Land of Lost Knowledge: An Initial Investigation into Projects Lost Knowledge

Márcia Lima
Institute of Computing
Universidade Federal do Amazonas
Manaus, Brazil
marcia.lima@icomp.ufam.edu.br

Iftekhar Ahmed
Department of Informatics
University of California
Irvine, USA
iftekha@uci.edu

Tayana Conte
Institute of Computing
Universidade Federal do Amazonas
Manaus, Brazil
tayana@icomp.ufam.edu.br

Elizamary Nascimento
Institute of Computing
Universidade Federal do Amazonas
Manaus, Brazil
elizamary.souza@icomp.ufam.edu.br

Edson Oliveira
Department of IT
Secretaria de Estado da Fazenda
Manaus, Brazil
edson.cesar@sefaz.am.gov.br

Bruno Gadelha
Institute of Computing
Universidade Federal do Amazonas
Manaus, Brazil
bruno@icomp.ufam.edu.br

***Abstract*—*Background:* Software development teams adopt various communication tools to support coordination and team interaction during the software development process. Among many other communication channels, developers' use instant messaging to discuss ideas, decisions and other project related issues with team members. Due to the informal nature of instant messaging, many of these discussions and decisions are lost. This situation could be even more critical in startups and other software companies that rely more heavily on instant message tools or other informal communication channels.**
***Aims:* This work investigates the effectiveness of using a semi-automatic approach for identifying, extracting, and determining a project's lost knowledge that was discussed using unstructured communication tools such as instant message.**
***Methodology:* We employed data-mining techniques to automatically retrieve discussions from instant message logs and showed them to the project managers to identify lost knowledge from two startup companies.**
***Results:* Our results demonstrate that the data-mining technique was capable of retrieving sentences with relevant issues discussion; reaching a precision of 75% at the first 10 relevant sentences evaluated. Moreover, the qualitative analysis conducted involving project managers shows an association of retrieved sentences with the project's lost knowledge.**
***Conclusion:* Our findings indicate that automated approaches can be used to identify such lost knowledge in software development projects. Follow-up interviews revealed the interest of PMs in adopting such automated tools in other projects.**

***Index Terms*—Communication Tools, Software Knowledge, Data Mining, Empirical Study**

I. Introduction

Software development teams adopt various communication channels [1]–[4] to support the collaborative development model and coordinate tasks. Not only distributed but also collocated software teams use these channels. Researchers have investigated the artifacts generated through using these channels such as bug report descriptions, source code linguistic data, requirements documents, mailing lists content, and chat messages [1]–[5]. However, recent studies show that software development teams are increasingly using social media for communication purposes [6] instead of using the traditional communication channels. Alkadhi *et al.* [1] found that chat messages have replaced emails in some development teams.

When teams use informal channels such as instant messaging (IM) for communication, relevant discussions pertaining to the software development and management resides in IM log files. According to Alkadin *et al.* [1], [2] IM log files are rich source of information and can help in identifying important issues. Along with many other things, these log files contain crucial decisions, design discussions, and issues related to the project that may not be preserved and implemented midst of various discussions happening among the team members.

Startups need to rapidly evolve and adapt to an uncertain market. Due to such dynamic nature it may not be always possible for startups to rely on traditional communication channels [7], [8]. Hence, startups rely more heavily on IM tools for facilitating interaction between team members. Since projects lose knowledge (from here on we will refer this as Project's Lost Knowledge (PLK)) even when using established communication mechanisms [9], [10], it's more likely that startups relying on IM would lose even more. Moreover, startups should be more affected by PLK compared to established companies, since forgetting about an important feature or a crucial design decision can lead to a failed software or even a failed company in the worst case. However, to the best of our knowledge, no prior research investigated how prevalent PLK is in startups, nor what are the impacts of PLK for startups. We aim to close this gap through a mix of automatic and manual analysis of the communication of development teams from two software startup companies. We start by using a ranking algorithm to retrieve relevant information from IM logs. Then we conduct a study involving the project managers to identify the PLK. We also follow up by contacting the

978-1-7281-2968-6/19/$31.00 ©2019 IEEE

managers to understand what were their perceptions regarding the lost knowledge.

