# Process Mining Applied in a Software Project Development with SCRUM and ProM

Ezequiel O. Ramos and Rogério Rossi

***Abstract* — The software product is high used by the society in general and its development complexity are inputs of this research that gears into the software development processes. The mapping and modelling of software processes, as well as their standardization are not trivial tasks in the industry of software. Therefore, process mining practices can be useful for discovering or validating processes. This article presents a hypothetical software development project that uses the agile SCRUM method, Jira software, Jenkins pipeline and a process mining tool called ProM. As the project team generates many records using the software development tools, these records are considered event logs and it is be used during process mining activities. ProM allows users to identify processes from the event logs and is used with the records generated by Jira and Jenkins. The visualization of a possible process derived from the use of these event logs is presented when using the ProM tool and the Flexible Heuristics Miner algorithm. In conclusion, process mining can be useful to discover or validate existing software processes during the execution of a software project, also allowing these processes to be standardized to be used in future projects.**

***Key words* — Agile, Process Mining, ProM, SCRUM Method, Software Project.**

## I. Introduction

Software Engineering and the entire software development area, which covers peopleware, hardware, programming languages, specific tools for software development and integrated development environments, can undergo evolutions constantly, depending on the evaluated component. These evolutions and updates favor software development, one of the most consumed products of many sectors (finance, government, logistics, public health, etc.) and by society in general.

The entire need for engineering specific processes for software engineering has undergone constant and substantial changes, with the advent of tools and new mechanisms that seek to favor the production of software. Agile methodologies, tools associated with agile methods, DevOps, DevSecOps, etc., aim to collaborate with the domain of software process engineering, offering means of increasing productivity and enabling the integration of the teams involved.

The complexity that can be verified in the domain of software processes, can be better treated with the use of process mining practices, so that these can, sometimes, favor the discovery of new processes, or provide improvements to existing software processes.

Process automation is notorious in several business areas and in several areas where engineering practices are applied, not being different for software engineering. With many automated software development processes, whether managerial or technical, there is a wide generation of records related to activities related to these processes. These records can be characterized as event logs, the basic element for process mining practices.

According to [1] software development methods are trying to offer an answer once again to the eager business community that asks for faster and more agile software development processes. In general, software projects that use agile practices or methods do not show improvements in their processes, usually due to increased productivity.

In this sense, this article aims to present, through a hypothetical software development project, the integration of process mining practices during the project development, which is conducted using the agile SCRUM method and the tool called ProM, a tool for activities related to process mining that favors mining activities for all kind of process discovery or enhancement.

Possibly when conducting a software project, a high number of records can be generated by different tools that are used during software development. These records, considered as event logs, favor the practices of process mining. According to [2], the analyses that derive from process mining practices can be used to improve the performance of processes or to verify compliance with the rules and regulations currently used in an organization.

Thus, in order to meet the objective proposed for this article, it is organized as follows: section two presents a conceptual review of process mining; section three presents two related works that address practices and a methodology for applying process mining; section four presents the characteristics of a hypothetical software project that uses the agile SCRUM method and process mining practices; section five focuses on the project's specific process mining activities and presents analysis of the results; and section six presents the final considerations of the article.

## II. Literature Review

The literature review is restricted to the concept of process mining because it is a concept that is still little discussed and treated by the software engineering community. In this way, the main characteristics of process mining are presented, considering its perspectives, types, and tools for mining.

Submitted on July 31, 2023.
Published on September 18, 2023.

E. O. Ramos, University of São Paulo/Continuing Engineering Education Program (PECE), Brazil.
(e-mail: eoliveira1991@hotmail.com)

R. Rossi, University of São Paulo/Continuing Engineering Education Program (PECE), Brazil.
(e-mail: rogeriorossi@usp.br)

*DOI: http://dx.doi.org/10.24018/ejeng.2023.8.5.3089*

The activity related to process modeling using any technique or tool can be complex, regardless of the type of process to model, whether business or software, for example. Carrying out process modelling activities manually has become an uncommon task, leading to many failures. Therefore, the support of specific tools and techniques has become more common. Business process modelling or BPM and its notation, BPM notation or BPMN, has become quite common for process modelling, as it disciplines and standardizes activities.

However, recent studies based on the notions of process modelling have presented forms of modelling in an automated or semi-automated approaches, through process mining [3]. According to [4] process mining refers to the act of discovering, improving real processes (that is, non-assumed processes) and monitoring, extracting knowledge from event logs available in current (information) systems.

Many daily activities are recorded digitally, from hospital consultations and bank withdrawals, scheduling appointments and medical examinations, purchasing airline tickets, services with any type of government agency, among others; all these records culminate in event logs that can be used in process mining practices. Process mining enables some process discovery actions, or improvements in existing processes, by identifying bottlenecks, verifying violations, and streamlining activities, among others.

Process mining fundamentally relies on an event log. The applied techniques and algorithms require events that refer to a real activity performed by a user. Event logs can include various information, such as the user who performed the activity, date and time of its execution, and other specific data for that event.

