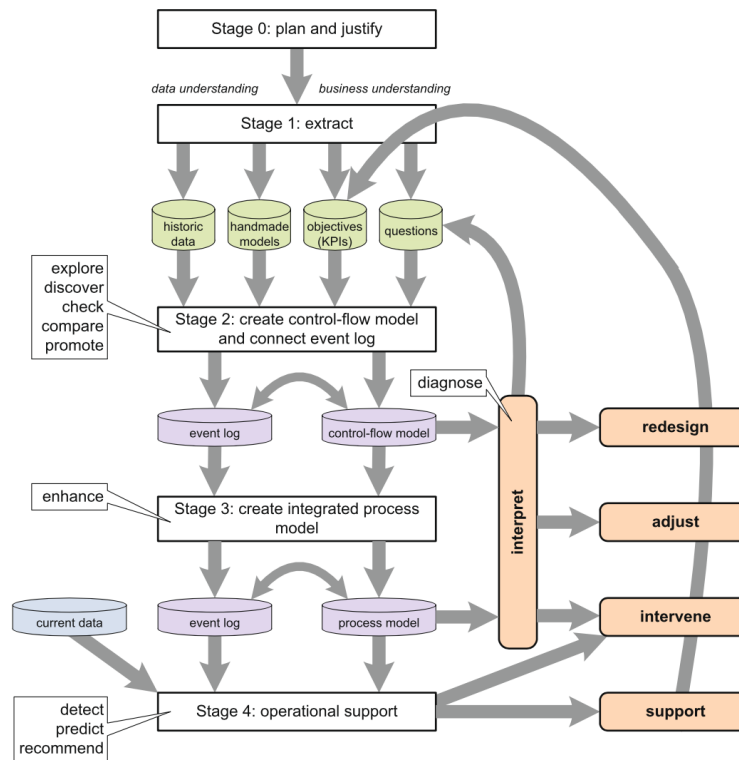


Figure 2.9: L* life-cycle model [54].

Stage 4: Operational support

The tools used for this operation were Disco and ProM and the outcome was a proposal of process redesign that would improve the current process.

2.5.1 Methodology

As some activities of distribution processes require decision-making several deviations are generated. Nevertheless, these activities are registered by software tools and event logs are generated keeping data of good quality. Regarding the stages of the L* life-cycle model, the investigation was performed as described afterward.

Stage 0: Project Planning

Before starting a project, the scope must be defined between the organization and the stakeholders, the problem domain must be understood, and all the goals should be clearly defined. This study was performed in a non-alcoholic beverage company in Chile and the research team found challenges about the quality of the data given by the company. They received inconsistencies and duplicates in data, overwritten records, and historical data deleted.

Stage 1: Data extraction

The log created was defined to have the fields: Unique case id, name of the activity, start timestamp, output timestamp, and name of the executor. As good data was not obtained, synthetic logs generated 1000 cases. The application to generate logs was developed using some established

procedures, drafts of other processes, job routines, and several deviations that were common to happen, according to the people involved in the processes. Timeouts and activities were randomly introduced to simulate real-life operations. Another characteristic of these processes is the necessity to improve the current control system, one goal was to obtain a more transparent distribution process using process mining discovery techniques.

Stage 2: Discovery and Conformance Analysis

The first step is to analyze to what extent the data must be cleaned and then perform a statistical description. Some statistics measures include the number of events, cases, and resources in the log as well as some values like frequencies, duration of activities, standard deviation, mean, mode, median, and so on.

The research listed some algorithms used for process discovery from a control-flow perspective:

1. Alpha Miner (AM): Recommended for noiseless and complete event logs because of its sensitivity with incomplete data [54].
2. Heuristics Miner (HM): More robust than Alpha miner, it can represent the results in heuristics nets [56] and causal nets [7].
3. Genetic Miner (GM): Similar to Heuristics Miner as it is represented with heuristics net as well [15].
4. Fuzzy Miner (FM): Allows clustering and simpler visualization, recommended for less structured event logs [30].
5. Transition Systems (TS): Only recommended if the Transition System Analyzer plugin will be used in ProM[4].

Stage 3: Model Integration

As the process is relatively structured some perspectives can be developed:

1. Performance Perspective: It identifies bottlenecks, measures service levels as well as occurrence and frequency of events.
2. Organizational Perspective: It focuses on the organization structure, usually this means classifying people in roles and organizational units, and their relationships.
3. Case Perspective: If some unidentified routes do not belong to the reference model this perspective can analyze each case and identify the possible deviations.

Stage 4: Operational Support

This approach involves combining knowledge from historical event data (post-mortem) and ongoing cases (pre-mortem).

Depending on the planned goal, minor adjustments and process interventions can be made. The results should be discussed with the process owner and the stakeholders. If some changes are needed in the process, the whole process mining exercise should be done and then the new results should be compared with original results to prove if there is a benefit.

2.5.2 Results

Process discovery was realized using ProM and the Alpha Miner algorithm but fitness was not good. Heuristics Miner application increased the fitness value and when using causal nets generated a more generalized model. Genetic Miner had similar fitness values as Heuristics Miner. Fuzzy Miner generated the best results, and on the other hand, Transition System produced a model with some flaws. The performance analysis found three possible bottlenecks. The results of organizational perspectives techniques identified two sets of three roles that do the same work regarding handover of work technique. The working together technique confirmed that all the individuals work with each other, and finally, there was not found any reassignment or subcontracting.

A redesign was made using a system that uses barcode and radio frequency tracking. Once this new data was generated, new results were obtained with the Disco tool. The comparison with the initial results indicates that for Alpha and Heuristics Miner the fitness metric increased, but for Genetic Miner the value decreased. Bottlenecks were found again as well as shared work for some roles from the organizational perspective. Working together confirmed a shared work between individuals, and no reassignment was found. Nevertheless, this time subcontracting relationships were identified.

The results showed evidence of very long times and deviations from the ideal model and confirmed that distributed process conforms to a semi-structured Lasagna process as long it has a clear sequence of activities. The most appropriate algorithms for this application were Fuzzy Miner and Heuristics Miner.

The study used versions 5.2 and 6.2 of ProM and Disco. Regarding ProM, a certain level of expertise is recommended to select the plugins to work and to correctly interpret the results. On the other hand, Disco is a good option for starters with a better user interface. If user experience improves this would lead to higher utilization of the tools in the industry and for researchers as well.

Moreover, operational support techniques could add value in predicting cases that will not fulfill desired time deadlines, and proactive corrections can be performed. The authors of the survey recommend identifying relevant data, avoid inconsistencies, back up data regularly, and update models.

Chapter 3

Methodology

3.1 Introduction

As mentioned in Chapter 1, the main objective of the present work involves the discovery and recommendation of enhanced processes that are part of certification projects developed by Strong-step company. For this purpose, process mining and sequential pattern mining algorithms were executed for every process phase of CMMI, ISO 27001, and GDPR projects. This chapter presents the methodology followed in this thesis project to obtain the desired results.

All the data extraction, preparation, modeling, and evaluation phases were realized using the Jupyter Lab framework v3.0.7 and Python programming language, v3.9.1. PM4Py Python library was used to generate process mining models and a Python wrapper of the SPMF tool was used to get the most frequent sequences with sequential pattern mining algorithms. SPMF is open-source software written in Java that includes a very robust implementation of pattern mining algorithms[24].

Even the processes of the current scenario can be identified as Lasagna processes, and therefore, could follow the L* life-cycle that is described in Chapter 2, a different approach was chosen for this project. As some operations related to data cleaning and data preparation were important due to the noise in the database, using a CRISP-DM methodology seemed more convenient. Cross-Industry Standard Process for Data Mining (CRISP-DM) is one of the most used methodologies in the data mining field [43].

The CRISP-DM methodology consists of the following phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. For this project, the final phase is not considered. Future work can focus on the deployment of this solution with SCRAIM. Hence, the methodology for the current experimentation would be as shown in Figure 3.1

The methodology starts understanding the business domain and the business objective so that some goals can be established. Then, after collecting information a data understanding phase is executed. Some description and exploration tasks are done to verify the quality of the given information. Next, in the data preparation phase, data is cleaned and filtered to be used for each

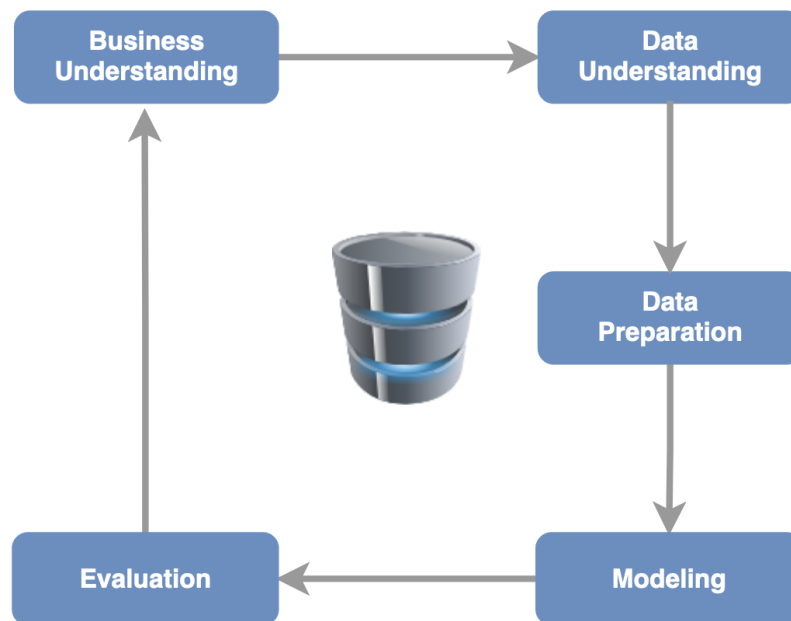


Figure 3.1: Experimentation methodology.

phase process. After that, the data is formatted and prepared for modeling. A different modeling phase is required for each technique, one for process mining and another for sequential pattern mining. Finally, the process models are evaluated with the client and for process mining outcome some metrics are used. The whole process repeats until it reaches a consistent model that is accepted by the stakeholders.

3.2 Business Understanding

3.2.1 Business Objective

The main objective of the present study is to find the ideal process which a phase should follow in a certification project. Strongstep wanted to discover the most repeated activities but also undetected activities that can be included in new project implementations. This can be identified as a process update exercise as some processes would need to revise their structure. Time optimization among activities was a concern to be considered, but the quality of data and the limited time for the research led to keep this for future work.

3.2.2 Mining Goals

As there are three types of projects, first, all the phases for every project were considered. This approach was not ideal because it would mean getting a lot of models as some processes are labeled in different ways. To solve this difficulty, the processes of CMMI and ISO 27001 were

unified, this agreement was reached with Strongstep support. Hence, the final models correspond to 7 phases in CMMI and ISO27001 projects and 3 phases in GDPR processes.

For CMMI and ISO 27001 projects, the phases are Kickoff, Initial Gap Analysis, Workshops, Process Monitoring, Intermediate Gap Analysis, Final Gap Analysis, and Project Closure. For GDPR projects, the phases are Preparation and Planning, Execution and Implementation, and Project Closure. The last phase is common between the two variants.

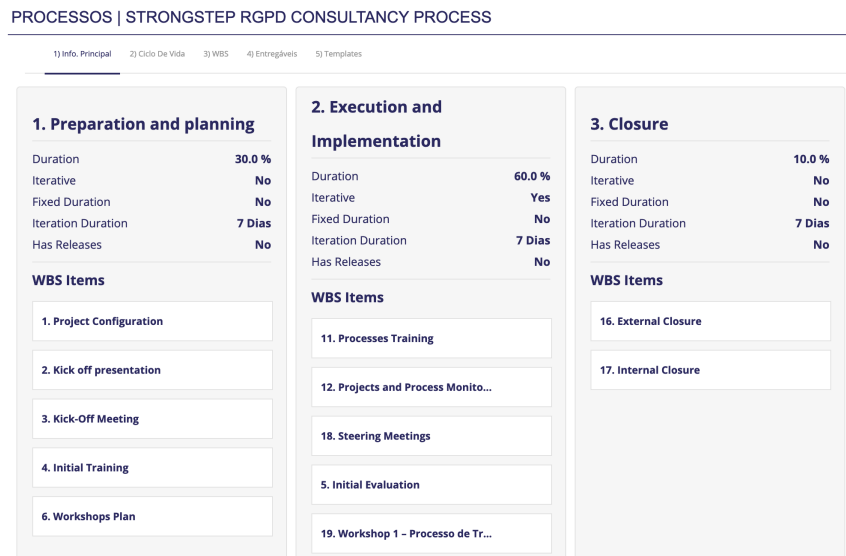


Figure 3.2: GDPR phases represented in SCRAIM.

In Figure 3.2, it can be seen the GDPR process phases that are represented in the SCRAIM tool.

To reach the process mining goals it was performed a process discovery execution using the SCRAIM event logs provided by Strongstep. This led to obtaining several process models for every phase. Then, a conformance checking evaluation was realized for every model and a process enhancement exercise was manually done with Strongstep feedback.

The sequential pattern mining objective was to verify the sequences that are most repeated in every phase, considering that all the processes should be sequential. After having these results, it is possible to perform a comparison with the results of process mining implementation.

The result of the process mining experimentation is represented as Petri nets and Directly-Follows graphs in Chapter 4. The outcome of the sequential pattern mining experimentation is a set of activities for each phase, corresponding to the most repeated pattern. The sequences with higher support are prioritized.

