

Article

Applying Process Mining: The Reality of a Software Development SME

Silvia Jaqueline Urrea-Contreras ^{1,*}, Maria Angelica Astorga-Vargas ², Brenda L. Flores-Rios ¹ , Jorge Eduardo Ibarra-Esquer ² , Felix F. Gonzalez-Navarro ¹ , Ivan Garcia Pacheco ³  and Carla Leninca Pacheco Agüero ³

¹ Instituto de Ingeniería, Universidad Autónoma de Baja California, Mexicali 21280, Mexico; brenda.flores@uabc.edu.mx (B.L.F.-R.); fernando.gonzalez@uabc.edu.mx (F.F.G.-N.)

² Facultad de Ingeniería, Universidad Autónoma de Baja California, Mexicali 21280, Mexico; angelicaastorga@uabc.edu.mx (M.A.A.-V.); jorge.ibarra@uabc.edu.mx (J.E.I.-E.)

³ División de Estudios de Posgrado, Universidad Tecnológica de la Mixteca, Oaxaca 69000, Mexico; ivan@mixteco.utm.mx (I.G.P.); leninca@mixteco.utm.mx (C.L.P.A.)

* Correspondence: jaqueline.urrea@uabc.edu.mx

Abstract: One of the challenges the organizations confront is to extract data from the information systems to know the reality of their processes to improve their efficiency. In this study, the application of Process Mining is addressed as an opportunity in the specific context of an SME dedicated to software development, implementing the L* life cycle model methodology from a layered Software Engineering approach. This research is carried out based on process improvement in an initial SME project. Subsequently, it is compared with a second project, using different Process Mining perspectives such as control flow, case, organization, and time, with the aim of extending the process model. This holistic view allows not only to better understand the processes involved, but also to identify and analyze the similarities and differences between the two projects. As a result, the Process Mining analysis shows crucial aspects such as the representation of integrated models, traces on sequences of actions, and the interaction of activities with specific roles and deviations in the flow of activities that compromise the quality of the process and the product. At the same time, the challenges that emerged during the improvement cycle are highlighted. These challenges cover issues such as data extraction, fluid communication between those involved, and the documentation associated with the processes. This study contributes to the body of knowledge of Process Mining. Likewise, the case study results offer a vision for other SMEs seeking to incorporate Process Mining as part of their improvement strategies.

Keywords: process mining; small and medium-sized enterprises; software development processes



Citation: Urrea-Contreras, S.J.; Astorga-Vargas, M.A.; Flores-Rios, B.L.; Ibarra-Esquer, J.E.; Gonzalez-Navarro, F.F.; Garcia Pacheco, I.; Pacheco Agüero, C.L. Applying Process Mining: The Reality of a Software Development SME. *Appl. Sci.* **2024**, *14*, 1402. <https://doi.org/10.3390/app14041402>

Academic Editor: Andrea Prati

Received: 18 October 2023

Revised: 21 November 2023

Accepted: 27 November 2023

Published: 8 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The successful implementation of Process Mining in large companies has motivated small and medium-sized enterprises (SMEs) to make efforts to extract data from their information systems in order to compare their documented processes with the processes observed by mining techniques [1] and expertise in process management. In the software industry, SMEs represent more than 80% of the workforce [2] and have specific characteristics depending on the capability level of their processes and the integration of their human, material, and technological resources. Software development SMEs face cost and tight delivery deadlines projects [3], the need to use various specific tools (such as management systems) to carry out the software development process [4], and their adaptation to a dynamic business environment [5,6]. For them, this reality represents a significant challenge in the implementation of Process Mining [5,7].

Process Mining, as a growing discipline of data science, aims to extract event data recorded in information systems [8–11]. Most of the data is scattered across various systems,

without an explicit relationship between them and the executed process activities [12], resulting in unstructured information. Software process management systems, such as the Issue Tracking System (Jira) and the Version Control System (Github), require additional practices to link their stored data. In the absence of an explicit link between management systems, the available data will need efforts for extraction and structuring through tools or Application Programming Interfaces (APIs).

Data extraction can be applied after understanding the data structure and system design [13]. The activities Extraction, Transformation, and Load (ETL) of event logs are the first step to applying Process Mining [8,11,13]. The extracted data are transformed into a standard data format and loaded into a system or database [11] as event logs. A software process event log is used in Process Mining to describe the information of events and attributes [14], which can vary depending on the type of process management system. Some attributes of software processes may include the identifier and name of the activity (requirements, architecture, design, implementation, testing, operation, etc.), the description, the date and time it was recorded, the author (the developer who recorded it), the status (open, close, in progress, etc.), the priority, and the assignment to a specific team or person. This set of attributes can be used to perform three types of Process Mining: discovery, conformance, and enhancement [11].

Discovery allows obtaining real process models from the event log with the application of algorithms; for example, Inductive Miner, Heuristics Miner, and Fuzzy Miner [11]. Conformance compares the event log and the process model (documented or discovered) through conformance-checking techniques (rule checking, token replay, and alignments) that produce an explicit description of the consistent and deviated parts of the process [15]. By linking an event log and a process model through a conformance technique, the understanding of the process can be improved, including techniques for enhancement [16]. Process enhancement aims for models to be accurate and have high fitness. Generally, the literature on enhancement mining is divided into two types: process extension and process improvement [11,16].

Process extension focuses on accuracy and aims to incorporate different perspectives such as data (case), resource (organizational), and time perspectives, based on the presence of attributes associated with events [11,16]. Their inclusion in process models allows for refining the model specifications, thus obtaining extended models with higher levels of accuracy [16]. Regarding process improvement, it seeks process models (i) to better reflect reality, and/or (ii) to only allow executions that are valid from a domain viewpoint, and/or are correlated to better performances [16].

Process improvement activities in the software SME industry are mainly based on developers' perceptions and little support is given to make process-wise data-driven decisions [4]. Mining the data generated by the execution of a process's activities allows for understanding the reality of an SME and making suggestions for process enhancement. These improvements can prevent the software organization from experiencing inefficiencies, deviations, and risks in its process model [13].

