

An Exploratory Study on Usage of Process Mining in Agile Software Development

Sezen Erdem^{1(✉)} and Onur Demirörs^{2,3}

¹ ASELSAN INC., Ankara, Turkey
erdem@aselsan.com.tr

² Computer Engineering Department,
Izmir Institute of Technology, Izmir, Turkey
demirors@metu.edu.tr

³ School of Computer Science and Engineering,
University of New South Wales, Sydney, Australia

Abstract. Agile software development methods have become popular in the software development field during the last decade. Majority of software organizations develop or claim to develop software based on agile methods. Process mining is a process management technique that allows for the analysis of business processes based on the event logs. The aim of process mining is to discover, monitor and improve real processes, but not assumed processes, by extracting knowledge from event logs readily available in information systems. Process mining can be used to discover agile processes followed in organizations/projects to determine the actual processes followed. Process mining can also establish the necessary evidences for assessing or measuring the agility of organizations. This study explores the usability of process mining methods in agile software development context. The results of an exploratory case study on using process mining techniques in a software project managed by Scrum are depicted. We also discuss the benefits of the process mining techniques used and compare different tools utilized.

Keywords: Process mining · Agile software development · Process discovery · Process conformance checking

1 Introduction

Process is defined as sequence of interdependent and linked activities to convert inputs into outputs. Manufacturing industries have focused on improving processes as it enables improvement of cost, cycle time and reliability at the same time. Software development society has also been working on developing methodologies to improve processes. Although the frameworks such as SPICE and CMM have significant success they also pose significant challenges [1–4]. Over the last two decades, agile software development methods have become very popular in software development area. Agile methods which are based on iterative development foundations brings more light weight and people centric view point when compared with the traditional approaches such as waterfall. Manifesto for Agile Software Development advices the agile teams to

value individuals and interactions over processes and tools [5]. These principles frequently lead to development processes which are not formalized. Some agile methods as Scrum and XP prescribe a set of practices. Frequently these practices are not applied as they should or interpreted by agile teams in such a way that the outcomes are unpredictable [6, 7]. Determination of the sequence of events and the techniques used by each personnel independently may result in inconsistency, instability, and unpredictability. Moreover, different development teams in an organization may interpret the agile method rules differently which leads to interoperability problems between the projects of an organization. So it is vital to draw some borders. However having defined processes also does not guarantee that the personnel follows the processes with high fidelity. A method to extract the actual processes followed by agile teams can be valuable and will help to visualize consistency, stability, interoperability and repeatability problems. Extracting actual process might also help organizations to assess their agility.

Process mining is a process management technique that allows for the analysis of business processes based on the event logs. The aim of process mining is to discover, monitor and improve enacting processes, by extracting knowledge from event logs readily available in today's information systems [8]. Process mining can provide the right tools to discover agile processes followed by agile teams to understand the reality of the organization. Process mining can also be a base for constructing an assessment framework to measure the agility of the organizations.

In this study, we performed an exploratory case study to evaluate the usage of process mining in agile software development context. A scrum project in a defense industry company is selected as target and traces of product backlog items and bug records are analyzed to mine their actual state flows. Results are compared with the predefined flows. This study is important to show the applicability of process mining techniques in agile context. Agile processes leave less traces when compared with the traditional software development approaches like waterfall. We have observed that, observing the process fidelity is difficult in agile contexts and working on process mining in agile context also requires a data collection methodology. However, we have also observed that process mining can be used to extract the actual processes, and organizations can be aware of their agile maturity using the results.

The remainder of this paper is organized as follows. In Sect. 2, a brief information about process mining is given and its usage areas in agile software development is discussed. In Sects. 3 and 4, an exploratory case study as an introduction to mining the actual process in a scrum project and its results are shared with the readers. And in Sect. 5, concluding remarks and future works are discussed.

