Process Mining Model Quality in Software Development Case Studies

An Analysis

Marianne Schmolke[1]

Abstract: With increasing digitalization speed, the need for knowledge about how Software Development can be improved, increases as well. Frequently encountered challenges are skill deficits, problematic behavior or activity sequences, and deviations from common and expected paths. Process Mining provides promising opportunities to remedy these issues. Process Mining applied to Software Development activities for example is an analytics method that can provide insights before a process has been executed as there is a huge amount of log files to be potentially analyzed. As promising case studies in this field have been conducted, there is a need to find quality measures to effectively learn from the results found. Therefore, a systematic analysis has been applied to the relevant sources using the relevant criteria of the Comprehensive Process Model Quality Framework (CPMQF).

Keywords: Process Analytics, CPMQF, Software Engineering, Model Quality Measures

1 Introduction

Practical informatics is a field which is expected to produce solutions to real-word problems. Software development in practical informatics therefore aims at improving certain use case scenarios and classically at producing viable software products. As this is a primary focus, the quality of case study process models seems to have less importance. As technologies have come up to ease the analysis of large data amounts, e.g. log data in a significant manner, this priority does not seem applicable anymore.

This paper aims at providing guidelines for good Process Mining Case Study Models in order to enrich their fit-for-use-level. In case of a high fit-for-use level models can be reused on a strong foundation and explanations based on these models increase their impact for the scenarios in question.

1 Hochschule Wismar, Abteilung Wirtschaftsinformatik, Philipp-Müller-Str. 14, 23966 Wismar, Marianne.Schmolke@hs-wismar.de

https://doi.org/10.30844/AKWI_2022_09

A powerful instrument of automatically constructing process models form log data is Process Mining. Generally, Process Mining aims at building a process model in order to describe the behavior contained in event logs of information systems ([Va05]). Event logs are produced by process-aware information systems (e.g. Workflow Management Systems). Usually, these event logs contain information about the start and completion of process steps together with context data, e.g. who executes a task and what is necessary to execute the task.

[Ca20] found that only 1 study (3.12%) in their Literature Review on Software Development Analytics in Practice used Process Mining methods and tools.

That means that there hasn't been a sufficient amount of studies indicating empirical quality as result of collective experience in the Process Mining Analytics domain.

2 Related work

2.1 Overview of Secondary Studies

A set of literature analyses has been conducted in order to structure the field:

[Ru07] focus on technical possibilities of mining software process information. They work with abstracted and algorithmically processed event logs. Their output is the characterization of developers in the software engineering process as well as insights about problematic behavior or activity sequences.

[No16] analyze classical Process Mining applications. They extract hidden information about the Software Development process and analyze event logs of the types: problem report, change request, feature request and information request. They then identify typical application fields, case study categories and the challenges: Process knowledge, data quality, tool integration, result evaluation and usability.

[Ca16] conduct a meta-analysis on Process Mining in Software Development. They identify high or low detail and accurateness of Software Development processes and investigate process and social network patterns of relevance for generating improvement suggestions.

[Do17] conduct a systematic mapping study on Process Mining Software Processes. They analyze 40 papers dealing with the relevant research topics which can be classified into Micro and Macro Process papers. They aim at identifying the structure of and connections between core Process Mining Software Process elements.

[Ur21] have the goal to identify the perspectives of Process Mining, its definitions, the

application of Process Mining within the Software Engineering and successful practical implementations in the Software Development process. They analyze 12 papers, most importantly complying to the search criteria: (“Process Mining” OR “Process Mining Perspective” OR “Software Engineering”) AND (“Control-flow perspective” OR “Organizational perspective” OR “Case perspective” OR “Time perspective”). As a core result they extract several types of definitions and degrees of practical application.

An analysis which has not been conducted is the application of a process model framework particularly to Process Mining Software Development process models. To close this gap is the overall goal of this paper.

2.2 Detailed findings from primary studies

The body of literature encompasses a set of findings regarding the well-founded application and interpretation of Process Mining models in Software Development:

The approach was mostly evaluated as beneficial to Process Management System (PMS) implementation conditions. [Ru07] identified the possibility of a ‘faster PMS creation’. [Ru14b] suggested to embed the model into the development lifecycle due to effectiveness, [Ca19] found that their model accuracy was sufficient. The other works did at least not falsify this assumption and presented theoretical or practical benefits e.g. as optimization examples.