The objective of this study is to compare the findings of a first improvement cycle [17] from the Process Mining of a second project in a software development SME. For this purpose, event logs were analyzed by applying Process Mining techniques and perspectives. The similarities and differences between Project 1 and Project 2 were analyzed, based on the representation of their respective integrated models, the traces, and the interaction of the activities with the roles. On the other hand, both experiences allowed the identification of the challenges that arise in Process Mining in a software development SME. For this reason, this study is considered a contribution to the software SME industry seeking to incorporate Process Mining as part of their improvement strategies.

This paper is structured as follows: Section 2 presents the background related to Process Mining in software development and SMEs; Section 3 details the Process Mining methodology L* from a layered Software Engineering approach; Section 4 expounds on the implementation of the methodology L* with the characterization of the software de-

velopment SME case study; Section 5 shows the results of the extended model of Project 2 with Process Mining perspectives. Section 6 presents a discussion. Finally, conclusions and future work are presented in Section 7.

2. Related Work

The application of Process Mining in the context of the SMEs is particularly relevant. Wijnhoven et al. [1] propose the Workaround Identification, Classification, and Evaluation (WICE) method for the organizational impact of workarounds in SMEs using Process Mining. WICE was implemented in a case study of a European SME where some workaround indicators were identified by applying the perspectives of control flow (single event, deadlocks, lacking or low number of event logs for an activity, activity or decision not mentioned in design), organizational (resources not present, resources not mentioned in the process design, resource over- or under-utilization, resources not used for assigned activity), and time (process time-out, fast/slow processing time, short/long response time, fast successive event inflow/outflow). Wijnhoven et al. [1] exposed the difficulty of defining the process due to the lack of documentation related to the processes and that not all of the process was implemented in a system; this is a common problem among SMEs. Additionally, it is mentioned that Process Mining represents a complexity for SMEs because its implementation may seem more feasible due to their relatively small and manageable processes, but formal process designs related to the process may not be profitable for SMEs. Finally, it is mentioned that the extraction of data by Process Mining is limited by the records in the system.

Eggert et al. [5] present a case study on the application of Process Mining in a German information technology SME, with the aim of identifying specific challenges that SMEs face when implementing Process Mining. The results reveal 13 specific challenges for SMEs and propose seven guidelines to address them. This study contributes to the understanding of the application of Process Mining in SMEs and shows similarities and differences with larger companies. The study highlights the importance of considering the specific characteristics of SMEs and provides practical recommendations to address the identified challenges.

In [18], an approach is presented to improve Software Development Methodologies (SDM) activities by combining two sources of data: stakeholders' perceptions of SDM and the application of Process Mining with PM2 methodology to records stored in software development tools. The approach was evaluated through a case study in an Austrian software development SME that uses Scrum as an agile methodology and Jira as a task-tracking system for its developers. Jira is an issue-tracking and project management tool widely used in the software industry. It was specified that data collection involved interviews with management, direct observation of their workday, surveys of all developers working on the observed project, and collection of corresponding Jira event records. They also note that due to the variability of software development tool records, it is not possible to define a step-by-step procedure for analyzing them.

On the other hand, Vidoni [19] conducted a systematic literature review on Mining Software Repositories (MSR) where 146 studies were identified. It was found that MSR studies often do not follow a systematic approach for repository selection, and many do not report on selection or data extraction protocols. It was determined that the selection of a repository is determined by the type of data that needs to be collected; for example, error data will require selecting repositories such as Jira or Bugzilla, while version control data could be directed to GitHub, Azure or BitBucket. Additionally, the paper mentions limited API data as a peril.

Further, the use of process mining software tools is important; in [20], a comparative analysis methodology was employed for a supply chain SME in order to evaluate five Process Mining software tools (Apromore Community Edition, ProM, Celonis, myInvenio, and Disco) through eleven specific criteria to identify the tool that best suits the needs of the SME. The methodology offers three different approaches for comparison: ontology, decision tree, and Analytic Hierarchy Process (AHP). It also provides a framework that

allows users to perform comparisons between any number of Process Mining software tools, allowing the methodology to be tailored to the specific needs of each user. This integrated and flexible approach benefits informed decision making in the area of Process Mining and its application in SMEs.

The studies presented highlight the importance of considering the specific characteristics of SMEs in the implementation of Process Mining. Wijnhoven et al. [1] and Eggert et al. [5] have highlighted the complexity of implementing this discipline in SMEs. On the other hand, the application of Process Mining in software development and management systems presents important opportunities, such as the improvement in the implementation of software development methodologies and the identification of specific challenges that SMEs with Process Mining, as evidenced in the studies of Vavpotič et al. [18] and Vidoni [19].

Process Mining is a practical discipline, but its application in SMEs requires a consideration of the identified challenges (Table 1) and the adaptation of approaches to take advantage of its full potential in small and agile business environments.

Table 1. Identified challenges in the application of Process Mining in SMEs.

| Challenge | Ref. |
|---|------|
| Lack of documentation related to processes | [1] |
| Limitations in data extraction due to incomplete records in the system | [1] |
| Preparation of event log data | [5] |
| Knowledge gathering (domain) | [5] |
| Communicating the results | [5] |
| Creation of the awareness in the organization for Process Mining, its benefits, and costs | [5] |
| Shifting manpower | [5] |
| Variability of records from software development tools | [18] |
| API limitations of system records | [19] |

3. Methodology

To guide the Process Mining of Project 2, as in Project 1 [17], the L* life-cycle model methodology [11] was implemented (Figure 1) from a layered Software Engineering approach, represented by 3 phases: (1) Preprocessing, (2) Process Mining, and (3) Mining perspectives. These 3 phases are carried out through 5 steps [11]. In the Preprocessing phase, step (1) obtain an event log, where data is extracted from information systems. In the Process Mining phase; step (2) creates or discovers a process model, focusing on process discovery techniques and algorithms; and step (3) connects events in the log to activities in the model, which is essential in projecting information onto models and adding perspectives, events in the log, and model activities can be connected using the replay technique (conformance). Finally, in the Mining perspectives phase, step (4) extends the model, where Process Mining perspectives such as organizational, case, and time are integrated; and step (5) returns the integrated model, which can be used for various purposes, such as obtaining a holistic view of the process and serving as input for other tools.