3.3 Data Understanding

Once the goals are set and the business logic is understood, the data understanding phase starts with the following activities: Data Collection, Data Description, Data Exploration, and Data Quality Verification.

3.3.1 Data Collection

The data used was entirely collected from Strongstep. Each project generated an event log in CSV format containing all the activities developed, the time when these activities were executed, the correspondent phase, and other related data. All this information was obtained from SCRAIM software, the project management software of Strongstep.

Projects were divided into those from 2020 and 2021 and projects made before 2020, i.e., old and new projects. Old projects are already finished and some new projects are still in execution. One important aspect to note is that old projects did not follow a CMMI, ISO 27001, or GDPR process but a Scrum methodology process. This means that activities were not associated with a certification phase but with a Scrum Sprint phase. Thus, the amount of valid data decreased and was not enough to work. To solve this data scarcity problem two solutions were proposed: generate synthetic data based on available information or change the Sprint tag to a certification phase tag, i.e., CMMI, ISO 27001, or GDPR phase identification.

The second proposal was chosen and performed by Strongstep. The reason to choose this option was the fact that synthetic information could not generate data with good quality, and having the know-how of Strongstep resulted in a better approach as they could align the activities with their respective processes.

Enough data was available for working on the defined process mining goals after performing the alignment operation previously described. The final data belonged to 6 CMMI projects, 12 ISO 27001 projects, and 17 GDPR projects. In the end, 35 projects were considered, 18 were part of the cataloged as new projects, and 17 were part of the old ones.

3.3.2 Data Description

Some data description operations were done to understand the information that Strongstep provided. It is important to mention that throughout the thesis project, there was always support from Strongstep to clarify any concerns related to the data and the processes. The final database consists of 35 projects merged in one data frame of 1602 entries and 77 columns.

For process mining experimentation three types of variables are crucial: case id, activity, and timestamp. The case id property orders all the activities for every case, for this exercise a case corresponds to a project instance. The activity value is the name of the action performed, and the timestamp is the time when each activity was executed. This time variable is essential because it orders the activity sequence.

Another important variable is the phase variable as it will filter just the activities referred to a specific phase, after all, the process to optimize is the phase process and not the whole project process. Two variables were added to the database: one to identify each project instance, i.e., the case id, and another to identify the project type (CMMI, ISO 27001, or GDPR).

Some variables were not necessary to the analysis and were dropped. Once all the columns that are not pertinent to the process mining execution were dropped the data frame ends with 1497

non-null values with the following five variables: Activity, Phase, Timestamp, Project Type, and Project Name.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1497 entries, 789 to 104
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Activity    1497 non-null   object
1   Phase       1497 non-null   object
2   Timestamp   1497 non-null   object
3   ProjType    1497 non-null   object
4   ProjName    1497 non-null   object
dtypes: object(5)
memory usage: 102.5+ KB
```

Figure 3.3: Data frame entries and columns.

In Figure 3.4, all five variables are described by their type. Pandas tool shows all variables as object types but here it can be seen that all are of string type. Hence, all are categorical variables.

	type	str	int	float	decimal	NaN	Nc2float	Total
Activity	<class 'str'>	1497	0	0	0	0	1497	1497
Phase	<class 'str'>	1497	0	0	0	0	1497	1497
Timestamp	<class 'str'>	1497	0	0	0	0	1497	1497
ProjType	<class 'str'>	1497	0	0	0	0	1497	1497
ProjName	<class 'str'>	1497	0	0	0	0	1497	1497

Figure 3.4: Variables type of the data frame.

Figure 3.5 shows a visualization of the data frame for the last 5 entries. The first column is an index value, the following variables correspond to activity, phase, timestamp, project type, and project name values, respectively.

	Activity	Phase	Timestamp	ProjType	ProjName
1597	Acompanhamento RH/revisão de Documentos	Acompanhamentos Piloto - 2020-04-08	2019-12-19	ISO 27001	95.1 PROCENSUS
1598	Ponto de situação interna (Joana)	Acompanhamentos Piloto - 2020-04-08	2019-12-18	ISO 27001	95.1 PROCENSUS
1599	Ponto de situação	Acompanhamentos Piloto - 2020-04-08	2019-12-20	ISO 27001	95.1 PROCENSUS
1600	Acompanhamento	Acompanhamentos Piloto - 2020-04-08	2020-01-02	ISO 27001	95.1 PROCENSUS
1601	Acompanhamento Marketing	Acompanhamentos Piloto - 2020-04-08	2020-01-02	ISO 27001	95.1 PROCENSUS

Figure 3.5: Data frame visualization.

3.3.3 Data Exploration

In this phase, the first exercise consisted of presenting a statistical analysis of all variables.

	VarName	Nan#	str#	N	Class#	Mode	ModeN	MinClass	MinClassN
0	Activity	0	object	1497	658	Steering Meetings	23	[Procensus] Acompanh	1
1	Phase	0	object	1497	199	Acompanhamentos Pilo	75	3. Closure - 2021-04	1
2	Timestamp	0	object	1497	652	2019-06-07	20	2018-10-29	1
3	ProjType	0	object	1497	3	ISO 27001	714	CMMI	353
4	ProjName	0	object	1497	35	96.1 NOMADTECH	179	106.2 EFFIZENCY GDPR	13

Figure 3.6: Statistical analysis of data frame.

Figure 3.6 shows a statistical description of all values. First, it is shown the number of entries for all variables, and as it was concluded before, there are no numerical variables. The number of classes is very high for activities and timestamps because the description of the activities and the timestamp vary considerably. For the phase, the number of the class is also high because this value contains not only the name of the phase but also the date when it finishes or is intended to finish, this provokes more classes to appear. Finally, the project type variable contains 3 classes for CMMI, ISO 27001, and GDPR, and the project name variable contains 35 classes for the 35 existent projects. More statistics like the most repeated (mode) and the least repeated are also shown. As all the variables are categorical, other statistical parameters like mean or standard value are not available.

Another result of the exploration phase is shown in Figure 3.7. In this figure are listed all the 35 projects with the number of entries for each one in the data frame. Each project name is formed with an identification number provided by Strongstep, the name of the company to audit, and in some cases the type of the project.

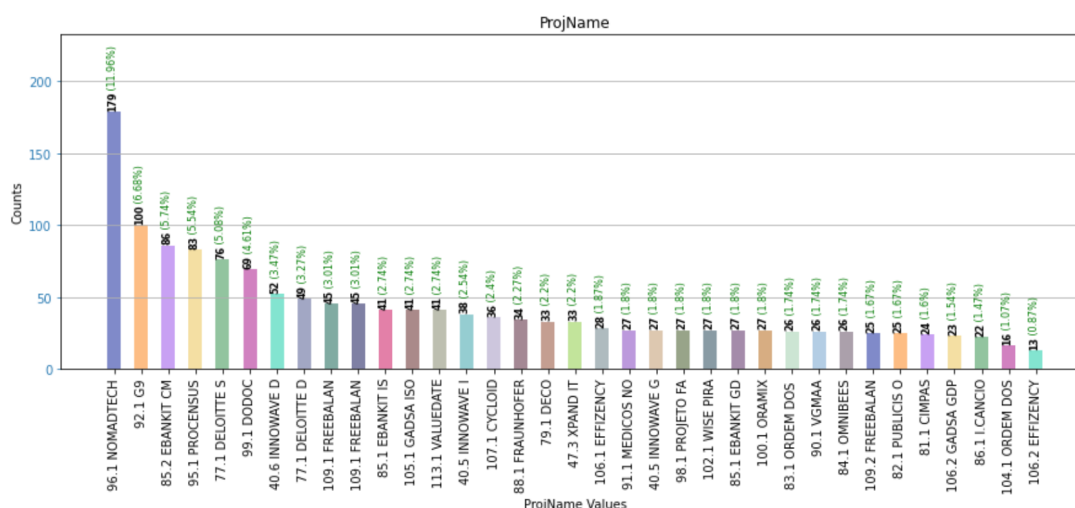


Figure 3.7: Number of entries for project name.

Figure 3.8 shows the occurrences for every type of project. This is not related to the number of projects but the number of entries for each type of project, i.e., some projects have more activities than others.

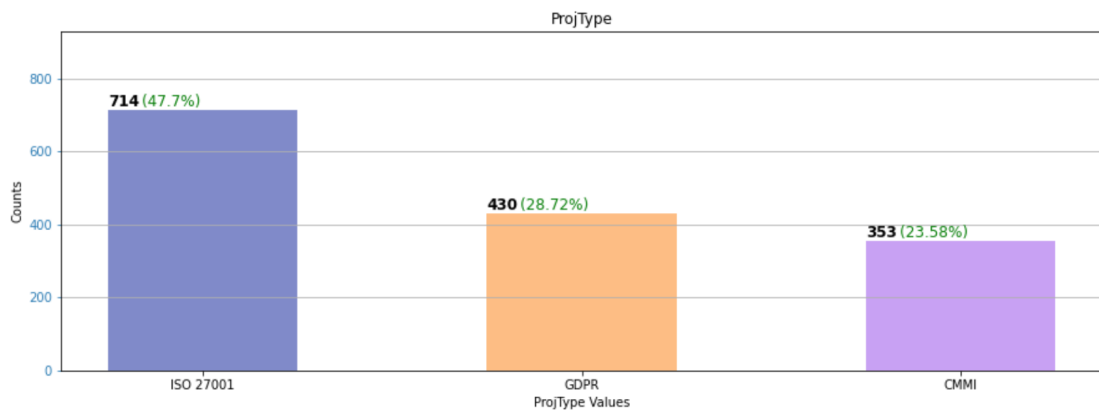


Figure 3.8: Number of entries for project type.

3.3.4 Data Quality Verification

The previous steps showed a database with a lot of variation within its values. As the main goal consists of finding a common process that suggests a particular sequence of activities, a cleaning phase is essential to achieve that purpose. The consistency of the model will closely depend on the quality of the data.

Even data scarcity problem was solved, some preprocessing tasks are needed to increase quality in data and guarantee robust models. Another issue found in this phase was the repetition of timestamps in activities that are occurring at different times, this gives an idea of parallel processes occurring, when in fact, this is not happening. In the next phase, some preparation activities are described.

3.4 Data Preparation

3.4.1 Data Selection

The first step in the data preparation phase consists of selecting the necessary data to build the requested model. Selecting data involves a filtering operation to choose all the activities that correspond to a specific phase. In some cases, there is a need to execute a merge action among different phases. For instance, ISO 27001 and CMMI projects have related phases that have different labels. Hence, data of both cases should be concatenated and then filtered to generate the correspondent model.

Figure 3.9 shows the result of selecting activities with different labeled phases that should be treated as a unique phase. To obtain this outcome, data containing the string “Kick” in the phase variable are selected, guaranteeing that all the entries with labeled phases like “Kickoff”, “Kick-off”, and “Pre-Kickoff” are gathered in one unique data frame.

A merge example is presented forwards. Figure 3.10 shows the concatenation of two previous data frames that contain the information of phases labeled as “Initial Gap Analysis”, “Initial Evaluation”, and “Avaliação Inicial”. Although these phases are defined with different labels they refer

	Activity	Phase	Timestamp	ProjType	ProjName
1435	Planeamento do Projeto	Kickoff - 2017-09-19	2017-09-17	ISO 27001	40.5 INNOWAVE ISO
1434	Kickoff Interno	Kickoff - 2017-09-19	2017-09-18	ISO 27001	40.5 INNOWAVE ISO
1437	Pre - KickOff	Kickoff - 2017-09-19	2017-09-19	ISO 27001	40.5 INNOWAVE ISO
1436	Kickoff do Projeto	Kickoff - 2017-09-19	2017-09-20	ISO 27001	40.5 INNOWAVE ISO
156	Team Definition	Pre-Kickoff - 2017-11-14	2017-11-07	CMMI	77.1 DELOITTE DEVELOPMENT

Figure 3.9: Data selection for Kickoff phase.

to the same Initial Gap Analysis phase. In this case, CMMI and ISO 27001 activities perform the same Initial Gap Analysis phase but have different values for that variable.

	Activity	Phase	Timestamp	ProjType	ProjName
1453	Preparação da Avaliação Inicial	Avaliação Inicial - 2018-06-29	2017-09-22	ISO 27001	40.5 INNOWAVE ISO
1457	Avaliação Inicial	Avaliação Inicial - 2018-06-29	2017-09-23	ISO 27001	40.5 INNOWAVE ISO
1456	Relatório Avaliação Inicial	Avaliação Inicial - 2018-06-29	2017-09-24	ISO 27001	40.5 INNOWAVE ISO
1455	Avaliação Inicial - Pre - Apresentação Resultados	Avaliação Inicial - 2018-06-29	2017-09-25	ISO 27001	40.5 INNOWAVE ISO
1454	Avaliação Inicial - Apresentação dos Resultados	Avaliação Inicial - 2018-06-29	2017-09-26	ISO 27001	40.5 INNOWAVE ISO
163	Initial Gap Analysis	Initial Gap Analysis - 2018-12-31	2017-11-21	CMMI	77.1 DELOITTE DEVELOPMENT
164	Report Presentation	Initial Gap Analysis - 2018-12-31	2017-11-23	CMMI	77.1 DELOITTE DEVELOPMENT
165	Report Preparation	Initial Gap Analysis - 2018-12-31	2017-11-23	CMMI	77.1 DELOITTE DEVELOPMENT
166	Analyze gaps	Initial Gap Analysis - 2018-12-31	2017-11-23	CMMI	77.1 DELOITTE DEVELOPMENT

Figure 3.10: Data selection for the Initial Gap Analysis phase.

As Figure 3.10 showed previously, there are some activities or phases with English values and others labeled in Portuguese. In the next phase, some cleaning activities are described to achieve a homogeneous data frame that could reach more consistent models.

3.4.2 Data Cleaning

Figure 3.11 presents some activity values with some text that is not part of the activity variable but is related to its phase description. This is the case for the two last entries that have an “Avaliação Inicial” text at the beginning of the activity description. Hence, some text replacement operations are needed in the activity values. Performing these cleaning operations helps to correct all the inconsistencies found.

Figure 3.12 shows the result of a cleaning operation. This time the values are in English and the initial description text that was related to the phase is deleted for the two last entries. The

	Activity	Phase	Timestamp	ProjType	ProjName
1453	Preparação da Avaliação Inicial	Avaliação Inicial - 2018-06-29	2017-09-22	ISO 27001	40.5 INNOWAVE ISO
1457	Avaliação Inicial	Avaliação Inicial - 2018-06-29	2017-09-23	ISO 27001	40.5 INNOWAVE ISO
1456	Relatório Avaliação Inicial	Avaliação Inicial - 2018-06-29	2017-09-24	ISO 27001	40.5 INNOWAVE ISO
1455	Avaliação Inicial - Pre - Apresentação Resultados	Avaliação Inicial - 2018-06-29	2017-09-25	ISO 27001	40.5 INNOWAVE ISO
1454	Avaliação Inicial - Apresentação dos Resultados	Avaliação Inicial - 2018-06-29	2017-09-26	ISO 27001	40.5 INNOWAVE ISO

Figure 3.11: Example of data before cleaning.

main focus of this data cleaning phase is to align the Activity variable values, considering that the algorithms will use this information to build the requested model.

	Activity	Phase	Timestamp	ProjType	ProjName
1453	Gap Analysis Preparation	Avaliação Inicial - 2018-06-29	2017-09-22	ISO 27001	40.5 INNOWAVE ISO
1457	Initial Gap Analysis	Avaliação Inicial - 2018-06-29	2017-09-23	ISO 27001	40.5 INNOWAVE ISO
1456	Gap Analysis Report	Avaliação Inicial - 2018-06-29	2017-09-24	ISO 27001	40.5 INNOWAVE ISO
1455	Pre - Results Presentation	Avaliação Inicial - 2018-06-29	2017-09-25	ISO 27001	40.5 INNOWAVE ISO
1454	Results Presentation	Avaliação Inicial - 2018-06-29	2017-09-26	ISO 27001	40.5 INNOWAVE ISO

Figure 3.12: Example of data after cleaning.

3.5 Process Mining Modeling

Once the cleaning operation finishes is time to prepare data to generate the model. PM4Py is the leading tool for process mining in Python [13] and is used for this modeling phase. To work with this library is necessary first to create a log object and preprocess the data to build the models.

3.5.1 Modeling Preparation

Four variables are needed to create a PM4Py event object: case id, activity, timestamp, and data frame. For our case, the data frame would be the one selected for its corresponding phase, while the other variables will correspond to the ProjName, Activity, and Timestamp variables, respectively.

After generating this event object, an XES file is created. XES stands for eXtensible Event Stream and corresponds to the IEEE Standard 1840-2016 for achieving interoperability in event logs and event streams. The standard defines XES as "a language to transport, store, and exchange event data" [20].

Finally, a PM4Py log is generated. The XES file is imperative to create this log object. Now is possible to build different models like Petri nets, heuristic nets, directly-follow graphs, or process trees using PM4Py functions.

Figure 3.13 presents data of the Initial Gap Analysis phase transformed into a set of traces, i.e., the result of generating the XES file. Each trace corresponds to a project instance, while each