Event logs are processed by different tools, such as: 1) ProM (Process Mining Workbench), an open-source tool that features several plug-ins for different purposes; and 2) DISCO commercial tool that implements its own algorithm; and by different algorithms that perform mining activities such as: Inductive Visual Miner, Heuristic Miner, and Flexible Heuristics Miner. The analyses that result from the mining stages can be observed from some perspectives according to [4] and [5]:

1) Control Flow perspective,
2) Organizational perspective,
3) Resource perspective,
4) Performance perspective, and
5) Information perspective.

The activities for carrying out process mining are grouped into three types of process mining, according to [4]:

1) Discovery – that allows generating the process model from the processing of the event log, being one of the most used types of data mining that algorithms are capable of discovering real processes simply based on examples of execution of the logs.

2) Conformance – that allows checking the conformity of the current process, the formal model that is in force is compared with the model generated from the execution logs.

3) Enhancement – a type of process mining to improve the existing process model based on the executions of the event logs.

## III. Related Works

Some works can be considered regarding the use of process mining, more specifically applied to software engineering, such as [5]–[7]; another work, according to [8], presents process mining practices applied to processes in other areas or business activities. In this article, one of the related works presented address the integration of process mining with software development [9], and another work deals with a methodology for the use and application of process mining [2].

The related work presented by [9] considers a framework to analyze the ticket resolution process related to software maintenance. The framework helps identify inefficiencies by mining events from software repositories considering multiple perspectives. This analysis helps the managers in their decision making and process improvement actions due to a detailed view of the analyzed process.

Gupta *et al.* [9] present the framework with a series of steps that initially considers an interview of managers to identify the process challenges encountered during the management of the organization's various software projects. The authors consider that managers and teams have a vision of the challenges and problems, without proper understanding of the extent of the problem. The framework presented by [9] considers three stages: 1) Research to identify the challenges encountered by teams working on software development; 2) Classification of problems according to the perspective of process mining; 3) Importance research.

The case study reports the application of the framework in a global Software Technology company called InfoSys, where interviews with Software Project Managers were carried out in order to identify the challenges. In the end, a list of 30 software process management challenges were defined, considering the software development cycle, and classified into 8 categories referring to process mining perspectives. The challenges were more associated with the software maintenance process.

To carry out the analysis, the repositories that contain the ticket event logs were considered. This analysis phase was divided into three activities: 1) data extraction; 2) transformation; and 3) multi-perspective process mining. For the specific activity of process mining, the Fuzzy Miner algorithm was used to discover process models, bottleneck analysis, compliance analysis, etc. A generic algorithm was also used to identify deviations between defined and current processes.

Based on the analysis of the InfoSys repositories [9], it was identified that about 57% of the tickets (593,497 tickets) have user input records, causing an increase in ticket resolution time, so a Predictive System was proposed to reduce such entries with unnecessary information, reducing ticket resolution time.

The result of analyses based on event logs from the ticket repository allowed detecting problems and reducing the opening of software maintenance tickets, and consequently allowed improvements in the quality of the software maintenance process.

The other related work presented by [2] proposes a methodology for the application of process mining to support projects that aim to improve compliance with rules and regulations or process performance called PM$^2$.

