Mapping Process Mining Techniques to Agile Software Development Perspectives

Cyrine Feres[1][a], Sonia Ayachi Ghannouchi[1][b] and Ricardo Martinho[2][c]
[1]*High Institute of Management of Sousse, University of Sousse, Tunisia*
[2]*INESCC - DL, ESTG, Polytechnic of Leiria, Portugal*

Keywords: Process Mining, Agile Software Development Processes, Mapping.

Abstract: Agile Software Development (ASD) processes have surfaced as an effective alternative for more efficient software project management. They concentrate on a set of informal best practices instead of a standardised process, making it difficult to determine the degree of real implementation in an organization. Process Mining (PM) can play a key role in such analysis by discovering the software development process model followed in a certain set of software projects, and by analysing event logs that report the projects' executed tasks. These discovered processes can then be compared to standardised ASD methods such as Scrum and eXtreme Programming (XP), and improved accordingly. Motivated by this, we present in this paper a literature review revealing the state of the art of Process Mining and its usage in ASD processes, but under a correlation between the three main research areas of PM (discovery, conformance, and enhancement), and the main ASD process perspectives including organisational/team, control-flow, quality, time, cost & risk, and data. We then analyse and discuss the results of this review quantitatively and qualitatively and prospect future opportunities for research accordingly.

1 INTRODUCTION

Over the last two decades, software engineers have been constantly looking for better ways to create high-quality software in a timely manner. The popularity of Agile Software Development (ASD) processes, combined with collaboration tools, has emerged as a flexible solution to these challenges (Erdem and Demirörs, 2017; Erdem et al, 2018).

As with other kinds of business processes, ASD processes can be analysed from several process perspectives. For instance, the *time* perspective in the Scrum framework is reflected by sprints with a fixed duration, daily scrum meetings, effort estimations set in sprint planning meetings and sprint reviews and retrospectives duration and scheduling. The *resources* perspective mainly includes Developers, the Scrum Master and the Product Owner, as well as their assignments to tasks. The *control-flow* perspective can be reflected by the sequence of activities performed for a certain software project. In a Scrum scenario, this could mean starting with prioritizing the user stories in the product backlog, picking them to the sprint backlog, developing them and showing them to the stakeholders, while collecting feedback and improvements for the next sprint.

Process Mining (PM) has been used successfully in a variety of fields, including software engineering and, particularly, ASD processes (Urrea-Contreras et al, 2021). In this case, PM is based on event logs that can be collected from software project management information systems and/or other data sources, and may include the project ID, the tasks executed, their corresponding user stories, the assigned team members, and their start and end timestamps, among other data. From here, PM techniques can address three major purposes: discovery, conformance checking and enhancement of ASD processes. Taking Scrum as an example, PM discovery techniques can be used to identify important metrics such as mean number of sprints per user story, most efficient

[a] https://orcid.org/0000-0002-6645-8490
[b] https://orcid.org/0000-0001-9583-9797
[c] https://orcid.org/0000-0003-1157-7510

Feres, C., Ghannouchi, S. and Martinho, R.
Mapping Process Mining Techniques to Agile Software Development Perspectives.
DOI: 10.5220/0011850500003464
In *Proceedings of the 18th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2023)*, pages 428-435
ISBN: 978-989-758-647-7; ISSN: 2184-4895

developer per type of user story, or simply the most common sequence of tasks performed to develop and finish a user story. Conformance checking can elicit deviations against a standard Scrum sequence of events, and enhancement can be reflected by adjusting developer assignments to certain types of tasks, or even to help estimate user story points.

We can find in literature systematic reviews and mappings on the usage of PM in ASD processes, such as the ones presented in Erdem and Demirörs (2017), Erdem et al (2018) and Arias et al (2018). Nevertheless, it is not obvious to understand, from these contributions, which ASD process perspectives have been under analyses with PM, particularly for which purpose. In this paper we perform a literature review on the use of PM in ASD processes, but in the form of a mapping between PM techniques and ASD process perspectives. In this way, we intent to provide an insight on how research has addressed the correlation between these two fields. This means that we'll be matching *discovery*, *conformance* and *enhancement* PM techniques with commonly addressed ASD process perspectives such as organisational/team, control-flow, quality, time, cost, risk, and data.