```

<?xml version="1.0" encoding="utf-8" ?>
<log>
  <extension name="Concept" prefix="concept" uri="http://www.xes-standard.org/concept.xesext" />
  <extension name="Time" prefix="time" uri="http://www.xes-standard.org/time.xesext" />
  <string key="origin" value="csv" />
  <trace>
    <string key="concept:name" value="105.1 GADSA ISO" />
    <event>
      <string key="concept:name" value="Initial Gap Analysis" />
      <string key="Phase" value="Gap Analysis - 2020-11-18" />
      <date key="time:timestamp" value="2020-10-09T00:00:00+00:00" />
      <string key="ProjType" value="ISO 27001" />
      <int key="@index" value="891" />
    </event>
    <event>
      <string key="concept:name" value="Gap Analysis Report" />
      <string key="Phase" value="Gap Analysis - 2020-11-18" />
      <date key="time:timestamp" value="2020-10-10T00:00:00+00:00" />
      <string key="ProjType" value="ISO 27001" />
      <int key="@index" value="890" />
    </event>
    <event>
      <string key="concept:name" value="Pre - Results Presentation" />
      <string key="Phase" value="Gap Analysis - 2020-11-18" />
      <date key="time:timestamp" value="2020-10-11T00:00:00+00:00" />
      <string key="ProjType" value="ISO 27001" />
      <int key="@index" value="889" />
    </event>
    <event>
      <string key="concept:name" value="Results Presentation" />
      <string key="Phase" value="Gap Analysis - 2020-11-18" />
      <date key="time:timestamp" value="2020-10-12T00:00:00+00:00" />
      <string key="ProjType" value="ISO 27001" />
      <int key="@index" value="888" />
    </event>
  </trace>
</log>

```

Figure 3.13: Example of XES file for the Initial Gap Analysis phase.

activity is identified as an event. Hence, a trace is a set of events, and the log is formed by a set of traces.

3.5.2 Algorithms Selection

Process discovery seeks to find a fit model that describes a defined process. Several algorithms can be used for this purpose, PM4Py library implements Alpha, Heuristics, and Inductive miners. The following table describes some features for each miner. Table 3.1 shows that Alpha Miner is the least favorable option to work while Inductive Miner has more advantages than the other choices.

Alpha	Heuristics	Inductive
Weak against noise	Deals with noise	Deals with noise
Model might not be sound	Model is not guaranteed to be sound	Model is sound
Cannot handle invisible tasks	Does not guarantee to handle invisible tasks	Detects invisible tasks
Cannot handle short loops	Detects short loops	Detects loops

Table 3.1: Comparison of process discovery miners implemented in PM4Py

The Alpha Miner is not considered because it is not strong against noise. The current database is prone to human error and needs a more robust algorithm to develop process discovery models. The Heuristics Miner is a good option as it can deal with noise [57], although, it does not guarantee

sound models. Soundness property indicates that if a process is flooded with k tokens, all these k tokens should get to the final activity [3].

The Inductive Miner is the most used process mining algorithm according to PM4Py documentation [2], the principal advantage is the guarantee of having sound models [34]. PM4Py implements two variants of the Inductive Miner apart from the miner itself. These variants are Inductive Miner infrequent (IMi) and Inductive Miner directly-follows (IMd).

Inductive Miner infrequent creates a sound model in a shorter time by applying the Pareto principle, moreover, compared with the Inductive Miner, the infrequent variant has more precision but a lower fitness according to [36]. On the other hand, Inductive Miner directly-follows uses a Directly-Follows graph and a divide-and-conquer strategy to work with models that require processing a big amount of data guaranteeing scalability and soundness [35].

The chosen miner for building the model is Inductive Miner infrequent because of its higher precision. Some experimental comparisons were done between Inductive Miner infrequent and Heuristics Miner, as well as Inductive Miner infrequent and the other variants of Inductive Miner. All the results showed a better precision when using the infrequent variant. Even the research documentation mentioned the fitness quality metric would decrease in the current project the fitness of the models using inductive miner infrequent did not decrease in a considerable way and in some cases was better than the other algorithms.

3.5.3 Model Visualization

There are some possible representations to visualize models using the PM4Py library, these include process trees, heuristic nets, Directly-Follows graphs, and Petri nets. As it was explained in the previous chapter, Petri nets are the most used graphs in process discovery operations, nevertheless, Directly-Follows Graphs (DFG) are used as well in this phase to show a simpler model to the stakeholders.

Directly-Follows graphs represent the activities within their nodes and their edges correspond to directly-follows relationships between the activities. Their edges can represent metrics like frequency or performance, for this case, frequency is an important metric to show due to the requirement of identifying the most repeated activities among the different processes. However, Van der Aalst in [5] warns about the limitations of Directly-Follows graphs on some process mining tools. The main concern aims to the removal of low frequent activities for improving performance which causes a misleading in some results.

Figure 3.14 shows an example of a Petri net using Inductive Miner infrequent for the Preparation and Planning phase of GDPR projects while Figure 3.15 presents a Directly-Follows Graph for the same case. As the Directly-Follows Graph is easier to visualize, this model representation is shown in the next chapter to discuss the results of process mining implementation.

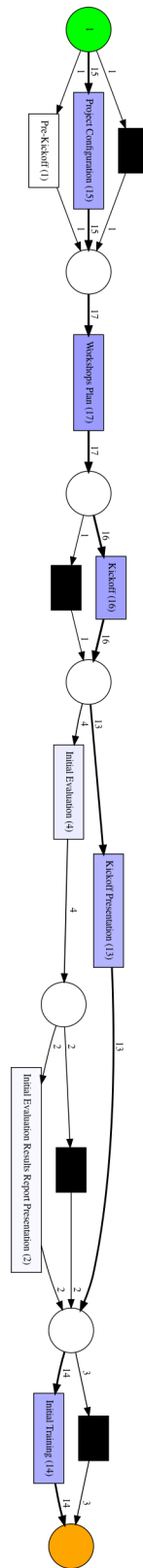


Figure 3.14: Petri net model for Preparation and Planning phase.

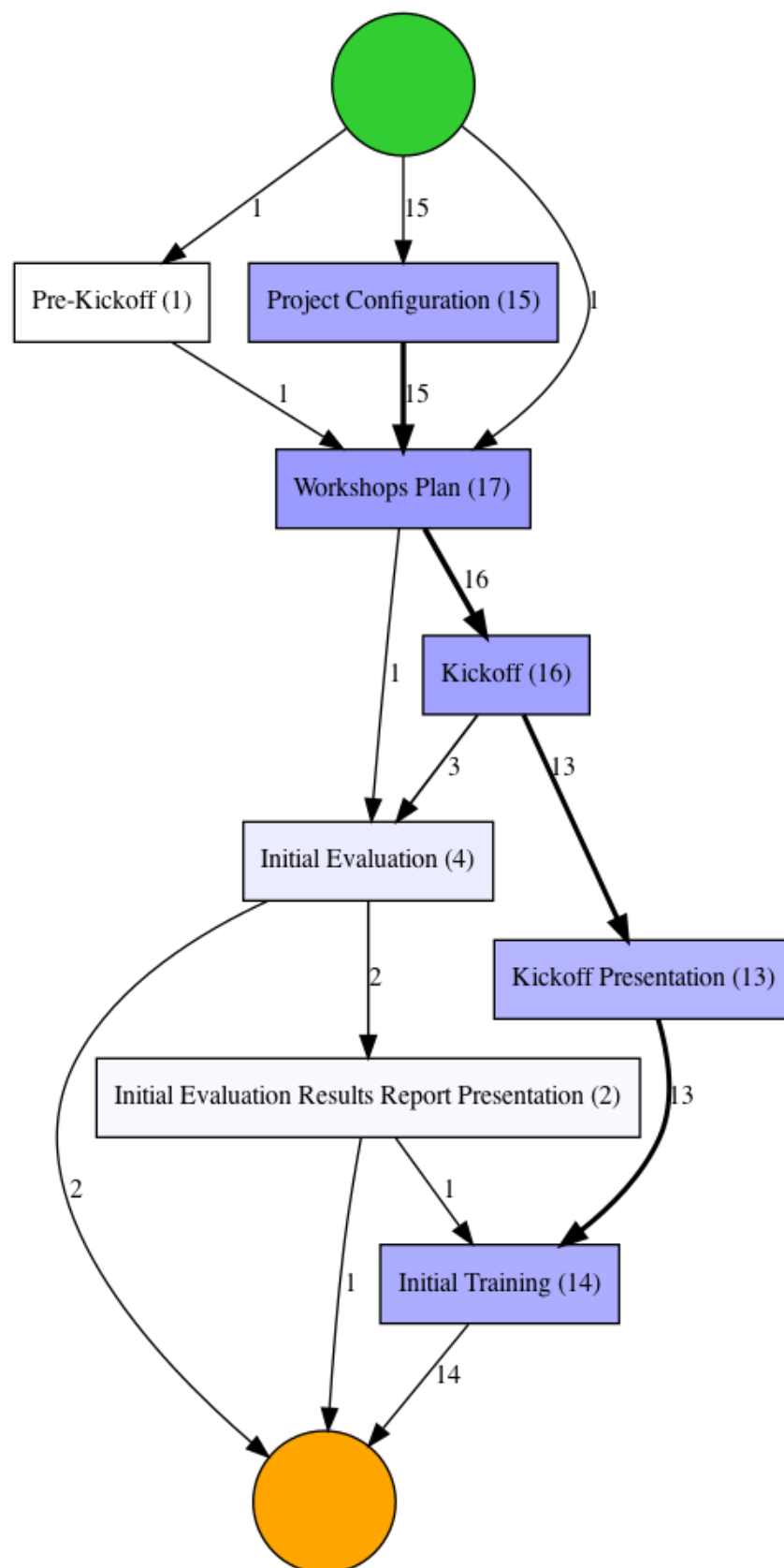


Figure 3.15: Directly-Follows Graph model for Preparation and Planning phase.

3.6 Sequential Pattern Mining Modeling

This second technique applied in this project corresponds to sequential pattern mining. The goal of this phase is to perform a pattern mining operation in all the project phases considering that the processes are supposed to be sequential. SPMF is a data mining library developed in Java that specializes in frequent pattern mining. It is an open-source project that started in 2009 thanks to the effort of Philippe Fournier-Viger. Nowadays, it continues adding the State of The Art algorithms in fields like itemset mining, high-utility pattern mining, periodic pattern mining, time series mining, sequential pattern mining, among others [24].



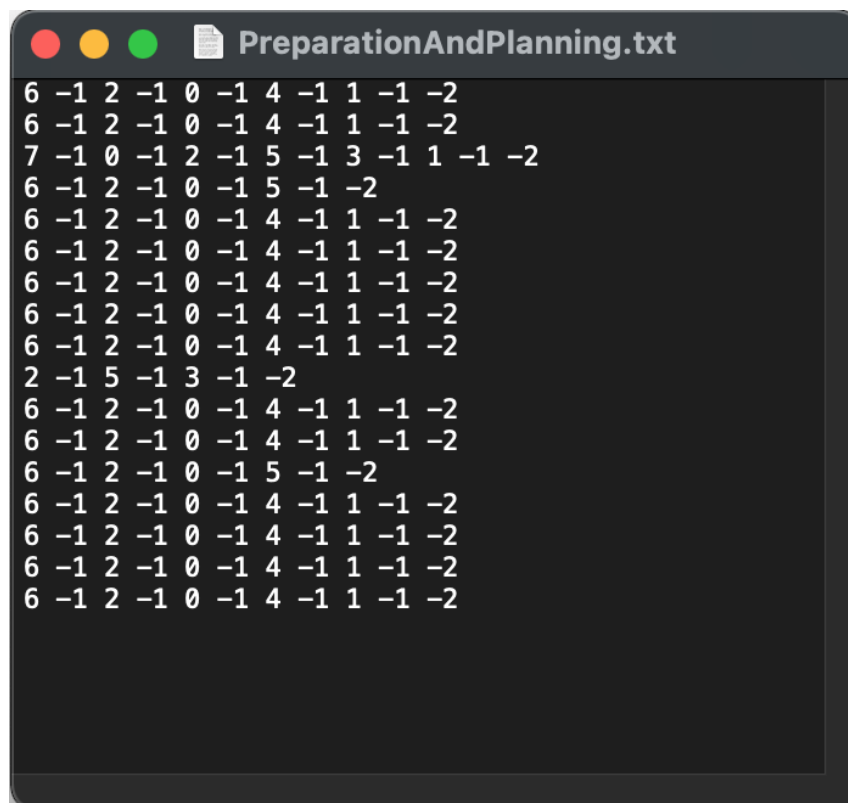
Figure 3.16: SPMF graphical interface.

Figure 3.16 presents the user interface of the SPMF tool. It can also be used as a command-line tool or with wrappers for Python and R. The figure shows a first field where the algorithm is chosen. Then, it is necessary to select a text file with the information of the sequence database and a text file that will have the output of the mining operation. Next, the minimum support is

requested, and some other optional values can be written, such as minimum and maximum pattern length. The output of the pattern mining operation can be seen in the text file created or in the same SPMF graphical interface.

3.6.1 Modeling Preparation

As seen before, to use the SPMF tool a text file is needed. Nevertheless, this file should follow a specific format to work. Even the SPMF tool has some internal features to transform CSV files to its format the data conversion to this format was developed using Python functions.



