Exploring Automation of the Development of Requirements from User Feedback

by

Ze Shi Li
Master of Science, University of Victoria, 2020
Bachelor of Computer Science, University of Victoria, 2018

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in the Department of Computer Science

We acknowledge and respect the Lək^{w}əŋən (Songhees and X^{w}sepsəm/Esquimalt) Peoples on whose territory the university stands, and the Lək^{w}əŋən and W̱SÁNEĆ Peoples whose historical relationships with the land continue to this day.

Exploring Automation of the Development of Requirements from User Feedback

by

Ze Shi Li
Master of Science, University of Victoria, 2020
Bachelor of Computer Science, University of Victoria, 2018

Supervisory Committee

Dr. Daniela Damian, Supervisor
(Department of Computer Science)

Dr. Neil Ernst, Supervisor
(Department of Computer Science)

Dr. David Lo, Outside Member
(School of Computing and Information Systems, Singapore Management University)

ABSTRACT

In modern software development, products and services collect heterogeneous feedback from end users on various platforms such as app stores, social media, forums, and videos. This user feedback is a source for identifying emerging needs, bugs, and potential features. As organizations shift toward rapid release cycles and continuous delivery models, the volume and breadth of user feedback have increased significantly. Traditional requirements elicitation techniques, such as interviews and surveys, remain time-consuming, stakeholder-dependent, and difficult to scale. Moreover, newer mediums like TikTok, YouTube, and Reddit have introduced informal, crowd-driven forms of feedback that are often unstructured and scattered across platforms. This has created a pressing need for scalability and methodological support to analyze and synthesize large-scale user feedback. This dissertation addresses this challenge by exploring scalable, AI-driven approaches to feedback analysis in requirements engineering.

For my first research goal, I aimed to investigate how software organizations manage user feedback. I conducted a grounded theory interview study with 40 practitioners from 32 companies to explore how organizations manage user feedback. My analysis identified many feedback channels and activities. Synthesizing these, I propose a life cycle of managing user feedback along with best practices for managing large-scale crowd feedback.

For my second research goal, I explored approaches to automate the development of new requirements from user feedback. For textual feedback, such as Reddit and app store reviews, I applied large language models (LLMs) to identify requirements relevant feedback and important themes from the data. This LLM-based approach was more efficient and less laborious than manual analysis for the same purpose. Additionally, I examined automating the analysis of video-based feedback. I extracted transcripts and on-screen text and employed deep learning classifiers to detect requirements-relevant content. My work shows that AI models can identify multimodal user feedback for requirements insights at scale.

Given the rising ubiquity of generative AI tools, my third research goal was to explore how we can use such generative AI tools to help automate the development of requirements from user feedback. I began by developing a theory that outlines the factors (i.e., motives and challenges) influencing AI adoption in software teams at both the individual and organizational levels through 26 interviews. Understanding these

factors details how generative AI tools could be introduced and supported in practice. Finally, I conducted a think-aloud study with requirements practitioners and product managers to understand how they use generative AI tools during the development of new user requirements from feedback. Participants were observed forming prompts and integrating AI-generated suggestions while analyzing user feedback and formulating requirements. This study highlighted the observed practitioners' practices.

To summarize, this dissertation highlights the findings across all studies, which culminate in a process for AI-assisted development of new requirements from user feedback. This conceptual model synthesizes the lifecycle of user feedback management, automation techniques for multi-modal analysis, and the socio-technical factors shaping tool adoption. This model offers both theoretical and practical contributions by providing scalable, human-centered strategies for transforming crowd-driven user feedback into requirements.

Contents

Supervisory Committee **ii**

Abstract **iii**

Table of Contents **v**

List of Tables **ix**

1 Introduction **1**
- 1.1 Research Goals and Research Questions 5
- 1.2 Contributions 8
  - 1.2.1 Publications 8
- 1.3 Structure 9

2 Background and Related Work **11**
- 2.1 Organizational Practice for Feedback Collection and Analysis 11
- 2.2 Automated User Feedback Analysis 13
- 2.3 AI for Software Engineering and Requirements Engineering 16

3 Industry Perspective on Analyzing User Feedback **19**
- 3.1 Research Methodology 20
  - 3.1.1 Participant Recruitment and Selection 20
  - 3.1.2 Interview Design 22
  - 3.1.3 Data Analysis 22
  - 3.1.4 Member Checking 23
- 3.2 Findings: Managing User Feedback 23
  - 3.2.1 Why User Feedback Matters 24
  - 3.2.2 The Life Cycle of Managing User Feedback 27
  - 3.2.3 Best Practices in Managing User Feedback 33