We provide insight into how such results were obtained by using the PRISMA method (Page et al, 2021) which consists in a set of recommended activities on a flow diagram, towards the development of a systematic review. Next, we present the main results under a matrix-based heat map (cross-reference between PM techniques and ASD process perspectives). We then analyse these correlations from both quantitative and qualitative points of view. Our goal is to identify the most and least studied correlations, as well as their specific applications and goals, in order to uncover research opportunities in these fields.

The paper is structured as follows: brief overviews of ASD processes and PM are given in section 2. The research method is presented in section 3, and our matrix-based analysis is illustrated in section 4, including quantitative and qualitative descriptions of the analysed research works. Finally, in section 5 we discuss the results and present future research perspectives on these themes.

2 BACKGROUND

In this section, we first provide an overview of ASD processes and their different perspectives. Then we highlight the main techniques used in PM and their purpose.

2.1 Agile Software Development Processes

Agile Software Development (ASD) processes are becoming more popular in the information systems field. Nowadays, a significant proportion of software organizations develop software that uses ASD processes (Garousi et al, 2015). ASD processes allow project managers and workers to conform to constantly changing contexts while not imposing rigid control. ASD projects are typically established by small teams over a shorter period of iterations (Beck et al, 2001). Customers are frequently delivered working software to test during the ASD process. As a result, there is a high probability that changes will occur throughout the development process, and can even be decided at late stages of development to ensure customer satisfaction (Marquez et al, 2018). The Manifesto for ASD processes recommends stakeholders to motivate agile teams, guarantees coordination and communication among teams and customers, and holds meetings to effectively and efficiently transmit information between themselves (Beck et al, 2001).

ASD processes can also be considered as a conjunction of process elements from several perspectives. For instance, the organizational/team perspective can deal with team members and their assignments to tasks.

The control-flow perspective is concerned in identifying the tasks executed and their corresponding sequences, while the time perspective can refer to the duration of events such as tasks, sprints and meetings. The quality perspective can address the quality of a software release measured through, for example, customer satisfaction assessment. Other commonly referred perspectives include cost, risk and data, which can be reflected in business values assigned to user stories, prioritising user stories that are more uncertain and under-defined, and managing documentation of the software product, respectively.

2.2 Process Mining

Process Mining (PM) emerged in the last decade with several research studies that have been carried out, and the trend is impressive. PM is a process management research field that analyses business processes using event logs as the starting point. It must include a minimum of three elements that describe the execution of activities: an identifier of the case for a certain activity executed (for instance, the software project to which it belongs), a label for

the name of the activity and its (start time) timestamp (Urrea-Contreras et al, 2021). Other information that can be part of an event log includes resources used to initiate or operate the activity (e.g.: operators/team member(s), roles, materials, equipment), cost, risk, quality, and data produced.

The aim of PM is to discover, monitor, and improve real processes (rather than assumed processes) by extracting data from readily available event logs in today's information systems and data sources (van der Aalst, 2016). These types of mining can be focused on several process perspectives such as time, control-flow, resources and case perspectives (van der Aalst, 2016). PM is useful for several reasons, namely (Van der Aalst, 2012):

- Provides insight into organisational business processes;
- Enables the organisation's business processes to be tested (whether processes are executed according to the rules and within boundaries);
- Helps in optimising and enhancing business processes and performance;
- Allows for the evaluation and improved decision-making for running cases in a certain business process.

These business processes can be diverse in their domain and evidently include ASD processes as well.

3 RESEARCH METHOD

In this section, we illustrate how we used the PRISMA statement[1] method to perform a literature review on the correlation of PM and ASD processes.

3.1 PRISMA Statement

Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) (Page et al, 2021) consists of a series of checks on a flow diagram (as illustrated in Figure 1), considering essential items for the development of a systematic review. Briefly, it foresees the definition of search strings and the choice of the research sources, the adoption of selection criteria unfolding the results of the search, and then the extraction of the different limitations of the related research works. Figure 1 shows the summary of results obtained from the application of the PRISMA statement to our domains of knowledge.