*DOI: http://dx.doi.org/10.24018/ejeng.2023.8.5.3089*

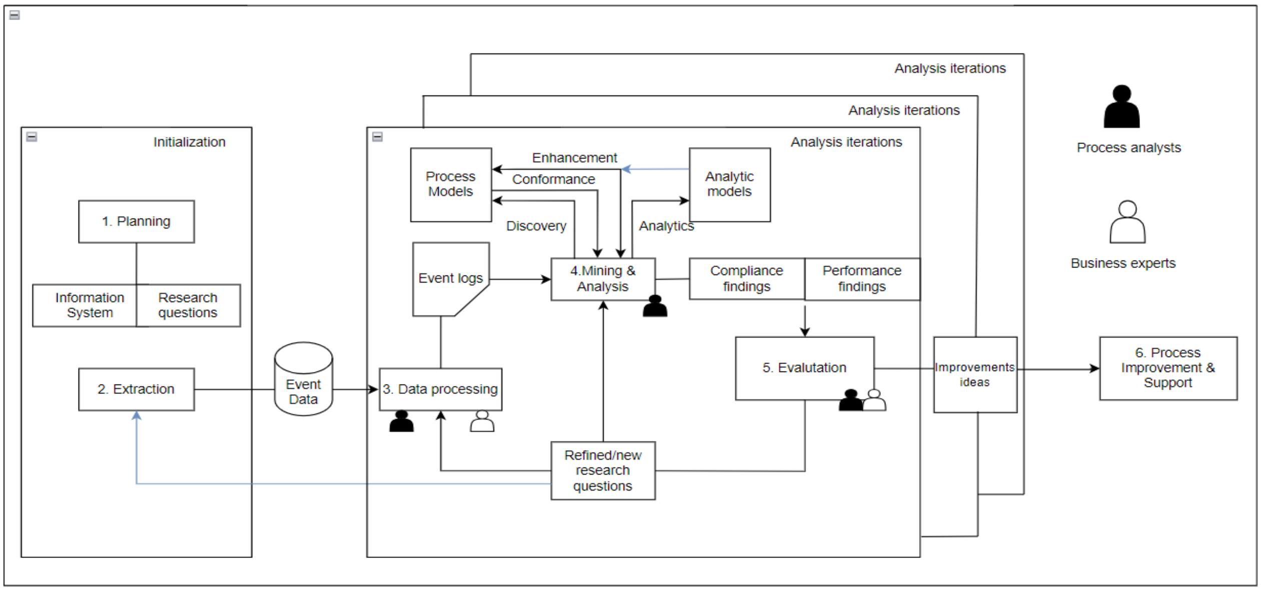


Fig. 1. PM$^2$ Structure [2].

According to [2], PM$^2$ aims to translate into concrete research questions the needs that must be refined and answered in an iterative way, favoring the discoveries of processes that are the basis of process mining practices.

The PM$^2$ Methodology (Fig.1) has six phases which correspond to:

1) Planning (to determine the Research Questions, considering the organization's business processes);

2) Extraction (extraction of events, using the Information System to obtain event logs);

3) Data Processing (creating event logs with different views of the events collected in the Extraction);

4) Mining & Analysis (answer the Research Questions, through the application of Process Mining techniques applied to event logs);

5) Evaluation (aiming to evaluate the results obtained according to the objectives defined in the Planning);

6) Process Improvement & Support (using the results obtained in the iterations to modify the actual execution of the process).

Phases 3, 4, and 5 are performed iteratively through Analysis Iterations. The phases proposed by PM$^2$ are briefly described below.

## IV. Characterizing the Software Project

This section presents the characteristics of a hypothetical software project that uses the SCRUM method, as well as the integration proposal through project sprints that deal with process mining practices within the project. The software to be delivered refers to an application for renting vacation properties that features specific user and property management functions, integration with a specific geolocation system and payment management.

Process mining is integrated into the project, where some sprints are planned to specifically carry out process mining activities during the project, and these sprints are carried out concurrently with the other sprints related to software development.

In this way, this section presents the project team, the requirements of the software (macro view), the product backlog, a sprint plan for the project according to the proposed backlog, a macro schedule of the main activities, and the infrastructure to carry out the project.

### *A. Project Team*

To carry out the project, the project team is made up of professionals in the area of software engineering and process engineering. The group of software engineering professionals is: the product owner, a scrum master and software engineers assigned to technical activities and development. The group of professionals for process mining also consider the product owner and the scrum master, for activities aimed at analyzing mining results, and a professional specialized in process mining, its related techniques, and tools.

Software engineering professionals are responsible for software coding, software unit tests, integration tests, performance tests, and versioning control, among others; and the professional of process mining is responsible for all mining activities, such as collecting event logs, processing using specific tools and algorithms for process mining, prior evaluation of results, presentation, and discussion with other members of the team, and decision-making arising from mining activities.

### *B. Project Infrastructure*

Addressing the technological infrastructure for carrying out the project is fundamental for two reasons: 1) it allows clarifying, even under a hypothetical domain, how the software product will be built, and which tools will be used in the software development; and 2) it allows to show how the records of development activities are generated using specific tools that will be the event logs.

Therefore, the GitHub platform is considered for storing and versioning the code, which, according to [10], corresponds to a collaborative code hosting website, built based on the version control system called Git.

In addition to using GitHub, for code automation activities the Jenkins pipeline is considered, which helps in continuous integration and continuous delivery (CI/CD) that, according to [11], corresponds to a tool capable of automating software construction and testing processes and monitoring any external work. Jenkins allows different teams to collaborate in different ways in building the same software [12].

The use of storage in the cloud approach characterizes the server base for data storage. For [13] cloud computing enables virtualized resources using different technologies, such as: web services, virtualization, and multi-tenancy. In the specific case of this project, a dedicated cloud server is also considered for storing event logs to be used during process mining practices.

The control of project management activities is carried out using the Jira Software as it is a work item tracker, and is also used to track software bugs and tasks in general, commonly used in projects that use agile methods [14].

### C. *Software Development Phases*

The phases of the project that are highlighted follow the agile SCRUM method that allows to create a systematization of actions and provide a plan of general activities for the project. Therefore, the phases considered for carrying out this software development project are:

1) Product Backlog definition;
2) Sprint definition;
3) Execution of sprints;
4) Retrospective of sprints.

Each of these phases are presented in the following subsections, emphasizing the focus is on process mining activities.

#### 1) *Product Backlog definition*

For this project, the definition of the product backlog is divided into four categories as presented in Table I:

1) Infrastructure, that deals with hardware and software configuration activities to support software development;

2) Front end, that considers the interfaces of the software, user interaction mechanisms with the software;

3) Backend, that considers the need to define the software's business rules, as well as the architecture and logical processing;

4) Process Mining, considering process mining activities that must be carried out throughout the project, through specific sprints, for possible discovery of processes from event logs.