3.3 Threats to Validity . . . 38
3.4 Discussion . . . 38
3.4.1 Implications for Practitioners . . . 40
3.4.2 Implications for Researchers . . . 40
3.5 Conclusion . . . 41

4 Experiments Towards Automating User Feedback from Text Based Sources 44
4.1 Exploration through Lens of Privacy Requirements . . . 44
4.1.1 Methodology . . . 47
4.1.2 Findings . . . 50
4.1.3 Narratives as Explanations of Variations in Privacy Discussions 54
4.1.4 Threats to Validity . . . 59
4.2 Exploration through Lens of Inclusiveness Requirements . . . 60
4.2.1 Motivation . . . 63
4.2.2 Methodology . . . 65
4.2.3 **RQ2.4**: A Taxonomy of Inclusiveness . . . 68
4.2.4 **RQ2.5**: Inclusiveness across different Sources of User Feedback 69
4.2.5 **RQ2.6**: Automated Identification of Inclusiveness user feedback 70
4.2.6 Threats to Validity . . . 71
4.3 Conclusion . . . 72

5 Experiments Towards Automating User Feedback from Video Based Sources 74
5.1 Methodology . . . 75
5.1.1 Data collection . . . 76
5.1.2 Preprocessing . . . 76
5.1.3 Analysis of Videos . . . 77
5.1.4 Manual Labeling . . . 80
5.1.5 Automated Analysis of User Feedback . . . 81
5.2 Findings . . . 84
5.2.1 **RQ3.1**: How can video-based social media be used to identify requirements relevant user feedback? . . . 84
5.2.2 **RQ3.2**: What are the main user feedback themes that we can identify? . . . 86

5.3 Discussion . . . 87
5.3.1 Videos: A Source of Requirements Relevant User Feedback . . 87
5.3.2 Implications for Practitioners . . . 89
5.3.3 Implications for Research . . . 90
5.4 Threats to Validity . . . 91
5.4.1 Construct Validity . . . 91
5.4.2 External Validity . . . 92
5.4.3 Internal Validity . . . 92
5.5 Conclusion . . . 92

6 Understanding Adoption of LLM Based Tools 94
6.1 Methodology . . . 96
6.1.1 STGT: Basic Stage of Data Collection and Analysis . . . 96
6.1.2 STGT: Advanced Stage of Theory Development . . . 100
6.2 A Theory of AI Tool Use and Adoption for Software Development . . 101
6.2.1 Individual Motives . . . 102
6.2.2 Individual Challenges . . . 105
6.2.3 Organizational Motives . . . 109
6.2.4 Organizational Challenges . . . 111
6.3 Push-Pull Relationships between the Motives and Challenges . . . 114
6.3.1 Culture of sharing AI knowledge vs. Negative judgment and lack of support: . . . 114
6.3.2 Providing and promoting vs. Not paying the cost: . . . 115
6.3.3 Providing guidance and training vs. Lack of prompting skills and usage guidelines: . . . 116
6.4 Discussion . . . 117
6.4.1 Increasing adoption of AI tool(s) in organizations . . . 118
6.4.2 Directions for Future Research . . . 120
6.4.3 Threats to Validity . . . 120
6.5 Conclusion . . . 122

7 Tool Based Automation of User Feedback 123
7.1 Methodology . . . 124
7.1.1 Study Design . . . 124
7.1.2 Participant Recruitment . . . 127

7.1.3 Data Collection 129
7.1.4 Data Analysis 130
7.2 Findings 130
7.2.1 How Requirements Practitioners Use ChatGPT for the Development of New Requirements from User Feedback 132
7.2.2 Challenges of Using AI Tools in Requirements Development . 137
7.3 Discussion 140
7.3.1 Recommendations for Practitioners: Strategies for Leveraging AI Tools 141
7.3.2 Recommendations for Researchers 143
7.3.3 Threats to Validity 144
7.4 Conclusions 146

8 Conclusion and Future Work **148**
8.1 Contributions 149
8.2 How can a Requirements Practitioner Use Machine Learning and AI Tools to develop new requirements from CrowdRE User Feedback . . 150
8.3 Final Conclusion 153

