Identifying Software Process Management Challenges: Survey of Practitioners in a Large Global IT Company

Monika Gupta*, Ashish Sureka*, Srinivas Padmanabhuni†, Allahbaksh Mohammedali Asadullah†
*Indraprastha Institute of Information Technology, Delhi, India
{monikag, ashish}@iiitd.ac.in
†Infosys Ltd., Bengaluru, India
{srinivas_p, allahbaksh_asadullah}@infosys.com

***Abstract*—Process mining consists of mining event logs generated from business process execution supported by Information Systems (IS). Process mining of software repositories has diverse applications because vast data is generated during Software Development Life Cycle (SDLC) and archived in IS such as Version Control System (VCS), Peer Code Review (PCR) System, Issue Tracking System (ITS), and mail archives. There is need to explore its applications on different repositories to aid managers in process management. We conduct two phase surveys and interviews with managers in a large, global, IT company. The first survey and in-person interviews identify the process challenges encountered by them that can be addressed by novel applications of process mining. We filter, group and abstract responses formulating 30 generic problem statements. On the basis of process mining type, we classify identified problems to eight categories such as control analysis, organizational analysis, conformance analysis, and preventive analysis. The second survey asks distinct participants the importance of solving identified problems. We calculate proposed Net Importance Metric (NIM) using 1262 ratings from 43 participants. Combined analysis of NIM and first survey responses reveals that the problems mentioned by few practitioners in first survey are considered important by majority in the second survey. We elaborate on possible solutions and challenges for most frequent and important problems. We believe solving these validated problems will help managers in improving project quality and productivity.**

***Index Terms*—Process Mining, Qualitative Study, Software Development Life Cycle, Software Repositories**

I. Introduction

Mining software repositories is one of the fastest growing field aimed at solving real problems encountered by practitioners and bringing value to industry. It describes a broad class of investigations into the software repositories to uncover interesting and actionable information [1] [2]. Process mining (intersection of Business Process Management and Data Mining) consists of mining event logs generated from business process execution supported by IS [3]. It bridges the gap between traditional model-based process analysis and data-centric analysis techniques. Many process mining framework and tools such as ProM[1] (open source) and Disco[2] (commercial) are used to derive process model and process mine data from different perspectives. Process mining includes process discovery, process performance analysis, conformance verification, case prediction, history based recommendations and organizational analysis [4]. It has already been applied to analyze business processes from multiple domains [5].

Process mining of software repositories has diverse applications and is an area that has recently attracted several researchers' attention due to availability of vast data generated and archived in multiple IS such as SCM, ITS, VCS, and mail archives during software development. It can provide Capability Maturity Model Integration (CMMI) assessors with relevant information and can support existing software process assessment and improvement approaches [6]. Some of the business applications of process mining software repositories are: uncovering runtime process model [7] [8], discovering process inefficiencies and inconsistencies [7] [9], observing project key indicators and computing correlation between product and process metrics [10], extracting general visual process patterns for effort estimation and analyzing problem resolution activities [11].

Managers work in a fast-changing, hyper competitive, interconnected global market place. They can no longer rely on intuition, use of analytics is necessary to improve the quality of decision making. The research objective of the work presented in this paper is to identify the process challenges that community in practice would such as to be addressed by novel applications of process mining. To achieve this, we conduct survey and interviews with managers in a large, global, software company. We adopt the methodology consisting of two surveys proposed by Begel *et al.* [12].

1) We identify the challenges encountered by managers while managing software projects that can be addressed by novel applications of process mining, by conducting first survey and an interview study with purposively sampled target population of 300 practitioners. We receive 130 response items from 46 participants that are filtered and grouped into 30 unique problems.
2) We classify identified problems to eight process mining types based on the technique to be used for addressing

[1] http://www.processmining.org/prom/start

[2] http://www.fluxicon.com/

978-0-7695-5594-2/15 $31.00 © 2015 IEEE
DOI 10.1109/MSR.2015.39

them and investigate benefits of solving problems from each category.
3) In the second survey, we determine the importance of solving identified challenges by calculating proposed Net Importance Metric (NIM) using 1262 ratings collected from 43 participants. We elaborate on solution feasibility of most important and frequent challenges identified by combined analysis of NIM and first survey responses.

Effectively, we gather a range of validated process management challenges spanning across the whole SDLC. The findings highlight the need for process mining of software repositories from various perspectives to enable more objective continuous process assessment and improvement of software projects.

II. Related Work and Research Contributions

In this section, we discuss closely related work (to the research presented in this paper) and present the novel research contributions of this research paper in context of the existing work. We organize closely related work into two following lines of research:

A. Process Mining of Software Repositories

We present some work where process mining is applied on software repositories from different perspectives such as control-flow, organizational and conformance checking. Samalikova *et al.* investigate CMMI from a process mining perspective and identify model components for which process mining techniques can be applied [6]. Results of a case study on change control board process illustrate that process mining can provide CMMI assessors with the relevant information [6]. Kim *et al.* propose a distributed workflow mining approach to discover workflow process model incrementally amalgamating a series of vertically or horizontally fragmented temporal work-cases [13]. Sunindyo *et al.* propose an observation framework that supports OSS project managers in observing project key indicators such as checking conformance between the designed and the actual process models [10]. Knab *et al.* present an approach of extracting general visual process patterns for effort estimation and analyzing problem resolution activities for ITS [11]. Ashish *et al.* present a generic framework for software process intelligence involving mining and analysis of software processes [14].

Gupta *et al.* present an application of integrating and process mining three software repositories (ITS, PCR and VCS) from control flow and organizational perspective [15]. Poncin *et al.* present a framework called as FRASR (FRamework for Analyzing Software Repositories) that facilitates combining and matching of related events across multiple repositories such as mail archives, subversion, and bug repositories, followed by assignment of role to each developer [16]. Song *et al.* apply process mining technology to common event logs obtained from five different IS for behavior pattern mining [17].

B. Study to Identify Practitioners' Questions

We conduct a literature review for studies adopting research methodologies involving steps such as surveys, interviews and categorization, to identify questions asked by practitioners. Most closely related research to the work presented in this paper is by Begel *et al.* in which they catalog 145 questions grouped into 12 categories which software engineers would like data scientists to investigate [12]. They also identify the most important and most unwise problems based on rating survey with 607 Microsoft engineers [12]. Phillips *et al.* interview seven release managers at a large software company to identify the information required for integration decisions when releasing software, organized around 10 key factors [18]. Fritz *et al.* interview eleven developers to identify the questions they want to ask but for which the support to integrate different kinds of project information is lacking [19]. LaTozo *et al.* propose 19 problems from their own experience as software developers and survey other developers to determine the seriousness of each problem [20]. Sillito *et al.* categorize 44 different kinds of questions that developers ask during change task on code base [21].