2 Process Mining

Process mining is a relatively young research discipline. The techniques related with process discovery has its roots in various disciplines such as data mining, computational intelligence and machine learning. The earliest study on process discovery is attributed to Cook et al. [9–11]. Agrawal et al. are also early pioneers of process mining [12]. A number of research studies on process mining have been conducted during the last

decade and the trend is spectacular. Aalst provides a comprehensive overview of the state-of-the-art in process mining in his books [13, 14]. There are various methods and algorithms proposed for different purposes. Cook and Wolf examined the use of statistical analysis methods (Rnet, Ktail and Markov) for use in mining tasks [15]. Also heuristic approaches [16], genetic algorithms [17], fuzzy mining techniques [18] and cluster analysis [19] are used in process mining. Akman and Demirors [20] studied on applicability of the process discovery algorithms for software organizations and their findings provide insight on how process discovery and mining algorithms could be effectively used.

As the capabilities of information systems and features of CASE tools are improved, it become possible to record and analyze flow of software development process. Each action generates some event log data, each shareholder in development process leave some footprints that can be traced to extract information. However, the challenge is to exploit data in a meaningful way. As a research discipline which sits between computational intelligence and data mining on the one hand, and process modeling and analysis on the other hand, the aim of process mining is to extract such meaningful information. Starting point for process mining is an event log. All process mining techniques assume that it is possible to sequentially record events such that each event refers to an activity and is related to a particular case [8]. Process mining studies can be categorized with respect to their purpose: discovery, conformance checking and enhancement. Process mining also covers different perspectives, control-flow, organizational, case and time perspectives.

Agile software projects are generally developed by small teams and in short iterations. Agile methods are more light weight, more people centric and leave less traces when compared with the traditional approaches such as waterfall. Process mining techniques can be used to analyze application of agile methods. Scrum is the most popular agile methodology around the world and prescribe a set of practices for the teams as team formation, roles, meeting schedules. Application life cycle management tools have built-in agile templates to help teams to follow their jobs. Activities of agile teams can be mined to discover what is going on and how is going on. Process discovery will be beneficial to extract the steps followed by agile teams, required inputs to progress, intermediate outputs generated inside the iterations and roles that has taken places through development. It also becomes possible to compare application of agile methods by different agile teams with process discovery. Caldeira [21] made a research to increase the awareness of software developers about their development process and reveal improvement opportunities by mining event logs of development environments. In this study, they plan to discover the process of an agile team by mining the data generated by their development tools. Rubin et al. [22] also has a work related with process mining and agile development. The work describes a bottom-up approach, which takes event logs (e.g., trace data) of a software system for the analysis of the user and system runtime behavior and for improving the software. It does not deal with the development processes but the improvement of software functionality by using process mining techniques in an agile manner.

3 Case Study

3.1 Case Study Design

In order to test the applicability of process mining in agile software development, we have planned an exploratory case study in an organization who utilizes an agile methodology.

The research questions are determined to be:

- (1) Is it possible to trace event logs generated through a project developed using agile methodologies?
- (2) Is it possible to extract the flow of actions in terms of agile events?
- (3) Is it possible to provide useful feedback to the software development team about their processes?

The case selection criteria determined to be:

- Software development project managed by Scrum. Scrum is the most popular agile methodology and it has predefined rules that can be used for conformance checking. Also sprints generate time periods to analyze and improve the process.
- At least 5 employees will be in Scrum Team and at least 6 sprints should happen. Otherwise meaningful data for analysis cannot be obtained.

We have planned the following activities:

- Collect data from the tools used by the organization.
- Analyze data and extract state flow information for Product Backlog Items and Bugs
- Transform state flow information into a format readable by process mining tools
- Run process mining analysis with Disco and ProM tools (as they are the most frequently used tools)
- Evaluate the results

3.2 Application of the Plan

Case study is performed in an organization which is a leader system developer in the defense industry sector and have been performing development projects for over 40 years. Case study is conducted on a software development division of the company which develops C4I software. A software project which is developed by following SCRUM methodology is selected to conduct case study. The details of the project is given in the following table (Table 1).