#### 2) *Sprints definition*

From the activities belonging to the product backlog, it was determined what should be developed during the project, as well as the priorities of the activities of development and implementation of the final product. Based on the estimates made by the project team and the infrastructure available to carry out the project, it was defined that the project should be carried out considering ten development sprints and three process mining sprints, lasting approximately two weeks each.

As this project has a specific characteristic compared to the other projects carried out by the team, which refer to process mining activities that will be carried out concurrently with the software development activities, the team defined the order of the sprints, paying attention to the needs and specificities of the process mining sprints throughout the project. Therefore, software development sprints and process mining sprints were defined for the project, as shown in Fig. 2.

Software development sprints address the activities of development, considering what was defined in the product backlog, that is, infrastructure, front-end and back-end activities. Process mining sprints aim to conduct process mining activities as defined in the product backlog.

Sprint planning assumes a parallelism between software development and process mining sprints. The process mining sprints run parallel to the second, fifth and ninth development sprints.

#### 3) *Sprints execution*

With a specific focus on process mining sprints, the execution of these three sprints planned for process mining practices integrated into the software development project, is presented in section V(A).

TABLE I. PRODUCT BACKLOG

| Infrastructure | Front End | Back End | Process Mining |
|---|---|---|---|
| Configure application and database servers | Define user management interfaces | Create user management services | Event logs identification for process mining |
| Configure Database | Define vacation properties management interfaces | Create vacation properties management services | Event logs extraction |
| Configure Jenkins | | Create geolocation services | Event logs processing |
| Configure Github | | Create vacation properties rental services | Process mining analysis and discussion |
| Configure Process Mining tools | | Perform process mining activities | |
| Configure database servers for process mining tasks | | | |

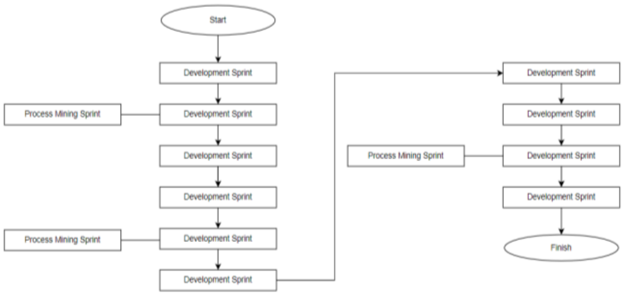


Fig. 2. Sprints for development and process mining activities.

#### 4) *Sprints retrospective*

The sprint retrospective is performed exclusively for process mining sprints, being presented as part of the results analysis described in section V(B).

## V. RESULTS AND DISCUSSION

The process mining activities that are performed during software project development are the collection of event logs for mining and processing, concluding with the analysis of possible results.

### *A. Sprints Execution*

As presented in Fig. 2, the first process mining sprint occurs concurrently with the second software development sprint. This process mining sprint deals with setting up the environment using ProM. This sprint also determines the objective of mining, the sources of event logs, and the specific algorithms used for mining, etc.

When considering PM$^2$ methodology [2] for process mining, there are six phases presented in Table II, which presents the relationship of each of these phases with the process mining sprints, and with the subsections of the article these phases are considered.

#### *1) Event logs collection*

Event logs correspond to basic items digitally identified and stored on servers and datasets linked to each of the tools used for software development. Event logs are essential for process mining activities and determine a possible view of the processes' performance, and these views are possible using algorithms linked to specific process mining tools.

It is feasible to determine the specific objective regarding the mining actions and that the event logs can allow achieving the results according to the specified objectives. However, it is essential to define which bases will be used to collect event logs to be used during process mining.

TABLE II. PROCESS MINING PHASE – PM$^2$ METHODOLOGY

| PM$^2$ methodology phases | PM sprints | Article subsections |
|---|---|---|
| 1. Planning | 2 | V.A.1 |
| 2. Extraction | 5, 9 | V.A.2 |
| 3. Data Processing | 5, 9 | V.A.3 |
| 4. Mining and Analysis | 5, 9 | V.A.3 |
| 5. Evaluation | 9 | V.B |
| 6. Process Improvement & Support | 9 | V.B |

For the process mining actions that are carried out in this project, the event logs of the Jenkins and the event logs collected from the Jira Software tool records are considered. It is important to point out that any tools used for the development of software, which present these characteristics of records stored in digital media, can be inputs for process mining practices. The determination of the event log bases is related to the search that is intended to be carried out from the mining actions.

The Jenkins pipeline generate several types of logs that can be considered for mining actions, with event logs referring to:

1) the execution time;
2) the date and time of execution;
3) the user who started the execution;
4) successful executions;
5) unsuccessful executions;
6) what project is under execution.