The goal of this study is to investigate whether and how we can identify PLK from IM logs effectively. Specifically, this study aims at answering the following research questions:

- ***RQ 1:*** *Can we extract relevant discussion from developers IM logs using automated techniques?*
- ***RQ 2:*** *Can we identify PLK from the relevant discussions by interviewing project managers?*

II. Related Work

A. Software Project Repositories: Data and Knowledge

Software development process produces different data sources resulted from the teams' daily interaction and evolutionary changes to software artifacts [11]. Chen *et al.* [11] classifies these sources into structured and unstructured. Structured sources have a known form, such as source code parse trees, execution logs and traces, mailing list metadata and chat log metadata. The unstructured sources mostly use natural language text such as: bug report descriptions, source code linguistic data, requirements documents, mailing lists content, and chat messages. Researchers have investigated both structured and unstructured sources. Instant messages (IM) tools are used for quick questions and clarifications, scheduling or coordination team' activities, and, social purposes, however the main purpose of workplace IM is to discuss work [12].

While analyzing different communication mechanisms, researchers found that knowledge is lost in projects. Soria and Hoek [9] found that not all of the knowledge generated during design meetings are captured because spoken knowledge evaporates. Burge and Brown [10] found that design rationale decisions are usually not captured and are therefore lost. They proposed an Eclipse plug-in that integrates the rationale with the source code. Manteuffel *et al.* [13] developed a tool to deal with architectural decision lost knowledge. They emphasize that architectural decisions are often not documented but reside in the architects mind as tacit knowledge. Kleebaum *et al.* [14] aim to support developers' tasks using summarization techniques to promote capture and use of decision knowledge into developers' daily work. Codoban and Ragavan [15] found that developers frequently refer to software history to gain knowledge. They propose a software history model that gives identity to Version Control System data history.

Bavota [16] highlighted the growth of unstructured data and describes the application of Mining Unstructured Data (MUD) in software engineering. They discuss three techniques for MUD, unstructured data repositories available for mining, and potential applications of MUD in software engineering. Alkadhi *et al.* [2] investigated the use of content analysis and machine learning techniques for extracting rationale from chat messages. Viviani *et al.* [3] used design-related keywords to automatically extract design information from pull request discussions. Francois *et al.* [4] proposed a Knowledge Trace Retrieval (KTR) system to retrieve elements of problem solving and design rationale from business emails.

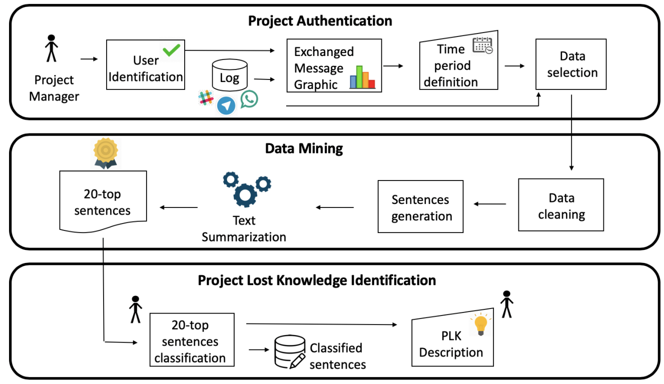

Fig. 1. The PLK identification approach process

Our goal is to identify PLK from IM logs. To the best of our knowledge no prior study tried to use a mix of automated and semi-automated techniques to extract relevant sentences from the developer's IM in order to identify the PLK.

B. Ranking Algorithms

Ranking algorithms are used for automatically sorting objects according to their relevance. Gambhir and Gupta [17] found that many Information Retrieval (IR) problems such as text summarization are by nature ranking problems. We use a generic text summary technique to automatically identify and extract PLK from developers' chat message. Position-based measures, such as $Precision@k$ and Mean reciprocal rank (MRR), are used to evaluate the performance of ranking models [18]. Precision is the probability that a retrieved sentence is relevant. $Precision@k$ ($P@k$) is the fraction of relevant results out of the first k returned [19]. It captures the ranking quality for applications where only the first k results matters. MRR measure denotes the rank position of the first relevant document, if all relevant documents are at the top of the ranked list the MRR value is 1 [18].