Regarding the databases considered for the search, ten were selected as data sources to collect and obtain the largest number of studies. These included: Springer Link, IEEE Xplore, Google Scholar, Science Direct, Web of Science, Base Search, Research Gate, Scopus, Scinapse, and Science.gov. For searching these databases, we considered the following search string: “Process mining” AND “Agile Software Development” OR “Process mining” AND “Agile Software” OR “Process mining” AND “Agile Software Lifecycle” OR “Process mining” AND “Scrum” OR “Process mining” AND “Extreme Programming” OR “Process Mining Perspectives” AND “Agile Software Development” OR “Process Mining” AND “Software Engineering” OR “Process Mining” AND “Business Process”. After collecting and counting the results, we divided the study selection process into two stages: 1) search results were evaluated by reading the title and abstract. The main criterion in this stage was the inclusion of words and context regarding PM and ASD processes; and 2) the remaining results were filtered, based on the following inclusion criteria:

- Scope related to PM in the context of ASD;
- Full text is available;
- Written in English;
- Published between 2009 and 2022.

Before the selection process was completed, the references of the selected papers were checked to find related papers and to increase the size of the result set.

Information from all papers was gathered in the form of an Excel file, resulting in 4444 papers + 4 duplicates (removed in the Screening section of PRISMA). The resulting papers were firstly screened based on the title (where we excluded 3315), and then by keywords and abstract (excluded 1129 more) resulting in 18 papers for a full-text reading. In the next section we provide details on these final results, considering their mapping onto ASD process perspectives and PM techniques.

4 SYSTEMATIC MAPPING

In this section, we begin by describing and exemplifying the benefits PM can bring to ASD processes, then we analyse the final 18 results from our PRISMA method.