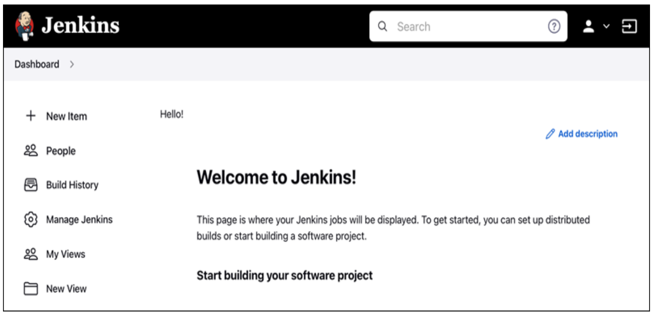
When considering the Jira Software tool, some types of event logs can be collected, such as:

1) activities deadline;
2) name of activities;
3) responsible for carrying out the activities;
4) activities with expired deadlines.

#### *2) Event logs extraction*

The extraction of event logs occurs according to the selected databases mentioned above, in the specific case of this project, logs obtained through the Jenkins and Jira tools are used, that is, logs generated during software development and stored in these tools.

Fig. 3 shows the Jenkins tool interface with its options menu (left side) where the Manage Jenkins option is identified. From this menu option, it is possible to access the functionalities for managing Jenkins. The Status Information Functionality is relevant for logs showing various information, such as: System Information, Load Statistics, System Log. From the System Log option, it is possible to retrieve all the execution information performed via Jenkins, as shown in Fig. 4.

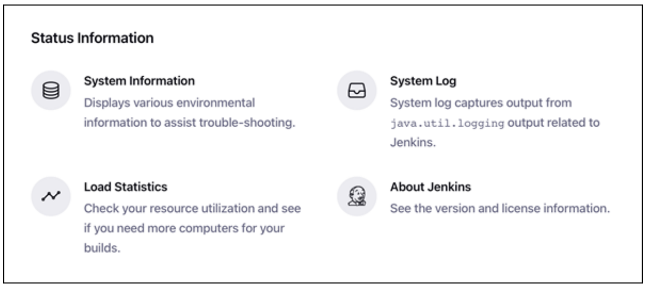


Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Fig. 3. Interface Jenkins



Status Information

System Information
Displays various environmental information to assist trouble-shooting.

System Log
System log captures output from `java.util.logging` output related to Jenkins.

Load Statistics
Check your resource utilization and see if you need more computers for your builds.

About Jenkins
See the version and license information.

Fig. 4. Jenkins – status information.

From the execution of the System Log option, with the availability of the “All Jenkins Logs” option, via the Windows operating system, it is possible to verify two types of files generated from the Jenkins tool, one that represents the console outputs (Jenkins.out) and another that represents the errors of the executed executions (Jenkins.err).

As for the event logs generated from the Jira Software tool, these are obtained using the Troubleshooting and support tools option, then using the Create support zip option that allows the generation of an event log file, as shown in Fig.5. The options Jira_Install/logs and Jira_Home/Log are checked, where specific storage of event logs that are generated when using the tool is verified.

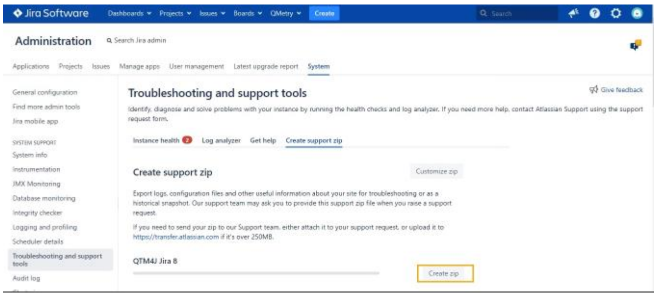
#### *3) Event logs processing*

Event log processing occurs during the specific process mining sprints planned for the project. After extracting the event logs, they are processed using a specific process mining tool, which in the case of this project is the ProM tool. The ProM tool exclusively presents process mining algorithms and can be used to carry out the process mining activities planned for the project.

The process mining activities that are carried out using ProM require that the mining objective be previously determined, as this objective has an implicit relationship with the previously collected event log bases. For this project specifically, the objective is to discover processes from the event log, that is, to verify if from the logs it is possible to determine some type of process that is related to the development of the software, favoring its implementation as a standard process to be used in future projects of the organization.

From the event logs that were collected through the Jenkins and Jira tools, it is possible to search or mine the processes. This mining takes place based on logs used with ProM and some specific process discovery algorithm. Event logs referring to project management are collected from Jira Software, so from these logs it is possible to identify a process, based on the mining that takes place. It is important to point out that process discovery may be related to the number of event logs. Possibly, the quantity allows identifying or consolidating possible discoveries.

From the event logs obtained from Jira, it is possible to discover some kind of management process, since this tool stores records of management activities carried out during the project's development sprints.



Troubleshooting and support tools

Identify, diagnose and solve problems with your instance by running the health checks and log analyzer. If you need more help, contact Atlassian Support using the support request form.

Create support zip

Export logs, configuration files and other useful information about your site for troubleshooting or as a historical snapshot. Our support team may ask you to provide this support zip file when you raise a support request.

If you need to send your zip to our Support team, either attach it to your support request, or upload it to https://transfer.atlassian.com if it's over 250MB.

Fig. 5. Event Logs from Jira.