General challenges and advantages of Process Mining Software Development were inter alia:

- [Ma18] e.g. found that it is hard to evaluate if a team is self-organized based on the information contained in an event log. There is still the need for more heuristics to identify typical log characteristics of self-organized teams.

- [Ma21] e.g. found that the longest path of executive deletion commits had to be done manually. They as well-found similar sequences of commits. Additionally, most and least used activities and traces easy to find.

The optimization of developing environments can inter alia be supported by these findings: [Po11] recommend the separation between the preprocessing and the analysis step. The reuse of analysis techniques is recommended.

- [Se14] found that the identification of important user roles, ‘check requirement’, ‘define specification’, ‘define solution’, ‘check impact’, ‘define testing strategy’ need to be mandatory. Additionally, the context and details need to be understood.

Insights regarding the developer learner models are:

- [Ca19] found that the more proficient a team the less complex their models.
- [Ca21] found these developer fingerprints: high performers: cautious coder, cautious coder/test skipper, insecure/testers, insecure/debuggers, balanced coders/confident; low performers: general coding limitations (2), limited python/algorithmic skills (2).

Specifically, teams using Scrum could profit from the following workflow optimizations. [Ma18] found that loops deserve special attention and identified the possibility to:

- check whether all issues are prioritized and assigned to some resources;
- identify Scrum roles, even though they were not recorded in the case-handling system by analyzing the organizational perspective;
- assess cross-functionality of the team based on patterns in a dotted chart (they were evaluated with the project manager).

2.3 Increasing Process Mining Model Result Quality

Regarding the more technical perspective these works have to be considered, as they provide practical guidelines regarding (partly specific) thresholds: [Ze18] find that there is no specific threshold most suitable for sequence encoding, as this greatly depends on the event log. They observe that using lower threshold values, e.g. 0–0.4, leads to less complex models. They recommend to use a lower threshold value first, which also reduces computation time due to a smaller constraint body size, and based on the obtained result increase or decrease the threshold value if necessary. Recommendations for choosing the right dependency measure, frequency and best dependency measure applying the particular approach of the Heuristic Miner can be found in [We06].[2]

2 A dependency measure above the dependency threshold value [WE06]; frequency higher than the positive observations threshold value (ibid.), and a dependency measure for which the difference regarding the best dependency measure is below the relative to best threshold value (ibid.).

An example of applying a framework to assess Process Mining result quality is [Fa22]. They apply the SEQUAL framework to analyze whether syntactic, semantic, or pragmatic issues occur including deadlocks, lack of synchronization, superfluous activity, labeling issues, line crossing, and crooked alignment. The approach used here which is elaborated in the next chapter is based on the model extracted from previous works in [Me18].

3 Approach

3.1 Applied Framework

The focus lies on papers where Process Mining and Software Development are an essential component of the article – in the sense that the terms appear in the document title.

The criteria of the Comprehensive Process Model Quality Framework are assessed regarding their applicability to the Process Mining domain and then applied to the models found according to these search criteria:

Reviewed (full paper) English articles containing at least one Process Model including the terms “Process Mining” AND “Software Development” OR “Software Engineering” in Title from or key words from 2007 on from the following databases: ACM Digital Library (dl.acm.org), IEEE Xplore (ieeexplore.ieee.org), Proquest (proquest.com), ScienceDirect (sciencedirect.com), Wiley Online Library (onlinelibrary.wiley.com), Springer (link.springer.com) and Google Scholar (scholar.google.com).

3.2 Model analysis

The model developed in [Me18] encompasses the state-of-the-art body of knowledge regarding process model quality.[3] The aim is to extract the criteria applicable to Process Mining models generated on a scientific basis.