In the context of the existing work, the study presented in this paper makes the following novel contributions:

1) A list of 30 process management challenges covering diverse perspectives (8 categories) that community in practice would like process mining team (specializes in process mining of software repositories) to solve. We enlist the benefits of solving problems from each category to motivate both, researchers and industry.
2) Identify more important problems by calculating NIM using responses of second survey. Therefore, focus on solving more important problems.
3) Combined analysis of NIM and number of first survey response items covered for each problem, to make better sense of identified problems. It brings out the interesting finding that while many problems are mentioned by few participants in first survey (less frequent), they are considered more important when presented to participants in second survey.
4) Investigation of most important and frequent problems to identify the gap and challenges involved in solving those problems.

III. Research Methodology

We conduct two-phase online survey and an interview study at Infosys Limited[3], a large, global, software company having more than 170000 software professionals. It is a CMM 5 company with well defined processes in place. We adopt research methodology by Begel *et al.*, with some contextual modifications [12]. Fig. 1 depicts methodology adopted for this research work:

- Survey and interviews to identify the process management challenges encountered by managers and benefits of solving those challenges.

[3] http://www.infosys.com/

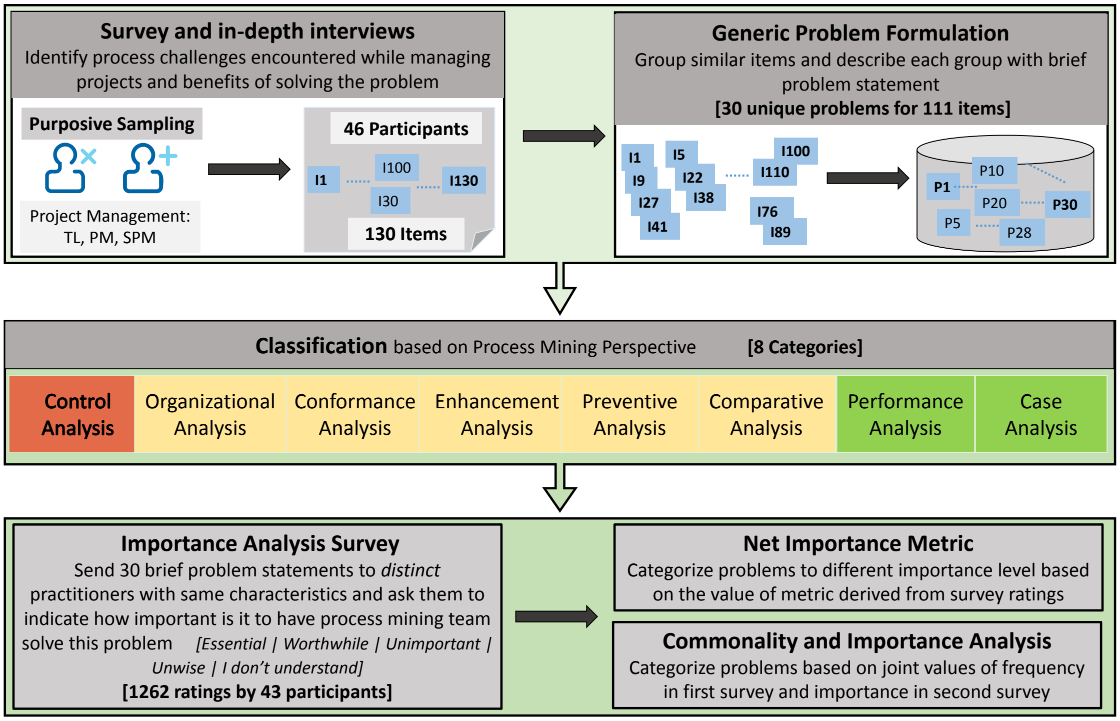
Survey and in-depth interviews
Identify process challenges encountered while managing projects and benefits of solving the problem
Purposive Sampling
Project Management: TL, PM, SPM
46 Participants
I1
I100
I30
I130
130 Items
Generic Problem Formulation
Group similar items and describe each group with brief problem statement
[30 unique problems for 111 items]
I1
I5
I9
I22
I27
I38
I41
I100
I110
I76
I89
P10
P1
P20
P30
P5
P28
Classification based on Process Mining Perspective [8 Categories]
Control Analysis
Organizational Analysis
Conformance Analysis
Enhancement Analysis
Preventive Analysis
Comparative Analysis
Performance Analysis
Case Analysis
Importance Analysis Survey
Send 30 brief problem statements to distinct practitioners with same characteristics and ask them to indicate how important is it to have process mining team solve this problem [Essential | Worthwhile | Unimportant | Unwise | I don't understand]
[1262 ratings by 43 participants]
Net Importance Metric
Categorize problems to different importance level based on the value of metric derived from survey ratings
Commonality and Importance Analysis
Categorize problems based on joint values of frequency in first survey and importance in second survey

Fig. 1: Research methodology with major steps as: 1. Phase I survey to identify the challenges encountered by practitioners, group them and formulate generic problem statements, 2. Classification based on process mining perspective (number in bracket indicates total problems in each category) and 3. Importance analysis survey followed by commonaity and importance analysis.

- Classification of identified challenges based on process mining perspective.
- Survey to determine the importance of solving each identified problem.

Participants for both the surveys are purposively sampled to ensure that they meet the criterion of having project management experience. Here, the eligible participants consist of professionals such as Team Lead (TL), Project Manager (PM), Senior Project Manager (SPM), Group Project Manager (GPM), Principal Technical Architect (PTA) and others, who are responsible for significant project management activities for the software project. In the first survey, we ask practitioners to solicit the challenges that they would like process mining team to solve, using the data stored in software repositories during SDLC. We also conduct interviews to better understand the process challenges encountered by them. We filter, group and develop generic problem statements to describe the identified challenges. As shown in Fig. 1, we classify them to various process mining types to highlight the diversity of responses. Thereafter, in the second phase of the study, we investigate the importance of solving each problem by conducting a survey with distinct participants. Unlike Begel *et al.* [12], we include all the problems in survey without splitting because we have total 30 problems which is almost same as the average number of questions in their each split. We evaluate the proposed importance metric and present interesting commonality-importance analysis. For reference and benchmarking, we have made the surveys, collected responses and consolidated lists for both the phases publicly available[4].