A Tool Based Automation of User Feedback Data **155**

Bibliography **167**

List of Tables

Table 3.1 Example Coding of Raw Quotes 20
Table 3.2 Participants and Their Roles and Organizations 42
Table 3.3 User Feedback Sources 43
Table 3.4 Best Practices for Managing User Feedback 43

Table 4.1 Categories used to group similar subreddits 50
Table 4.2 Result from classifying posts from all subreddits 51
Table 4.3 Major privacy concerns and associated subreddit categories (from Table 4.1) 52
Table 4.4 Total number of inclusiveness-related user feedback in the 5 types of Apps from Reddit, Google Play Store, and X 69
Table 4.5 Results of Classifying between Inclusiveness and Non-Inclusiveness using GPT4-o Mini 71

Table 5.1 Products used for the analysis 77
Table 5.2 Results of Different Deep Learning Models on Classifying between Relevant vs Irrelevant. AUC is area under curve. 83
Table 5.3 Result from labeling and Classifying the Dataset 86
Table 5.4 Requirement Relevant Themes 87

Table 6.1 Interviewee Demographics 100
Table 6.2 Motives that increase AI tool use and adoption 103
Table 6.3 Challenges that limited AI tool use and adoption 104

Table 7.1 Study Participants. Product experience refers to the number of years that a participant has conducted requirements or product related work, such as a product manager, requirements engineer, or business analyst. 124
Table 7.2 Base follow-up interview questions asked after each session . . . 128
Table 7.3 Example Coding of Raw Quotes 131

Table 7.4 Practices for using AI Tools by Requirements Practitioners . . . 132
Table 7.5 Challenges experienced when using AI Tools by Product Practitioners 137

Table A.1 Reddit User Feedback about Zoom 155

Chapter 1

Introduction

Since the 1980s, software engineering has been known to be difficult and challenging [1]. Part of what makes software engineering so challenging is understanding what to build for all involved parties. User feedback serves as a crucial aspect for the continuous improvement and development of software products. With the advent of agile methodologies and software organizations moving towards continuous delivery models [2], the feedback loop between development and release has become more immediate. However, managing and using the feedback from end-users remains a significant challenge for many organizations.

Requirements engineering (RE) is a fundamental aspect of software engineering that involves the identification, documentation, and maintenance of software requirements. The goal of RE is to ensure that the software product meets the needs and expectations of its end-users and other involved parties. Common RE practices include [3] elicitation, interpretation, negotiation, documentation, validation/verification/ and change management. In my work, I focus on the RE practices of elicitation and analysis, which relate to identifying and analyzing the software requirements that should be developed.

With the rapid rise of new mediums and platforms that support users to express their opinions and discuss software products [4], it is increasingly important to consider these user concerns to fulfill user needs. These user needs are referred to as "user feedback", which are derived from "current and potential stakeholders wherever customers share or exchange their experience with a particular product and the extent to which it meets their needs, because requirements can be derived from such statements" [5]. User feedback from multiple sources can contain requirements, relevant content, such as features, bugs, and other aspects about software products.

Traditional requirements elicitation approaches have long relied on labour-intensive and end-user-centric approaches such as conducting interviews, surveys, and document reviews with key stakeholders to elicit potential requirements. These requirements elicitation sessions may take place over the course of hours and days and strongly relied on the willingness of stakeholders to give access, and the interviewing ability of requirements engineers to elicit the right information. While thorough, these approaches are time-consuming and prone to human error. Another drawback to this approach is that once information was collected, eliciting additional or updating requirements from stakeholders was often at the mercy of gaining additional access or receiving further communication lines to stakeholders.

Moreover, traditional requirements elicitation approaches may struggle to capture rapidly evolving user needs, particularly in rapid development cycles practices in many organizations [6]. However, new sources of user feedback have afforded organizations the ability to continuously collect product feedback.

User feedback is a vital source of information that contains the pertinent information from end-users which provides insights into user needs, preferences, and pain points. Requirements engineers, product managers, and other product-focused employees can make informed decisions about feature development and bug fixes. Proliferation of digital platforms such as app stores, social media, and product forums, the volume and types of user feedback have exponentially increased in comparison to elicitation surveys or interviews. Furthermore, traditional requirements engineering often struggles with the scale and diversity of user needs in modern applications, especially those with large user bases.