	Title	Publication	Year	Source	Ref
a)	Using Process Mining in Software Development	Journal	‘11	IEEE	[Le11]
b)	Conformance analysis on Software Development: an experience with process mining	Journal	‘11	Int. J. of Business Process Integration and Management	[Cr11]
c)	Process Mining Multiple Repositories for Software Defect Resolution from Control and Organizational Perspective	Conference	14	MSR 2014	[Gu14]

Tab. 1: Papers (P) I

3 High resolution PDF of the model: https://www.janclaes.info/downloads/CPMQF.pdf

	Title	Publication	Year	Source	Ref
d)	Process Mining support for Capability Maturity Model Integration-based software process assessment, in principle and in practice	Journal	14	J. Softw.: Evol. and Proc.	[Sa14]
e)	Applied process mining in Software Development	Conference	14	SACI	[Se14]
f)	The Discovery of the Implemented Software Engineering Process Using Process Mining Techniques	Journal	16	IJACSA	[Za16]
g)	An Exploratory Study on Usage of Process / Mining in Agile Software Development	Conference	17	SPICE	[Er17]
h)	A new perspective on Process-oriented Software Engineering based on BPMN Process Mining	Conference	17	AITEM	[Ca17]
i)	Assessing Agile Software Development Processes with Process Mining: A Case Study	Conference	18	IEEE Business Informatics	[Ma18]
j)	Evaluating coding behavior in Software Development processes: A process mining approach	Conference	19	ICSSP	[Ar19]
k)	Assessing Software Development Teams' Efficiency using Process Mining	Conference	19	ICPM	[Ca19]
l)	Profiling Software Developers with Process Mining and N-Gram Language Models	Journal	21	J. o. Systems and Software	[Ca21]
m)	Using process mining for Git log analysis of projects in a Software Development course	Journal	21	Educational and Information Technologies	[Ma21]

Tab. 2: Papers (P) II

Title	Publication	Year	Source	Ref	Title
n)	Application of Constructive Modelling and Process Mining Approaches to the Study of Source Code Development in Software Engineering Courses	Journal	21	J. o. Communications Software and Systems	[Sh21]
o)	Business process analysis of programmer job role in Software Development using process mining	Conference	22	ISICO	[Fa22]

Tab. 3: Papers (P) III

Regarding the papers not presented in the selection (Tab. 1 - Tab. 3): [Ru14_a] illustrates how Process Mining models can look like in agile development and [Ru14_b] in Software Development in general without focusing on the specific outcome of a model.

In [Sa09] and [Ba18] no process model has been generated. [Po11] does not stress the software engineering/ development perspective. These terms do not appear in title or key words in contrast e.g. as in [Gu14].

These criteria were not assessed: Proactive modeling is not tested in detail – as the recommendation to use Process Mining is being met by all analyzed studies. The syntactical rules of Process Mining methods are applied, i.e. they are not questioned regarding their quality. The optional guidelines are left out in order to focus on the mandatory guidelines.

The following criteria (C) were assessed from which a selection was kept for the analysis:

10 Process Modeling Guidelines – Error probability and understandability

1. Use no more than 31 nodes -> applicable
2. Use no more than 3 in-or outputs per connector -> not applicable
3. Use no more than 2 start and end event -> fulfilled by all models
4. Model as structured as possible -> mechanism of automatic model generation
5. Use design patterns to avoid mismatch -> Process Mining can discover design patterns, e.g. [Ba03] but the models are not adjusted by using design patters. That means, this criterion does not apply.
6. Avoid OR-joins and OR-splits -> not applicable
7. Minimize the heterogeneity of connectors -> not applicable
8. Minimize the level of concurrency -> not applicable
9. Use verb-object activity labels -> applicable
10. Decompose a model with more than 31 nodes -> not applicable

These criteria were extracted:

1. Use no more than 31 nodes
9. Use verb-object activity labels

C|P	a)	b)	c)	d)	e)	f)	g)	h)	i)	j)	k)	l)	m)	n)	o)
1	y	x	y	y	y	y	x	y	y	x	y	y	y	y	y
9	0[4]	y	x	x	x	y	y	x	x	x	y	x	y	y	y

Tab. 4: Applicable Criteria; legend: x=not fulfilled, 0=not applicable y=fulfilled

These criteria sets were not assessed: The Gateway Complexity Rules cannot be influenced in a Process Mining model. The gateways that are mined have no alternative in the representation.

Regarding the patterns used, the explanation given regarding criterion 5 from the "Process Modeling Guidelines – Error probability and understandability" applies.

"Reaching GoM – Basic guidelines -> Syntactic correctness" does not apply as assessment criterion to Process Mining. The models have their own validation mechanisms.