To execute the methodology, in step 1, during the ETL process, APIs were implemented for data extraction and filtering. A Python script was used to obtain data from the information systems project management involved in the software development project, Jira and Github. The script enabled connection with the systems' APIs, which allowed obtaining detailed information about the process. Once the data was extracted and filtered, it was stored in the NoSQL database MongoDB to perform the necessary queries and obtain the event log. In step 2, the process model in a Petri net was discovered by applying a discovery algorithm to the event log. As proposed by L*, in step 3, the Token Replay conformance technique was applied to connect the events in the log to the activities in the discovered model and generate a validated model. Step 4 involved extending the validated model with the case, organizational, and time perspectives to better characterize the process

through the available process attributes and establish an integrated model. Finally, step 5 involved using the integrated model to create an overview of the mined process.

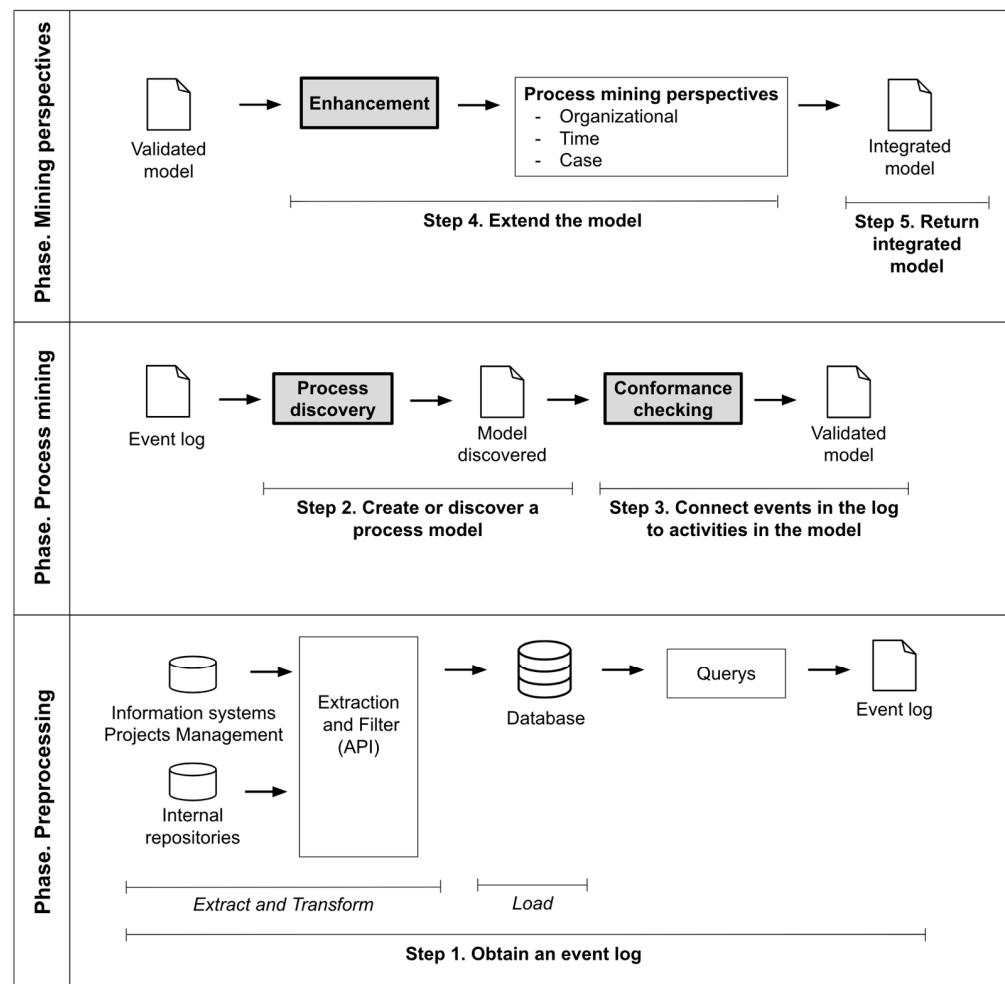


Figure 1. Application of the L* Methodology for software development projects.

4. Implementation

The motivation of this study is to demonstrate the applicability and the utility of Process Mining in the practical context of the software SME industry. This study emerges from the need presented by the case study to know the reality of its software development process and seek improvement actions in order to be competitive in the market and achieve maturity-level certifications, as well as to be the first SME with a Process Mining experience in the region.

In this section, the case study is first characterized. Then, the execution of the L* Methodology (Figure 1) for the Preprocessing and Process Mining phases is presented. The Mining perspectives phase is detailed in Section 5.

4.1. Case Study: Process Mining in a Software Development SME

The software development SME case study is located in Baja California, Mexico, established since 2011. It has a portfolio of projects at regional, national, and international level. The specific methodologies for software development and maintenance adopted by the SME are Scrum [21] and Kanban [22].

4.2. Process Mining in the Project 1

Project 1 presents the analysis of Process Mining perspectives based on the extraction of 481 events from Jira and SIMIo repositories, which are supporting tools for the software development process of an SME [17]. The traceability between these repositories was carried out manually, involving the correlation of SIMIo events with Jira issue keys. This allowed for defining important attributes to describe the events, including the issue key, issue status, timestamp, role, actor, and software development stage.

For an enhancement representation with an extension of the execution of the activities, an integrated model was made (Figure 2). The combination of control flow, organizational, case, and time perspectives has allowed us to identify trends and areas of improvement in the software development process. This comprehensive approach not only provided deep insight into the software process, but also laid the foundation for informed decision making in the first SME improvement cycle in order to compare results with a second project. Detailed information about the results is shown in Section 5.3, where the comparison between both projects is presented.