"CrowdRE" has emerged to tackle these challenges by focusing on user feedback as a primary source of requirements and recognizes that the collective insights from a large number of users can provide a deeper understanding of what is needed in a software product [7]. The field of "CrowdRE" is a growing field of requirements research that encourages the collection and analysis of online user feedback from the crowd [5]). Specifically, "CrowdRE" [5] refers to

> *"a semi-automated requirement engineering (RE) approach for obtaining and analyzing any kind of 'user feedback' from a 'crowd', to derive validated user requirements"*

Improving the body of knowledge in CrowdRE can pay dividends in improved requirements elicitation for software organizations [5, 8] as organizations gain an in-

creased understanding of user concerns from analyzing vast amounts of user feedback [9]. CrowdRE research has explored additional feedback sources such as app reviews [10, 11, 12], product forums [13], Twitter [14], Reddit [15], and Facebook [16]. Studies in this area have explored acquiring user insights from app reviews [10, 11, 12] and vision videos [17].

Despite the advances from recent studies to leverage the "crowd" via increased involvement of crowd-based discussions, the software industry still lacks empirical structured guidance for synthesizing user feedback from multiple CrowdRE channels such as app reviews or videos. There are a few studies that explored feedback collection methods in software engineering [18, 19], but in limited capacities. Overall, these previous studies provide some early insights regarding feedback collection, but lack details about how organizations can handle all the different user feedback sources. Therefore, we need to establish a more detailed and structured set of practices for managing user feedback based on empirical evidence. In particular, new immersive approaches for users to express their concerns include media such as videos, which are quite popular for communication [20]. For example, TikTok and YouTube are among the world's most popular video-based sharing platforms [20, 21].

The proliferation of tools and processes that explore these sources of user feedback has made it more amenable for organizations to elicit requirements from the crowd. Despite its importance, managing user feedback effectively presents several challenges.

The volume of feedback can be significant, making it difficult for developers to identify and prioritize the most critical issues. Unlike traditional approaches of requirements elicitation where a small number of stakeholders are involved, when an organization opens the elicitation to the entirety of the "crowd", the labor costs of analyzing the amount of feedback are significant, if not impossible for smaller resource-constrained organizations.

User feedback is often unstructured [22], coming in various forms such as text, images, and videos. However, collecting and tracking user feedback is also a necessary obligation, given that feedback is shown to have the ability to quickly become popular and "trending". These "trends" may snowball [23] into issues that ultimately lead to significant financial losses.

Recent advancements in artificial intelligence (AI), particularly in large language models (LLMs) offer promising solutions to these challenges. AI models, often those involving large models such as GPT-4[1] and Llama [24] have shown a broad range of

[1]https://openai.com

abilities from question answering, summarization, and text generation. These models can analyze vast amounts of unstructured data, identify patterns and trends, and generate meaningful insights.

Software products are used by a wide range of individuals with different backgrounds, preferences, and expectations. Capturing and addressing these requirements is crucial for developing software that caters to diverse end users. Among the many types of requirements, non-functional requirements (i.e., architecturally significant) stand out as particularly difficult to handle. Non-functional requirements, such as privacy or security, not handled properly, can lead not only to user dissatisfaction but also to serious legal and regulatory consequences. Prior work has emphasized the organizational challenges of designing for privacy, especially in smaller companies that lack formal processes or resources for privacy compliance [25]. Similarly, while not traditionally treated as a non-functional requirement, it is increasingly framed as a quality attribute that intersects with usability, accessibility, and fairness [26]. These issues disproportionately affect marginalized users. However, AI models may provide more systematic and scalable methods for requirements elicitation and analysis. For example, classification can help identify requirements and relevant feedback, while computer vision can identify useful information inside a user-created video.

For individual developers, AI tools can assist with code generation, testing, and debugging, thereby increasing productivity and job satisfaction [27]. Studies have shown that AI tools help reduce repetitive tasks [28, 29], and individual developers reported experiencing a reduction in mental energy. The adoption and integration of AI tools in software organizations is not without challenges.

However, preliminary studies have also revealed that practitioners often lack trust in AI generated outputs [30]. Therefore, before exploring the use of AI tools to support CrowdRE and automate requirements engineering tasks, it was necessary to investigate the overall AI tool adoption in software engineering.

Understanding how organizations effectively adopt AI tools is beneficial for my research toward understanding how AI tools can support the requirements elicitation and analysis process. More importantly, it is paramount that we also build an understanding of how product managers, product owners, and other requirements practitioners can use these AI tools for the development of new requirements from user feedback.