1. Document the meta-model. The overall meta-model for Process Mining can be found in [Po11] modified form [Do11].
2. Use a tool for automatic syntactic checks -> Does not apply to Process Mining where the model already is automatically generated.

Reaching GoM – Basic guidelines -> Semantic correctness

1. Use clearly defined domains and scopes -> applicable
2. Use internal and external feedback loops -> applicable
3. Try to capture the 'soft' process-related issues -> applicable

C|P	a)	b)	c)	d)	e)	f)	g)	h)	i)	j)	k)	l)	m)	n)	o)
1	y	x	y	0	y	y	y	x	y	y	y	y	y	y	y
2	x	x	y	y	y	y	y	y	y	y	y	y	y	y	y
3	x	x	y	y	y	y	y	y	y	y	y	y	y	y	x

Tab. 5: Applicable Criteria
x=not fulfilled, 0=not applicable y=fulfilled

4 Markov-Chain notation

In the following, the models found in the papers listed in (Tab. 1 - Tab. 3) are assessed.

[Le11] (a) explore the domain of a Software Development Process in a large-sized Brazilian Software Company. The scope is to exclude the performance perspective due to lack of duration data of the logs. As this is not the focus of the study, it does not cause issues. Regarding the internal and external feedback loops (C2) [Le11] already reflect the gap: "A set of activities being executed in a process instance are only a subset of the activities defined in the formal process. So, a deeper conformance analysis is still required." Therefore, this criterion is not fulfilled. 'Soft' process-related issues can be interpreted as less quantifiable, i.e. perceptional and social aspects, mostly addressed by change management initiatives in software projects. [Le11] relate to rather cost-related issues. The category 3 could have been considered more thoroughly.

[Cr11] (b) clearly state that they mainly want to obtain an understanding of the challenges involved in the process. They refer to a specific Software Development project but do not abstract the investigated domain or scope. The authors thoroughly reflect their results which confronted them with certain interpretation challenges. They state that further investigation is needed. Due to the lack of evidence in the model, conclusions on 'soft' process-related issues were not possible.

[Gu14] (c) discover a runtime process model for a bug resolution process spanning three repositories using the process mining tool, Disco, and conduct process performance and efficiency analysis. They identify bottlenecks, and detect basic and composite anti-patterns. In addition to control flow analysis, they mine event logs to perform an organizational analysis and discover metrics such as the handover of work, subcontracting, joint cases and joint activities. Regarding C2 they use a broad selection of metrics to test the validity of the model which can be seen as an equivalent to feedback cycles. 'Soft' process-related issues are addressed e.g. by differentiating generalists and specialists and observing that social performers contribute more actively.

[Sa14] (d) deliberately choose a broad scope using 'complexity' of the selected processes as one criterion. This approach results in a more general model than if the scope definition was more restricted. Therefore, this criterion is not applicable here. Regarding the feedback, several other analyses for additional projects were conducted in order to validate their model. They closely observe the corresponding real world process. They used their insights for actively enabling the development team considering the 'soft' process-related issues.

[Se14] (e) analyze the case of a software product release involving an external company. They explain their scope in detail and define all relevant elements clearly. Their real-world model is statistically analyzed in a way that the internal feedback criterion is fulfilled. They address 'soft' process-related issues by e.g. identifying training necessities and discovering that the support team did not ensure if the project team has understood the context or the details of the requirements.

[Za16] (f) focus on a subset of the project life cycle by the MS Visual Studio Scrum 3.0, i.e. the work items: feature, product backlog item, bug, and task. They apply quality dimensions to the model (Pareto Front model). The log is well described according to accepted thresholds whereas much more behavior than recorded can occur. The 'soft' process-related issues are discovered by interpreting correlations. It is assumed that the scrum team is not adhering to the process definition. Whereas specific work items have to be analyzed individually.

[Er17] (g) present a model of an agile Scrum team executing Product Backlog Items, i.e. user stories or specifications inter alia. Scope and definition are clear.

The relevant patterns are analyzed by the team as a feedback mechanism and 'soft' process-related issues are considered, e.g. the analysis is used as an approach to reflect one's own work as an agile team member.