A. Survey to Collect Problems

The first survey aims to identify the challenges encountered by practitioners which they would like process mining team to solve for them. This phase of the study includes a pilot study to improve the survey followed by final data collection.

1) Pilot Study and Survey Design: We conduct pilot study to improve the survey before sending it to the target audience. Initially, we included only the definition of process mining to quickly introduce a participant with process mining and set the context. Question asked in the survey is as follows:

> *Process Mining of software repositories involves extraction of useful information from event logs recorded by Information Systems (such as Bug Tracking System, Version Control System, SCMs, E-mails, Peer Code Review System) used during the SDLC.*
>
> *Suppose you are given an opportunity to benefit from the expertise of process mining specialists' team. The team can process mine the data stored in repositories during SDLC, and solve problems related to software development and maintenance process.*

[4] https://github.com/Mining-multiple-repos-data/QualitativeStudy

> **Please list up to three problems that you encounter during the software development process management and you would like process mining team to solve for you. Also, mention the benefits you will have if the problem is solved.**

The survey is given to three participants (one SPM, one PM and one TL) and we observe them while they are filling the survey. Two of them (PM and TL) ask for some example applications of process mining as they find it difficult to understand the kind of challenges that meet the criteria. While SPM directly fills the survey mentioning the challenges such as high attrition rate, and difficulty in requirements gathering, these are not aligned with our objective. Therefore, we add the following example process mining applications along with the definition to give participant a better idea of the domain:

- **Performance Analysis:** Identify inefficiencies and imperfections such as most time consuming activities, rework, cause of delay, bottlenecks in the process.
- **Process Compliance Checking:** Detect inconsistencies with the defined process and flag anomalies.
- **Control Flow Analysis:** Discovery of actual process from the execution logs and analyze them to better understand the actual runtime process.
- **Organizational Analysis:** Analyze individuals, team coordination and interactions with the process.

Henceforth, the improved survey is sent to around 300 randomly selected participants out of all purposively sampled professionals who have project management experience. They are given an option to fill it online or indicate interest for an in-person discussion about the same.

2) Participants and Data Collection: We specifically target participants from different horizontals such as delivery, maintenance, development, quality, and Finacle (the banking product business division of Infosys), to capture the diverse perspectives and problems of the practitioners involved in management activities. Overall, 46 practitioners respond, out of which 12 opt for an in-person discussion. Response rate is around 15% which is comparable with the response rate of similar free text surveys [12]. Practitioners from different job levels ranging from TLs to delivery manager participated and added to the wholesample of problems where process mining can help. We group individuals based on similarity in role and observe that out of all participants, around 26% are TLs, 32.6% are PMs, 28.3% are SPMs or Senior Technology Architect and rest are others such as delivery manager, group leader and quality manager. The total work experience of the respondents ranges from 8 to 21 years. Project management experience is up to 14 years where 25 participants have been performing project management activities and handling teams for more than 5 years. Four respondents choose not to mention their experiences.

The first author conducts a semi-structured interview using the survey questionnaire as guide. The interviewer gives interviewee an overview of process mining followed by the objective of the study. Most participants describe a scenario along with the context that is noted by the interviewer. Survey responses also include long statements describing the scenario and benefits. Very few responses state the questions directly. A major difference between the survey responses and interviews is in terms of context understanding. During the interview, interviewer has the opportunity to probe and ask follow up questions for better understanding. Conducting interviews and getting a detailed idea on perspectives discussed in the interview helps us to better interpret the survey responses. We gather 130 items as a result of this survey and interview exercise. We focus on exploring variety and richness of the questions rather than the frequency. We discard 19 points in survey responses that are simple comments and not aligned with the study. For example, the following responses are filtered out:

- *"Lack of product culture in services industry."*
- *"Clarity of non functional requirements, migration projects do not have proper testing"*,
- *"Impact elements are not easily defineable."*

3) Generic Problem Statement Formulation: We observe that many points though expressed differently have essentially similar underlying problem. On the basis of similarity in problem, we openly group (groups unknown) valid 111 items resulting into 30 groups. We notice that more than one problem is stated in some items, thus included in multiple groups. As shown in Fig. 1, a problem statement is created for each group. 30 problem statements listed in Table I cover 111 items and are therefore more abstract than the original grouped items.

For example, for problem statement: *During issue resolution, detection and analysis of PING-PONG patterns due to bug tossing between developers to reduce resolution time*, the following three responses (as stated by respondent) belong to the group:

- *"Quite often an issue is reported as defect. However, developers do not consider it to be a defect or state that it belongs to some other component. A lot of efforts go waste in this tossing around. Can we improve the process to reduce such cases?"*
- *"Unnecessary delays and blame game. Reporting of defects often leads to delay and blame game. Understand the patterns with cause to avoid such situations."*
- *"Ping-Pong patterns between various teams when an issue is reported."*

Similarly, we abstract other groups using brief problem statements. Each statement has **task** (or actual problem) as the first component followed by **cause and benefit** as the second component. We intentionally structure it like this considering the second phase of the study because we want the problem to act as a trigger. For instance, in the above example, *detection and analysis of Ping-Pong patterns* is the **problem** and *due to bug tossing between developers to reduce resolution time* is **cause and benefit**. Original detailed points for every group are not presented due to limited space and hence made publicly available.

TABLE I: List of generic problem statements along with count of positive responses $(+ve)$, count of negative responses $(-ve)$, number of items from first survey in group $(C(P1))$ and process mining category to which the problem belongs.