[1] https://www.prisma-statement.org

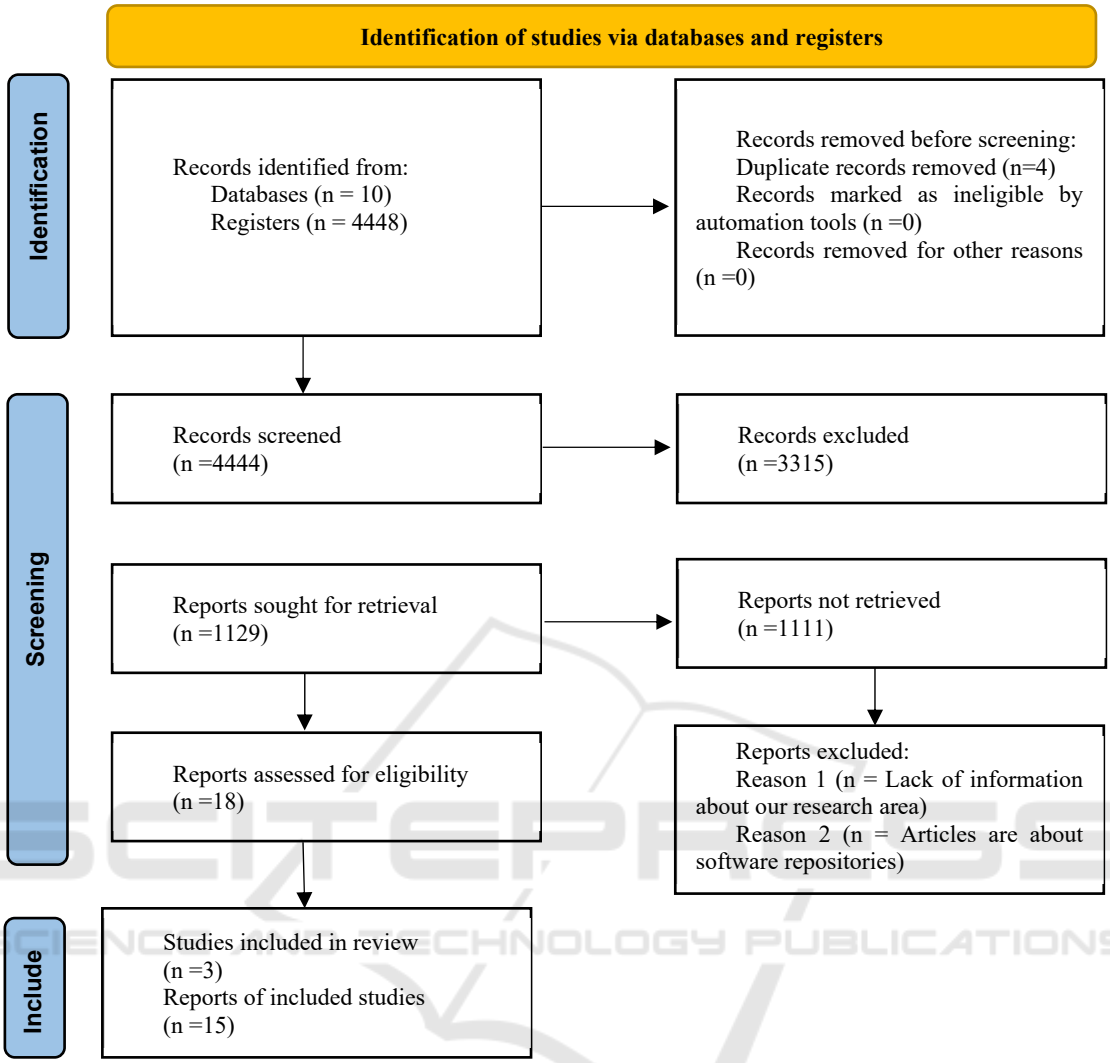

Figure 1: Literature review results found and filtered according to the PRISMA statement.

4.1 PM in ASD Processes

Process Mining offers tools for discovering agile processes used by agile teams to clarify an organization's reality, and can also be used to improve the process quality during agile software development processes (Erdem and Demirörs, 2017).

Currently, ASD processes/frameworks' best practices such as those represented by Scrum[2], Behavior Driven Development (BDD)[3], or eXtreme Programming (XP)[4] are commonly built into software project life cycle management tools to assist teams in carrying out their software projects. From here, event logs on the execution of these projects can be registered and later collected to be fed into Process Mining algorithms. These can then output several results, which include performance graphs illustrating control-flow and time-related values, common teams and their collaborations, or even frequency graphs to account for rework, cost, risk, and other quality attributes. As a side contribution of this paper, we present in Figure 2 a concrete illustration of how results from PM can help ASD processes. For this, we also present the well-known Scrum framework, decorated with some examples of results originated from the application of PM techniques, as well as the decisions they can support to enhance the Scrum ASD process. In this Figure, the main Scrum concepts are associated with examples of tuples in the following form:

[2] https://scrumguides.org/
[3] https://dannorth.net/introducing-bdd/
[4] http://www.extremeprogramming.org/

