

Improving Software Maintenance Ticket Resolution Using Process Mining (Extended Abstract)

Monika Gupta

Indraprastha Institute of Information Technology Delhi, India
monikag@iiitd.ac.in

Software maintenance is a crucial activity in software industry and consumes a major portion of the expenditure on software. Software maintenance refers to the modification of software product after delivery and is required to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment. Ever-changing customer needs and rapid technical progress highlight the need to continuously improve software maintenance process to make it more effective and efficient.

The work in this thesis focuses on analyzing and improving software maintenance process by exploring novel applications of process mining and predictive analytics. While process mining helps to discover the process reality, using predictive analytics helps recommend suitable actions to mitigate the inefficiencies in a proactive way.

To identify the potential opportunities for improvement in software process management by mining data repositories, we first conducted qualitative interviews and surveys of over 40 managers in a large global IT company. The survey provided us with a list of over 10 maintenance process challenges encountered by practitioners, and benefits that may accrue by addressing them. The survey is published in MSR 2015 [10].

This thesis addresses a few of the identified challenges pertaining to the software maintenance process. We have conducted a series of case studies on large real world data (commercial and open source) to evaluate the usefulness of the proposed solution approaches. Overall approach of the thesis is published as doctoral symposium paper [2][3].

The main contributions of the thesis are as follows:

– **Analyzing the Maintenance Ticket Resolution Process to Identify the Process Inefficiencies**

Ticket resolution is an important part of software maintenance process. As identified from the survey, there is a need to analyze the data generated during ticket resolution process to capture process reality and identify the process inefficiencies.

We have proposed a framework for analyzing software repositories for ticket resolution from diverse perspectives, by applying process mining. The framework has three main steps: 1. data extraction from multiple repositories and integration, 2. transformation of the data to an event log, and 3. multi-perspective process mining from the event log. Using multi- perspective pro-

cess mining, we discover the process model which captures the control flow, timing and frequency information about events. We then studied inefficiencies such as self-loops, back-forth, ticket reopen, timing issues, and effort consumption. We also analyze the degree of conformance between the designed and the run time (discovered) process model.

We conducted a series of case studies on open-source Firefox browser, Core project, and open-source Google Chromium project. The data on tickets was obtained from Issue Tracking System (ITS) for the project (e.g. Bugzilla). We also used repositories for Peer Code Review (PCR) system and Version Control System (VCS), where available. For each of the project, separate analysis was done, from which we also made some general observations. For example, in Google Chrome, we observed that for around 14% cases, ticket is instantiated in ITS after patch submission in PCR or commit in VCS (ideally, for traceability reasons, a ticket's life cycle should start from issue reporting in ITS followed by patch submission in PCR and commit in VCS), and for these tickets the number of patch revisions thus resolution time is higher. In Firefox and Core, we found that a significant percentage of tickets undergo multiple developer reassignment causing delays in resolution. Also, we identified two categories of tickets (wontfix and worksforme) which consume the maximum ticket resolution effort. We noted that several issues in these categories get reopened signaling the need for improvement in identifying such tickets. The proposed multi-perspective process mining framework and the case studies to evaluate the proposed approach is presented in the thesis, and is published in ISEC, APSEC and MSR [7][8][9].

– **Reducing User Input Requests in the Maintenance Ticket Resolution Process**

A ticket is required to be resolved in the defined service level resolution time, measured using the service level clock. Failure to meet this requirement leads to a penalty on the service provider. After a ticket is assigned to an analyst (person responsible for servicing the tickets), they can ask for user inputs to resolve the ticket. When user input is requested, the service level clock stops in order to prevent spurious penalty on the service provider. However, this waiting time adds to the user-experienced resolution time and degrades user experience. Therefore, in this work, we aim to reduce the user input requests to make the ticket resolution faster.

We first applied the multi-perspective process mining framework on the tickets of a large global IT company and found that around 57% of the tickets have user input requests in the life cycle, causing user-experienced resolution time to be almost twice as long as the measured service resolution time. We observed that user input requests are broadly of two types - real, seeking information from the user to process the ticket and tactical, when no information is asked but the user input request is raised merely to pause the service level clock. We propose a machine learning based system that pre-emptively asks a user at the time of ticket submission to provide additional information that the analyst is likely to ask thus, reducing real user input

requests. We also propose a rule-based detection system to identify tactical user input requests.

The proposed system that predicts the information needs has an average accuracy of 94 – 99% across five cross validations while traditional approaches such as logistic regression and naive Bayes have accuracy in the range of 50 – 60%. The detection system identifies around 15% of the total user input requests as tactical with a high precision. Together the proposed preemptive and detection systems efficiently bring down the number of user input requests and improve the user-experienced resolution time. This work is published in the Empirical Software Engineering journal [5].

– **Discovering Underlying Maintenance Ticket Resolution Process Interactions using Unstructured Data from Execution Logs**

Process mining uses largely structured data viz. event logs and does not leverage the rich information from unstructured data such as comments and emails. This work is motivated by the need to explore unstructured data generated during process execution to capture underlying process interactions to help in making effective process improvement decisions.