ID	Problem Statement	+ve	−ve	C(P1)	NIM	Category
1	Identify BOTTLENECKS and inefficiencies causing delay to take remedial actions and have better estimation in future.	42	0	7	0.92	Performance
2	Enable early detection and PREVENTION OF DEFECTS instead of fixing them during the later stage by understanding patterns of escaped defects.	41	0	9	0.91	Case
3	Avoid putting efforts on LESS SIGNIFICANT ACTIVITIES by identifying redundant or unnecessary steps of process.	41	1	5	0.89	Enhancement
4	Automatic ADAPTION OF PROCESS according to different project specifications that is, design process based on knowledge of similar successful projects instead of selecting process only on the basis of experience.	43	0	2	0.85	Preventive
5	Inspect REOPENED issues to identify the root cause and recommend verification for future issues based on learning from issues reopened in the past.	41	1	6	0.84	Case
6	Need for efficient TASK ALLOCATION mechanism by considering individuals skills, interests, and expertise as well as team compatibility for better utilization of resources.	40	2	13	0.83	Organizational
7	Approvals are part of software development lifecyle (SDLC) and need better management. Design a process for seamless approvals to reduce delays.	40	3	2	0.79	Preventive
8	Mechanism for CONTINUOUS PROCESS EVOLUTION based on best practices of individuals who exercise the process. Therefore, we improve process by encouraging on-the-job learnings of people rather than dependence on process designers.	39	2	6	0.76	Enhancement
9	Improve effectiveness of CODE REVIEW PROCESS AND STANDARDIZATION by redesigning check list and updating code analyzers based on the defects reported during testing.	39	2	7	0.74	Enhancement
10	Facilitate BETTER INTEGRATION between different silos by reconstructing the process thus, reduce rework happening due to differences in understanding.	36	3	6	0.71	Organizational
11	Handle CHANGING TEAMS seamlessly by analyzing interaction pattern between team members and team dynamics.	38	2	2	0.70	Organizational
12	Design a technique to TRACE ADHERENCE WITH REQUIREMENTS and adapt process automatically with changing requirements.	36	3	2	0.69	Conformance
13	PEOPLE VS PROCESS: Identify which factor contributed to what extent towards the success and failure of project.	39	4	2	0.69	Comparative
14	Simplify tracking of the whole REVIEW PROCESS to identify inefficiencies quickly.	37	4	2	0.67	Control
15	During issue resolution, detection and analysis of PING-PONG patterns due to bug tossing between developers to reduce resolution time.	36	3	3	0.67	Control
16	Improve PROJECT PLANNING AND ESTIMATION by complimenting it with the insights derived from event log mining of similar projects done in the past.	38	5	3	0.66	Preventive
17	Investigate the LEAD TIME for issue resolution by analyzing issue resolution process from TIME PERSPECTIVE. Therefore, we can have timely resolutions.	37	5	2	0.64	Performance
18	Design of more meaningful QUALITY METRICS by understanding runtime process practices to precisely identify the scope of improvement.	35	5	3	0.63	Enhancement
19	Equip novice with the KNOWLEDGE OF EXPERIENCED PRACTITIONERS by associating efficiency of adopted process with the experience of practitioners.	36	4	4	0.63	Comparative
20	Facilitate in-depth understanding of point where things went wrong by deriving and understanding actual process at a MORE GRANULAR LEVEL.	36	5	1	0.63	Control
21	Continuous check on SCHEDULE ADHERENCE is a complex task. Design an automated way to track and preempt if any deviations.	36	6	1	0.60	Conformance
22	Relate bugs with the ACTUAL STAGE OF INCEPTION by understanding issue resolution life cycle along with other relevant attributes.	34	5	3	0.60	Control
23	Uncover DEVIATIONS between the actual process followed by the team and the defined process, their cause, impact on overall outcome and identify the set of people exhibiting more deviations.	36	6	8	0.59	Conformance
24	INTEGRATE MULTIPLE STANDALONE SYSTEMS used during SDLC to solve data and process redundancy challenges, and obtain a holistic view.	34	5	5	0.57	Preventive
25	Analyze code review life cycle to identify developers who are not reviewing their code properly before they submit it for external review and the deviations from defined checklist. It will help take corrective actions and reduce defects during testing phase.	33	7	2	0.53	Conformance
26	Mechanism to manage and keep track of SVN check-ins process that is, activity sequence for merging and branching as it is very important and can help take informed decisions.	27	5	1	0.49	Control
27	Capture the ACTUAL STATUS (reality) of project or any task by discovering runtime process from event logs instead of current manual practice.	31	8	1	0.48	Control
28	Trace the complete flow and understand WHICH ISSUE LEADS TO WHICH CODE CHANGE by analyzing event logs for issue resolution in combination with the code modified in VCS.	31	9	1	0.45	Control
29	Perform COMPARATIVE ANALYSIS OF TICKETS along dimensions such as component, owner (analyst), reporter, type such as performance, regression and security, final resolution such as duplicate, invalid and fixed, and turnaround time to derive useful insights for improvement.	32	10	7	0.44	Comparative
30	Identify the group of ACTIVE VS INACTIVE CONTRIBUTORS, GENERALIST VS SPECIALIST by analyzing performance of individuals participating in the process.	24	17	2	0.14	Organizational

$C(P1)$ in Table I corresponds to the number of items in each group. We notice that some of the groups have only one item while some have as many as 13. However, majority of them have two to seven members. Though the focus is on exploring variations not on frequency, interestingly, some of them are stated by multiple participants. It is an indicator if the problem is comparatively more common and intuitive. Here, problem statement six and two are mentioned by 13 and 9 respondents respectively indicating that these are the most common problems faced by practitioners.

B. Classification

The shortlisted problems in Table I are classified into various categories such as control flow, organizational, case, information, application and time, based on the process mining perspective [4] [5] [22] [23]. In Table I, 17 out of 30 problem statements belong to the categories mentioned as example in the survey to stimulate the respondent. Remaining problems belong to the following categories:

1) **Case Analysis**: Focuses on properties of a case such as its path, people working on it [4].
2) **Preventive Analysis**: Focus is on using insights derived from event log mining for better planning and process design even before the execution happens.
3) **Comparative Analysis**: Process mining is performed to compare multiple processes from various perspectives along different dimensions such as experience of actors (novice vs experienced) and turnaround time [24] [25].
4) **Enhancement**: Improve or extend the existing process model using information about actual process recorded in the event log [4].

From Table I, we notice that out of all, seven problems involve analysis from control perspective making it the most frequent category. To the best of our knowledge, it is one of the most explored perspective in literature as well [15]. Fig. 1 depicts all categories with number of members in each of them. We present the benefits of solving problems from each category as suggested by respondents in the first survey. It requires consolidation at two levels: first on the basis of points grouped under one problem statement, and second for the problems belonging to the same category (as indicated by the values in the bracket).