Specifically, how are they using such tools in the industry, what sort of challenges

may manifest, and how can we, as researchers, focus on making their lives easier? By answering the aforementioned questions, this dissertation aims to fill a critical gap in the current literature and practice, providing empirical insights that can support organizations and practitioners in leveraging the newer sources of feedback and best approaches to applying AI tools.

1.1 Research Goals and Research Questions

Three main research goals motivated my dissertation. 1) My first goal was to develop an empirical understanding of how software organizations manage user feedback. Specifically, I aimed to derive more insights regarding the sources of user feedback that organizations consider or are important for their software. Additionally, I further intended to investigate organizational practices regarding how feedback is administered once collected. This led to my first research question.

RQ1 How do software organizations manage user feedback to improve existing products?

For this research question, I conducted a grounded theory study to understand how organizations manage user feedback in the industry. I interviewed 40 people from 32 organizations who represented participants from a wide and diverse set of roles and industries. Understanding where and how users provide feedback is the first step towards effective management. This involves mapping out various feedback channels and identifying the characteristics of feedback from each source. In addition, I need to identify all the activities involved in managing user feedback from collecting raw data to developing the product. This RQ was described in Chapter 3.

2) Identifying that the volume of user feedback and manual processing was a crucial challenge for practitioners, I discerned my second research goal. I aimed to investigate techniques to automate large-scale user feedback analysis using deep learning models. This goal led to two research questions, which were detailed in Chapters 4 and 5.

RQ2 How can we automate the development of new requirements from textual based user feedback?

To address the research question, I first conducted a study to analyze user feedback from Reddit. The study resulted in understanding how narratives play a vital role in shaping users' privacy concerns and influencing the types of requirements they express. By collecting over 4.5 million Reddit posts from 62 software product-related and 4 privacy-related subreddits, and applying unsupervised clustering, I identified 9 privacy-related requirements. Building on this, I co-authored a paper with my colleague Nowshin Nawar Arony, which extended my approach to inclusiveness-related requirements by analyzing over 10 million posts across Reddit, Google Play, and X (previously known as Twitter) [31]. Nowshin focused on developing a taxonomy of inclusiveness-related feedback. I focused on fine-tuning LLMs (e.g., GPT-4o mini) to automatically identify inclusiveness-related feedback. The rest of the paper is presented with her permission. These studies show how automated techniques using LLMs can elicit diverse requirements from textual feedback at scale.

RQ3 How can we automate the development of new requirements from video-based user feedback?

For this research question, I conducted a data-driven exploratory study to analyze video-based user feedback from TikTok and YouTube. By extracting and processing audio, visual text, and metadata from over 6,000 videos covering 20 products across four industries. I co-authored a paper with my colleagues Manish Sihag, and Dr. Amanda Dash [32]. Manish manually analyzed and labeled the video-based feedback to identify the emerging user feedback themes. Dr. Dash contributed by developing the sampling algorithm used in the computer vision pipeline to extract representative frames from video content. I applied deep learning models such as GPT-2 and RoBERTa to classify video content as requirements relevant or irrelevant with high accuracy (up to 97%). The rest of the paper is presented with their permission. I applied clustering (BERTopic) to identify recurring feedback themes, including feature ratings, bug reports, performance issues, and design suggestions that organizations can further refine into requirements.

3) As I approached the completion of my second research goal, the technology industry underwent significant transformations. In particular, large language model-infused generative AI tools such as ChatGPT were launched to the public [33]. These tools quickly gained immense popularity and challenged the status quo of using earlier

deep learning models for classification, topic modelling, and basic data analysis tasks. Since my earlier goal was to explore how to investigate techniques to automate user feedback analysis to alleviate practitioner manual work, I expanded on this goal to account for the evolving changes in the technology landscape. My third research goal aimed to explore how we can leverage generative AI tools for the development of new requirements from user feedback. This research goal is addressed in Chapters 6 and 7.

RQ4 What impacts generative AI adoption and use in software engineering?

To address this research question, I first developed empirical insights on adopting generative AI tools in software engineering, as understanding this adoption provides a foundation to later study its impact on the development of new requirements from user feedback. I conducted a socio-technical grounded theory study [34] with 26 interviewee participants. My work involved finding more empirical evidence for what challenges exist for integrating AI tools into software organizations. It also involved developing an understanding of the impact of AI tools on productivity and organizational culture.

RQ5 How are requirements practitioners leveraging generative AI tools to conduct the development of new requirements from user feedback?