In the case study, the data from Microsoft TFS is analyzed to discover flow of Product Backlog Items (PBI) and Bugs from creation to closing. An application is developed in C# with MS Visual Studio to analyze and transform info into XES. Following steps are executed:

- Microsoft TFS has a development API and supports developing application with C#. Developers can create queries to extract data from TFS database. We developed an application to access the Scrum project data. We queried history of product

Table 1. Project information

Agile method	SCRUM
Scrum development team size	6 person
Number of sprints	9
Sprint length	4 weeks
Agile management tool	Microsoft TFS

backlog items and bugs. History data contains all of the changes made on PBIs and bugs. History data is processed to extract state transitions and data is filtered and only parts of the data required for process mining analysis are extracted.

- Processed data is not in the format that process mining tools can understand. Our application also transforms the data into XES format to feed the process mining tools.
- Disco and ProM tools are used to analyze the data. Input is feed as XES files.
- Results are evaluated by discussion.

4 Case Study Results

In the Sprint 1 planning meeting, the team defined a flow to execute PBIs (Fig. 1) and Bugs (Fig. 3). The results of process mining analysis based on the data collected through 9 sprints are compared with the defined flows (Fig. 2).

When the actual flow and defined flow for PBI and Bug states are compared, the patterns are pretty much similar to each other. However there are some patterns which are not expected to occur in normal execution (Fig. 4):

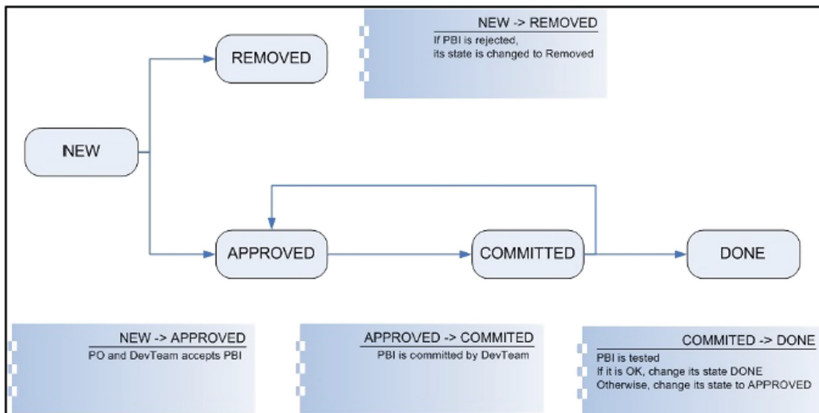


Fig. 1. Defined flow for PBI execution

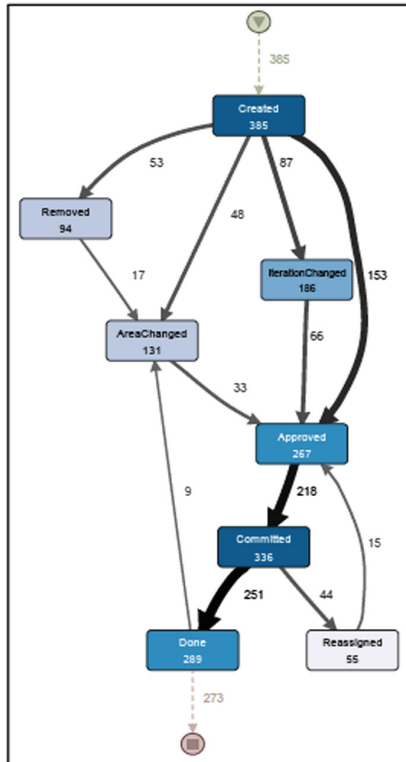


Fig. 4. Actual flow for Bug execution (Disco)

- PBI state change from Done to Reassigned
- PBI state change from Removed to Reassigned
- PBI area change at Done state
- Bug area change at Done state

These patterns are analyzed by the team and some are signed as noise (made by mistake). There are also some patterns which are real execution models. These patterns signals that the team made some mistakes in the execution. The team changed the state of some PBI to Done. But at later iterations, it is realized that some of these PBIs are not really done. They have misinterpreted the Scrum method. All these misuse or misinterpretation patterns are opportunities for the team to improve their process and also opportunity to detect the point where they leave Scrum rules.