The most important benefits of solving problems from each category based on specific problems mentioned in Table I are as follows:

1) **Control Analysis** (7 problem statements):
   - Minimize unnecessary efforts and delay by detecting the point where process went wrong and its overall impact to take timely corrective actions.
   - Reduce rework as the process reality visualized in more reliable way to quickly identify the inefficiencies and phases more prone to error.
2) **Organizational Analysis** (4 problem statements):
   - Improve coordination between members to avoid issues and delays due to differences in understanding.
   - More productivity, smoother project execution, better management of resources, avoid rework because of wrong allocation, efficient management of new teams.
   - Identify more efficient and low performing individuals to augment training requirements and enable objective appraisal.
3) **Conformance Analysis** (4 problem statements):
   - Reduce bugs and client escalations because of deviations from requirements and drafted schedules as adherence checked continuously.
   - Identify the cause for inefficient performance from process perspective, that is, see if the process is followed religiously or not and the impact of those deviations.
   - Control risk because of deviations more effectively and mitigate their ill effects at every stage.
4) **Enhancement Analysis** (4 problem statements):
   - Tailor the process with empirical establishment based on cost benefit analysis to find the right balance of resources deployed and outcome.
   - Increasing usability and better management of quality audits would go a long way in enabling more usage of information systems, that would eventually benefit the entire team. Easy to take corrective actions
   - Better knowledge management and learn from as-is process.
   - Ensure processes are not overhead and adapted to the need by continuous improvement.
   - Help in reducing review and rework effort due to delayed discovery of defects.
5) **Preventive Analysis** (4 problem statements):
   - Empirically convince team for more religious meaningful process adoption. Improve trust, awareness and efficiency for the processes.
   - Save a good amount of time, make the project execution consistent, reduce risk of effort overrun helps avoid penalty clauses. Manual rework in creating project plan will be reduced.
   - Large extent of data and process redundancies will be solved.
6) **Comparative Analysis** (3 problem statements):
   - Improve learning curve – equip novice with the best practices of experienced efficient professionals by understanding experience association with the process adopted and efficiency.
   - Make things more and more people independent. Find best practices which compensate low skilled people by comparing impact of people and process.
   - Reduce number of tickets, turnaround time, and improve performance based on multiple parameters.
7) **Performance Analysis** (2 problem statements):
   - Improve throughput and efficiency of the system.
   - Help in better planning and timely issue resolutions. Product release can happen on time without a fuss.
   - Effort overrun and wastage can be avoided.

- Timeline which is always a challenge can be managed more efficiently by addressing the identified bottlenecks.

8) **Case Analysis** (2 problem statements):
   - Balance between rework and verification. This will help in better utilization of resources and taking informed steps.
   - Allow team to review their mistakes and strengths by understanding cases over a period of time.
   - Improve system response and reduce cost.

Effectively, we obtain a list of 30 process mining problems with benefits of solving them. It presents a range of perspectives (categories) spanning across multiple phases of SDLC. Now, we conduct the second survey where the objective is to validate if others also feel that solving the identified problems is important. This helps establish the importance and need for process mining of software repositories to solve some of the identified problems.

C. Importance Survey

A survey is designed to understand if the identified problems are worth solving.

1) Survey Design: The survey has all the 30 problems with 5 options as:
[*Essential* | *Worthwhile* | *Unimportant* | *Unwise* | *I don't understand*], where the question asked is:

> *We have identified some process related problems encountered by practitioners while managing IT projects. In your opinion, please indicate how important it is to have a process mining team solve this problem.*

We decide for the above options because it meets the need for both positive (*Essential* and *Worthwhile*) and negative (*Unimportant* and *Unwise*) scale with variation in intensity. Since the problems are succinct, the participant may find it difficult to understand them without a detailed context. Therefore, the fifth option allows expression of a problem not understood. As an introduction, we include only the definition of process mining in this survey as the focus is to validate the identified problems. We mention the significance of each option as following to avoid any differences in understanding:

- *Essential:* Problem poses many challenges and should be dealt on a high priority.
- *Worthwhile:* Important to solve and will help the managers.
- *Unimportant:* Its not worth solving the problem.
- *Unwise:* Strongly discourages the team to solve the problem.
- *I don't understand:* Problem statement not clear or difficult to understand.

We do not mention that the problems are identified by surveying their colleagues to ensure that the responses are not influenced to the best level. Also, we ask the following basic details: current role, total work experience and total project management experience. To ensure that they can mention any other challenge, if missing, we add a free text question in the end as:

> *Mention any other process related problems that you encounter and not listed in the above questionnaire*

The final two page online survey is administered to the target population.

2) Participants: A consolidated list of 30 problems (shown in Table I) is sent to 160 distinct randomly selected practitioners with the same characteristics of having project management experience and having handled teams. We sent the survey to distinct participants, that is, no overlap with the participants of the first survey, to avoid any bias. We received responses from 46 people out of 160 with the response rate of around 29%, which is almost double the response rate of previous survey. One of the major reasons for the increased response rate is that the second survey is closed type where the respondent has to select one out of the given options. While answering all the questions is preferred, respondent can answer as many as possible due to high number of questions. We got 1262 responses instead of 1290 (43×30) that means only 28 (2%) are unanswered. Participants cover a broad spectrum in terms of role, type of projects, and experience. We have participants with total work experience from 6 years up to 25 years with median as 12.5 years. Similarly, the project management experience is as low as 1 year and for some it is as high as 15 years. Median project experience is 6 years which is quite high. Likewise, we have responses from diverse horizontals within the organization such as Finacle, manufacturing, maintenance, support, development, and quality. We map different role titles for different horizontals to TL, PM, and SPM based on similarity in responsibilities and job level. The role distribution is as follows: TL - 14%, PM - 49%, SPM - 26%, and the rest includes GPM and senior product line manager. Hence we capture opinions of diverse professionals making our study rich.

3) Survey Results: Out of 1262, very few responses (only 42) are *I don't understand*. Two reasons are possible for this: either the respondent does not have experience with that kind of project and therefore, finds it difficult to understand it or our summarization is not clear. Interestingly, problem statement $P26$ from Table I is indicated as not understandable by 9 respondents which essentially signals that the formulation requires more clarity. We notice that up to three respondents have selected the option of *I don't understand* for other statements. It is mainly because of diversity in roles, experiences, and domain.

From Fig. 2, positive $(+ve)$ refers to *Essential and Worthwhile* and negative $(-ve)$ implies *Unimportant and Unwise*. A count of both $+ve$ and $-ve$ for each problem statement is presented in Table I. We observe from Table I that maximum responses are positive. Positive response highlights the need for process mining specialists' team to work towards solving the problem. Negative responses are significantly less which can be attributed to the fact that the initial set of problems is coming from practitioners with emphasis on variations. Therefore, none of the listed problem is absolutely irrelevant.