[Ca17] (h) use the domain of a manufacturing company to conduct a conformance analysis regarding an initially BPMN-mined process. They do not name the scope of the initially mined model. Still, their conformance testing is conducted in a mainly quantitative manner. They conclude that nonconformities can be difficult to detect if only predefined KPIs are used to control the process. Nevertheless, the 'soft' process-related issues, such as, discussing the warehouse's personnel oversight ('can generate a big delay'+ 'probably represents a local problem') are considered.

[Ma18] (i) analyze two projects in an IT organization providing software products and services whereas Jira Software and Scrum are used. They apply a set of mining methods to test the models. They list several conclusions which also include 'soft' process-related issues.

[Ar19] (j) create high- and low-performer models of Software Development students of an advanced Java course. They use grading as a measurement to classify the students with a certain mined process profile. They consider 'soft' process-related issues by e.g. pointing out graphical feedback possibilities.

[Ca19] (k) create a model of practitioner activities executed in an IDE, including its artifacts and additional details on the ecosystem of components supporting the process. They use the internal and external validity criterion to assess the quality of their results and find clear threats to validity, i.e. idle time could not be interpreted and due to study organizational reasons the students might not have been the adequate surrogates. They clearly find 'soft' process-related issues, e.g.: the less complexity in the models, the more proficient students were in the task.

[Ca21] (l) create developer profiles according their development process.

They use N-gram language models and text mining in addition to the Process Mining models. The developer classifications include 'soft' process-related issues.

[Ma21] (m) create models of the project development process from the perspective of Git log attributes. Their feedback method was a manual analysis of the discovered model. They find 'soft' process-related issues such as inadequate distribution of work as a reason for deadline issues.

[Sh21] (n) construct a model about the task to write a program to calculate the minimum number of coins needed to give change to a customer. They use a formalized approach including code history and a direct experiment with students. They consider 'soft' process-related issues, e.g. factors complicating the program development process- which are leaving certain decisions to the teacher in the end (i.e.: grading).

[Fa22] (o) work on finding out the flow carried out by programmers in Software Development courses. They apply the SEQUAL framework to assess the quality of the process model. E.g. one of their main result components (issues of business process quality assurance) does not consider 'soft' process-related issues – as it is not in their scope.

Reaching GoM (Guidelines of Modeling) – Basic guidelines -> Relevance

1. Have sufficient constructs in the meta-model to represent the elements of the real world -> Applies to Process Mining automatically as the real-world processes are the center of the mining activity.
2. Use continuous feedback loops -> The constant repetition of mining activities can be considered as feedback loops.

Reaching GoM – Basic guidelines -> Economic efficiency

1. Use process scope definitions and clear objectives and targets -> similar to C1 of "Reaching GoM – Basic guidelines -> Semantic correctness". The objective and target are similar throughout all papers as they are all related to process mining in software engineering – aiming at finding optimization scenarios.
2. Use business frameworks -> Frameworks such as Porter's 5 Forces Model or BCG Growth-Share Matrix inter alia are clearly related to classical business goals. Optimizing with Process Mining is not only limited to these. Optimization goals in Software Engineering cannot be clearly categorized into traditional business frameworks. E.g. software architectures can be assessed in terms of technical debt ([Li19]), indicating the long-term vision on its profitability. Clearly quality can often be valued opposed to a short-term cost-benefit ratio.
3. Select only relevant users to participate in feedback loops->This criterion does not apply because the feedback loops are rather automatized by the mining approach.
4. Re-use models -> This is a core element of Process Mining.
5. Use reference models -> This is a core element of Process Mining.
6. Use state of the art modeling tools -> Different contexts require different mining algorithms. They are chosen for optimization reasons which comply to the state-of-the-art requirement.

Summary of applicable criteria:

1. ***Use no more than 31 nodes***
9. ***Use verb-object activity labels***
1. Use clearly defined domains and scopes
2. Use internal and external feedback loops
3. Try to capture the 'soft' process-related issues

C|P	a)	b)	c)	d)	e)	f)	g)	h)	i)	j)	k)	l)	m)	n)	o)
1	y	x	y	y	y	**y**	x	y	y	x	**y**	y	**y**	**y**	y
9	0^5	y	x	x	x	**y**	y	x	x	x	**y**	x	**y**	**y**	y
1	y	x	y	0	y	**y**	y	x	y	y	**y**	y	**y**	**y**	y
2	x	x	y	y	y	**y**	y	y	y	y	**y**	y	**y**	**y**	y
3	x	x	y	y	y	**y**	y	y	y	y	**y**	y	**y**	**y**	x