During the case study, we faced with some difficulties especially in extracting data and transforming it into a format that can be processed by process mining tools. Also in analysis phase, there are many algorithms that can be run on data. The decision of which algorithm best fits with the data to generate successful results is a non-trivial issue.

There exist many tools with process mining capability as ProM, Disco (Fluxicon) ARIS Process Performance Manager (Software AG), Comprehend (Open Connect), Discovery Analyst (StereoLOGIC), Flow (Fourspark), Futura Reflect (Futura Process Intelligence), Interstage Automated Process Discovery (Fujitsu), OKT Process Mining Suite (Exeura), Process Discovery Focus (Iontas/Verint), ProcessAnalyzer (QPR), Rbminer/Dbminer (UPC) and Reflect one (Pallas Athena). Among several available tools, ProM and Disco are selected for the use in analysis since they are popular and also available for academic evaluation purpose. Both ProM and Disco have powerful analysis capabilities. Disco provides a simpler user interface and a filtering mechanism which has clear representation compared to ProM. ProM has many plug-ins and serves a high number of alternative to run analysis which makes the tool very powerful. But this can also make the tool complex. Since our work is not comprehensive, we cannot make a detailed comparison between the tools. The result generated by both tools are nearly the same. But as our work progress through analysis of agile processes, we will experience detailed features of the tools and have chance to compare them.

5 Conclusion and Future Works

Process mining traces the footprints and outputs generated through the application of development process to extract knowledge about the processes. Agile methods have become a de facto standard in software development over the last decade. However, due to the developer centric nature of methods, process fidelity issues may not be given required importance which might lead to critical problems. Process mining can be used as a beneficial tool for analyzing the application of agile methods in organizations. Case study mentioned in this paper has shown that agile methods leaves many evidences to be tracked to extract knowledge about real process. By analyzing the data, beneficial results to make agile team aware of their process can be obtained and also opportunities to diagnose the failing parts of the applied process.

However, reliability of the data is the most important criteria in the correctness of the analysis. Data collection is in the hearth of the process mining. The quality of process mining results heavily depend on the input. Therefore, event logs should be treated as first-class citizens in the information systems supporting the processes to be analyzed. In order to benefit from process mining, organizations should aim at event logs at the highest possible quality level. As a future work we plan to develop a data collection model for process mining in agile methods (especially Scrum) to make efficient process mining analysis.

Although generally used for discovery, process mining is not limited with discovery. Discovery is just one of the three basic form of process mining. Conformance checking and enhancement can be used to extract knowledge to discover failure points in the real processes and generates improvement opportunities. Conformance checking can also be used for measuring the agility rates of organizations. As a future work, with the completion data collection model for process mining in agile methods, an assessment framework for measuring the agility of organizations based on process mining can be constructed.

This is an exploratory work, our future plans are constructing a framework for collecting data in agile software context for process mining and generating conformance checking methodology for assessing organizations/projects agility. Although this study is a preliminary work on using process mining in agile software development, our observations provided us motivation to continue the research to generate a data collection model for process mining in agile methods and a mining based assessment model to measure agility.