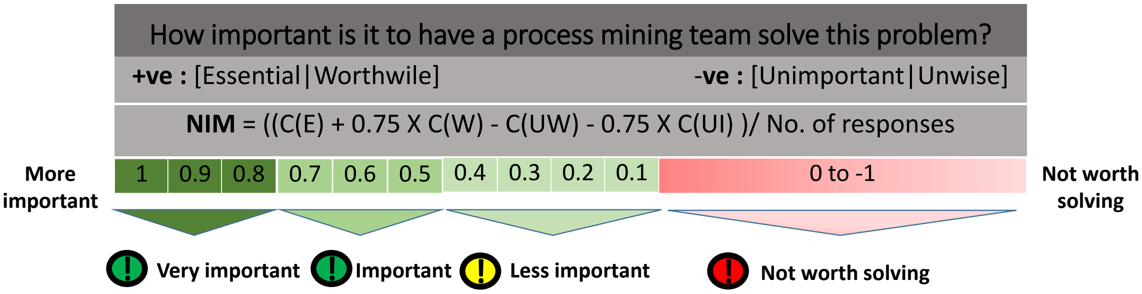
How important is it to have a process mining team solve this problem?
+ve : [Essential|Worthwile]
-ve : [Unimportant|Unwise]
NIM = ((C(E) + 0.75 X C(W) - C(UW) - 0.75 X C(UI))/ No. of responses
More important
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0 to -1
Not worth solving
Very important
Important
Less important
Not worth solving

Fig. 2: Importance Analysis Framework.

Interestingly, the distribution of negative responses is not uniform. While problem statement $P30$ has as many as 17 negative responses, many statements $(P1, P2$ and $P4)$ have no negative response. Some of the possible reasons for negative response to a particular problem are:

- The problem is already solved to a satisfactory level and it is not worth spending resources in solving it further. It is based on a remark, *"We already have solution for this"*, mentioned by a participant as comment along with a question where *Unwise* is marked as an option.
- The participant has never faced that problem and feels it is not really a problem. For example, we got similar comments such as *"Some of the listed problems are project and domain specific"*, from couple of respondents.

We receive few comments in the free text question. No new problem emerged as a result of those comments. Most of them are general remark on the listed problems where the most common are the following:

- *"Need to identify the data required for solving these problems thus, ensure we have system in place to automatically fetch required details."*
- *"Most of the identified problems are associated with the maintenance phase, can we apply process mining to improve requirements phase which is a perennial problem."*

D. Net Importance Analysis

We propose *Net Importance Metric* (NIM) to objectively measure the importance of solving each problem where the count of positive and negative responses are taken as input parameters (refer to Fig. 2), that is,

Net Importance Metric,

$$\boxed{NIM = \frac{C(E) + 0.75 \times C(W) - C(UW) - 0.75 \times C(UI)}{C(E) + C(W) + C(UW) + C(UI) + C(DU)}}$$

where,
$C(E)$ = Number of responses as *Essential*,
$C(W)$ = Number of responses as *Worthwhile*,
$C(UW)$ = Number of responses as *Unwise*,
$C(UI)$ = Number of responses as *Unimportant*, and
$C(DU)$ = Number of responses as *I don't understand.*

Metric NIM measures the net importance of solving a problem based on responses from the participants. We use this metric to determine the joint effect of positive and negative responses. *Essential* expresses stronger need to solve a problem as compared to *Worthwhile*. *Unwise* strongly discourages the team to spend efforts in solving a problem while *Unimportant* reflects the same with a comparatively weaker intensity. Therefore, we use multiplier factor of $1, 0.75, -1$, and -0.75 to scale the intensity of positive and negative responses respectively. The difference between $(+ve)$ and $(-ve)$ responses is normalized for comparison because all the problems do not have same number of ratings.

As depicted in Fig. 2, the value of NIM lies between -1 and 1 where 1 corresponds to the *most important*, 0 corresponds to the *least important*, and less than 0 corresponds to not worth solving. Value of NIM is less than 0 (indicated in red) for problems with more negative responses thus, solving those problems is not recommended. Problems with NIM value up to 0.5 (indicated with light green color) are less important while problems with values highlighted in dark green color correspond to highly important ones. It is based on the fundamental perspective that some problems have more impact and need to be dealt on a high priority. Even though some problems exist, they do not require immediate attention. In Table I, problems are arranged in decreasing order of NIM. In terms of importance, we make the following observations from Table I:

1) The first six problems have NIM value greater than 0.8. Therefore, solving these problems is **very important**. We notice that 40 or more responses out of 43 are *Essential* or *Worthwhile* with up to two responses as negative. It means that the majority of the managers feel that these problems need to be solved.
2) Majority of the problems (19 out of 30) have value within the range of 0.5 to 0.7, that is, **important**. The reason is that while most participants have responded positively, *Worthwhile* is a more frequent choice. Only a few responses are negative.
3) Only the last 5 problems belong to the category of **less important**. We observe many negative responses, many *Worthwhile* and a few *Essential*. Therefore, concentrating

on these problems is not recommended as compared to others.

Effectively, all the listed problems are validated by experienced professionals. Practitioners believe that solving some problems is more important as compared to others. Therefore, as a process mining specialist one can make an informed choice on which problems to focus on and solve first.

E. Commonality and Importance

Total members in group of each problem statement, that is, $C(P1)$ along with the NIM gives an idea on commonality and importance of problems. Here, we consider statements with equal to or more than 5 members as common and with NIM value more than 0.5 as important (refer to Fig. 2). We can have the following combinations and classification of problems from Table I:

1) **Common and Important:** Problems mentioned by many participants in phase I of the study and also positive responses outweigh negative responses. This is the most preferred set of problems to be solved. For example: $P1$, $P2$, $P3$, $P5$, $P6$, $P8$, $P9$, $P10$ and $P24$
2) **Common and Unimportant:** The impact of a problem even though encountered by many people may not be very high. As inferred from multiple responses the comparison of process along various dimensions such as process adopted by different bug owners and process for resolution of different bug types is an interesting problem (refer to $P29$ in Table I). However, it is not very important in terms of direct benefit to the overall improvement.
3) **Uncommon and Important:** We notice that there is a large set of problems which are mentioned by few people during the open survey. But when these problems are presented to other practitioners, the importance and benefits of solving them is appreciated. Problems from Table I belonging to this class are $P4$, $P7$, $P11$, $P12$, $P13$, $P14$, $P15$, $P16$, $P17$, $P18$, $P19$, $P20$, $P21$, $P22$, and $P25$.
4) **Uncommon and Unimportant:** Some problems are very specific and expressed by few people. Infact they are not considered important by many other practitioners. Hence, giving preference to these problems is not recommended. For instance: $P26$, $P27$, $P28$, and $P30$.