```

6 -1 2 -1 0 -1 4 -1 1 -1 -2
6 -1 2 -1 0 -1 4 -1 1 -1 -2
7 -1 0 -1 2 -1 5 -1 3 -1 1 -1 -2
6 -1 2 -1 0 -1 5 -1 -2
6 -1 2 -1 0 -1 4 -1 1 -1 -2
6 -1 2 -1 0 -1 4 -1 1 -1 -2
6 -1 2 -1 0 -1 4 -1 1 -1 -2
6 -1 2 -1 0 -1 4 -1 1 -1 -2
6 -1 2 -1 0 -1 4 -1 1 -1 -2
2 -1 5 -1 3 -1 -2
6 -1 2 -1 0 -1 4 -1 1 -1 -2
6 -1 2 -1 0 -1 4 -1 1 -1 -2
6 -1 2 -1 0 -1 5 -1 -2
6 -1 2 -1 0 -1 4 -1 1 -1 -2
6 -1 2 -1 0 -1 4 -1 1 -1 -2
6 -1 2 -1 0 -1 4 -1 1 -1 -2
6 -1 2 -1 0 -1 4 -1 1 -1 -2

```

Figure 3.17: SPMF format in the input text file.

Figure 3.17 shows the SPMF format to perform frequent mining operations. The activities are identified with integers dividing every itemset with "-1", even a few algorithms can work with strings, most of the algorithms work with integers. To indicate that a sequence ends a "-2" should be written.

In the current project, a sequence will represent a project phase or a trace, compared to the process mining operation. The sequence database would be all the existent project instances for the selected phase, or for process mining, the log containing all the traces.

3.6.2 Algorithms Selection

Once the data is formatted and the input text file generated, the Python wrapper could be used to generate the sequential pattern mining outcome.

There are plenty of algorithms that can be used in the current exercise, but according to this study [23], the best are CM-SPAM, CM-SPADE, and CM-ClaSP. All of them use the previously mentioned co-occurrence pruning technique to optimize the use of SPAM, SPADE, and ClaSP algorithms. Even CM-ClaSP would be a better choice as it is an algorithm for mining closed sequential patterns which means that the results will not be redundant.

Comparing the results of CM-SPAM with other algorithms like CM-SPADE, PrefixSpan, or CM-ClaSP, the conclusion reached is that the result does not change considerably, and having the chance of defining a minimum length for the output pattern helps to reduce the number of output patterns. The definition of minimum support, on the other hand, helps to find sequences with a good balance between support and length. For that reason, the CM-SPAM algorithm is chosen as the technique to use for this approach.

Figure 3.18 shows the outcome of the Preparation and Planning phase using the SPMF graphical interface. It shows a sequence of numbers divided by a "-1" that do not help to identify the activities, meanwhile, Figure 3.19 shows an array of activities as the final outcome. This is the main reason for using the Python wrapper instead of the user interface.

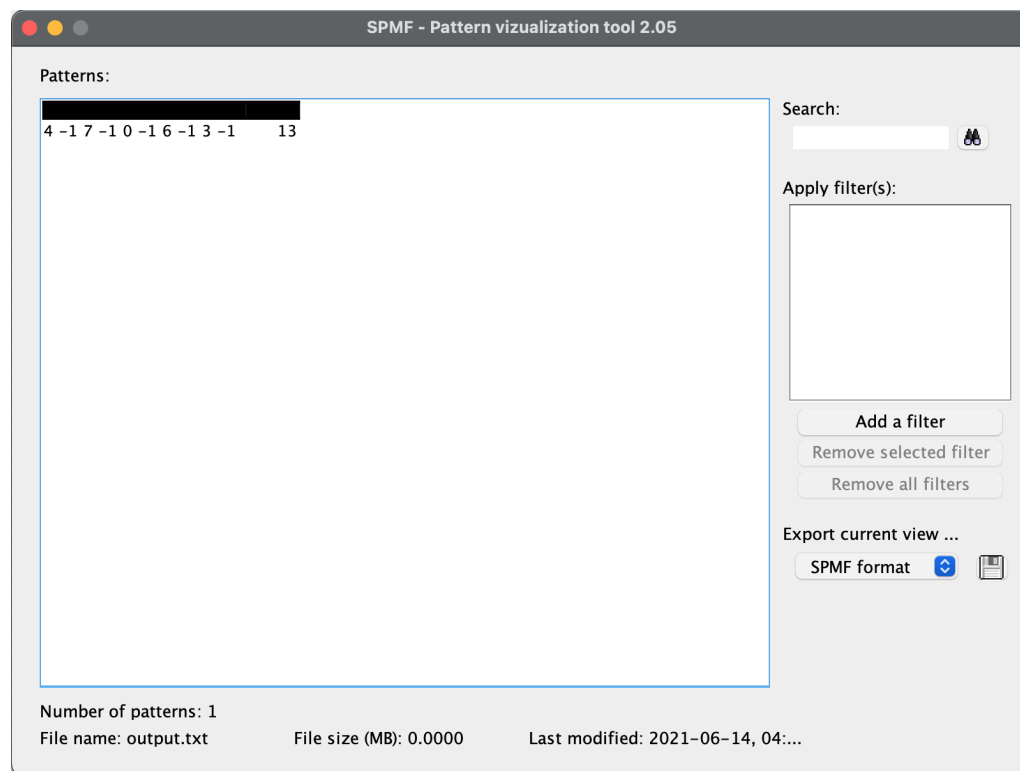


Figure 3.18: SPMF result with graphical interface for the Preparation and Planning phase.

```

>/Users/joseb/Developer/process-optimization-dissertation/spmf.jar
===== SPAM v0.97a- STATISTICS =====
Total time ~ 17 ms
Frequent sequences count : 1
Max memory (mb) : 6.74218751
minsup 9
=====

  pattern  sup
-----
0 [4, 7, 0, 6, 3]  13

...