III. Methodology

In this work, we investigate developers' IM logs collected from two different software startups. Figure 1 shows the overall process conducted in order to enable the PLK identification. We describe the data collection method and the phases of the applied research method to determine the PLK.

A. Data Collection

We collected and analyzed the IM log of two different development teams: Team A and Team B, from two startups companies. Though both teams were collocated, team members had different work times. Both teams used IM tool to support the software development process and some management activities. We chose to analyze one project for each team. We selected projects that had delivered a current operational version of a software that the startup companies want to evolve. Therefore, the companies wanted to check for any lost knowledge in these projects.

Team A is composed of 5 members: 1 business manager, 1 project manager, 2 developers, and 1 tester. Team A is specialized in the development of crowd interaction through technology for entertainment purposes. They dealt with incomplete and evolving requirements. The team's interaction was conducted through in-person meetings and IM tools. We collected Team A's IM log ranging from November 17, 2017 to October 18, 2018. 104 days and 3,325 messages in total were analyzed. In this period, the team was engaged in a project for enabling crowd interaction through mobile devices for a nationwide folklore festival. This team works for a two-years software startup, composed of eight employees. The team used evolutionary prototypes lifecycle to develop the system and adopted Swift and Iconic as programming languages.

Team B is composed of 6 members: 1 projects owner, 2 scrum master, 1 interface designer, 2 developers, and 1 tester. Team B works on developing general software solutions for other companies. In the analyzed project, the team was engaged in a project related to an e-commerce mobile application. For this project, Team B was collocated, except for one of the developers. The team used IM tool for communication, discuss the software designs, and for scheduling meetings. Team B also used a Kanban board and some documentation, such as prototypes. We collected IM log ranging from March 4, 2016 to November 29, 2018. 285 days of conversation and 5,256 exchanged messages in total were analyzed. This team works for a five-years software startup company, composed of nine employees, adopting Scrum as a software development process and Java as the programming language.

B. Semi-automated Approach for relevant sentence identification

To answer the *RQ 1- Can we extract relevant discussion from developers IM log using automated techniques?* - we applied text summarization algorithm on the collected IM log files. Applying text summarization algorithm in natural language requires data pre-processing in order to optimize the algorithms execution [4], [17], [20]. We conducted the following pre-processing steps: Splitting the entire chat log into individual sentences, removing stopwords, non-alphabetic data, and administrative messages sent by the chat message tool. The text summarization algorithm used is based on term relevance measured using Term Frequency (TF) and Inverse Document Frequency (IDF) [21]. $TF \times IDF$ measure is used to compute the relevance score of each sentence with the whole document [4], [20]. We used $TF \times IDF$ to retrieve relevant sentences that could be associated with the PLK.

The text summarization algorithm provided a ranked list of important discussion by the development team. This ranking was calculated based on the relevance score of 1,726 different sentences (845 sentences from Team A, 881 sentences from Team B). A total of 240 relevant sentences were identified and extracted, as shown in Table I. In order to measure the effectiveness of text summarization algorithm, we asked project managers from each project to classify the relevance of the 20-top sentences in their respective project. The ranking

TABLE I
EVALUATION METRICS VALUES

	Team A	Team B
#Evaluated Sentences	845	881
#Displayed Sentences to PMs	100	140
MRR	1	1
P@3	100%	85%
P@5	88%	80%
P@10	74%	75%
% of analyzed days	47.11%	20%

algorithm was evaluated using $Precision@k(P@k)$ [19] and MRR [18] measures. In order to measure the precision value of the 3-top, 5-top and 10-top results, we calculated the $P@3$, $P@5$ and $P@10$ values.

C. Semi-automated Approach for PLK determination

To answer the *RQ 2: Can we identify PLK from the relevant discussions by interviewing project managers?* - we need to determine if relevant sentences can be associated with PLK. This step introduces some unique challenges: 1. It requires semantic analysis of the data; 2. It requires prior knowledge about the project; and, 3. It involves human judgment, given that the concept of relevance depends on the evaluators' perspective, role and maturity. So the project managers evaluated the results from the semi-automatic phase, assessing the relevance of extracted sentences to determine the PLK.