Addressing the issues related to specific managers will be insufficient. Having identified that broader systematic problem exists, interventions can be designed and applied to address such problems and hence, aid managers.

IV. Gaps and Challenges

We identify gaps and challenges for identified problems by studying literature. While research is lacking for some of the identified problems, some have already been studied by researchers and solution needs translation to usable tool. In Table II, we elaborate on top five problems classified to most common and frequent problems category. Solving some of these important problems (such as $P2$) needs designing of new tools and techniques because the in-depth research itself is lacking. Whereas others such as $P1$ and $P6$ can be solved by direct application of existing process mining techniques which have not been explored for software repositories so far. Problems such as $P3$ and $P5$ need to be studied further before designing any usable tool.

To solve some of these problems, we may need data that is not readily available and sometimes not stored automatically in the system. As mentioned by many interviewee and survey participants, the availability of required data is one of the major challenge. However, they understand and agree with the benefits of solving a problem even if it requires storing some extra information. If researchers' can solve the problem and tell them the exact data requirements, they can start capturing that as well.

General challenges in process mining of software repositories

Some of the issues that are encountered while performing process mining in general [4] [32] and that will be valid for process mining of software repositories are as follows:

- *Noise:* Logged data may be incorrect or incomplete.
- *Merging of event data:* Access to multiple heterogeneous IS to analyze the process.
- *Hidden tasks:* Tasks are not explicitly defined and captured in data.
- *Different perspectives:* Additional information requirement to process mine from diverse perspectives.
- *Concept drift:* Process is changing while being analyzed for reasons such as change in requirements, and to make up for deviation from project planning and estimates.
- *Operational support:* Unavailability of good quality data sources in real-time for online operational support activities: detect, predict, and recommend.
- *Corrective actions:* Process mining helps identify the inefficiencies. However, most of the times no silver bullet solution can be suggested to address them. Actionable information is derived which can be used according to project requirements.

V. Implications

The previous work highlights the potential of process mining of software repositories [16]. Many process discovery algorithms are proposed and software repositories are process mined from multiple perspectives such as control flow and organizational perspective [15]. The focus has been on issue resolution process (maintenance). Based on the results reported in this paper, we suggest several applications of process mining on different software repositories to benefit the managers and help them achieve better project maintainability, stability, productivity and quality. Researchers can directly utilize these problem statements to extend the applicability of process mining in software engineering.

1) Application at various stages of SDLC: We notice that the problems mentioned by managers, which are sought to be solved by novel applications of process mining span across various stages of SDLC. Starting from requirements adherence to maintenance phase, it can help enhance existing process

TABLE II: ID refers to problem ID in Table I. Tick indicates if the major gap is in process mining research or tools. Comments provide additional information.

ID	Research	Tool	Comments
P1		✓	Mining event logs to identify bottlenecks is well studied [26]. Also explored for issue resolution process [7] [15]. Need to extend the same for other repositories. Tool to automate delay detection for software repositories will help.
P2	✓		Risks are predicted from the logs of past process executions [27], however, not explored to predict escaped defects.
P3	✓	✓	Though analysis of missing and unnecessary activities that negatively affects the process is performed on synthetic logs still needs investigation for software repositories considering the differences in their properties [28].
P5	✓	✓	Reopening of tickets analyzed from event logs however, not with a view to design recommendation system [7] [29] [30].
P6		✓	Event logs are process mined from organizational perspective [31]. Customize for software repositories and design a tool to utilize its application as task allocation system.

capabilities. Some of the mentioned problems are confined to single repository while many others requires data from multiple repositories simultaneously.

2) Process mining from diverse perspectives: We classify identified problems into 8 different process mining perspectives. We can address several problems from each perspective and all of them contribute differently to benefit the project quality.

VI. Study Limitations

In our study, we survey and interview participants performing managerial roles for diverse project types such as development projects, maintenance projects and support. Nevertheless, they work in the same organization using similar processes and guidelines. Though having diverse participants from the same organization enables deeply exploring experiences from several perspectives, organizational culture may create a bias. Even though the organization is huge and CMM 5, the problems encountered by managers may not be as important to managers of other companies with different organizational cultures. Since the questions identified covers a vast spectrum of process mining types, we can expect them to reflect problems encountered by managers which process mining specialists' team can solve.

We included some examples to illustrate process mining applications in the first survey. They can shape their thoughts and make them think mainly in the direction of similar challenges. While we notice that many challenges (such as $P1$, $P6$, $P12$, $P14$) belong to the categories included as example, several new classes emerged as a result of the study.

We notice that there are few problems from requirements gathering phase (also pointed out in a comment for second survey). Based on our understanding from interviews, practitioners' response is limited by the scope of process mining. Since requirements gathering is communication intensive phase resulting into Software Requirement Specification (SRS) document, it is difficult to find event logs thus, refrained them from soliciting requirement phase challenges.

Some problems are directly stated by respondents and others are inferred from the long statements received in the responses. Informal discussion during the interview complimented the process of generic problem formulation but the interview is conducted with a small set of respondents. During the second survey, importance may be influenced by individual differences among the participants themselves.

VII. Conclusion

Process mining of software repositories involves working with different kinds of IS and identifying the exact questions to be answered to improve the process. We conducted a two-phase online survey study and an interview study to identify the challenges encountered by managers and the importance of solving those problems. We have identified 30 different questions that could be categorized into 8 categories based on the type of analysis to be performed. In addition to this, we conducted an importance analysis survey with distinct participants to validate the importance of solving a particular problem. Investigation of most important problems helps to identify gaps and sets requirement to transform existing research to usable tools for efficient software process management. To the best of our knowledge this is the most comprehensive catalog of questions published to date which process mining team should answer. Our results highlight the need to apply process mining on software repositories to effectively and efficiently manage projects. We believe that the results provide an important input to researchers to solve a problem by novel applications of process mining and help managers.

Acknowledgement

The presented research work is supported by Prime Minister Fellowship awarded to the first author. The authors are thankful to the participants from two surveys, Veerabhadra Rao Chilukuri and Gopakumar M.G. for their help with the surveys. We acknowledge SERB, CII, and Infosys Limited for the support.