Tab. 6: These criteria where assessed per case study model

The criteria automatically fulfilled by applying Process Mining have are:

- Use no more than 2 start and end event
- Model as structured as possible
- Document the meta-model
- Have sufficient constructs in the meta-model to represent the elements of the real world
- Use continuous feedback loops
- Re-use models
- Use reference models
- Use state of the art modeling tools

An applicable quality indicator is to have at least four criteria fulfilled. This applies to the majority of models. There does not seem to be a connection between time of publication and quality although more recent studies generate high quality (5 times "y") models.

5 Markov-Chain notation

4 Conclusions

In general, it can be stated that the Process Mining approach itself already embodies characteristics that increase the quality of the resulting models. Regarding the criteria which can be assessed particularly per model it can be stated that they are not applicable to create a clear order of the selected studies. Nevertheless, they help to frame the applied model in order to predict their practical relevance and weigh the conclusions drawn in interpretation of the connected results. A particular conclusion can be drawn from assessing the work of [Fa22]: They work on finding out the flow carried out by programmers in Software Development courses. They apply the SEQUAL framework to assess the quality of the process

model. E.g. one of their main result components (issues of business process quality assurance) does not consider 'soft' process-related issues – as it is not in their scope. This finding can be interpreted as a hint to further differentiate the term 'soft' process-related issues, e.g. to also define 'counter-intuitive' process-related issues.

References

[Ar19] Ardimento, P., Bernardi, M. L., Cimitile, M., Maggi, F. M.: Evaluating coding behavior in Software Development processes: A process mining approach. In: IEEE/ACM International Conference on Software and System Processes (ICSSP) 2019, pp. 84-93.

[Ba18] Bala, Saimir, Mendling, J.: Monitoring the Software Development process with process mining. In: International Symposium on Business Modeling and Software Design, Cham 2018. Springer.

[Ca16] Caldeira, J., Brito e Abreu, F.: Software development process mining: Discovery, conformance checking and enhancement. In: IEEE/10th International Conference on the Quality of Information and Communications Technology (QUATIC) 2016, pp. 254-259.

[Ca17] Carmignani, G., Cimino, M. G. C. A., Failli, F., Santone, A., Vaglini, G.:A new perspective on Process-oriented Software Engineering based on BPMN Process Mining. In: XIII Convegno dell'Associazione Italiana di Tecnologia Meccanica AITEM 2017, pp. 1-10.

[Ca19] Caldeira, J., Brito e Abreu, F., Reis, J., Cardoso, J.: Assessing Software Development Teams' Efficiency using Process Mining. 10.1109/ICPM.2019.00020, pp. 65-72. 2019.

[Ca20] Caldeira, J., Cardoso, J., Ribeiro, R., Werner, C.: Software Development Analytics in Practice: A Systematic Literature. arXiv preprint arXiv:2101.06733. 2020

[Ca21] Caldeira, J., Brito e Abreu, F., Reis, J., Cardoso: Profiling Software Developers with Process Mining and N-Gram Language Models. arXiv preprint arXiv:2101.06733. 2021.

[Cr11] Da Cruz, J. I. B., Ruiz, D. D. Conformance Analysis on Software Development: An Experience with Process Mining. International Journal of Business Process Integration and Management, 5(2), 2011, pp. 109-120.

[Do17] Dong, L., Liu, B., Li, Z., Wu, O., Babar, M. A., Xue, B. A Mapping Study on Mining Software Process. In: IEEE/24th Asia-Pacific Software Engineering Conference (APSEC) 2017, pp. 51-60.

[Er17] Erdem, S., Demirors, O.: An Exploratory Study on Usage of Process Mining in Agile Software Development. 187-196. 10.1007/978-3-319-67383-7_14, 2017.

[Fa22] Fauzi, R., Andreswari, R.: Business Process Analysis of Programmer Job Role in Software Development Using Process Mining. Procedia Computer Science, 197, pp. 701-708, 2022.