Team A's project manager (PM-A) choose five different periods of interest to identify the PLK. These five periods totaled 49 days. Team B's project manager (PM-B) selected seven different periods of time, totaling 57 days, to conduct his analysis and identify the related PLK. The last line of Table I identify the percentage of analyzed days determined by PMs interesting. Relating to Team A's IM log, the PM-A analyzed 47.11% of the days on which communication occurred. The PM-B analyzed 20% of the days on which Team B exchanged messages. An evaluation screen displayed the 20-top most relevant sentences to the managers who then commented on whether each of the sentences were associated with PLK or not. Managers were then asked to describe the PLK identified from their perspective.

D. Manual Data Analysis

We first categorized the different types of software issues registered in IM log files. To do so, two of the authors coded a random sample of IM's sentences separately; after that, they discussed the differences in their coding. We measured the inter-rater agreement using Cohens Kappa Coefficient [22], which were 0.61 and 0.64, for Team A and Team B, respectively. These values indicate substantial agreement according to the interpretation proposed by Landis and Koch [23].

Table II summarizes the categories of identified relevant software issues in IM logs. We also asked the PMs to categorize the topics of their identified PLK using the same categories.

TABLE II
CATEGORIES OF TOPICS DISCUSSED IN IM

Category	Description
Requirement	The features, services, and constraints of a software product
Design	Software design
Development	Configuration, implementation
Deployment	Software installation
Delivery	Software delivery
Testing	Testing and testability
Acceptance	Users' feedback
Support	Assistance on the installation process, and troubleshooting
Company Business	Company's management
Management	Product's management
Marketing	Product's marketing
Product Vision	Future plans and similar applications
Clarification	Generic questions
Documentation	Project documentation in and out-code

E. Follow up interview

We contacted the managers, to understand if they made any changes to their communication process to reduce PLK. We asked the following questions: (1) Did you get surprised to see that you were losing knowledge?;(2) What did you do afterwards once you learned that software knowledge was being lost?; and, (3) Would you be willing to use a tool that automatically captures the lost knowledge in other projects?

IV. RESULTS

In the following section, the collected and observed results for the research questions stated above are presented.

A. Can we extract relevant discussion from developers IM log using automated techniques? (RQ1)

We applied text summarization algorithm on IM log files collected from the projects. Based on the relevance of the sentences a ranking was calculated of 1,726 different sentences (845 sentences from Team A, 881 sentences from Team B). A total of 240 relevant sentences were identified and extracted, as shown in Table I. Once project managers from each project classified the relevance of the 20-top sentences in their respective project. Team A' project manager highlighted that all the 3-top sentences are related to relevant issues. Considering the 10-top sentences, 74% of them were considered relevant. Team B' project manager highlighted that 85% of the 3-top sentences and 75% of the 10-top were relevant, as shown in Table I. Table I also shows that the proposed approach achieved the best MRR value, this means that all the first ranked sentences were considered relevant to the project according to the managers' perspective.

Observation 1: We can extract relevant discussion from developers IM log using automated techniques with high precision.

B. Can we identify PLK from the relevant discussions by interviewing project managers? (RQ2)

To answer RQ2, we asked project managers to describe the PLK they identified after they read the 20-top sentences. Also, PMs manually classified their quotations to reveal the PLK related topics.

To illustrate the process of PLK determination, let's consider the following sentences that the project manager of TEAM A (PM-A) classified as relevant (for brevity, we did not include the complete sentences transcription):

- *"...we have already seen that it is not enough to let users have the option to adjust what we need (time, luminosity, etc.) ... we have to do it automatically..."*
- *" we discussed, and we noticed that the 3-sec side effect is speedy and could it not synchronize ..."*
- *"... I had an interesting idea. If we combine a bracelet with a mobile app? The cell phones would be available for photos and other functions..."*

After identifying the relevant sentences, we asked PM-A if the sentences reminded him of some PLK. PM-A described the PLK according to his point of view:

1) *"Less user intervention required. This could simplify user interaction with the app."*
2) *"Decision to change app' effects speed and the reason for blocking the app..."*
3) *"It emphasized the need to merge visual and sounds effects. The team did not implement this issue yet."*
4) *"Although during this period I did not identify lost knowledge, I remembered negotiations for the using the app in a concert."*
5) *"Lighting test to check colors displayed on the cellphone screen. We also needed to check the feasibility to distinguish each color."*

One can notice that the PM-A identified some important projects decisions in his quotations. These decisions can be used in order to support the software requirements, code design, updates, User Interface (UI) design and evolution.