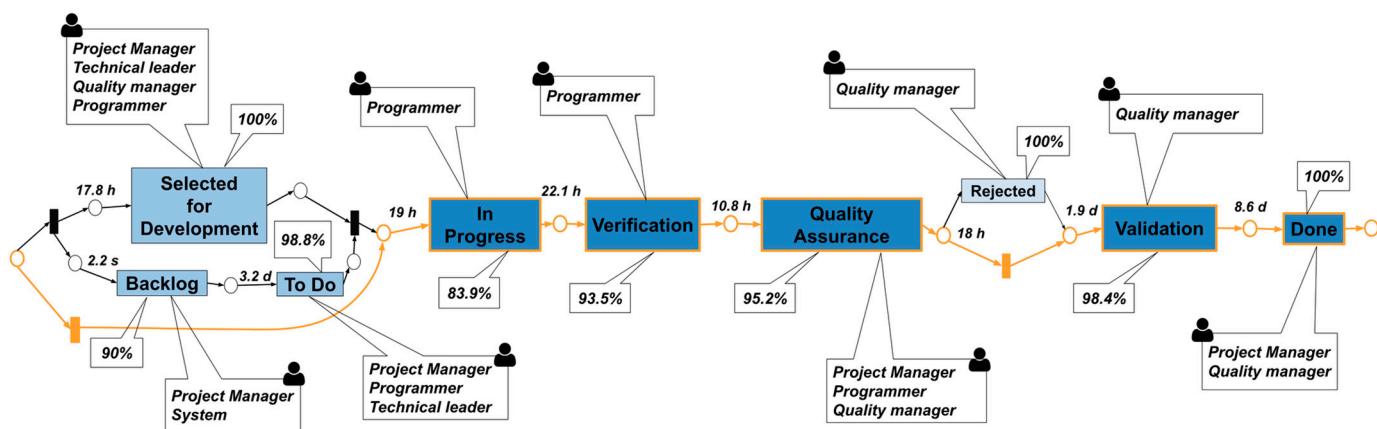


Figure 2. Project 1 integrated model.

4.3. Process Mining in the Project 2

Project 2 corresponds to a Web application for scholar services management that includes the functionalities of registration and management of affiliates, catalog of affiliations, change logs, and payroll management, among others. This project started in December 2021 and is still under development. The project team was made up of the roles of Project Manager, Technical Leader, Programmer, and Quality Manager.

For the case study, the statuses of the Kanban board presented by the project in Jira were analyzed (Table 2), in order to provide insight into the software development process. By studying and analyzing the statuses, information was obtained about how activities are performed and how tasks flow in the process. Each status represents a distinct stage in the life cycle of a task or item in the project. From creation in the Backlog to completion in the Done status, each status reflects an action or set of actions that must be taken to move forward in the SME software development process.

Table 2. Description of the Jira statuses used by the SME.

| Jira Status | Description |
|--------------------------|--|
| Backlog | Comprises a prioritized list of pending tasks in the project, used to plan and manage the future activities of the development team. |
| Selected for Development | Groups the elements of the Backlog that have been chosen to be worked on in the current sprint or iteration. |
| To Do | Contains the tasks that have not yet started and are pending to be assigned to team members for execution. |

Table 2. Cont.

| Jira Status | Description |
|-------------------|--|
| In Progress | Presents the tasks or work items that are currently being actively worked on by team members, i.e., are in the process of development. |
| Verification | Indicates tasks or work items that have been completed by the development team and are in the process of being reviewed. |
| Quality Assurance | Shows the tests and reviews of the tasks to ensure their quality. |
| Validation | Involves the tasks validated after they have gone through the development and quality control process. |
| Rejected | Contains the tasks or work items that do not meet the standards or requirements and need corrections before moving forward in the process. |
| Done | Comprises completed tasks ready for delivery and deployment. |

4.4. Preprocessing

Once the activities were identified, Preprocessing began, which corresponds to Phase 1 of the methodology. Preprocessing is necessary to transform the records from systems into a format that can be analyzed by Process Mining techniques [18]. In this way, Step 1 is to obtain an event log is presented through two sub-steps: the development of a script for ETL activities (Section 4.4.1) and obtaining an event log (Section 4.4.2).

4.4.1. ETL Activities

REST API and GraphQL were used for the extraction and filtering process, with GraphQL being chosen due to its ability to access data through queries and its flexibility in defining the required information. The query results were initially saved in a JSON format to structure the data before being finally stored in the non-relational database MongoDB. Firstly, issues from the project in Jira (Figure 3) were extracted and filtered, obtaining the issue ID, creation date, and creator information as attributes.

```
query = {
  'jql': 'project = %s' % (project_name),
  'startAt' : offset,
  'maxResults' : maxResultGetList
}
```

Figure 3. Query used to obtain the list of issues in the project.

After obtaining the list of issues related to the project, the data from the changelog field was queried to extract the Jira status field. This field contains information about changes in status for each issue, which allows for a detailed record of movements of issues over time and the ability to analyze the control flow of the development process. Filters were established through the API to obtain specifically the required status. In this way, attributes such as the ID of the issue, the predecessor state, the posterior state, the date on which the change was made, and who made the change were obtained. Once the data was filtered, it was stored together with the attributes of the first extraction related to the ID of the issue (Figure 4).

```

changelog_csv_array = []
changelog_csv_array.append
([issue["key"], "", "To Do",
issue["created"], issue["creator"]])

for change in changelog:
    for item in change["items"]:
        if(item["field"] == 'status'):
            changelog_csv_array.append
            ([issue["key"], item["fromString"],
            item["toString"], change["created"],
            change["author"]["displayName"]])
return changelog_csv_array

```

Figure 4. Filtering and linking the attributes obtained from Jira.

For the records associated with coding activities, a query was implemented to filter those issues that are related to Jira and Github (Figure 5). Attributes were obtained with the details of the commits such as the commit reference, creation date, who created it, and the URL of access to the change.

```

graphQLQuery = """query MyQuery {
  jira {
    issue(id: "ari:cloud:jira:4e503e73
      id
      devInfoDetails {
        commits {
          details {
            created
            entityUrl
            providerCommitId
            author {
              name
            }
          }
        }
      }
    }
}"""
% (issue["id"])

```

Figure 5. Query to obtain commits per issue.

4.4.2. Obtain an Event Log

After implementing the ETL activities, the information was extracted from MongoDB by defining queries to select attributes from the event log. As mentioned before, the issue status was set as an activity, allowing each activity instance to be related to a case. In this sense, 9 activities that take place during the software development process and are recorded in the information systems were identified. Each activity represents a specific task and was recorded every time it was performed during the development process. Table 3 shows the activities in the event log ordered from highest to lowest frequency. The most frequent activity was “In Progress”, followed by “Quality Assurance” and “Verification”. With the above, the event log identified a total of 4055 events in 702 cases, with the attributes ID Issue, Activity, Timestamp, Role, and Actor.

Table 3. Activities in the software project event log ordered by frequency.