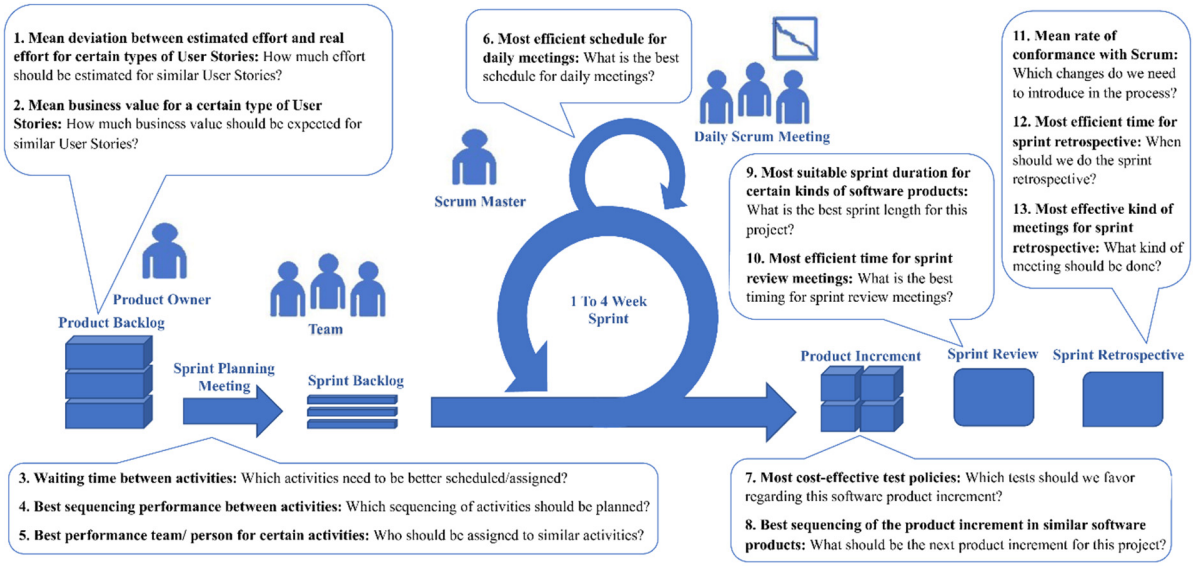

Figure 2: PM applied to the Scrum framework.

<Result obtained from PM>: *<decision that can be supported by this result>*

For instance, for the Product Backlog, we can measure the mean deviation between estimated effort and real effort for certain types of User Stories (tuple number 1 in Figure 2) so that we can estimate how much effort should be considered for similar User Stories. For tuple 2, we can obtain from PM the mean business value for a certain type of User Stories, to have a reference value for future ones.

For the Sprint Planning Meeting, we can consider the waiting time between similar activities in past projects (for instance, through a performance graph from PM), as well as the best sequencing performance between activities and best performance team for certain activities (tuples 3, 4 and 5), so that we can plan the next sprint accordingly.

For Daily Scrum Meetings (tuple 6), we can use, for instance, Multi-perspective PM techniques (Mannhardt et al, 2015) to retrieve the most efficient schedule for daily meetings, based on past projects' data (time of day and comparative amount of work done further).

For the Product Increment, we can measure the most cost-effective test policies, so that we can decide which kind of tests we should favour (tuple 7). Also, we can mine through PM the best performance sequencing of product increments in similar software projects, to better decide the next product increment (tuple 8).

Tuples 9 and 10 are related to the Sprint review event of Scrum. Similarly, we can measure the number of user stories that surpassed one sprint till concluded, to assess the most suitable sprint duration for similar projects (tuple 9).

This can be done, for instance, by using the Heuristics Miner from PM (Weijters et al, 2006), considering the sprint ID as the case identifier, and evaluating user stories that appear in more than one sprint. We can also determine the best timing for sprint reviews, based on this result, in conjunction with data relative to the week day/time of each sprint review event. This also applies in tuples 12 and 13 for Sprint retrospective events, in order to derive the time and kind of these meetings. Here, we can also use, for instance, the Inductive Visual Miner algorithm (Weijters et al, 2006) to analyse conformance and deviations from a standard Scrum process, to infer the changes in the process needed for a particular project (tuple 11).

4.2 Quantitative Analysis

In Table 1, we present these 18 research works in the form of a grey-shaded heat map, where darker cells represent more papers addressing a certain mapping of PM technique versus ASD process perspective. We can conclude that most of these works are focused on the *organisational/team* and *control-flow* perspectives of ASD process, and with a higher frequency on the application of PM *discovery* techniques (10 and 14 research works, respectively). The second most addressed correlations are those related to the *quality* and *time* perspectives, again with higher incidence on the application of discovery PM techniques (8 and 4 works, respectively). Less

referred correlations include the ones from the cost & risk and data perspectives, with 2 research works per correlation cell. In the following section, we provide deeper details on the exact contributions for each of these research works by PM techniques.

4.3 Qualitative Analysis

The most commonly application of PM to ASD is process discovery, with the purpose of uncovering the really accomplished processes in organisations.