For this research question, I conducted a user study with 10 requirements practitioners. As part of trying to understand how practitioners are using generative AI tools, the user study involved an exercise developing new requirements from 20 user feedback posts from Reddit about *Zoom*. My work involved identifying practices that practitioners use when applying AI tools for the development of new requirements from user feedback. Moreover, I uncovered several challenges that persist when practitioners use AI tools for this purpose. Finally, I provided recommendations for practitioners using AI tools.

For this work, I perform several studies that use both qualitative and quantitative approaches on both industrial organizations and user feedback in the wild. My work involves grounded theory studies to build theory about organizational practices for handling user feedback, empirical studies on user feedback from different sources, and a think-aloud study towards supporting the identification of trends from user feedback.

1.2 Contributions

The objective of my research is to provide contributions that are relevant and useful for both practitioners and researchers. In this work, I describe my findings on how organizations manage user feedback in practice and how practitioners can apply AI tools to help automate their development of new requirements from user feedback. My work has shown that online feedback sources can provide an avenue to identify requirements.

The first contribution of my work is a series of practices and insights into how software organizations effectively collect and manage user feedback, particularly from online social media sources. This contribution characterizes the current practices used by software organizations and provides insights for other practitioners to adopt in their work. My second contribution consists of empirical evidence of eliciting requirements and relevant information from a variety of types of user feedback, including textual and video-based sources.

The third contribution of this dissertation is empirical evidence for motivations and challenges inhibiting practitioners from adopting AI tools in software development. Since the goal of my work is to assist organizations with their requirements elicitation and analysis through automation, my fourth and final contribution provides empirical insights for analyzing user feedback with AI assistance, identifying and modifying groupings of user feedback, and deriving actionable requirements for software organizations. By uncovering these empirical insights, I bring forth practices and challenges from leveraging AI tools for the development of new requirements from user feedback. My work details gaps that future research may explore regarding generative AI tools in requirements engineering.

1.2.1 Publications

Below I present the list of publications that have resulted from this research.

[1] Z. S. Li, N. N. Arony, K. Devathasan, M. Sihag, N. Ernst, and D. Damian, "Unveiling the Life Cycle of User Feedback: Best Practices from Software Practitioners," In Proceedings of the 46th IEEE/ACM International Conference on Software Engineering (pp. 1-13).

[2] Z. S. Li, M. Sihag, N. N. Arony, J. B. Junior, T. Phan, N. Ernst, and D. Damian, "Narratives: the Unforeseen Influencer of Privacy Concerns," in 2022 IEEE 30th

International Requirements Engineering Conference (RE), 2022, pp. 127–139.

[3] M. Sihag, Z. S. Li, A. Dash, N. N. Arony, K. Devathasan, N. Ernst, A. Albu, and D. Damian, "A data-driven approach for finding requirements relevant feedback from TikTok and Youtube," 2023 IEEE 31st International Requirements Engineering Conference (RE).

[4] N. N. Arony, Z. S. Li, D. Damian, and B. Xu, "Unveiling inclusiveness-related user feedback in mobile applications," arXiv preprint arXiv:2311.00984, 2024. [Online]. Available: https://arxiv.org/abs/2311.00984.

[5] Z. S. Li, N. N. Arony, A. M. Awon, D. Damian, and B. Xu, "AI Tool Use and Adoption in Software Development by Individuals and Organizations: A Grounded Theory Study" in Review.

1.3 Structure

The rest of this manuscript is structured as follows:

Chapter 2 discusses the background and related studies for my research. It provides an overview of existing research on user feedback management in software development. It identifies gaps in the current literature and sets the stage for the subsequent chapters.

Chapter 3 describes my empirical research with industrial participants about how they manage user feedback to help improve existing products research. This study provided empirical grounding for my subsequent studies exploring how we can automate analyzing user feedback, which I describe in Chapters 4 and 5. This chapter presents the findings from the qualitative study with a large number of participants and the activities they conduct to manage user feedback.

Chapter 4 outlines the work I conducted to automate the analysis of textual based user feedback.

Chapter 5 outlines the work I conducted to automate the analysis of video based user feedback.

Chapter 6 discusses my study on understanding what impacts AI adoption and use in software engineering. This chapter was the first part of answering my second research goal, which is exploring how we can leverage AI tools. Subsequently, Chapter 7 explores using generative AI tools to assist in the development of new requirements from user feedback.