['Project Configuration',
 'Workshops Plan',
 'Kickoff',
 'Kickoff Presentation',
 'Initial Training']

```

Figure 3.19: SPMF result with Python wrapper for the Preparation and Planning phase.

The next chapter presents the results and a further discussion for each project phase as well as a comparison between process mining and sequential pattern mining approaches.

Chapter 4

Discussion of Results

4.1 Introduction

This chapter presents the results of process mining and sequential pattern mining techniques used for the current project.

Projects are divided into two types of workflows: CMMI and ISO 27001 and GDPR. For the first type exists 7 phases which are Kickoff, Initial Gap Analysis, Workshops, Process Monitoring, Intermediate Gap Analysis, Final Gap Analysis, Project Closure; and for the second type there are 3 phases which are Preparation and Planning, Execution and Implementation and Project Closure. The final phase, called Project Closure, is the only common phase between the two types.

The following section includes a brief explanation of every phase and a comparison between both approaches implemented. The process model as a Directly-Follows Graph is presented and compared with the pattern with the highest support. Then, some graphics are presented to compare the quality metrics of all the phases. The quality dimensions include fitness, precision, simplicity, and generalization.

Finally, the last section presents a general evaluation for both kinds of projects. It summarizes the support and pattern length of CM-SPAM algorithm implementation and presents the four quality metrics and the number of places, arcs, and transitions of Petri nets models to support the process mining assessment.

4.2 Results of Project Phases

The results are divided into two kinds of projects. CMMI and ISO 27001 projects are gathered in a single type as their phases are equivalents. On the other hand, the first two phases of GDPR projects include activities that are part of several phases of the CMMI or ISO 27001 projects so that there is a need to handle this type of project separately. The last phase of GDPR projects, Project Closure, is the only equal phase for all CMMI, ISO 27001, and GDPR projects.

4.2.1 CMMI & ISO 27001 Projects

4.2.1.1 Kickoff

It is the first phase of a project and includes planning activities of the certification workflow.

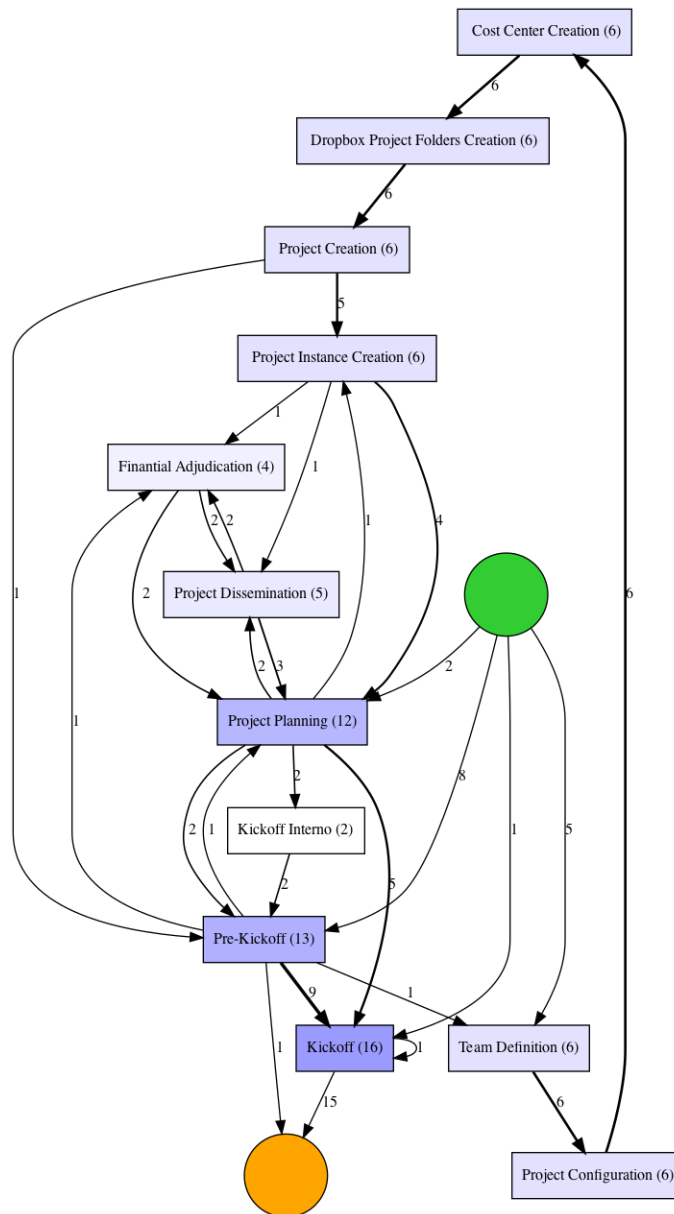


Figure 4.1: Directly-Follows Graph model of Kickoff phase.

The pattern with the highest support included just two activities:

- Pre-Kickoff
- Kickoff

As it can be seen in Figure 4.1 these are the most repeated activities within all the project instances.

Another recommended sequence includes the following activities:

- Team Definition

- Project Configuration
- Project Creation
- Project Planning
- Kickoff

This pattern has a support value of 0.35 against the 0.7 value of the first one. Nevertheless, it could be a good alternative if the project managers want to include more steps in the process as this sequence contains five activities against the two of the first recommendation.

Figure 4.1 shows that more activities (Cost Center Creation, Dropbox Folder Creation, Project Instance Creation) are present in at least five project instances but these activities are not part of the previously recommended pattern because are occurring at the same time. Project Configuration activity is conformed by Dropbox Project Folder Creation, Cost Center Creation, and Project Instance Creation. As these activities in some cases are logged being parallel, the CM-SPAM algorithm cannot gather all in one single sequence.

Another conclusion reached is that the pattern with the highest support corresponds mainly to the ISO 27001 project instances and the second pattern to the CMMI project instances. This can be verified as there are 13 instances for ISO 27001 projects and 5 instances for CMMI projects. Unifying both projects can lead to this kind of result as some different activities are done for every case, the advantage is that now a decision can be made to start managing the two cases as a single one.

4.2.1.2 Initial Gap Analysis

This phase executes a first evaluation of the company status regarding the certification objectives that need to be accomplished.

The sequential pattern mining outcome presents the following activities:

- Initial Gap Analysis
- Gap Analysis Report
- Pre - Results Presentation
- Results Presentation

For this case, the support is not high as is just 0.4. This low support can be explained as the project instances gathered four different versions of the same stage, two versions of CMMI projects and two versions of ISO 27001 projects. As it was explained in Chapter 3, some old projects needed to be included as a data scarcity problem appeared. This led to having two versions of the same phase in some cases.

Figure 4.2 shows that the four activities recommended by the CM-SPAM algorithm are also the most repeated ones. This process model has a considerable number of other activities that are present in very few instances, this would lead to a low precision for the Petri net of this phase.

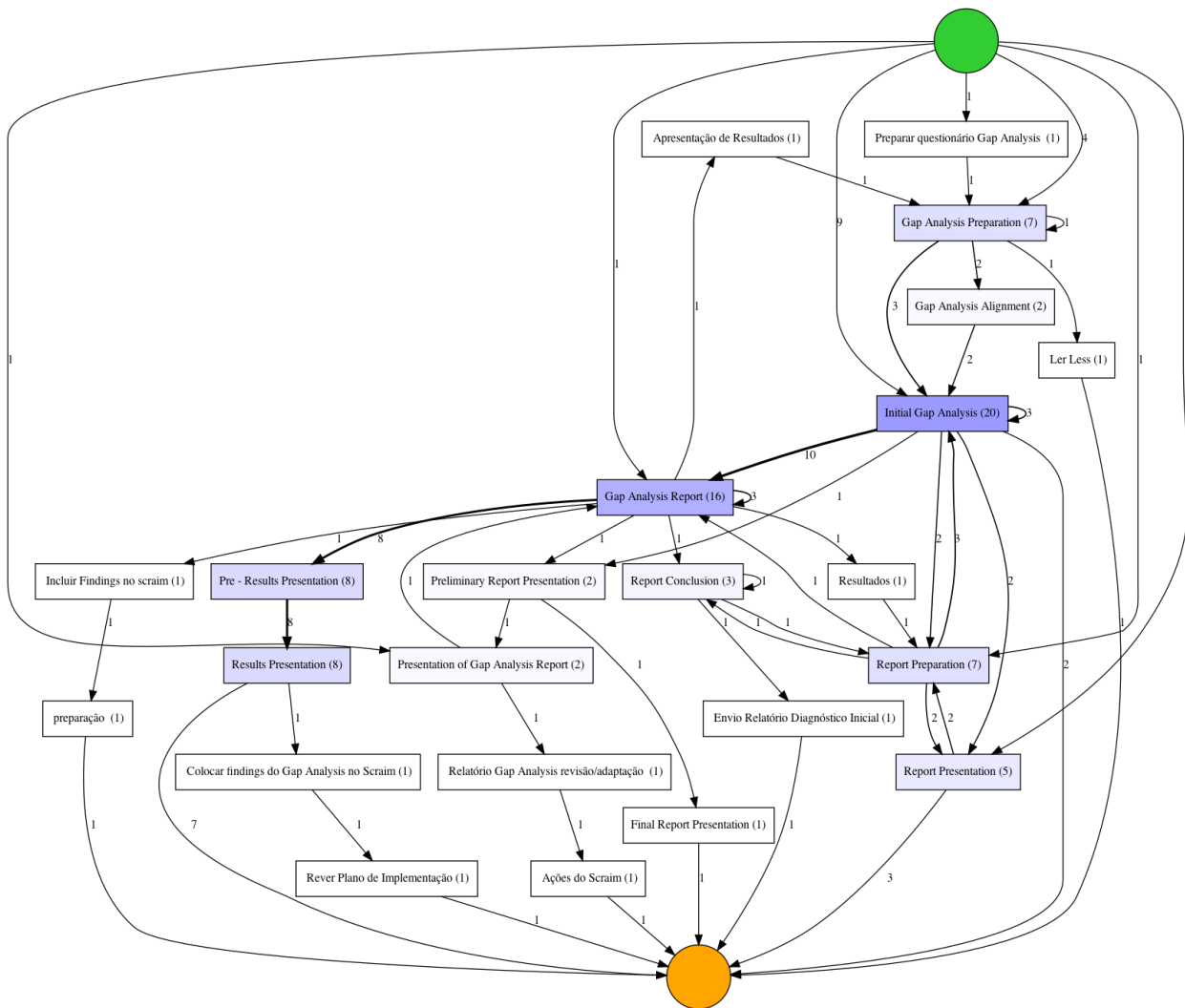


Figure 4.2: Directly-Follows Graph model of Initial Gap Analysis phase.

4.2.1.3 Workshops

This phase includes the activities that are done to guarantee the certification for the different areas such as Risk Management, Information Security Management, Integrated System Management, Incidents Management, Physical and Logical Controls, Information Security in Human Resources, Information Security in Marketing and Purchase and Legal for ISO 27001 projects. CMMI projects also include workshops to execute their evaluations but in this case, it depends on the areas that need to be evaluated. The Workshops phase presents a very inconsistent process model as the activities could vary in a lot of ways, for instance, a company could pay for five of the eight workshops that should be included in an ISO 27001 project. Also, the CMMI workshops are not attached to a known concern, it is uncertain which problem needs to be solved.

Having this kind of peculiarities make that no patterns should be recommended when applying sequential pattern mining implementation. The pattern with the highest support included the following activities:

- Workshop 1 - Information Security Risk Analysis and Treatment
- Workshop 2 - Information Security Management System And Integrated System Management
- Workshop 3 - Physical and logical controls
- Workshop 4 - Vulnerability problem and incidents Management
- Workshop 5 - Information Security in HR And Marketing

However, the support was 0.2 which is a very low value. The appendix includes the Directly-Follows Graph for this phase (Figure A.1) but is not shown in this section as it is not a consistent outcome.

4.2.1.4 Process Monitoring

Process Monitoring phase, the so-called "Acompanhamentos", are status meetings that are executed after the workshops. This phase is even more uncertain than the Workshops phase because here the monitoring meetings would depend on the necessity of the project. Some projects require more meetings than others to monitor the state of the project. Thus, a process is hard to define as this phase does not have a usual set of activities.

The process model of this phase is not presented as well, as is the most inconsistent of all, however, it can be seen in the appendix (Figure A.2). The pattern with higher support includes the activities:

- Acompanhamento
- Acompanhamento

This outcome represents a repeated label that is used for a status meeting activity. This pattern has a 0.3 support value and shows the lack of a good process recommendation.

4.2.1.5 Intermediate Gap Analysis

This phase is an intermediate evaluation to check the opportunities for improvement from the previous evaluations and to verify whether or not they have been addressed.

The most frequent pattern includes the following activities:

- Intermediate Gap Analysis
- Intermediate Gap Analysis Report
- Pre - Results Presentation
- Results Presentation

The support value for this pattern is 0.6. Figure 4.3 explicitly confirms that those four activities are the most recommended to be part of an optimal Intermediate Gap Analysis process. It can be seen also that just a few more activities are part of the Directly-Follows Graph. Plan and Preparation, Report Preparation, and Report Presentation are some activities that are part of CMMI projects but without the same importance as the initial ones.

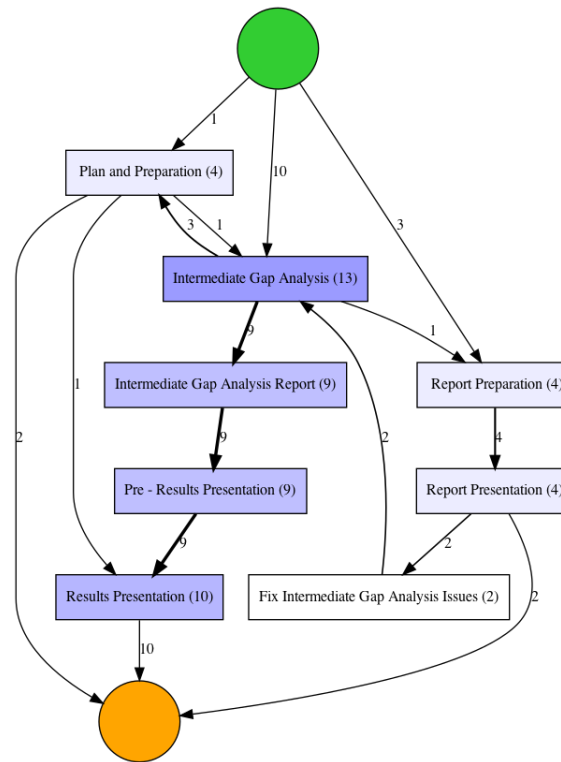