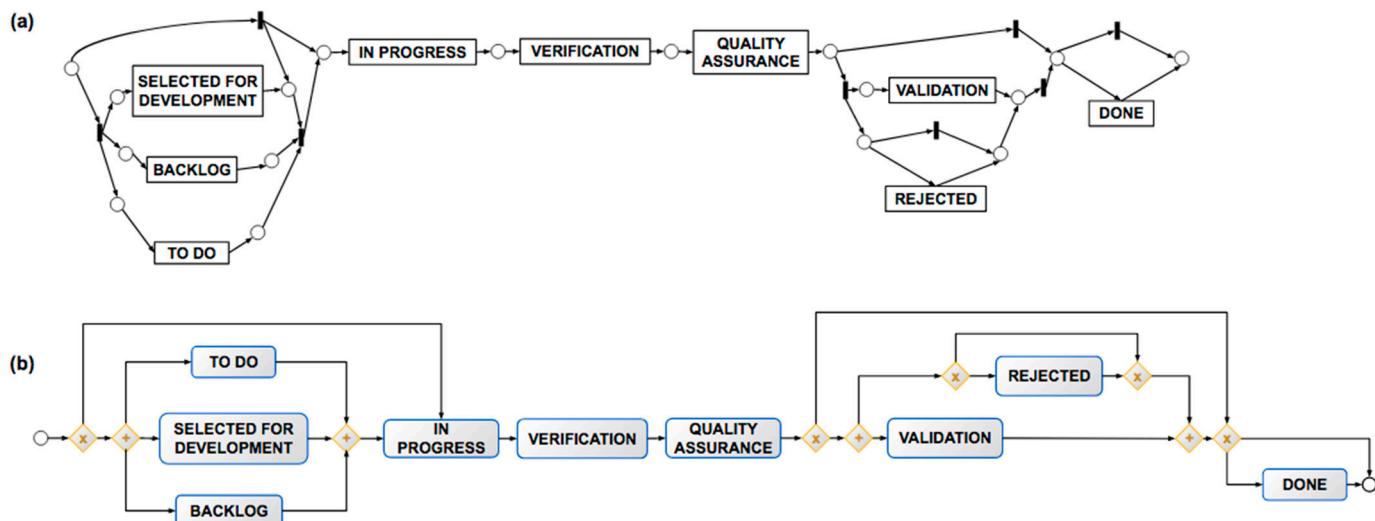
| Activity | Frequency |
|--------------------------|-----------|
| In Progress | 724 |
| Quality Assurance | 683 |
| Verification | 661 |
| Done | 441 |
| Validation | 362 |
| Selected for Development | 361 |
| To Do | 349 |
| Backlog | 275 |
| Rejected | 199 |
| Total | 4055 |

4.5. Process Mining

With the creation of the event log, the following phase is initiated, where the mining types were applied: discovery (Step 2: create or discover a process model) and conformance (Step 3: connect events in the log to activities in the model).

4.5.1. Discover a Process Model

Inductive mining was applied to generate the real process model in a visual format represented by Petri nets. The ProM tool was used to analyze the obtained event logs. Figure 6 shows the discovered model through a Petri net and its BPMN diagram with the different flows that the issue transition status presented according to the Kanban board.

**Figure 6.** Real process model of Project 2: (a) Petri net, (b) BPMN diagram.

After obtaining the process model, the traces generated from the event log were analyzed. A total of 144 traces were identified, where it was observed that the trace with the shortest length consisted of 1 event, indicating that it will begin its life cycle, while the trace with the longest length was composed of 17 events. Furthermore, it was identified that the error issue trace is the most frequent with 247 cases (Table 4), representing 35.19% of the event log. This trace is composed of activities $\sigma_1 = \langle \text{In progress}, \text{Verification}, \text{Quality Assurance}, \text{Done} \rangle$ (Figure 7). The second most common trace is related to development issues (Feature). Table 4 presents the first 12 and last 3 traces identified in the event log.

Table 4. Set of total traces present in the event log. a = Backlog, b = Selected for Development, c = To Do, d = In Progress, e = Verification, f = Quality Assurance, g = Rejected, h = Validation, i = Done.

| Trace | Cases | % of the Log |
|---|-------|--------------|
| $\sigma_1 = \langle d, e, f, i \rangle$ | 247 | 35.19 |
| $\sigma_2 = \langle a, b, c, d, e, f, h \rangle$ | 45 | 6.41 |
| $\sigma_3 = \langle a, b, c, d, e, f, g, h \rangle$ | 40 | 5.70 |
| $\sigma_4 = \langle d, e, f, h \rangle$ | 37 | 5.27 |
| $\sigma_5 = \langle a, b, c, d, e, f, g, h, i \rangle$ | 21 | 2.99 |
| $\sigma_6 = \langle a, b, c, d, e, f, h, i \rangle$ | 18 | 2.56 |
| $\sigma_7 = \langle b, c, d, e, f, h \rangle$ | 16 | 2.28 |
| $\sigma_8 = \langle a \rangle$ | 13 | 1.85 |
| $\sigma_9 = \langle d, c, d, e, f, i \rangle$ | 12 | 1.71 |
| $\sigma_{10} = \langle d, e, f, g, h \rangle$ | 12 | 1.71 |
| $\sigma_{11} = \langle b, c, d, e, f, i \rangle$ | 9 | 1.28 |
| $\sigma_{12} = \langle d, e, f, g \rangle$ | 9 | 1.28 |
| $\sigma_{142} = \langle a, b, c, d, e, f, g, i, h, i \rangle$ | 1 | 0.14 |
| $\sigma_{143} = \langle a, b, c, d, e, f, g, i \rangle$ | 1 | 0.14 |
| $\sigma_{144} = \langle b, d, e, f, i \rangle$ | 1 | 0.14 |



Figure 7. Sequence of activities for the trace with the highest presence in the event log.

4.5.2. Connect Events to Activities

To connect the events of the activities, conformance checking using the Token Replay technique was performed, which allows for diagnosing and quantifying discrepancies between the modeled behavior and observed behavior in the event log. Token Replay measures the fitness level at the event level, providing a result that ranges from 0 to 1, where 1 represents a perfect fitness condition [11]. The relationship of the issues in the event log with the activities in the process model was evaluated. The overall result obtained from the Token Replay analysis was 0.894 (89.4%), indicating that around 10.6% of the events show deviations compared to the process model. Table 5 presents the fitness results concerning each activity: Backlog (83.6%), Verification (90.3%), Validation (90.7%), Quality Assurance (92.7%), In Progress (94.2%), To Do (96.2%), and Selected for Development (99.7%). This indicates that there are missing activities that prevent them from being reproduced optimally in the traces and can affect the process flow.