Chapter 7 outlines my work exploring how requirements practitioners leverage AI tools in their work. This chapter presents the findings from the user study I conducted with 10 participants.

Chapter 8 discusses the implications of my work for requirements engineering practice. I further provide a process about *how requirements practitioners can use machine learning and AI tools to conduct the development of new requirements from user feedback*. This process details the overarching narrative across all of my studies in this dissertation. Finally, I suggest directions for future research.

Chapter 2

Background and Related Work

While we have a reasonably good understanding of traditional requirements elicitation [35, 36], we have a limited understanding of leveraging the growing body of online sources of user feedback. Previous studies have focused either on organizational practices from a high level collection perspective or on the automation of various aspects of the requirements elicitation and analysis process. In the following subsections, I discuss the related work on collecting and analyzing user feedback, analysis from various sources of user feedback, automating the processing of feedback, and the use of AI tools for software engineering tasks.

2.1 Organizational Practice for Feedback Collection and Analysis

Software organizations have long practiced requirements elicitation techniques such as interviews, questionnaires, surveys, etc., to collect feedback from end users [37]. While traditional methods are needed and effective to a certain extent [19], they fail to capture the full range of user needs and perspectives [38] as they often involve a one time or infrequent gathering of requirements. These traditional elicitation techniques are insufficient to identify and develop requirements from online communities because they assume co-located, stable stakeholders, do not scale to large user groups, and are not designed to handle the unstructured, noisy, and asynchronous nature of crowdsourced feedback [5]. In Figure 2.1, I show a graphic about how requirements are conducted following the traditional approach for elicitation and analysis.

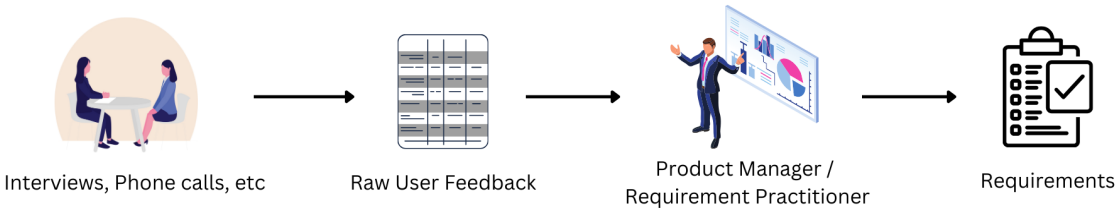


Figure 2.1: Traditional Elicitation and Analysis Approach

As described earlier, CrowdRE is *"a semi-automated requirement engineering (RE) approach for obtaining and analyzing any kind of 'user feedback' from a 'crowd,' with the goal of deriving validated user requirements."* [5] After the collection of data, one may develop requirements through manual [39] or automated approaches [40]. CrowdRE techniques have gained popularity in recent times as they help to gather a large set of user feedback from a variety of end users [8]. Utilizing the wisdom of a large distributed crowd, organizations can identify important requirements [41]. Groen et al. argue that CrowdRE can address the limitations of traditional RE methods, such as the limited scope and representation of user feedback [8]. Listening to the voices of users helps organizations reap the benefits of feedback [19].

Other industries extensively study the process of understanding user feedback from both traditional and online sources to improve their products and services on a regular basis [42, 43, 44, 45, 46]. For example, the pharmaceutical industry uses negative complaints from online sources to identify areas for improvement [46]. Similarly, studies in the retail pharmacy industry have developed frameworks to analyze unstructured social media data into supportive information to improve operations and service management [46]. They utilize statistical and sentiment analysis assisted by business analytical techniques to develop actionable information from most-discussed topics and negative comments. These findings support critical business operations such as customer services, marketing, and operations management [46]. The hotel and entertainment industry frequently relies on online reviews for insights into guests' general attitudes and feelings, employing this information to improve service quality [43]. Likewise, social media analytics are frequently used as a key source of feedback for retail companies [42]. Organizations ultimately benefit from a deeper understanding of user preferences and behavior [47] when they analyze the large breadth of user feedback. Therefore, for a software organization, understanding the complex and intricate processes of analyzing user feedback from the newer online sources is worth an organization's time and resources. as it leads to improved software quality,

more efficient development processes, and enhanced user satisfaction [48].

In the software industry, different sources of user feedback, including app reviews [49], forums [50], and vision videos [17], have been explored. However, understanding users, given the constantly evolving nature of the various sources of feedback, is more complex than simply "analyzing user feedback"; it involves the gathering and analysis of feedback from potentially diverse users with different needs, preferences, and expectations [51].