As for the event logs obtained from Jenkins, these correspond to technical software development activities, so from the processing of these event logs, the discovery that may occur favors some type of technical process related to software development.

Considering the characteristics of this software project and based on its development activities that are executed according to the product backlog and the team defined for the project, the event logs start to be generated at the beginning of the project. The logs generated during the development of the software can allow the second sprint of process mining to reach an approximate amount of one hundred thousand event logs from Jira Software and about one hundred and fifty thousand event logs from Jenkins. This amount tends to grow as the project evolves, allowing the extraction that occurs in the third process mining sprint to be around five hundred thousand event logs from Jira Software and around eight hundred thousand event logs from Jenkins.

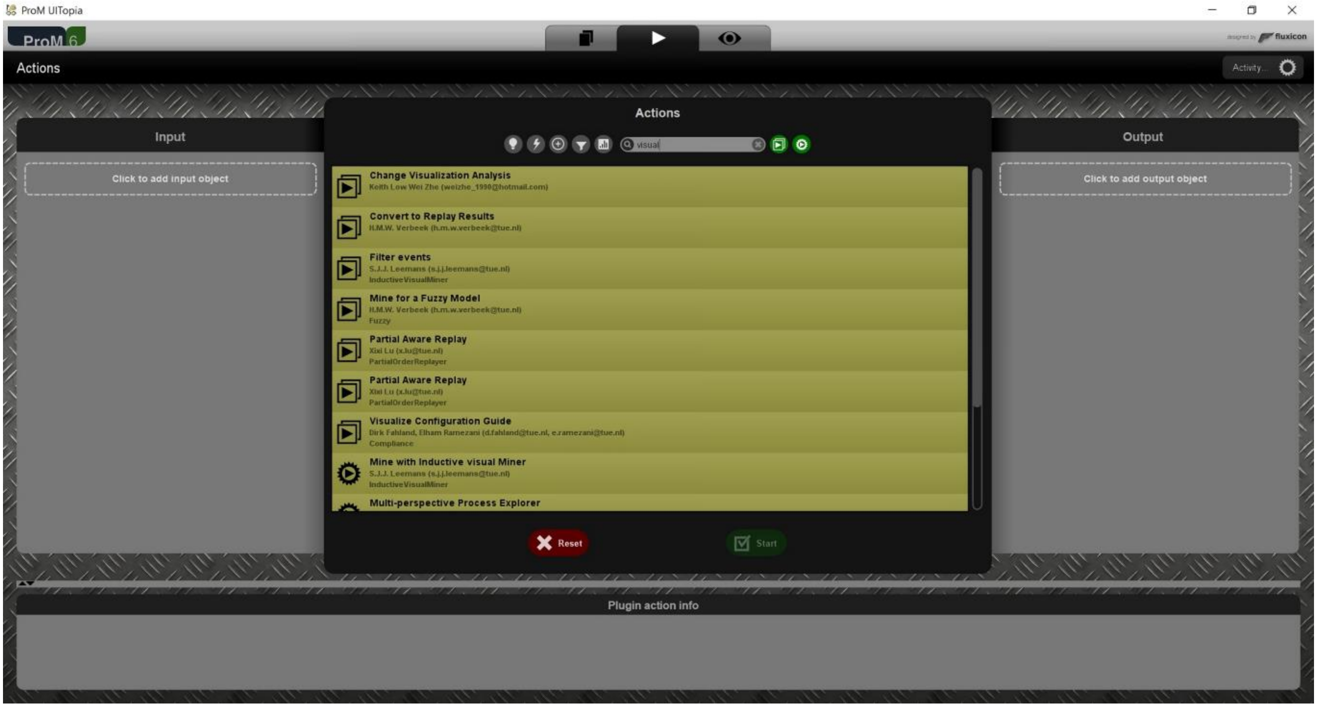
The processing of these event logs occurs during the sprints foreseen for the process mining activities, and what is processed using the ProM tool in the second sprint of process mining with these event logs can allow the discovery of some type of process, whether managerial or technical, considering the event logs that were processed. The third sprint of process mining has higher amounts of event logs, which may allow:

1) to validate possible processes discovered in the second sprint; and/or

2) discover previously unidentified new processes in the third sprint.

To carry out the processing using the ProM tool, the event logs are imported for processing and the algorithms are selected. ProM allows the use of some algorithms as presented in Fig. 6. To carry out the Discovery activity, it is possible to use, for example, the Fuzzy Miner and Flexible Heuristics Miner algorithms.

After selecting the desired algorithm, based on a previous analysis of the actions of this algorithm, the event logs are processed separately, in order to enable the identification of some process. The execution to be carried out depends on the characteristics of the event logs, the number of event logs and the algorithm used. After processing, it is possible to verify the result with various forms of visualization, depending on the algorithm selection, being possible graphical analyzes through petri nets, for example, which favors the possible identification and visualization of the process, as shown in Fig. 7.