Agile team activities can be collected to discover what is going on and how it is going on. Process discovery will be beneficial in extracting the steps taken by agile teams, required inputs to improvement, intermediate outputs developed within iterations, and roles that have evolved during development. This is advocated in the research works referred in the *discovery – organisational/team* correlation from Table 1. Caldeira and Abreu (2016) and Shani et al, (2019) focus on discovering the *time* perspectives which means the time spent on each activity, which activity is causing a bottleneck, what activity is done the most on a particular time, and if exists a particular moment where activities are slower to get done. This can indicate inefficiencies in the ASD process model, particularly in the early stages of software development. Activities that cause bottlenecks can be identified immediately with the event log streamed to the system. This will accelerate the entire development process, allowing software products to be delivered to customers more quickly.

Rubin et al. (2014b) discovered that the different interpretations of agile method rules by teams in an organization may result in interoperability issues between the organisation's projects. The works of Akman and Demirörs (2009) and Aalst (2015) focus on discovering the *control-flow* of the software process. This is achieved by analysing the order of activities and highlighting the most frequent activity paths in software development, as well as analysing how each task/activity follows each other in an event log and inferring a possible model for the behaviour captured in the observed process.

Caldeira et al. (2019) and Fauzi and Andreswari (2022) discuss that ASD perspectives such as *organisational/team* performance, *time*, and *control-flow* can reflect on the *cost and risk* of ASD processes. Urrea-Contreras et al. (2021) advocate, through a systematic literature review, that PM techniques are important for software development processes' management. The results illustrate a research gap in the successful execution of the case and time process perspectives. The systematic mapping study of Arias et al (2018) proves that PM discovery techniques (namely, performance analysis) were used in three different studies. To diagnose performance issues, a process model is discovered, and time information is annotated for this type of analysis. Lemos et al. (2011) explore the conformance checking of the formal software development process defined by a company and evaluate the application of PM for making such work less costly and more effective. The authors focused at the *control-flow* perspective in the presented conformance analysis. Rubin et al. (2014a) also correlate PM and ASD. This work describes a bottom-up approach for analyzing user and system runtime behaviour and improving software by using event logs (e.g., trace data) from a software system. They are concerned with development processes, but focused on the enhancement of software functionality through the use of PM techniques in an agile manner.

5 CONCLUSIONS

Motivated by the lack, to the best of our knowledge, of a systematic mapping/literature review correlating the two fields of ADS processes and PM, we used the PRISMA statement to perform a literature review. We identified 18 resulting research works, being the majority published within the last decade.

In Figure 1 we illustrated through Scrum how PM techniques can serve decision makers on the improvement of ASD process. We then correlated PM techniques with ASD process perspectives through a heat map (Table 1), to provide a better structured global analysis, as well as to identify most and least addressed research themes. From here, we can conclude that, within these research works, PM techniques applied are mostly related with discovery of the *organisational/team* and *control-flow* ASD process perspectives. Also, we can observe a significant difference between these correlations and the ones concerning the *cost & risk* and *data* perspectives. Even for the *time* perspective, less than ¼ of the research works check for the application of any PM technique.

Table 1: Correlation of Process Mining with Agile Software Development Process Perspectives.