Table 5. Fitness percentage with respect to the activity.

| Status/Activity | Ref |
|--------------------------|-------|
| Backlog | 83.6% |
| Verification | 90.3% |
| Validation | 90.7% |
| Quality Assurance | 92.7% |
| In Progress | 94.2% |
| To Do | 96.2% |
| Selected for development | 99.7% |
| Rejected | 100% |
| Done | 100% |

5. Results

In this section, we present the Mining perspectives phase (Figure 1), in which the results of the mining of the case, organizational, and time perspectives were analyzed (Step 4: extend the model). The integrated model of Project 2 (Step 5: return integrated model) was also obtained, as well as that of Project 1, in order to analyze the differences between both projects.

5.1. Extended Process Mining Perspectives

In the analysis of the case perspective, each issue was considered as an individual case or trace. As observed initially in the discovered model, the results show that the trace with the highest frequency in the event log (Figure 7) corresponds to error-type issues, and the second most present trace (representing 6.09% of the log) corresponds to Feature-type issues. Figure 8 shows that Feature-type issues involve a more detailed process that describes the stages of software development more clearly. It can also be observed that the trace does not include the Done activity, which reflects that some tasks related to the Project are still in progress and have not reached their final status. The latter is due to the fact that the project is not yet finalized.



Figure 8. Sequence of activities for the second most frequent event trace (Feature) in the event log.

From an organizational perspective, it is related to the roles present in the event log. Table 6 shows the relative frequency of roles with respect to the most frequent trace, in which the role of programmer has the highest frequency at 44.16%, followed by the Quality Manager at 23.0%, Technical Leader at 14.40%, and the Project Manager at 12.25%. For certain traces, the system role is present with 6.14% automated activities. This is because Jira as a project management tool is automated.

Table 6. Participation frequency and percentage.

| Role | Events Frequency | Participation Percentage |
|------------------|------------------|--------------------------|
| Programmer | 1791 | 44.16% |
| Quality Manager | 934 | 23.03% |
| Technical Leader | 584 | 14.40% |
| Project Manager | 497 | 12.25% |
| System | 249 | 6.14% |
| Total | 4055 | 100% |

Finally, the time perspective was analyzed, which refers to the measurement of time obtained through the analysis of the timestamp attribute. In the log, the timestamp elements correspond to MM/DD/YYYY HH:MM:SS. The average, minimum, and maximum time of the activities presented in Table 7 were obtained. The activities with the longest duration are Backlog, Selected for Development, To Do, and Validation, while the Done status is also considered an activity with a longer duration because it remains until the client approves the deliverable before being sent to production.

Table 7. Specification of average, maximum, and minimum time per activity.

| Status/Activity | Time | | |
|--------------------------|---------|---------|---------|
| | Average | Maximum | Minimum |
| Backlog | 2.2 d | 134 d | - |
| Selected For Development | 8.7 d | 130.2 d | - |
| To Do | 10.8 d | 133.1 d | - |
| In Progress | 15.1 h | 18 d | - |
| Verification | 16.9 h | 26.2 d | 1 s |
| Quality Assurance | 9.6 h | 16.1 d | 1 s |
| Rejected | 5 d | 27.1 d | 11 m |
| Validation | 10.3 d | 112.3 d | 11.8 m |
| Done | 4.9 d | 52.8 d | - |

5.2. Integrated Model

Taking the Petri net generated by process discovery (Figure 6), the integrated model representation (Figure 9) was created to show how the different perspectives are combined into a single model. From the control flow perspective, it is possible to observe the order in which activities are carried out within the event log, which is represented through the status of the issues. It can be observed that the most executed activities are In Progress, Verification, and Quality Assurance. In contrast, the activity with the least execution is rejected, which was only executed 199 times in the event log. The case perspective is represented by the most frequent trace in the event log. The organizational perspective shows the roles involved in each activity. Finally, the time perspective is represented by the average time of the activities.

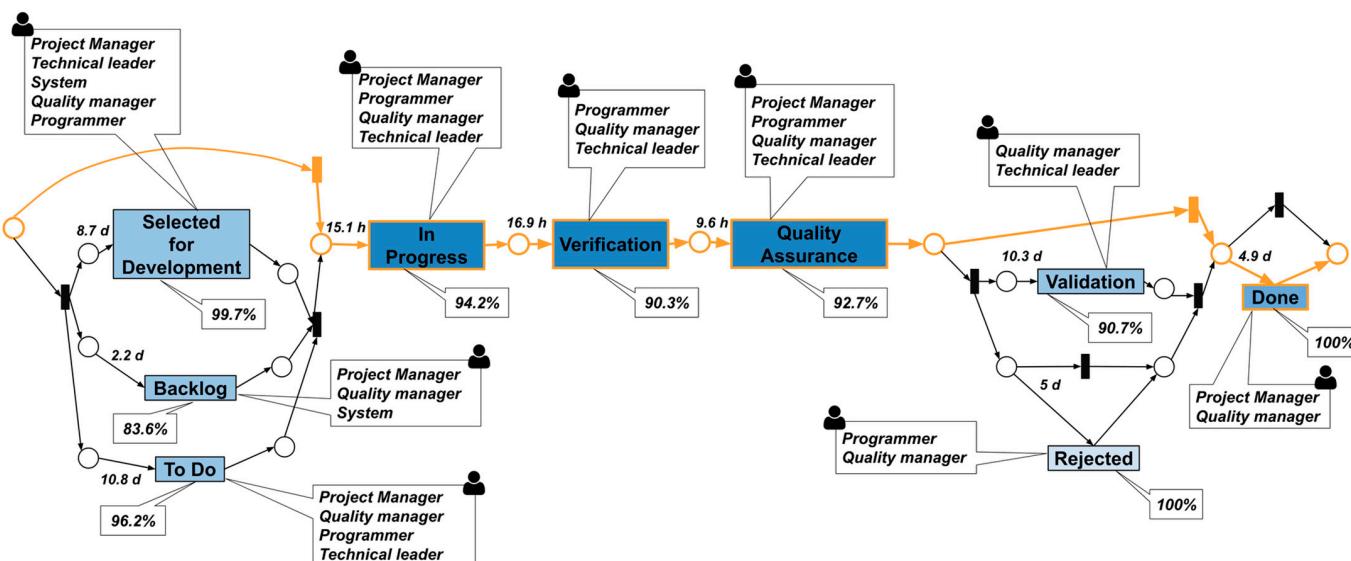


Figure 9. Project 2 integrated model.