Following, we describe some relevant sentences identified by PM-B:

- *"..features of the devices that can be used for android apk testing: (1) smart-phone; (2) Android versions from 5.0 release; (3) screen resolutions..."*
- *".. I will check if we already visualize the medicament description. If yes, I will start the delivery process today..."*
- *"But, can you create the testing cases? He needs testing the application. I thought to create rules for two users, for validation purposes. What do you think? This functionality is essential for project conclusion and the application deployment..."*

Similar to PM-A, PM-B also described the PLK identified:

1) *"Changes in design to improve the web service. Usability changes to improve the user experience..."*
2) *"Functional rules for particular clients were not specified. The team did not know how the app should treat these client's purchases. Developers had not implemented these rules."*
3) *"It was important to remember that the Ministry of Health defined the drugs description... The acceptance testing, identified the software nonconformity."*
4) *"Identified how user configuration profile were used. I did not remember that there was a case test for the functionality that deals with clients registration."*
5) *"Definition of two new requirements for the next software release. It also defined rules to deal with particular users purchase."*

The quotations of PM-B identified relevant PLK regarding software updates, testing, maintenance and code design. The identified PLK also allowed to understand some software requirements elicitation, identified when the PM-B says : "...

TABLE III
THE PROJECT'S LOST KNOWLEDGE TOPIC

PLK Topic	PLK Count PM-A	PLK Count PM-B
Requirement	1	7
Testing	2	3
Design	2	1
Product Vision	2	1
Management	1	1
Development	0	1
Delivery	0	1
Acceptance	0	1
Company Business	1	0

the Ministry of Health defined...". This issue identifies who defined the description messages displayed in the application. One can notice that the determined PLK can be used to promote the software knowledge transfer, that is important to disseminate software information through the team.

Observation 2: We can successfully identify PLK from the relevant discussions by interviewing project managers.

We also investigated the categories of most frequently PLK in IM discussions. As explained in Section III, we asked PMs to codify their PLK's quotations according to the categories identified in Table II. Table III summarizes the count of each topic found in the PLK identification. According to PM-A perspective, different PLK's categories were identified, such as design issues, testing, product vision, management, and requirements. The most common lost knowledge determined by PM-B perspective was related to requirements issues.

We also conducted a follow up interview of the PMs to understand what were their perceptions regarding the PLK. Below are some of the responses we received:

PM-A's response: *"The retrieved relevant sentences reminded me of important design decisions, testing issues, and business-related knowledge. We used the identified PLK to evolve the software... It was surprising to observe issues that I did not remember, even though I had selected only five periods to analyze."*

PM-B's response: *"I was surprised by the amount of lost knowledge I found during the PLK identification process because many decisions were made only in the chat tool. Although this specific project is over, we can use the acquired knowledge as a source of information to rescue decisions made and use the learning acquired in new projects."*

When asked about the possibility of adopting this technique in other projects both PMs showed interest:

PM-A's response: *"A tool that automatically extracts PKL beyond chat logs, such as emails, discussion forums, project management tools would be fantastic, and I would make it use as default to all the projects developed in my company."*

PM-B's response: *"It is crucial to take advantage of IM tools. IM logs contain different information and decisions registres. I would adopt a tool to automatically tells the team the PLK. If we had identified the PLK in this project, it probably could have led us to detect bugs, confirm requirements, and identify testing problems."*

V. DISCUSSION

This paper reveals a promising use of text summarization algorithms to select relevant sentences from developers' chat message in IM tools. While identifying PLK after reading the 20-top ranked sentences PMs highlighted different opinions and utilities for the identified PLK. On average, PMs took 22 minutes to evaluate each set of 20 relevant sentences and identify the PLK. An analysis of the quotations indicate that relevant sentences identified by the text summarizarion algorithm enables the PLK determination.