To achieve this, we extract topical phrases (keyphrases) from the unstructured data using an unsupervised graph-based approach. Keyphrases are then integrated into the event log, which then gets reflected in the discovered process model. This provides insights that cannot be obtained solely from structured data, which can be used to identify process improvement opportunities.

To evaluate the usefulness of the approach, we conducted case studies on the publicly available ticket data from a Dutch insurance company, and on the ticket data of a large global IT company. Our approach extracts keyphrases from the comments associated with the tickets with an average accuracy of around 80% across different data sets. This enabled us to succinctly capture the additional information in the comments regarding issues influencing ticket resolution process and often causing delays, like extra information required, priority, severity, etc. This allows the managers or the process analysts to make decisions about how to speed up the resolution process, e.g., implement a bot to capture the information or add a mandatory field in the initial ticket template thus reduce the delays incurred while waiting for information. This work is published at AI4BPM [4].

– **Runtime Monitoring in Changed Software as Compared to Previous Version**

To resolve a ticket, some code changes are made which can lead to an anomaly such as regression bugs. In this work, we aim to monitor and compare the execution behaviour of new version (after code change) with the previously deployed version to detect if ticket resolution has caused some anomalous behaviour thus reduce the post release bugs.

We propose an approach to discover execution behaviour for the deployed and the new version using the execution logs (which contain outputs of all the

print statements along with related information like time, thread ID, statement number, etc.). Differences between the two models are then identified and refined such that spurious differences, e.g., due to logging statement modifications, are eliminated. The differences are presented graphically as regions within the discovered behaviour model. This allows programmers to identify anomalous behaviour changes which are not consistent with code changes, thereby identifying potential bugs that may have been introduced during code change.

To evaluate the proposed approach, we conducted case study on Nutch (open source application), and an industrial application. We discovered the execution behaviour models for the two versions of applications and identified the differences between them. By manually analysing the regions, we were able to detect bugs introduced in the new versions of these applications. The bugs have been reported and later fixed by the developers, thus, confirming the effectiveness of our approach. This work is published in ICSOC [6].

In the thesis we have explored the potential of applying combination of process mining using various data sources and predictive analytics to improve various aspects of the maintenance process. We have applied the proposed approaches on a series of case studies on data sets of commercial and open source projects. Although we believe that the case studies are representative, to establish generalizability, the proposed approach should be applied on different data sets. To support the reproducibility of our case studies, the large part of data (with the data from the industrial partners being the only exception) have been made publicly available [1].

We believe that leveraging diverse data sources and applying analytics intelligently has more potential for process improvement. Information from other sources such as emails, chat logs, and screen recordings can further enhance process improvement. Such analysis usually focus on identifying the inefficiencies, but as we observed in the thesis, it can also lead to automation opportunities to make process more efficient.

References

1. Link to publicly available artifact. <https://github.com/Mining-multiple-repos-data/TicketExperimentalDataset>.
2. Monika Gupta. Nirikshan: process mining software repositories to identify inefficiencies, imperfections, and enhance existing process capabilities. In *Companion Proceedings of the 36th International Conference on Software Engineering*, pages 658–661, 2014.
3. Monika Gupta. Improving software maintenance using process mining and predictive analytics. In *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 681–686. IEEE, 2017.
4. Monika Gupta, Prerna Agarwal, Tarun Tater, Sampath Dechu, and Alexander Serebrenik. Analyzing comments in ticket resolution to capture underlying process interactions. In *Artificial Intelligence for Business Process Management*, 2020.

5. Monika Gupta, Allahbaksh Asadullah, Srinivas Padmanabhuni, and Alexander Serebrenik. Reducing user input requests to improve it support ticket resolution process. *Empirical Software Engineering*, 23(3):1664–1703, 2018.
6. Monika Gupta, Atri Mandal, Gargi Dasgupta, and Alexander Serebrenik. Runtime monitoring in continuous deployment by differencing execution behavior model. In *International Conference on Service-Oriented Computing*, pages 812–827. Springer, 2018.
7. Monika Gupta and Ashish Sureka. Nirikshan: Mining bug report history for discovering process maps, inefficiencies and inconsistencies. In *Proceedings of the 7th India Software Engineering Conference*, pages 1–10, 2014.
8. Monika Gupta and Ashish Sureka. Process cube for software defect resolution. In *2014 21st Asia-Pacific Software Engineering Conference*, volume 1, pages 239–246. IEEE, 2014.
9. Monika Gupta, Ashish Sureka, and Srinivas Padmanabhuni. Process mining multiple repositories for software defect resolution from control and organizational perspective. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, pages 122–131, 2014.
10. Monika Gupta, Ashish Sureka, Srinivas Padmanabhuni, and Allahbaksh Mohammedali Asadullah. Identifying software process management challenges: Survey of practitioners in a large global IT company. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, pages 346–356. IEEE, 2015.