Figure 4.3: Directly-Follows Graph model of Intermediate Gap Analysis phase.

4.2.1.6 Final Gap Analysis

This is the last evaluation made until the project finishes. In some CMMI projects, it is identified as SCAMPI A.

The sequential pattern mining outcome presented the following activities:

- Final Gap Analysis
- Final Gap Analysis Report
- Pre - Results Presentation
- Results Presentation

The support value, in this case, is also 0.6. Figure 4.4 shows a similar outcome to the last phase. It is possible to conclude that all Gap Analysis processes include the same four activities, i.e., the evaluation itself, the report, the pre- results presentation, and the presentation of the results. For this phase also appears Plan and Preparation, Report Preparation and Report Presentation activities in a lesser amount.

4.2.1.7 Project Closure

The final phase includes some activities to evaluate all the workflow and to highlight the lessons learned. As mentioned before, this is the only phase that is common to the 35 project instances, and although this could lead to having more discrepancies in the model, the results show the opposite.

The most frequent pattern includes just two activities:

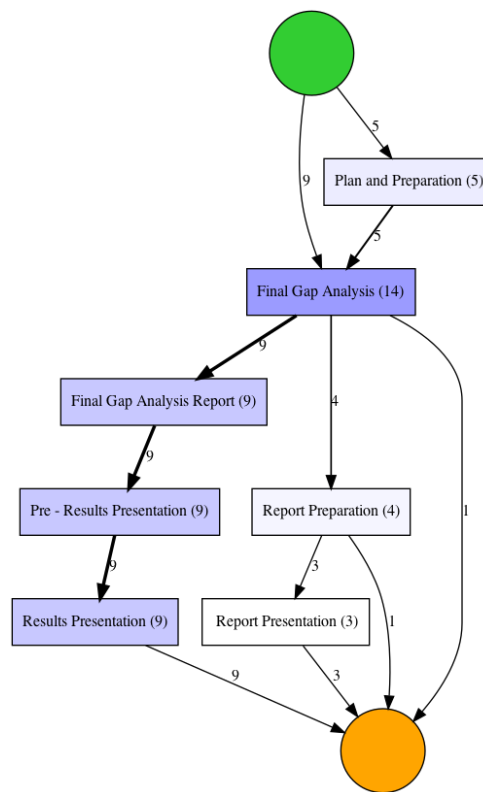


Figure 4.4: Directly-Follows Graph model of Final Gap Analysis phase.

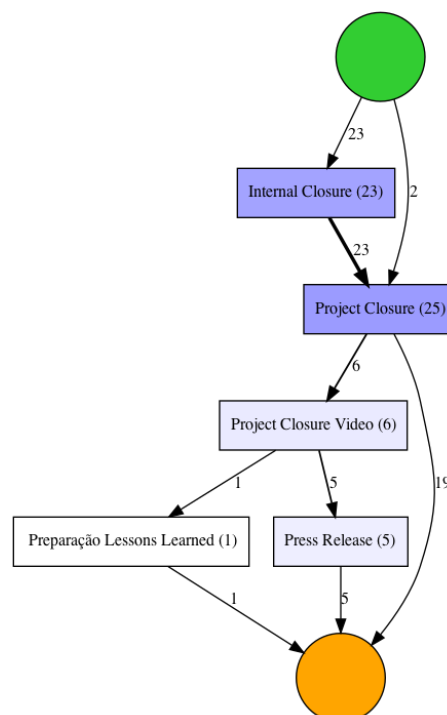


Figure 4.5: Directly-Follows Graph model of Project Closure phase.

- Internal Closure
- Project Closure

This pattern has a support value of 0.7. Figure 4.5 shows that there are other activities like Project Closure Video and Press Release that are occurring in some instances which are part of CMMI projects.

4.2.2 GDPR Projects

4.2.2.1 Preparation and Planning

The first phase of GDPR projects includes a similar phase as Kickoff but with different activities, as can be seen in Figure 4.6.

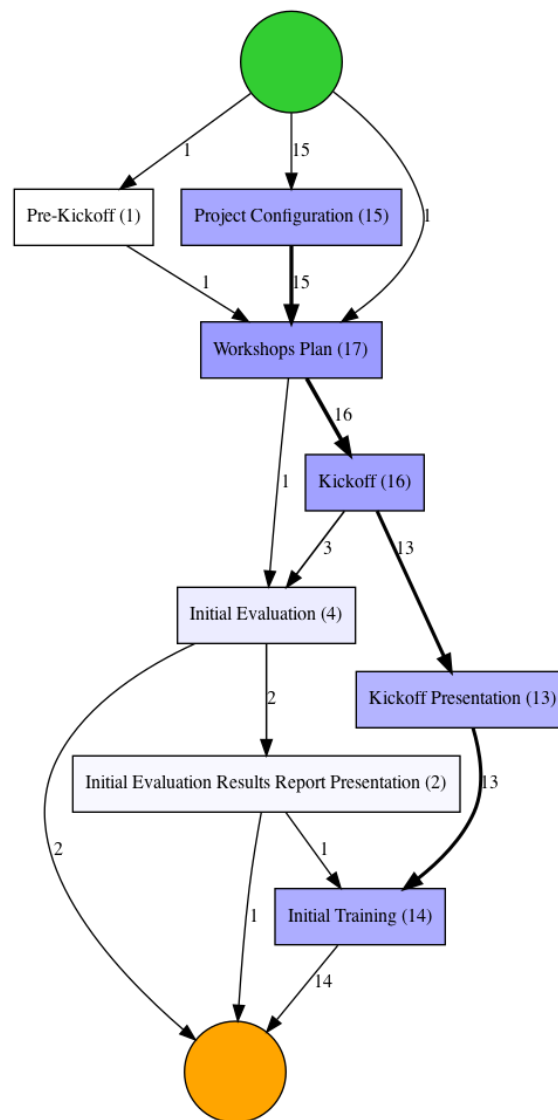


Figure 4.6: Directly-Follows Graph model of Preparation and Planning phase.

The most frequent sequence includes the following activities:

- Project Configuration
- Workshops Plan
- Kickoff
- Kickoff Presentation
- Initial Training

The support value for this pattern is 0.75. Figure 4.6 shows that these five activities are the most repeated and follow a consistent sequence.

4.2.2.2 Execution and Implementation

The second phase of GDPR projects can be seen as a big phase that includes Initial Gap Analysis, Workshops, Process Monitoring, and Final Evaluation. Considering the issues that phases like Workshops and Process Monitoring presented regarding the big variation of activities, the model obtained here is also very inconsistent. Thus, the process model is not presented in this section but in the appendix (Figure A.3).

An interesting pattern is obtained applying the CM-SPAM algorithm, this pattern includes the following activities:

- Initial Evaluation
- Processes Training
- Projects and Process Monitoring
- Final Evaluation Plan

The pattern has a considerable 0.45 support value. Nevertheless, it could not be recommended as a process of this phase includes more activities like workshops and process monitoring meetings.

4.2.2.3 Project Closure

The results of this phase are shown in subsection 4.2.1.7.

4.3 Analysis of Quality Metrics

4.3.1 Fitness

The replay fitness measure used to evaluate the process models in this project implements a token-based technique where a value between 0 and 1 represents the percentage of traces of the event log that can be replayed in the process model. This PM4Py implementation of this quality metric is detailed in [12] and the results of its application for the different process models are presented below.

Figure 4.7 shows that the values of fitness for all nine models were higher than 0.93. This leads to the conclusion that all the models could replay the behavior of the log very accurately.

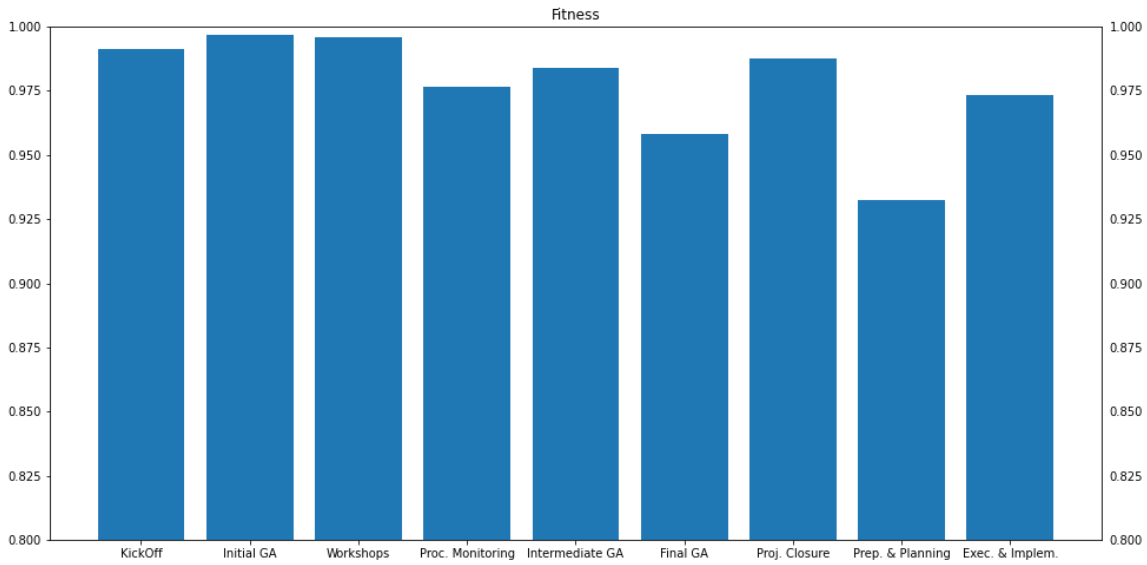


Figure 4.7: Fitness for all process models.

4.3.2 Precision

Precision measure defines the extent of events that are in the process model but not in the log, this metric deals with the underfitting problem. For the current case, a token-based approach was implemented using PM4Py respective function that uses the ideas of [41].

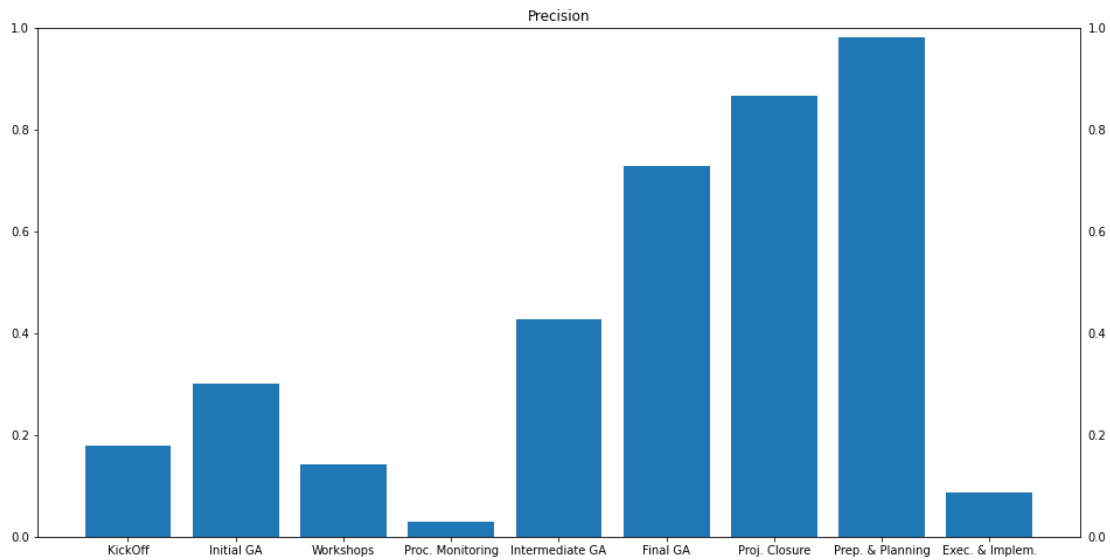


Figure 4.8: Precision for all process models.

Figure 4.8 presents precision values for all the phases with a lot of variation among them. Phases like Workshops, Process Monitoring, and Execution and Implementation were predictable to have such low values as the result of their process models did not meet expectations because of the wide diversity of activities. Thus, a common process could not be identified. There are other phases like Kickoff, Initial Gap Analysis, and Intermediate Gap Analysis that show precision

values lower than expected. This can be understood due to the merge of different CMMI and ISO 27001 phases to obtain each of these phases. For instance, the Kickoff phase merged some Pre-Kickoff and Kickoff phases of CMMI and ISO 27001 projects in a generic Kickoff phase.

4.3.3 Simplicity

The simplicity dimension describes how simple a model is to understand. This measure is not related to the event log and just focuses on the process model. The technique used for its implementation in PM4Py is based on the inverse arc degree principle which is described in [14].

All the values for simplicity are between 0.6 and 0.8 having some relationship between precision values as the process models with more precision, Final Gap Analysis, Project Closure, and Preparation and Planning, are the simplest process models. The same occurs for the less precise models, Workshops, Process Monitoring, and Execution and Implementation, with the worst values for simplicity as it can be seen in Figure 4.9.

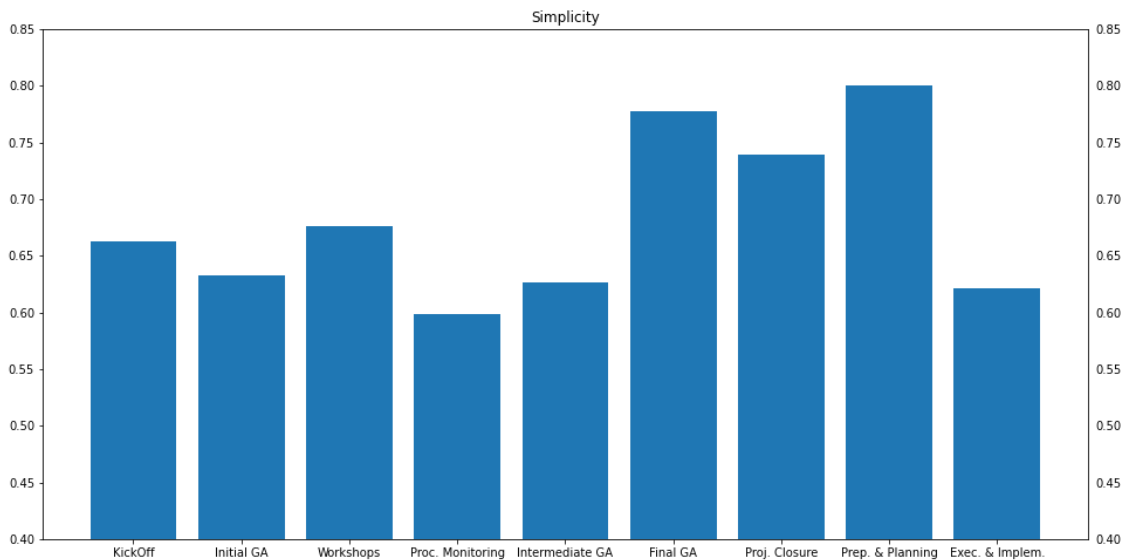


Figure 4.9: Simplicity for all process models.

4.3.4 Generalization

The generalization measure presents as well a relation between the behavior of the event log and the process model but this time evaluating the overfitting problem. The event log contains some traces of events that have already happened. Nevertheless, other traces can occur and unseen behavior should also be considered to not overfit the model with just actual data.

The PM4Py implementation of generalization is based on [16] and specifies that a model is general where the elements are visited often on a replay operation.

Figure 4.10 presents the results of the generalization dimension for all project phases. The lowest values are for Workshops and Process Monitoring phases and the highest values for Project Closure and Final Gap Analysis. These two phases have the best results for all quality metrics,

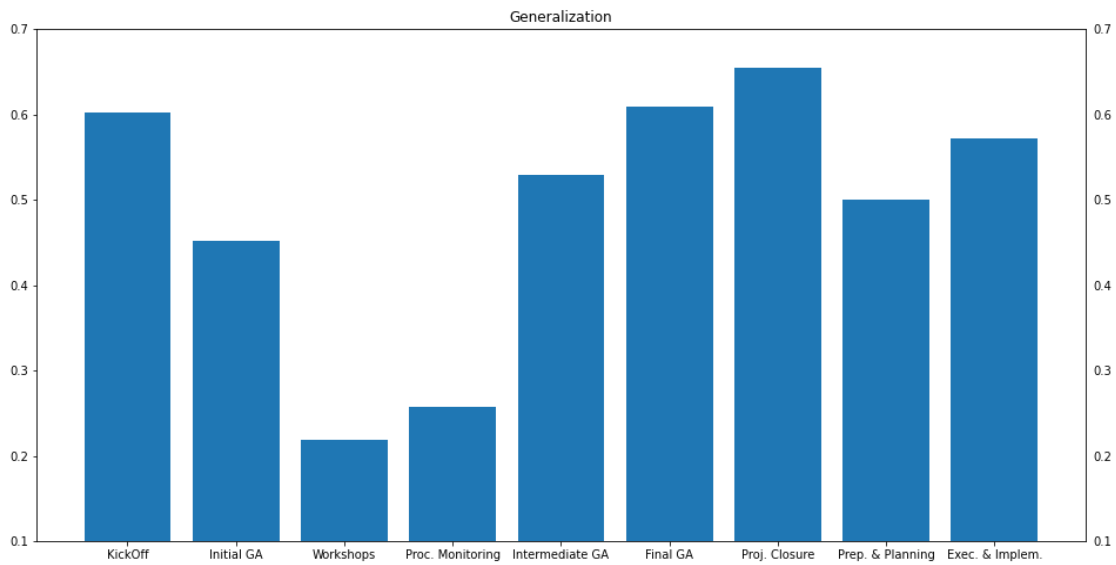


Figure 4.10: Generalization for all process models.

Preparation and Planning phase also has good results but the generalization value is not as good as Project Closure and Final Gap Analysis generalization values.

4.4 Overall Evaluation

This section shows two tables to visualize process mining and sequential pattern mining metrics for all phases and presents a final evaluation.

Table 4.1 shows the values of places, transitions, and arcs as well as the four quality metrics for all Petri nets. The models with more arcs tend to be those with less precision, e.g., Process Monitoring phase; while the three models with fewer arcs, Final Gap Analysis, Project Closure, and Preparation and Planning, are the models with the best precision and simplicity values.

Phase	Places	Transitions	Arcs	Fitness	Precision	Simplicity	Generalization
Kickoff	24	35	74	0.991	0.18	0.66	0.60
Initial G.A.	37	56	120	0.997	0.30	0.63	0.45
Workshops	170	235	502	0.996	0.14	0.68	0.22
Proc. Monitoring	320	405	968	0.976	0.03	0.60	0.26
Intermediate G.A.	17	20	48	0.984	0.43	0.63	0.53
Final G.A.	6	8	16	0.958	0.73	0.78	0.61
Project Closure	7	10	20	0.987	0.87	0.74	0.65
Prep. & Planning	7	9	18	0.932	0.98	0.80	0.50
Exec. & Impl.	57	81	180	0.973	0.09	0.62	0.57

Table 4.1: Petri nets properties and evaluations

The best model according to fitness dimension is the one for Initial Gap Analysis with 0.997. For precision metric, the best value is 0.98, which corresponds to the Preparation and Planning

phase. This phase model also has the highest value for simplicity which is 0.80. Regarding generalization, the best-assessed model is Project Closure with 0.65.

Table 4.2 presents a summary of the support and pattern length values for each pattern. The Workshops, Process Monitoring, and Execution and Implementation patterns are not recommended and not presented in the table.