Our findings suggest that many decision and rationales can be identified during the process of PLK identification which also creates the opportunity to do knowledge transfer, knowledge transformation and the inception of new ideas. Knowledge transfer can help to disseminate the knowledge throughout the team and reduce the chance of introducing inconsistencies and bugs in the code. The knowledge identification can enable members, specifically new ones, to acquire the necessary project related knowledge quickly and help them in on-boarding. Knowledge transformation permits refreshing the software knowledge and deepen team's understanding of the project. This can help conceptualization of new ideas and develop new ways to help company businesses [24].

The loss of knowledge leads to a variety of problems, such as loss of design rationale, decrease system understanding, degraded knowledge sharing, and makes difficult the software evaluation, and the estimation of changes impact [9], [13]. We also noticed that though the most frequent PLK was related to requirements (Table III), each project has a different distribution of the PLK topics. This highlights the fact that the project lose knowledge on different stages of the software development life cycle and PLK is not necessarily tied to any specific step.

While interviewing the PMs, we noticed they had a good experience while identifying the PLK. They highlighted that PLK is very important to promote software quality, acquire knowledge, and to produce new product releases. The PMs affirmed that they would like to adopt tools that automatically informs them about lost knowledge to support their companies' software development process.

VI. THREATS TO VALIDITY

We have taken care to ensure that our results are unbiased, and have tried to eliminate the effects of random noise, but it's possible that our mitigation strategies may not have been entirely effective.

Given that we examined only two projects, we cannot guarantee whether our findings generalize to all projects. Our categorization of discussions required manual labeling since this information is not readily available. Our labeling process included inter-rater reliability to reduce the threat of individual bias. As we had multiple researchers and we also had a high inter-rater agreement, we assume this should minimize the aforementioned threat.

In addition, this study only traced one strand of knowledge exchange, via IM, and did not look further to see if any of this knowledge could be captured in other media, like e-mail, Slack, commented code, backlog document or daily stand-ups.

We presented sentences to PMs using a ranking algorithm which uses $TF \times IDF$ value. Since $TF \times IDF$ is sensitive to misspelling, the ranking is not perfect. Since our study was an investigation to understand if projects lose knowledge and if we can identify the lost knowledge using a semi-automated approach, our major findings are not impacted by the imperfect ranking. However, we intend to investigate the TextRank algorithm's efficiency [25] as future work.

VII. Conclusion and Future Work

In this paper, we investigated whether projects lose knowledge and how to retrieve it from IM logs. Our analysis indicates that projects lose knowledge as decisions, reasons, and rationales are discussed in IM tools. These decisions simply fall through the cracks. We also found that data-mining techniques can effectively extract relevant information from discussions which can be used by team members to identify PLK.The proposed approach can support knowledge identification and transfer on startups and other software companies that rely on IM tools or other informal communication channels. On average, PMs took 22 minutes to evaluate the sentences for each period and identify the PLK. Other team members can also use this approach during the entire software life cycle time to support development tasks.

The importance of this work is two fold. First, our results show that we can use a semi-automatic approach for identifying the PLK and achieve a precision of 75%. While this is an important step forward, it serves as a call to action for future research to investigate other techniques that can further improve the accuracy. Second, it provides empirical evidence that projects lose knowledge and the process of PLK determination creates opportunity for knowledge transfer, acquisition, update and conception of new ideas. The PLK can be used to support the software products evolution and also to support the strategic decision-making process. We only investigated IM log for this study, however our results indicate the need for extensive research to understand how knowledge is lost in other non-official and official channels such as IRC, mailing list, Slack, etc. Also, we interviewed only the PMs to identify the PLK, incorporating feedback from the whole team could lead to more effective PLK identification and is another interesting future research direction.

Acknowledgement

We would like to thank the financial support granted by CNPq through processes number 423149/2016-4, 311494/2017-0, and 204081/20181/PDE. CAPES through process number 175956/2013, and PROAP 001. FAPEAM through process number PPP 04/2017.

References

[1] R. Alkadhi, M. Nonnenmacher, E. Guzman, and B. Bruegge, "How do developers discuss rationale?" in *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2018, pp. 357–369.

[2] R. Alkadhi, T. Lata, E. Guzmany, and B. Bruegge, *Rationale in development chat messages: an exploratory study*, IEEE, 2017.