References

1. Chrissis, M.B., Konrad, M., Shrum, S.: *CMMI for Development: Guidelines for Process Integration and Product Improvement*. Pearson Education, Upper Saddle River (2011)
2. Emam, K.E., Melo, W., Drouin, J.N.: *SPICE: the theory and practice of software process improvement and capability determination*. IEEE Computer Society Press, Los Alamitos (1997)
3. Uskarcı, A., Demirörs, O.: Do staged maturity models result in organization-wide continuous process improvement? insight from employees. *Comput. Stand. Interfaces* **52**, 25–40 (2017)
4. Tarhan, A., Demirors, O.: Apply quantitative management now. *IEEE Softw.* **29**(3), 77–85 (2012)
5. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Kern, J.: *Manifesto for agile software development* (2001)
6. Top, Ö.Ö., Demirörs, O.: Assessing software agility: an exploratory case study. In: Mitasiunas, A., Rout, T., O'Connor, R.V., Dorling, A. (eds.) *SPICE 2014*. CCIS, vol. 477, pp. 202–213. Springer, Cham (2014). doi:[10.1007/978-3-319-13036-1_18](https://doi.org/10.1007/978-3-319-13036-1_18)
7. Ozcan-Top, O., Demirörs, O.: A reference model for software agility assessment: AgilityMod. In: Rout, T., O'Connor, R.V., Dorling, A. (eds.) *SPICE 2015*. CCIS, vol. 526, pp. 145–158. Springer, Cham (2015). doi:[10.1007/978-3-319-19860-6_12](https://doi.org/10.1007/978-3-319-19860-6_12)
8. van der Aalst, W., et al.: *Process mining manifesto*. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) *BPM 2011*. LNBP, vol. 99, pp. 169–194. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-28108-2_19](https://doi.org/10.1007/978-3-642-28108-2_19)
9. Cook, J.E., Wolf, A.L.: Discovering models of software processes from event-based data. *ACM TOSEM* **7**(3), 215–249 (1998)
10. Cook, J.E., Wolf, A.L.: Automating process discovery through event-data analysis. In: *Proceedings of 17th International Conference on Software Engineering*, pp. 73–82 (1995)
11. Cook, J.E.: *Process discovery and validation through event-data analysis*, Ph.D. thesis, University of Colorado, Boulder, Department of Computer Science (1996)
12. Agrawal, R., Gunopulos, D., Leymann, F.: Mining process models from workflow logs. In: Schek, H.-J., Alonso, G., Saltor, F., Ramos, I. (eds.) *EDBT 1998*. LNCS, vol. 1377, pp. 467–483. Springer, Heidelberg (1998). doi:[10.1007/BFb0101003](https://doi.org/10.1007/BFb0101003)
13. Aalst, W.M.: *Process Mining: Data Science in Action*. Springer, Cham (2016)
14. Aalst, W.M.: *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer, New York (2011)
15. Cook, J.E., Wolf, A.L.: Discovering models of software processes from event-based data. *ACM Trans. Eng. Methodol. (TOSEM)* **7**(3), 215–249 (1998)
16. Weijters, A.J., Van der Aalst, W.M.: Rediscovering workflow models from event-based data using little thumb. *Integr. Comput.-Aided Eng.* **10**(2), 151–162 (2003)

17. de Medeiros, A.K., Weijters, A.J., van der Aalst, W.M.: Genetic process mining: an experimental evaluation. *Data Min. Knowl. Disc.* **14**(2), 245–304 (2007)
18. Günther, C.W., van der Aalst, W.M.P.: Fuzzy mining – adaptive process simplification based on multi-perspective metrics. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) *BPM 2007*. LNCS, vol. 4714, pp. 328–343. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-75183-0_24](https://doi.org/10.1007/978-3-540-75183-0_24)
19. Schimm, G.: Mining exact models of concurrent workflows. *Comput. Ind.* **53**(3), 265–281 (2004)
20. Akman, B., Demirörs, O.: Applicability of process discovery algorithms for software organizations. In: *35th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2009*, pp. 195–202. IEEE (2009)
21. Caldeira, J., e Abreu, F.B.: Software development process mining: discovery, conformance checking and enhancement. In: *2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC)*, pp. 254–259. IEEE (2016)
22. Rubin, V., Lomazova I., Aalst, W.M.: Agile Development with software process mining. In: *Proceedings of the 2014 International Conference on Software and System Process*, pp. 70–74. ACM (2014)