Phase	Support	Pattern Length
Kickoff	0.7	2
Initial G.A.	0.4	4
Intermediate G.A.	0.6	4
Final G.A.	0.6	4
Project Closure	0.7	2
Prep. & Planning	0.75	5

Table 4.2: Metrics of recommended sequential patterns

Considering the other 6 phases, the fewer values for pattern length belong to Kickoff, and Project Closure with just two activities. On the other hand, Preparation and Planning phase has the longest pattern. Regarding support values, the highest support among all the recommended values is Preparation and Planning with 0.75, while the lowest value corresponds to Initial Gap Analysis with 0.4. Sequential pattern mining technique could confirm and support the results obtained using the process mining approach, however, process mining implementation showed a more complete outcome for the current work.

Chapter 5

Conclusions and Future Work

This chapter gathers the conclusions reached by applying process mining and sequential pattern mining techniques to extract, update and optimize certification projects of Strongstep company. It also presents some recommendations and future work to be made with the insights of the current project.

5.1 Conclusions

The current use of process mining procedures demonstrated that this family of techniques has a strong foundation to deal with process inconsistencies. Starting with process discovery implementation, several process models were built for each project phase giving a visual representation of how the activities were related among themselves. For three of the nine phases, Workshops, Process Monitoring, and Execution and Implementation, the results did not meet expectations because of the nature of the data. A lot of different activities were found, therefore, finding a common process for all instances was not possible. However, process mining could reach good results for the other six phases where data was appropriate, and the best results were obtained where data used for building models did not come from merged data. This merge operation was executed at the data preparation phase due to the data scarcity problem that appeared after collecting the data.

PM4Py library showed that is a complete tool for process mining in Python and has several functions regarding process discovery and conformance checking operations, including working with event logs, building process models, and applying conformance checking techniques to evaluate the models. The miner used for process discovery technique was Inductive Miner infrequent and the models were visualized in Petri nets and Directly-Follows Graph. Finally, the models were assessed with quality metrics such as fitness, precision, simplicity, and generalization.

Sequential pattern mining outcome was the most frequent pattern for a phase applying the CM-SPAM algorithm. Some cleaning tasks need to be done to generate the sequential nature as it is supposed to occur in the real processes. The six phases that generated consistent process models, Kickoff, Initial Gap Analysis, Intermediate Gap Analysis, Final Gap Analysis, Project Closure, and Preparation and Planning, were the phases that for this approach obtained high-grade

results as well. Even this technique is specifically oriented to sequential processes, its application could support the decision of generating standard phase processes for future CMMI, ISO 27001, and GDPR projects of Strongstep.

The SPMF tool was very helpful to implement several algorithms and compare them to see which one was the best to use. However, it still needs to work on the casting of data. The tool comes with a function to transform CSV format to SPMF format, but for this project, it was not feasible to use it. Some preprocess functions were manually implemented to work with this tool. The Python wrapper of SPMF helped to maintain all the work in one single environment: the Jupyter Lab framework.

5.2 Future Work

The main difficulty found in this dissertation project was the lack of data quality as some activities have their description in different languages and some old data need to be used to avoid data scarcity, this led to the existence of more variety of activities, thus, more inconsistency in data. The outcome of a data mining model is always related to the quality of the available data, the same occurs for process mining. Some solutions that deal with data inconsistency and particularly with text anomalies like record linkage or entity matching, some research can be reviewed in [33] and [39]. Working with inconsistencies happens often when data come from real projects due to the human errors that occur when data registers are created in their respective management systems. Hence, this is now an important investigation field.

The stakeholders can also take care of how they are logging information and put special attention to the registration of activities in their project management tools. One of the main problems with the data was the lack of good timestamps, sometimes activities that were happening at different moments were logged as if they were occurring at the same time. The format of the time variable should be replaced from yyyy-mm-dd to yyyy-mm-dd-hh-mm because some activities are occurring the same day but at a different time, and as the format just admit a day format, the activities are represented as happening in a parallel way when that is not the case. Solving these issues will help to apply a more accurate sequential pattern mining implementation as well as an optimization task to define the exact time between activities and find bottlenecks or deadlocks. Another project focusing on time or human resources perspectives can be done, now that the current results include the most frequent sequential processes and process models for each project phase regarding CMMI, ISO 27001, and GDPR projects.

Appendix A

Inconsistent Directly-Follows Graphs

Some Directly-Follows Graphs are presented to notice the inconsistent outcome of their respective phases. The intention is to demonstrate the large number of activities within Workshops, Process Monitoring, and Execution and Implementation phases. Having so many bifurcations makes it impossible to recommend a standard process for these phases.

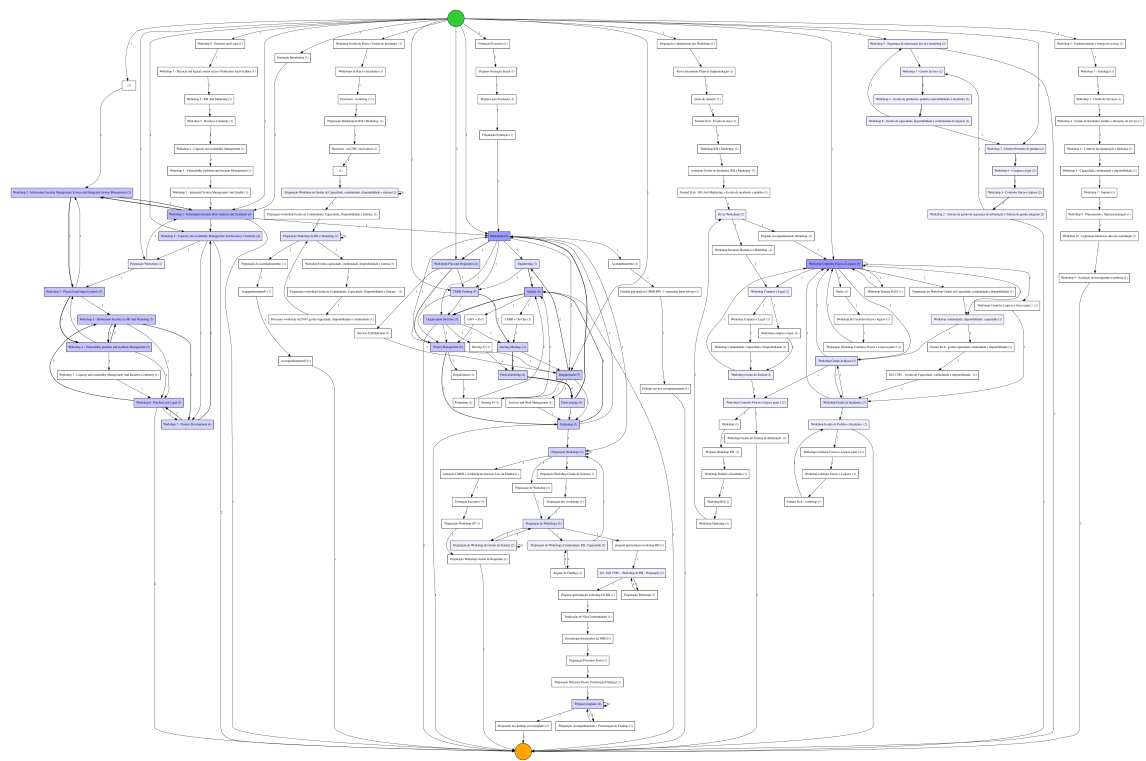


Figure A.1: Directly-Follows Graph model of Workshops phase.

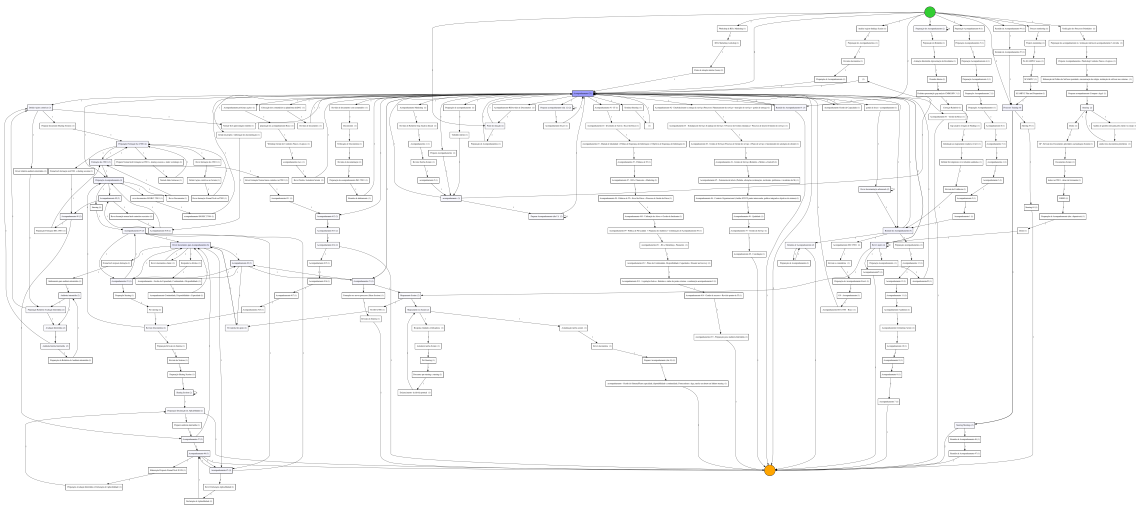


Figure A.2: Directly-Follows Graph model of Process Monitoring phase.

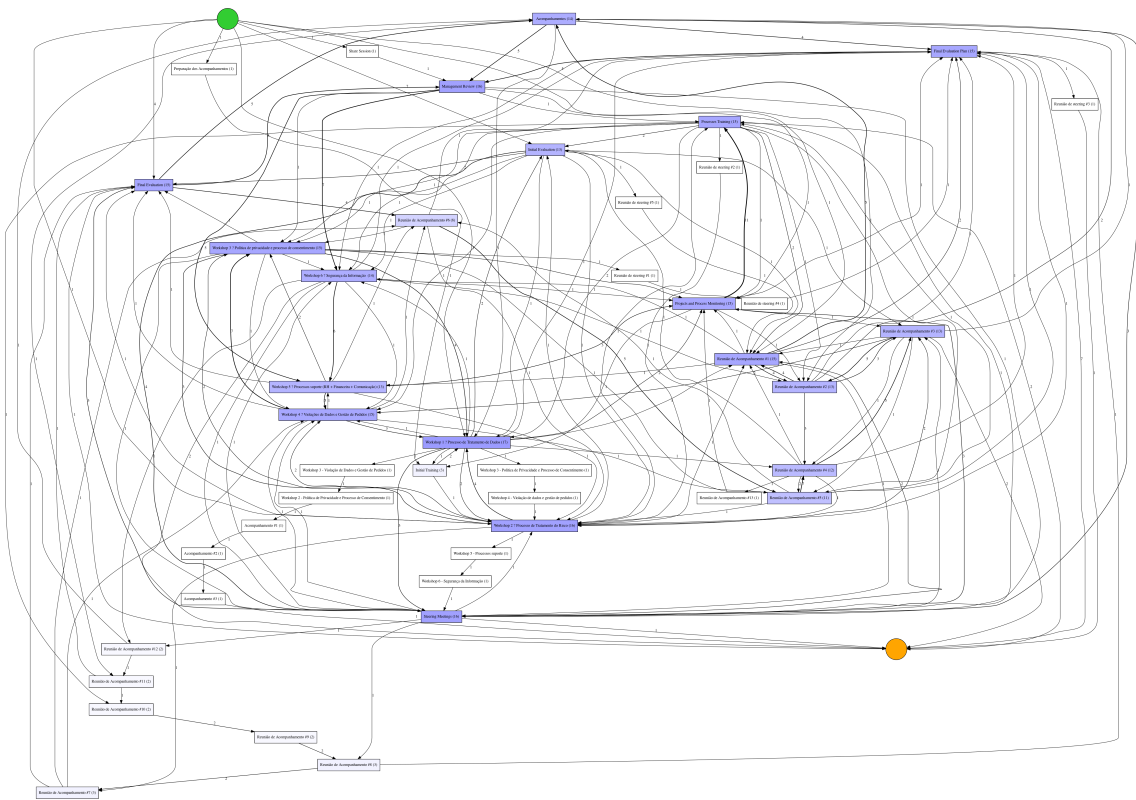


Figure A.3: Directly-Follows Graph model of Execution and Implementation phase.

References

- [1] The value of business process optimization: Are your processes working with or against you?, 2020. <https://www.d-wise.com/blog/the-value-of-business-process-optimization>. Accessed: 2021-01-03.
- [2] Process discovery documentation of pm4py - process mining for python, 2021. <https://pm4py.fit.fraunhofer.de/documentation#discovery>. Accessed: 2021-06-14.
- [3] W. M.P. Van Der Aalst. The application of petri nets to workflow management. *Journal of Circuits, Systems and Computers*, 8, 1998.
- [4] W. M.P. Van Der Aalst, M. H. Schonenberg, and M. Song. Time prediction based on process mining. *Information Systems*, 36, 2011.
- [5] Wil M.P. Van Der Aalst. A practitioner’s guide to process mining: Limitations of the directly-follows graph. *Procedia Computer Science*, 164, 2019.
- [6] Wil Van Der Aalst. *Process mining: Data science in action*. 2016.
- [7] Wil Van Der Aalst, Arya Adriansyah, and Boudewijn Van Dongen. Causal nets: A modeling language tailored towards process discovery. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6901 LNCS, 2011.
- [8] A. Adriansyah, J. Munoz-Gama, J. Carmona, B. F. van Dongen, and W. M.P. van der Aalst. Measuring precision of modeled behavior. *Information Systems and e-Business Management*, 13, 2015.
- [9] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. *Proc. of 20th International Conference on Very Large Data Bases, {VLDB’94}*, 1994.
- [10] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. *Proceedings - International Conference on Data Engineering*, 1995.
- [11] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. Sequential pattern mining using a bitmap representation. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.
- [12] Alessandro Berti and Wil Van Der Aalst. Reviving token-based replay: Increasing speed while improving diagnostics. *CEUR Workshop Proceedings*, 2371, 2019.
- [13] Alessandro Berti, Sebastiaan J. Van Zelst, Wil M.P. Van Der Aalst, and Fraunhofer Gesellschaft. Process mining for python (pm4py): Bridging the gap between process- and data science. *CEUR Workshop Proceedings*, 2374:13–16, 2019.