[3] G. Viviani, C. Janik-Jones, M. Famelis, X. Xia, and G. C. Murphy, "What design topics do developers discuss?" in *Proceedings of the 26th Conference on Program Comprehension*. ACM, 2018, pp. 328–331.

[4] R. Francois, M. Nada, and A. Hassan, *How to extract knowledge from professional e-mails*, IEEE, 2015.

[5] A. E. Hassan and T. Xie, *Mining software engineering data*, IEEE, 2010.

[6] H. Mushtaq, B. H. Malik, S. A. Shah, U. Bin Siddique, M. Shahzad, and I. Siddique, "Implicit and explicit knowledge mining of crowdsourced communities: Architectural and technology verdicts," *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, vol. 9, no. 1, pp. 105–111, 2018.

[7] J. Pantiuchina, M. Mondini, D. Khanna, X. Wang, and P. Abrahamsson, "Are software startups applying agile practices? the state of the practice from a large survey," in *International Conference on Agile Software Development*. Springer, Cham, 2017, pp. 167–183.

[8] C. Giardino, M. Unterkalmsteiner, N. Paternoster, T. Gorschek, and P. Abrahamsson, "What do we know about software development in startups?" *IEEE Software*, vol. 31, no. 5, pp. 28–32, 2014.

[9] A. M. Soria and A. van der Hoek, "Collecting design knowledge through voice notes," in *Proceedings of the 12th International Workshop on Cooperative and Human Aspects of Software Engineering*. IEEE Press, 2019, pp. 33–36.

[10] J. E. Burge and D. C. Brown, "Seurat: Integrated rationale management," in *Proceedings of the 30th International Conference on Software Engineering*, ser. ICSE '08. New York, NY, USA: ACM, 2008, pp. 835–838. [Online]. Available: http://doi.acm.org/10.1145/1368088.1368215

[11] T.-H. Chen, S. W. Thomas, and A. E. Hassan, "A survey on the use of topic models when mining software repositories," *Empirical Software Engineering*, vol. 21, no. 5, pp. 1843–1919, 2016.

[12] E. Isaacs, A. Walendowski, S. Whittaker, D. J. Schiano, and C. Kamm, "The character, functions, and styles of instant messaging in the workplace," in *Proceedings of the 2002 ACM conference on Computer supported cooperative work*. ACM, 2002, pp. 11–20.

[13] C. Manteuffel, D. Tofan, H. Koziolek, T. Goldschmidt, and P. Avgeriou, "Industrial implementation of a documentation framework for architectural decisions," in *2014 IEEE/IFIP Conference on Software Architecture*. IEEE, 2014, pp. 225–234.

[14] A. Kleebaum, J. O. Johanssen, B. Paech, R. Alkadhi, and B. Bruegge, "Decision knowledge triggers in continuous software engineering," in *Proceedings of the 4th International Workshop on Rapid Continuous Software Engineering*, ser. RCoSE '18. New York, NY, USA: ACM, 2018, pp. 23–26. [Online]. Available: http://doi.acm.org/10.1145/3194760.3194765

[15] M. Codoban, S. S. Ragavan, D. Dig, and B. Bailey, "Software history under the lens: A study on why and how developers examine it," in *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2015, pp. 1–10.

[16] G. Bavota, "Mining unstructured data in software repositories: Current and future trends," in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, vol. 5. IEEE, 2016, pp. 1–12.

[17] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artificial Intelligence Review*, vol. 47, no. 1, pp. 1–66, 2017.

[18] T.-Y. Liu, *Learning to rank for information retrieval*. Springer Science & Business Media, 2011.

[19] B. McFee and G. R. Lanckriet, "Metric learning to rank," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 775–782.

[20] S. Haiduc, J. Aponte, and A. Marcus, "Supporting program comprehension with source code summarization," in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2*. ACM, 2010, pp. 223–226.

[21] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for idf," *Journal of documentation*, vol. 60, no. 5, pp. 503–520, 2004.

[22] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[23] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *biometrics*, pp. 159–174, 1977.

[24] P. Carreteiro, J. B. de Vasconcelos, A. Barão, and Á. Rocha, "A knowledge management approach for software engineering projects development," in *New advances in information systems and technologies*. Springer, 2016, pp. 59–68.

[25] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004.