Process Mining Techniques	Agile Software Development Process Perspectives					
	Organisational /Team	Control-Flow	Quality	Time	Cost & Risk	Data
Discovery	Erdem and Demirörs (2017), Marquez et al. (2018), Caldeira et al. (2019), Fauzi and Andreswari (2022), Rubin et al. (2014b), Urrea-Contreras et al. (2021), Caldeira and e Abreu (2016), Erdemet et al. (2018), Zayed and Farid (2016), Arias et al. (2018)	Rubin et al. (2014a), Erdem and Demirörs (2017), Akman and Demirörs (2009), Marquez et al. (2018), Caldeira et al. (2019), Aalst (2015), Fauzi and Andreswari (2022), Rubin et al. (2014b), Keithand Vega (2017), Urrea-Contreras et al. (2021), Caldeira and e Abreu (2016), Zayed and Farid (2016), Arias et al. (2018), Lemos et al. (2011)	Rubin et al. (2014a), Erdem and Demirörs (2017), Marquez et al. (2018), Caldeira et al. (2019), Rubin et al. (2014b), Keithand Vega (2017), Shani et al. (2019), Lemos et al. (2011)	Marquez et al. (2018), Urrea-Contreras et al. (2021), Caldeira and e Abreu (2016), Shani et al. (2019)	Caldeira et al. (2019), Fauzi and Andreswari. (2022)	Sebu and Ciocarlie (2014), Zayed and Farid (2016)
Conformance checking	Marquez et al. (2018), Caldeira et al. (2019), Fauzi and Andreswari (2022), Ardimento et al. (2019), Caldeira and e Abreu (2016), Erdemet et al. (2018)	Rubin et al. (2014a), Marquez et al. (2018), Caldeira et al. (2019), Aalst (2015), Fauzi and Andreswari (2022), Rubin et al. (2014b), Caldeira and e Abreu (2016), Lemos et al. (2011)	Rubin et al. (2014a), Caldeira et al. (2019), Rubin et al. (2014b), Shani et al. (2019), Lemos et al. (2011)	Marquez et al. (2018), Caldeira and e Abreu (2016), Shani et al. (2019)	Caldeira et al. (2019), Fauzi and Andreswari (2022)	Sebu and Ciocarlie (2014), Zayed and Farid (2016)
Enhancement	Marquez et al. (2018), Fauzi and Andreswari (2022), Ardimento et al. (2019), Rubin et al. (2014b), Caldeira and e Abreu (2016), Erdemet et al. (2018)	Rubin et al. (2014a), Marquez et al. (2018), Aalst (2015), Fauzi and Andreswari (2022), Rubin et al. (2014b), Keithand Vega (2017), Caldeira and e Abreu (2016)	Rubin et al. (2014a), Rubin et al. (2014b), Keithand Vega (2017), Shani et al. (2019)	Marquez et al. (2018), Caldeira and e Abreu (2016), Shani et al. (2019)	Fauzi and Andreswari (2022)	Sebu and Ciocarlie (2014)

Qualitatively, we can observe that the majority of these research studies aim to demonstrate that PM can be a useful tool to improve the implementation of ASD process perspectives, namely:

- *Organisational/team*: Enhancing the process model with organisational structures that combine social network analysis, mapping resource behaviours, user collaboration, and role analysis;
- *Control-flow*: Defining a method for examining how each task/activity follows the next in an event log and deducing a possible model for the behaviour recorded in the observable process;
- *Quality*: Addressing software quality as one of the main goals of ASD process to obtain a high level of customer satisfaction, which can be affected by time, team performance, and cost of the project;
- *Time*: Concerning the frequency of events and their timing and used them to predict remaining and analysing the processing time or duration of an activity;
- *Cost & risk*: Associating costs and risks mainly to planning activities, which can also be highly affected by scope, time, and quality during the project;
- *Data*: representing all the information consumed and produced during one release of a software product in ASD process.

Taking these considerations into account, this paper can be used as a reference and brief guide for

future researchers working in the cross-field of PM and ASD processes.

REFERENCES

Rubin, V., Lomazova, I., & Aalst, W. M. V. D. (2014). Agile development with software process mining. *In 2014 Int. Conf. on Software and System Process (pp. 70-74),* IEEE..

Erdem, S., & Demirörs, O. (2017). An exploratory study on usage of process mining in agile software development. *In 2017 Int. Conf. on Software Process Improvement and Capability Determination (pp. 187-196)*. Springer, Cham.

Akman, B., & Demirörs, O. (2009). Applicability of process discovery algorithms for software organizations. *In 35th Euromicro Conf. on Software Engineering and Advanced Applications (pp. 195-202)*. IEEE.

Sebu, M. L., & Ciocarlie, H. (2014). Applied process mining in software development. *In 2014 9th IEEE Int. Symp. on Applied Computational Intelligence and Informatics (SACI) (pp. 55-60)*. IEEE.

Marquez, R., Mira da Silva, M., & Ferreira, D. R. (2018). Assessing agile software development processes with process mining: a case study. *IEEE 20th Conference on Business Informatics (CBI),* IEEE.