Change Visualization Analysis
Keith Low Wei Zhe (weizhe_1990@hotmail.com)

Convert to Replay Results
H.M.W. Verbeek (h.m.w.verbeek@tue.nl)

Filter events
S.J.J. Leemans (s.j.j.leemans@tue.nl)
InductiveVisualMiner

Mine for a Fuzzy Model
H.M.W. Verbeek (h.m.w.verbeek@tue.nl)
Fuzzy

Partial Aware Replay
Xixi Lu (x.lu@tue.nl)
PartialOrderReplayer

Partial Aware Replay
Xixi Lu (x.lu@tue.nl)
PartialOrderReplayer

Visualize Configuration Guide
Dirk Fahland, Elham Ramezani (d.fahland@tue.nl, e.ramezani@tue.nl)
Compliance

Mine with Inductive visual Miner
S.J.J. Leemans (s.j.j.leemans@tue.nl)
InductiveVisualMiner

Multi-perspective Process Explorer

Fig. 6. ProM Algorithms.

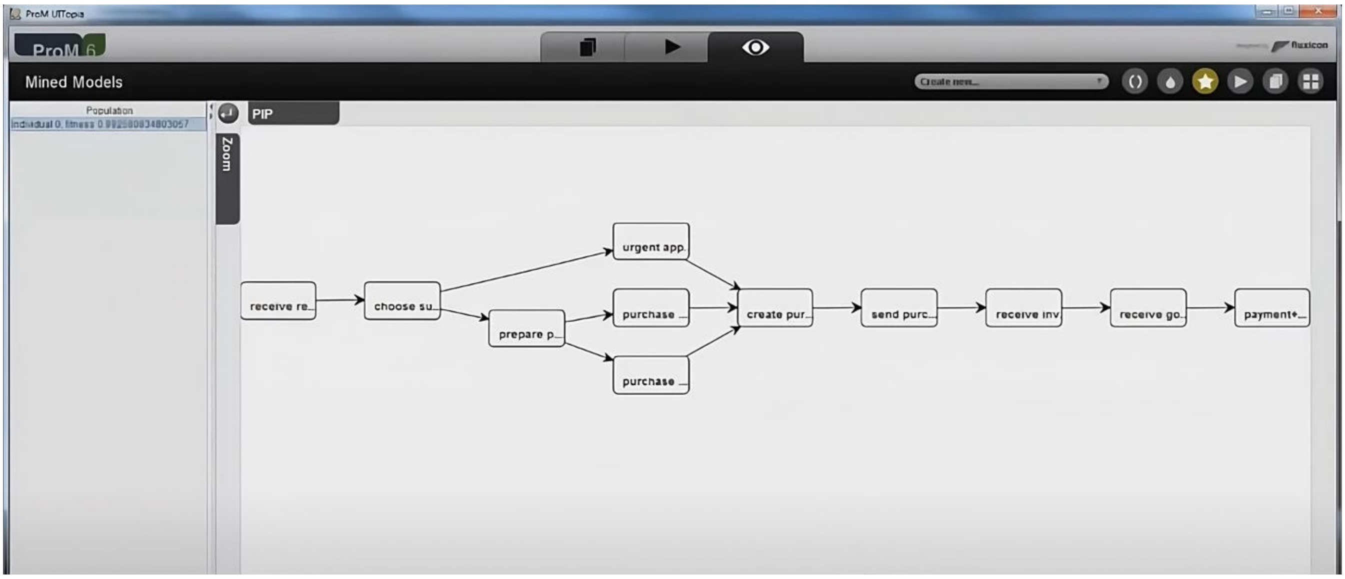


Fig. 7 Results of Flexible Heuristics Miner.

### B. *Analysis and Discussion (Sprints Retrospective)*

The results to be achieved with the integration of process mining practices in a software development project are highlighted in this section, allowing to verify that such integration can reveal important results for the project team and for the whole organization, since the processes discovered during the project may allow implementations, improvements, or adaptations of the organization's processes.

With the use of the ProM tool and the Flexible Heuristics Miner algorithm, it is possible to discover processes using event logs from various databases, since these event logs meet the restrictions of the algorithm. With the execution of the algorithm, it is possible to identify the result, as shown in Fig. 7, thus verifying a process model that can favor the implementation of this process identified as a standard process of the organization, or can validate an existing process used by the team in an ad hoc format.

The possibility of process discovery for the case of this project occurs separately, when considering the event logs obtained throughout the project through the Jenkins pipeline and Jira Software. Therefore, these findings favor processes of managerial aspects if considering the event logs from Jira Software and processes of technical/managerial aspects if considering the Jenkins event logs. Anyway, the processes can be identified from the event logs obtained based on the two tools.

It should be noted that other algorithms could be used with the ProM tool, such as: Inductive Visual Miner and Alpha Miner; and still others can be used if other tools, such as DISCO, can be considered.

Another relevant point to be highlighted regarding the use of process mining practices refers to the dynamics of the use of event logs, being able to use the tool and the algorithm with subsets of the event logs in order to verify the achieved results. Thus, for the case of this project, if one considers the third process mining sprint that could already count on a greater amount of event logs, the separation into subsets may somehow favor mining activities.