- [14] Fabian Rojas Blum. Metrics in process discovery. *Technical Report TR/DCC*, 2015.
- [15] Carmen Bratosin, Natalia Sidorova, and Wil Van Der Aalst. Distributed genetic process mining. *2010 IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010*, 2010.
- [16] J. C.A.M. Buijs, B. F. Van Dongen, and W. M.P. Van Der Aalst. Quality dimensions in process discovery: The importance of fitness, precision, generalization and simplicity. *International Journal of Cooperative Information Systems*, 23, 2014.
- [17] María Chamorro and Sergio Maturana. Method for applying process mining to the distribution of non-alcoholic beverages. *Proceedings - International Conference of the Chilean Computer Science Society, SCCC*, 0, 2013.
- [18] Jonathan E. Cook and Alexander L. Wolf. Discovering models of software processes from event-based data. *ACM Transactions on Software Engineering and Methodology*, 7, 1998.
- [19] W. M.P. Van der Aalst and A. J.M.M. Weijters. Process mining: A research agenda. *Computers in Industry*, 53, 2004.
- [20] M. Dramski. Extensible event stream format for navigational data. *Zeszyty Naukowe Akademii Morskiej w Szczecinie*, 2016.
- [21] Sebastian Dunzer, Matthias Stierle, Martin Matzner, and Stephan Baier. Conformance checking: A state-of-the-art literature review. *ACM International Conference Proceeding Series*, 2019.
- [22] Philippe Fournier-Viger, Jerry Chun, Wei Lin, Rage Uday Kiran, Yun Sing Koh, and Rincy Thomas. A survey of sequential pattern mining. *Data Science and Pattern Recognition*, 1:54–77, 2017.
- [23] Philippe Fournier-Viger, Antonio Gomariz, Manuel Campos, and Rincy Thomas. Fast vertical mining of sequential patterns using co-occurrence information. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8443 LNAI, 2014.
- [24] Philippe Fournier-Viger, Antonio Gomariz, Ted Gueniche, Azadeh Soltani, Cheng Wei Wu, and Vincent S. Tseng. Spmf: A java open-source pattern mining library. *Journal of Machine Learning Research*, 15, 2015.
- [25] Philippe Fournier-Viger, Antonio Gomariz, Michal Šebek, and Martin Hlosta. Vgen: Fast vertical mining of sequential generator patterns. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8646 LNCS, 2014.
- [26] Philippe Fournier-Viger, Cheng Wei Wu, Antonio Gomariz, and Vincent S. Tseng. Vmsp: Efficient vertical mining of maximal sequential patterns. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8436 LNAI, 2014.
- [27] Philippe Fournier-Viger, Cheng Wei Wu, and Vincent S. Tseng. Mining maximal sequential patterns without candidate maintenance. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8346 LNAI, 2013.

- [28] Fabio Fumarola, Pasqua Fabiana Lanotte, Michelangelo Ceci, and Donato Malerba. Clofast: closed sequential pattern mining using sparse and vertical id-lists. *Knowledge and Information Systems*, 48, 2016.
- [29] Chuancong Gao, Jianyong Wang, Yukai He, and Lizhu Zhou. Efficient mining of frequent sequence generators. *Proceeding of the 17th International Conference on World Wide Web 2008, WWW'08*, 2008.
- [30] Christian W. Günther and Wil M.P. Van Der Aalst. Fuzzy mining - adaptive process simplification based on multi-perspective metrics. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4714 LNCS, 2007.
- [31] Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8, 2004.
- [32] B. Chaitanya Krishna, A. Yeshwanth Srinath, N. Bhavani, and G. Jaya Sai. Analysing software quality using cmmi-2 with agile-scrum framework. *International Journal of Engineering and Technology(UAE)*, 7, 2018.
- [33] Hanna Köpcke and Erhard Rahm. Frameworks for entity matching: A comparison. *Data and Knowledge Engineering*, 69, 2010.
- [34] Sander J.J. Leemans, Dirk Fahland, and Wil M.P. Van Der Aalst. Discovering block-structured process models from event logs - a constructive approach. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7927 LNCS, 2013.
- [35] Sander J.J. Leemans, Dirk Fahland, and Wil M.P. Van Der Aalst. Scalable process discovery with guarantees. *Lecture Notes in Business Information Processing*, 214, 2015.
- [36] Sander J.J. Leemans, Dirk Fahland, and Wil M.P. van der Aalst. Discovering block-structured process models from event logs containing infrequent behaviour. *Lecture Notes in Business Information Processing*, 171 171 LNBIP, 2014.
- [37] He Li, Lu Yu, and Wu He. The impact of gdpr on global technology development. *Journal of Global Information Technology Management*, 22, 2019.
- [38] R. S. Mans, M. H. Schonenberg, M. Song, W. M.P. Van Der Aalst, and P. J.M. Bakker. Application of process mining in healthcare - a case study in a dutch hospital. *Communications in Computer and Information Science*, 25 CCIS, 2008.
- [39] Matthew Michelson and Craig A. Knoblock. Learning blocking schemes for record linkage. *Proceedings of the National Conference on Artificial Intelligence*, 1, 2006.
- [40] Tadao Murata. Petri nets : Properties , analysis and applications. *Proceedings of the IEEE*, 77, 2015.
- [41] Jorge Muñoz-Gama and Josep Carmona. A fresh look at precision in process conformance. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6336 LNCS, 2010.

- [42] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei Chun Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on Knowledge and Data Engineering*, 16, 2004.
- [43] Chapman Pete, Clinton Julian, Kerber Randy, Khabaza Thomas, Reinartz Thomas, Shearer Colin, and Rudiger Wirth. Crisp-dm 1.0. *CRISP-DM Consortium*, 2000.
- [44] Amogh Phirke and Jayshree Ghorpade-Aher. Best practices of auditing in an organization using iso 27001 standard. *International Journal of Recent Technology and Engineering*, 8, 2019.
- [45] Jan Recker, Michael Rosemann, Marta Indulska, and Peter Green. Business process modeling- a comparative analysis. *Journal of the Association for Information Systems*, 10, 2009.
- [46] Marcos Hernán Rivas and Sussy Bayona-Oré. Process mining algorithms for automated process discovery. *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, 2019, 2019.
- [47] Mohammadreza Fani Sani, Wil Van Der Aalst, and Sebastiaan J. Van Zelst. Improving the performance of process discovery algorithms by instance selection. *Computer Science and Information Systems*, 17, 2020.
- [48] Stefan Schönig and Stefan Jablonski. Comparing declarative process modelling languages from the organisational perspective. *Lecture Notes in Business Information Processing*, 256, 2016.
- [49] Soraya Sortino, Matteo Dell’Orto, Giorgio Cotroneo, Roberto Carbone, and Paolo Riccardi. Process mining: A database of applications. page 103, 2020.
- [50] Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1057 LNCS, 1996.
- [51] S. Suriadi, R. Andrews, A. H.M. ter Hofstede, and M. T. Wynn. Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs. *Information Systems*, 64, 2017.
- [52] Malte Thiede, Daniel Fuerstenau, and Ana Paula Bezerra Barquet. How is process mining technology used by organizations? a systematic literature review of empirical studies. *Business Process Management Journal*, 24, 2018.
- [53] W. M.P. van der Aalst, H. A. Reijers, A. J.M.M. Weijters, B. F. van Dongen, A. K. Alves de Medeiros, M. Song, and H. M.W. Verbeek. Business process mining: An industrial application. *Information Systems*, 32, 2007.
- [54] W.M.P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, volume 136. 2011.
- [55] Jianyong Wang, Jiawei Han, and Chun Li. Frequent closed sequence mining without candidate maintenance. *IEEE Transactions on Knowledge and Data Engineering*, 19, 2007.

- [56] A. J.M.M. Weijters and J. T.S. Ribeiro. Flexible heuristics miner (fhm). *IEEE SSCI 2011: Symposium Series on Computational Intelligence - CIDM 2011: 2011 IEEE Symposium on Computational Intelligence and Data Mining*, 2011.
- [57] A.J.M.M. Weijters, W M P van der Aalst, and A K Alves de Medeiros;. Process mining with the heuristicsminer algorithm. *Beta working papers*, 2006.
- [58] Xifeng Yan, Jiawei Han, and Ramin Afshar. Clospan: Mining: Closed sequential patterns in large datasets. *SIAM International Conference on Data Mining*, 2003.
- [59] Fitri Almira Yasmin, Faiza Allah Bukhsh, and Patricio De Alencar Silva. Process enhancement in process mining: A literature review. *CEUR Workshop Proceedings*, 2270:65–72, 2018.
- [60] Shengwei Yi, Tianheng Zhao, Yuanyuan Zhang, Shilong Ma, and Zhanbin Che. An effective algorithm for mining sequential generators. *Procedia Engineering*, 15, 2011.
- [61] Mohammed J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42, 2001.