Caldeira, J., e Abreu, F. B., Reis, J., & Cardoso, J. (2019). Assessing Software Development Teams' Efficiency using Process Mining. *In 2019 Int. Conf. on Process Mining (ICPM) (pp. 65-72)*. IEEE.

Aalst, W. V. D. (2015, August). Big software on the run: in vivo software analytics based on process mining (keynote). *In 2015 Int. Conf. on Software and System Process (pp. 1-5),* IEEE.

Fauzi, R., & Andreswari, R. (2022). Business process analysis of programmer job role in software development using process mining. Procedia Computer Science, 197, 701-708.

Ardimento, P., Bernardi, M. L., Cimitile, M., & Maggi, F. M. (2019). Evaluating coding behavior in software development processes: A process mining approach. *In 2019 IEEE/ACM Int. Conf. on Software and System Processes (ICSSP) (pp. 84-93)*. IEEE.

Keith, B., & Vega, V. (2017). Process mining applications in software engineering.In Trends and Applications in *Software Engineering: Proceedings of CIMPS 2016 5 (pp. 47-56)*. Springer International Publishing.

Rubin, V. A., Mitsyuk, A. A., Lomazova, I. A., & van der Aalst, W. M. (2014b). Process mining can be applied to software too!. *In 8th ACM/IEEE Int. Symp. On Empirical Software Engineering and Measurement (pp. 1-8),* IEEE.

Urrea-Contreras, S. J., Flores-Rios, B. L., Astorga-Vargas, M. A., & Ibarra-Esquer, J. E. (2021). Process Mining Perspectives in Software Engineering: A Systematic Literature Review. *In 2021 Mexican Int. Conf. on Computer Science (ENC) (pp. 1-8)*. IEEE.

Caldeira, J., & e Abreu, F. B. (2016). Software development process mining: Discovery, conformance checking and enhancement. *In 2016 10th Int. Conf. on the Quality of Information and Communications Technology (QUATIC) (pp. 254-259)*. IEEE.

Erdem, S., Demirörs, O., & Rabhi, F. (2018). Systematic mapping study on process mining in agile software development. *In Int. Conf. on Software Process Improvement and Capability Determination (pp. 289-299)*. Springer, Cham.

Zayed, M. A., & Farid, A. B. (2016). The discovery of the implemented software engineering process using process mining techniques. *Int. J. Adv. Comput. Sci. Appl, 1(7), 279-286.*

Shani, A. H. M., Sarno, R., Sungkono, K. R., & Wahyuni, C. S. (2019). Time performance evaluation of agile software development. *In 2019 Int. Seminar on Application for Technology of Information and Communication (iSemantic) (pp. 202-207)*. IEEE.

Arias, M., Marques, M. R., & Rojas, E. (2018). Using process mining in agile software development methodologies: a systematic mapping study. *In 2018 XLIV Latin American Computer Conf. (CLEI) (pp. 552-561)*. IEEE.

Lemos, A. M., Sabino, C. C., Lima, R. M., & Oliveira, C. A. (2011). Using process mining in software development process management: A case study. *In 2011 IEEE Int. Conf. on Systems, Man, and Cybernetics (pp. 1181-1186)*. IEEE.

Garousi, V., Coşkunçay, A., Betin-Can, A., & Demirörs, O. (2015). A survey of software engineering practices in Turkey. *Journal of Systems and Software, 108, 148-177.*

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). Manifesto for agile software development.

van der Aalst, W. M. (2016). Responsible data science: using event data in a “people friendly” manner. *In Int. Conf. on Enterprise Information Systems (pp. 3-28)*. Springer, Cham.

Van Der Aalst, W. (2012). Process mining. Communications of the ACM, 55(8), 76-83.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., ... & Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. Systematic reviews, 10(1), 1-11.

Weijters, A. J. M. M., van Der Aalst, W. M., & De Medeiros, A. A. (2006). Process mining with the heuristics miner-algorithm. *Technische Universiteit Eindhoven, Tech. Rep. WP, 166(July 2017), 1-34.*

Mannhardt, F. (2018). Multi-perspective Process Mining. In BPM (Dissertation/Demos/Industry) (pp. 41-45).