5.3. A Comparison between the Projects

Process Mining allowed the SME to identify the changes implemented so far between the development period of Project 1 (Figure 2) and Project 2 (Figure 9). Although these changes are still in the process of integration, their early identification allows working on improvement to achieve greater effectiveness in SME operations. The application of Process Mining techniques and perspectives allows the SME to make decisions based on data to improve its software development process and optimize its performance.

Table 8 presents a comparison between Project 1 and Project 2 with respect to data extraction, events, trace identification, conformance checking, and perspectives. Regarding data extraction, in Project 1, manual traceability between events was established, while in Project 2 a script was developed to automate ETL activities. In trace identification, Project 2 identified a significantly higher number of traces in the event log compared to Project 1, suggesting greater diversity or complexity in the flow of activities in the second project. In the compliance check, both projects achieved satisfactory overall fitness results, although Project 1 scored a slightly higher percentage (93.5%) compared to Project 2's 89.4%.

From a control flow perspective, Project 1 revealed no inconsistencies, while Project 2 identified an inconsistency in the Validation activity in the most executed trace, indicating a possible area for improvement in that regard. From the perspective of the case, in both projects, it was discovered that the activities of the software development process are focused primarily on correcting the error: Project 1 (32.26%) and Project 2 (35.19%). Second to the creation of new Feature requirements: Project 1 (25.19%) and Project 2 (6.41%). From the organizational perspective, unlike Project 2, it experienced high staff turnover and role assignment. Finally, from the time perspective, taking into account that the Done activity

is relevant to completed tasks ready for delivery and deployment, Project 1 had a longer duration with an average time (8.6 d) compared to Project 2 (4.9 d), which may be because, in Project 2, tasks were released without being validated.

Table 8. Comparative between Project 1 and Project 2.

| Data | Project 1 | Project 2 |
|----------------------------|---|--|
| Data extraction | Manual traceability established between events. | Development of a script for ETL activities. |
| Events | 481 events. | 4055 events. |
| Trace identification | 22 traces identified in the event log. | 144 traces identified in the event log. |
| Conformance checking | Overall fitness result 0.935 (93.5%). | Overall fitness result 0.894 (89.4%). |
| Control flow perspective | Discovery of the real flow of the company. | Inconsistency in the Validation activity in the most executed trace. |
| Case perspective | Error issues (32.26%). Feature issues (35.19%). | Error issues (25.19%). Feature issues (6.41%). |
| Organizational perspective | There were no changes in role assignment. | Rotation and change in role assignment. |
| Time perspective | Longer average duration in the Done activity (8.6 d). | Shorter average duration in the Done activity (4.9 d). |

These differences represent the importance of adaptability and efficiency in software project management and how they can influence process improvement results.

6. Discussion

The findings identified in both Project 1 and Project 2, based on the discovered model and trace analysis, coincide in that the most frequent trace corresponds to the Error type. Additionally, in Project 2, the inconsistency in the flow of events was observed when setting issues in the Done state without having executed Validation activities. On the other hand, when comparing resources with the organizational perspective in Project 2, changes are observed both in the project team and management. Therefore, even when SMEs establish a better flow in their activities, personnel changes remain a challenge [5]. In this sense, the loss of knowledge and experience of personnel was identified as a challenge, affecting the correct execution of the process model and, consequently, the quality of the work product. To overcome this challenge, it is important that SMEs have strategies to efficiently integrate new team members, as well as retain and leverage existing talent. Another challenge was the lack of documentation related to processes [1]. Documentation allows explicit knowledge to be transferred among team members. During the mining of Project 1, there was no formal document of the software development process, so the SME had to document it before starting mining activities.

Similarly, recording process activities through their flow between various systems proves to be a challenge. In this sense, during the preprocessing of the first mining, there was a limitation in the extraction [1] because the process records were in several systems without any interface between them. The lack of APIs, therefore, has been considered another challenge [19]. On the other hand, having data in different formats, the next challenge during ETL activities was to prepare event log data [5] to provide a unique and standardized format. This is also due to the challenge of variability in records among tools or support systems for software development [18], such as Jira and Github, which are linked but stored differently.

Furthermore, another challenge to highlight between Project 1 and Project 2 was communication. In Project 1, there was open and constant communication through meetings, which helped to address questions and concerns that arose about Process Mining. Maintaining this level of communication allowed the successful challenge of creating awareness [5], meaning achieving an understanding of the process by the senior management and other team members to communicate the results [5] with the purpose of applying them to improve SME processes.

However, in Project 2, there was inadequate communication, mainly carried out through messages and email. This was due to the change in project leadership, prioritizing compliance with customer delivery. This made it difficult to interpret the data, as a high level of knowledge of the business unit is required [5].

Therefore, another important challenge is the ability of SMEs to implement process improvements. This may require a significant change in the culture of the SME, as well as the allocation of resources and personnel training.

7. Conclusions and Future Work

Process Mining represents an opportunity for software development SMEs looking to refine their software development processes. Through trace analysis and workflow visualization, Process Mining not only identifies opportunities for improvement, but also enhances process optimization, improving operational efficiency and overall effectiveness of SMEs.

In addition, the extension of process models through the Process Mining perspectives of control flow, case, organizational, and time flow generates a holistic view, providing deep insights for both management and process improvement teams. This integration of perspectives guides informed, data-driven decisions.

Challenges such as lack of documentation, data extraction, constant shifting manpower, and variability in software development system records were identified as some of the obstacles that SMEs must face when implementing Process Mining. To overcome these challenges, it is necessary to establish constant communication among the different actors involved in the process, up to the top management. Communication is key to ensuring accurate data interpretation and to ensuring that Process Mining is aligned with the organization's goals and objectives.

It is important to highlight that Process Mining is not a single and definitive solution, but a continuous improvement process. The changes implemented in the processes are constantly integrated, which may lead to the emergence of non-optimal flows. Therefore, it is necessary to iteratively identify improvement opportunities to achieve greater effectiveness in SME operations.