## VI. Conclusion

Software processes are extremely relevant to software projects, whether these are formalized and defined as a standard process by the organization, or whether these are informal and used randomly by project teams. However, the ability to determine processes is not a trivial task, even by software team interviews, questionnaires, or observations in loco. In this way, the use of process mining practices can favor the identification of processes to be used in different projects that are executed by the organization.

The discovery of processes through digital event logs stored in tools that are used during software development allow the discovery of processes. However, it presents relevant limitations, as these logs come from different tools, making it difficult to see integrated processes. In order to perform the discovery of integrated processes, there is a need for a better deepening and application of algorithms using specific process mining tools.

Above all, process mining practices can also favor compliance actions or verification of existing processes, since given that the organization presents a standard process, process mining can be used to verify the use of existing processes.

It is relevant for those involved in process management, specifically considering software processes, to work on future research in the development of a methodology that addresses the use of process mining practices integrated into a software project, regardless of the software methodology adopted for the project.

## Conflict of Interest

Authors declare that they do not have any conflict of interest.

## References

[1] Abrahamsson P, Salo O, Ronkainen J, Warsta J. Agile software development methods: Review and analysis. *arXiv preprint* arXiv:1709.08439. 2017 Sep 25. doi: 10.48550/arXiv.1709.08439.

[2] van Eck ML, Lu, X, Leemans SJ, van der Aalst WMP. PM$^2$: a process mining project methodology. *Advanced Information Systems Engineering*. CAiSE 2015. Springer, Cham. 2015. doi: 10.1007/978-3-319-19069-3_19.

[3] Lemos AM, Sabino CC, Lima RMF, Oliveira CAL. Using process mining in software development process management: A case study. *2011 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, pp. 1181-1186, Anchorage, AK, USA, 2011. doi: 10.1109/ICSMC.2011.6083858.

[4] van der Aalst WMP, Adriansyah A, De Medeiros AKA, Arcieri F, Baier T, Blickle T, et al. Process mining manifesto. *Business Process Management Workshops: BPM 2011 International Workshops*, Springer Berlin Heidelberg, pp. 169-194, Clermont-Ferrand, France, 2012. doi: 10.1007/978-3-642-28108-2_19.

[5] Rubin V, Günther CW, van der Aalst WMP, Kindler E, van Dongen BF, Schäfer W. Process mining framework for software processes. *International Conference on Software Process, ICSP 2007: Software Process Dynamics and Agility*, Springer, Berlin, Heidelberg, pp. 169-181, Minneapolis, MN, USA, 2007 May 19-20. doi.org/10.1007/978-3-540-72426-1_15.

[6] Poncin W, Serebrenik A, van den Brand M. Process Mining Software Repositories. *15th European Conference on Software Maintenance and Reengineering*, Oldenburg, Germany, pp. 5-14, 2011. doi: 10.1109/CSMR.2011.5.

[7] Keith B, Vega V. Process mining applications in software engineering. *Trends and Applications in Software Engineering, CIMPS 2016, Advances in Intelligent Systems and Computing*, Springer, Cham, vol 537, 2017. doi:10. 10.1007/978-3-319-48523-2_5.

[8] Fernandez-Llatas C, Lizondo A, Monton E, Benedi J-M, Traver V. Process Mining Methodology for Health Process Tracking Using Real-Time Indoor Location Systems. *Sensors* 2015;15(12):29821–40. doi: 10.3390/s151229769.

[9] Gupta M, Serebrenik A, Jalote P. Improving software maintenance using process mining and predictive analytics. *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, IEEE, pp. 681-686, Shanghai, China, 2017. doi: 10.1109/ICSME.2017.39.

[10] Kalliamvakou E, Gousios G, Blincoe K, Singer L, German DM, Damian D. The promises and perils of mining github. *Proceedings of the 11th Working Conference on Mining Software Repositories*, pp. 92-101, 2014 May. doi: 10.1145/2597073.2597074.

[11] Larrea VGV, Joubert W, Fuson CB. *Use of Continuous Integration Tools for Application Performance Monitoring* [Internet]. *2015.* Available from: https://www.osti.gov/biblio/1185993.

[12] Seth N, Khare R. ACI (Automated Continuous Integration) using Jenkins: Key for Successful Embedded Software Development. *International Conference on Recent Advances in Engineering & Computational Sciences (RAECS)*, IEEE, pp. 1-6, Chandigarh, India, 2015. doi: 10.1109/RAECS.2015.7453279.

[13] Ali M., Khan SU, Vasilakos AV. Security in cloud computing: Opportunities and challenges. *Information Sciences*. 2015; 305: 357-383. doi: 10.1016/j.ins.2015.01.025.

[14] Filion L, Daviot N, Le Bel JP, Gagnon M. Using Atlassian tools for efficient requirements management: An industrial case study. *Annual IEEE International Systems Conference (SysCon)*, IEEE, pp. 1-6, Montreal, QC, Canada, 2017, doi: 10.1109/SYSCON.2017.7934769.