References

[1] H. Kagdi, M. L. Collard, and J. I. Maletic, "A survey and taxonomy of approaches for mining software repositories in the context of software evolution," *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 19, no. 2, pp. 77–131, 2007.

[2] A. E. Hassan, "The road ahead for mining software repositories," in *Frontiers of Software Maintenance, 2008. FoSM 2008.* IEEE, 2008, pp. 48–57.

[3] A. Weijters, W. M. van der Aalst, and A. A. De Medeiros, "Process mining with the heuristics miner-algorithm," *Technische Universiteit Eindhoven, Technical Report WP*, vol. 166, 2006.

[4] W. Van Der Aalst, A. Adriansyah, A. K. A. de Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. van den Brand, R. Brandtjen, J. Buijs *et al.*, "Process mining manifesto," in *Business process management workshops*. Springer, 2012, pp. 169–194.

[5] W. M. van der Aalst, H. A. Reijers, A. J. Weijters, B. F. van Dongen, A. Alves de Medeiros, M. Song, and H. Verbeek, "Business process mining: An industrial application," *Information Systems*, vol. 32, no. 5, pp. 713–732, 2007.

[6] J. Samalikova, R. Kusters, J. Trienekens, and A. Weijters, "Process mining support for capability maturity model integration-based software process assessment, in principle and in practice," *Journal of Software: Evolution and Process*, 2014.

[7] M. Gupta and A. Sureka, "Nirikshan: Mining bug report history for discovering process maps, inefficiencies and inconsistencies," in *Proceedings of the 7th India Software Engineering Conference*. ACM, 2014, p. 1.

[8] E. Kindler, V. Rubin, and W. Schäfer, "Activity mining for discovering software process models." *Software Engineering*, vol. 79, pp. 175–180, 2006.

[9] B. Akman and O. Demirors, "Applicability of process discovery algorithms for software organizations," in *Software Engineering and Advanced Applications, 2009. SEAA'09. 35th Euromicro Conference on*. IEEE, 2009, pp. 195–202.

[10] W. Sunindyo, T. Moser, D. Winkler, and D. Dhungana, "Improving open source software process quality based on defect data mining," in *Software Quality. Process Automation in Software Development*. Springer, 2012, pp. 84–102.

[11] P. Knab, M. Pinzger, and H. C. Gall, "Visual patterns in issue tracking data," in *New Modeling Concepts for Todays Software Processes*. Springer, 2010, pp. 222–233.

[12] A. Begel and T. Zimmermann, "Analyze this! 145 questions for data scientists in software engineering." in *ICSE*, 2014, pp. 12–13.

[13] K. P. Kim, "Mining workflow processes from distributed workflow enactment event logs," *Knowledge Management & E-Learning*, vol. 4, no. 4, pp. 528–553, 2013.

[14] A. Sureka, A. Kumar, and S. Gupta, "Ahaan: Software process intelligence: Mining software process data for extracting actionable information," in *Proceedings of the 8th India Software Engineering Conference*. ACM, 2015, pp. 198–199.

[15] M. Gupta, A. Sureka, and S. 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*. ACM, 2014, pp. 122–131.

[16] W. Poncin, A. Serebrenik, and M. van den Brand, "Process mining software repositories," in *15th European Conference on Software Maintenance and Reengineering*. IEEE, 2011, pp. 5–14.

[17] J. Song, T. Luo, and S. Chen, "Behavior pattern mining: Apply process mining technology to common event logs of information systems," in *IEEE International Conference on Networking, Sensing and Control.* IEEE, 2008, pp. 1800–1805.

[18] S. Phillips, G. Ruhe, and J. Sillito, "Information needs for integration decisions in the release process of large-scale parallel development," in *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*. ACM, 2012, pp. 1371–1380.

[19] T. Fritz and G. C. Murphy, "Using information fragments to answer the questions developers ask," in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1*. ACM, 2010, pp. 175–184.

[20] T. D. LaToza, G. Venolia, and R. DeLine, "Maintaining mental models: a study of developer work habits," in *Proceedings of the 28th international conference on Software engineering*. ACM, 2006, pp. 492–501.

[21] J. Sillito, G. C. Murphy, and K. De Volder, "Questions programmers ask during software evolution tasks," in *Proceedings of the 14th ACM SIGSOFT international symposium on Foundations of software engineering*. ACM, 2006, pp. 23–34.

[22] V. Rubin, C. W. Günther, W. M. P. Van Der Aalst, E. Kindler, B. F. Van Dongen, and W. Schäfer, "Process mining framework for software processes," in *Proceedings of the international conference on Software process*, ser. ICSP'07. Springer-Verlag, 2007, pp. 169–181.

[23] W. M. Van der Aalst and A. Weijters, "Process mining: a research agenda," *Computers in industry*, vol. 53, no. 3, pp. 231–244, 2004.

[24] T. Mamaliga, "Realizing a process cube allowing for the comparison of event data," Ph.D. dissertation, Masters thesis, Eindhoven University of Technology, Eindhoven, 2013.

[25] M. Gupta and A. Sureka, "Process cube for software defect resolution."

[26] E. Vasilyev, D. R. Ferreira, and J. Iijima, "Using inductive reasoning to find the cause of process delays," in *Business Informatics (CBI), 2013 IEEE 15th Conference on*. IEEE, 2013, pp. 242–249.

[27] R. Conforti, M. de Leoni, M. La Rosa, W. M. van der Aalst, and A. H. ter Hofstede, "A recommendation system for predicting risks across multiple business process instances," *Decision Support Systems*, vol. 69, pp. 1–19, 2015.

[28] G. Calderón-Ruiz and M. Sepúlveda, "Automatic discovery of failures in business processes using process mining techniques."

[29] S. Suriadi, C. Ouyang, W. M. van der Aalst, and A. H. ter Hofstede, "Root cause analysis with enriched process logs," in *Business Process Management Workshops*. Springer, 2013, pp. 174–186.

[30] M. Heravizadeh, J. Mendling, and M. Rosemann, "Root cause analysis in business processes," 2008.

[31] W. M. Van Der Aalst, H. A. Reijers, and M. Song, "Discovering social networks from event logs," *Computer Supported Cooperative Work (CSCW)*, vol. 14, no. 6, pp. 549–593, 2005.

[32] A. Tiwari, C. J. Turner, and B. Majeed, "A review of business process mining: state-of-the-art and future trends," *Business Process Management Journal*, vol. 14, no. 1, pp. 5–22, 2008.