As a future work, it is recommended to continue researching and developing solutions for the identified challenges. It is necessary to look for new strategies and techniques to overcome these limitations and ensure the quality of the information obtained through Process Mining. Additionally, it is important to foster awareness and a culture of continuous improvement in SMEs, so that Process Mining becomes a regular discipline in data-driven decision making.

Author Contributions: Conceptualization, S.J.U.-C., M.A.A.-V. and B.L.F.-R.; methodology, S.J.U.-C., M.A.A.-V. and B.L.F.-R.; software, S.J.U.-C.; validation, J.E.I.-E. and F.F.G.-N.; formal analysis, S.J.U.-C.; investigation, S.J.U.-C.; resources, F.F.G.-N.; data curation, S.J.U.-C.; writing—original draft preparation, S.J.U.-C., M.A.A.-V. and B.L.F.-R.; writing—review and editing, J.E.I.-E., I.G.P. and C.L.P.A.; visualization, S.J.U.-C., M.A.A.-V. and B.L.F.-R.; supervision, M.A.A.-V. and B.L.F.-R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Acknowledgments: The authors acknowledge Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Wijnhoven, F.; Hoffmann, P.; Bemthuis, R.; Boksebeld, J. Using process mining for workarounds analysis in context: Learning from a small and medium-sized company case. *Int. J. Inf. Manag. Data Insights* **2023**, *3*, 100163. [[CrossRef](#)]
- Suarez, D.R.; Leon, G.C. Las PyME de desarrollo de software. Modelos de mejora de sus procesos en Latinoamérica. *Rev. Espac.* **2019**, *40*, 9.
- Findikoglu, N.M.; Ranganathan, C.; Watson-Manheim, M.B. Partnering for prosperity: Small IT vendor partnership formation and the establishment of partner pools. *Eur. J. Inf. Syst.* **2020**, *30*, 193–218. [[CrossRef](#)]
- Choras, M.; Springer, T.; Kozik, R.; Lopez, L.; Martinez-Fernandez, S.; Ram, P.; Rodriguez, P.; Franch, X. Measuring and Improving Agile Processes in a Small-Size Software Development Company. *IEEE Access* **2020**, *8*, 78452–78466. [[CrossRef](#)]
- Eggert, M.; Dyong, J. Applying process mining in small and medium-sized IT enterprises—challenges and guidelines. In Proceedings of the Business Process Management: 20th International Conference, BPM 2022, Münster, Germany, 13–15 September 2022; pp. 125–142.
- Glodek, P.; Łobacz, K. Transforming IT small business—The perspective of business advice process. *Procedia Comput. Sci.* **2021**, *192*, 4367–4375. [[CrossRef](#)]
- Heidt, M.; Gerlach, J.P.; Buxmann, P. Investigating the Security Divide between SME and Large Companies: How SME Characteristics Influence Organizational IT Security Investments. *Inf. Syst. Front.* **2019**, *21*, 1285–1305. [[CrossRef](#)]
- Berti, A.; Van Der Aalst, W. Extracting multiple viewpoint models from relational databases. In *Lecture Notes in Business Information Processing*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 379. [[CrossRef](#)]
- Berti, A.; Park, G.; Rafiei, M.; Van der Aalst, W. An Event Data Extraction Approach from SAP ERP for Process Mining. In *ICPM Workshops*; Springer International Publishing: Cham, Switzerland, 2021; Volume 433, pp. 255–267.
- Kouzari, E.; Sotiriadis, L.; Stamelos, I. Enterprise information management systems development two cases of mining for process conformance. *Int. J. Inf. Manag. Data Insights* **2023**, *3*, 100141. [[CrossRef](#)]
- Van der Aalst, W. *Process Mining: Data Science in Action*; Springer: Berlin/Heidelberg, Germany, 2016.
- Delgado, A.; Calegari, D. Process and Organizational Data Integration from BPMS and Relational/NoSQL Sources for Process Mining. In Proceedings of the 17th International Conference on Software Technologies (ICSOFT), Lisbon, Portugal, 11–13 July 2022.
- Özdağoğlu, G.; Kavuncubaşı, E. Monitoring the software bug-fixing process through the process mining approach. *J. Softw. Evol. Process* **2019**, *31*, e2162. [[CrossRef](#)]
- Zhu, R.; Dai, Y.; Li, T.; Ma, Z.; Zheng, M.; Tang, Y.; Yuan, J.; Huang, Y. Automatic Real-Time Mining Software Process Activities From SVN Logs Using a Naive Bayes Classifier. *IEEE Access* **2019**, *7*, 146403–146415. [[CrossRef](#)]
- Carmona, J.; Dongen, B.; Weidlich, M. Conformance checking: Foundations, milestones and challenges. In *Process Mining Handbook*; Springer International Publishing: Cham, Switzerland, 2022; pp. 155–190.
- de Leoni, M. Foundations of process enhancement. In *Process Mining Handbook*; Springer International Publishing: Cham, Switzerland, 2022; pp. 243–273.
- Urrea-Contreras, S.J.; Flores-Rios, B.L.; González-Navarro, F.F.; Astorga-Vargas, M.A.; Ibarra-Esquer, J.E.; García Pacheco, I.A.; Pacheco Agüero, C.L. Process Mining Model Integrated with Control Flow, Case, Organizational and Time Perspectives in a Software Development Project. In Proceedings of the 2022 10th International Conference in Software Engineering Research and Innovation (CONISOFT), Cuapiaxtla, Mexico, 24–28 October 2022; pp. 92–101.
- Vavpotič, D.; Bala, S.; Mendling, J.; Hovelja, T. Software Process Evaluation from User Perceptions and Log Data. *J. Softw. Evol. Process.* **2022**, *34*, e2438. [[CrossRef](#)]
- Vidoni, M. A systematic process for Mining Software Repositories: Results from a systematic literature review. *Inf. Softw. Technol.* **2021**, *144*, 106791. [[CrossRef](#)]
- Drakoulogkonas, P.; Apostolou, D. On the Selection of Process Mining Tools. *Electronics* **2021**, *10*, 451. [[CrossRef](#)]
- Scrum-Institute. *Scrum Revealed Training Book*, 3rd ed.; International Scrum Institute: Wollerau, Switzerland, 2017; Available online: <https://bit.ly/3rKMF31> (accessed on 1 April 2023).
- Brechner, E. Agile project management with Kanban. In *Pearson Education*; Pearson Education: London, UK, 2015.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.