[Gu14] Gupta, M., Sureka, A., Padmanabhuni, S.: Process Mining Multiple Repositories for Software Defect Resolution from Control and Organizational Perspective. In: Proceedings of the 11th Working Conference on Mining Software Repositories 2014, pp. 122-131.

[Le11] Lemos, A. M., Sabino, C. C., Lima, R. M., & Oliveira, C. A.: Using Process Mining in Software Development Process Management: A case study. In: 2011 IEEE International Conference on Systems, Man, and Cybernetics, pp. 1181-1186.

[Li19] Lilienthal, Carola. Langlebige Software-Architekturen: Technische Schulden analysieren, begrenzen und abbauen. dpunkt. verlag, 2019.

[Ma18] Marques, R., da Silva, M. M., Ferreira, D. R.: Assessing Agile Software Development Processes with Process Mining: A case study. In: IEEE 20th Conference on Business Informatics (CBI) 2018, Vol. 1, pp. 109-118.

[Ma21] Macak, M., Kruzelova, D., Chren, S., Buhnova, B.: Using process mining for Git log analysis of projects in a Software Development course. Education and Information Technologies, 26(5), 5939-5969, 2021.

[Me18] De Meyer, P., Claes, J.: An Overview of Process Model Quality Literature - The Comprehensive Process Model Quality Framework. arXiv preprint arXiv:1808.07930, 2018.

[No16] Norambuena, BK, Zepeda, VV: Process Mining Applications in Software Engineering. Risti-Revista Iberica de Sistemas e Tecnologias de Informacao 21, 2016, 51-66.

[Po11] Poncin, W., Serebrenik, A., Van Den Brand, M.: Process Mining software repositories. In: IEEE/ 15th European conference on software maintenance and reengineering 2011, pp. 5-14.

[Ru07] Rubin, V., Günther, C., van der Aalst, W. M., Kindler, E., Dongen, B., Schäfer, W. Process Mining Framework for Software Processes. 10.1007/978-3-540-72426-1_15, pp. 169-181, 2007.

[Ru14a] Rubin, V. A., Mitsyuk, A. A., Lomazova, I. A., van der Aalst, W. M.: Process Mining Can be Applied to Software Too!. In Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement 2014, pp. 1-8.

[Ru14b] Rubin, V. A., Mitsyuk, A. A., Lomazova, I. A., van der Aalst, W. M.: Agile development with software process mining. In Proceedings of the 2014 International Conference on Software and System Process, pp. 70-74.

[Sa14] Samalikova, J., Kusters, R. J., Trienekens, J. J., & Weijters, A. J. M. M.: Process Mining support for Capability Maturity Model Integration-based software process assessment, in principle and in practice. Journal of Software: Evolution and Process, 26(7), pp. 714-728, 2014.

[Se14] Sebu, M. L., Ciocarlie, H.: Applied process mining in Software Development. In IEEE 9th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI) 2014, pp. 55-60.[Sh21] Shynkarenko, V., Zhevaho, O.: Application of Constructive Modeling and Process Mining Approaches to the Study of Source Code Development in Software Engineering Courses. Journal of Communications Software and Systems 2021, 17(4), pp342-349.

[Ur21] Urrea-Contreras, Silvia Jaqueline, et al. “Process Mining Perspectives in Software Engineering: A Systematic Literature Review.” 2021 Mexican International Conference on Computer Science (ENC). IEEE, 2021.

[Va05] Van der Aalst, W. M., Medeiros, A. K., Weijters, A. J.: Genetic process mining. In: International conference on application and theory of petri nets 2005 (pp. 48-69). Springer, Berlin, Heidelberg.

[We06] Weijters, A. J. M. M., van Der Aalst, W. M., & De Medeiros, A. A.: Process Mining with the Heuristics Miner-Algorithm. Technische Universiteit Eindhoven, Tech. Rep. WP, 166 (July 2017) , pp. 1-34, 2006.

[Za16] Zayed, M. A., Farid, A. B.. The discovery of the implemented software engineering process using process mining techniques. Int. J. Adv. Comput. Sci. Appl, 1(7), pp. 279-286, 2016.

[Ze18] van Zelst, S. J., van Dongen, B. F., van der Aalst, W. M., Verbeek, H. M. W.: Discovering Workflow Nets Using Integer Linear Programming. Computing, 100(5), pp. 529-556.