Social media platforms are known as catalysts for new innovation in organizations [52]. However, managing all the feedback and emergent trends that the feedback provides is a very complex task. Considering more online channels for feedback helps companies gain a more comprehensive picture of what their end-users need [53].

Previous studies investigating the industry practices employed by organizations have primarily reported on the feedback collection and analysis process [18, 19]. Organizations rely heavily on manual methods for collecting user feedback, which is often obtained through explicit channels, such as app reviews. Implicit feedback [54], which is unintentional feedback such as user software usage, is often overlooked in the feedback collection process. Johanssen *et al.* [18] interviewed 24 practitioners from 17 organizations and recommend improving user feedback collection through continuous integration, as it has the potential to diversify the requirements organizations work with.

In addition, Oordt and Guzman [19], through interviews with 18 software practitioners and surveys of 101 practitioners, found that companies are considering newer evolving feedback channels like social media to improve their feedback collection methods. Despite the efforts of various studies to learn about the utilization of user feedback in the industry [19, 18], they lack in providing a comprehensive and structured set of practices on how user feedback from various sources can be effectively managed and acted on.

2.2 Automated User Feedback Analysis

The services offered by CrowdRE aim to provide motivational tools (e.g., gamification techniques, forums, visuals) that can inspire stakeholders to actively participate in a crowd [9]. In Figure 2.2, I show a graphic about how requirements elicitation and analysis are conducted for CrowdRE.

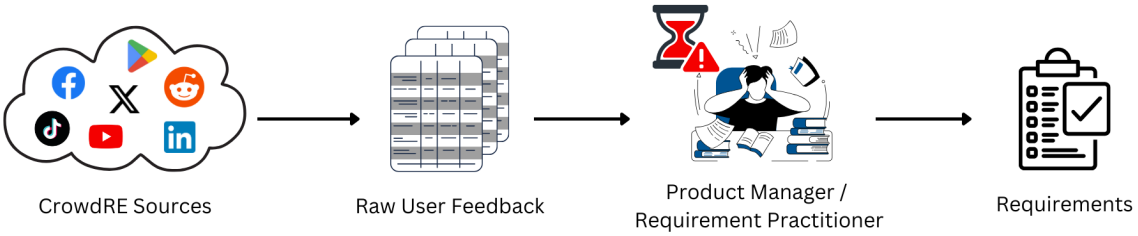


Figure 2.2: RE Elicitation and Analysis with CrowdRE

Online user feedback represents a rich, crowd-generated source of requirements-relevant information. Understanding how this unstructured feedback can be analyzed incorporated into the requirements engineering process is key to advancing current requirements practices. In this subsection, I review prior work that has examined applications, value, and challenges of using online feedback in requirements elicitation and analysis.

Several studies have explored how user-generated text from platforms like app stores, forums, and social media can inform software requirements, revealing patterns, feature requests, and bug reports. Pagano and Maleej [11] conducted a seminal study in textual user feedback analysis by identifying the patterns, topics, and quality of user feedback in over one million Apple App Store reviews and studied its impact on the software requirements. Other prior work has shown the importance of user feedback from software forums [55] and social media platforms [56, 57, 58], and even video platforms like YouTube and TikTok [59, 60, 61].

Tizard et al. [55] analyzed user reviews and feedback from the two product forums (VLC and Firefox) and concluded that product forums are a valuable source of consumer feedback that is essential for the evolution of the product. For social media, Kanchev et al. [56] performed a preliminary analysis of user discussions on Google Maps from Reddit discussions and presented examples of requirement-related artifacts. A recent study by Iqbal et al. [58] analyzed characteristics of Reddit posts about software applications and found out that more than half of the posts contain useful information such as bug reports and feature requests.

To scale user feedback analysis, researchers have proposed automation techniques that extract and organize user needs from large volumes of textual feedback. Automating feature extraction from app stores to find crowd-based requirements engineering was looked at in [10, 62]. For example, Guzman and Maalej [10] proposed to identify fine-grained app features by using collocations and sentiment analysis about the identified features and grouping them using topic modeling into a more mean-

ingful high-level feature. Guzman et al. [63] mined Twitter and found that 42% of tweets are software relevant and can be used to group, classify, and rank tweets about software.

Dabrowski *et al.* [64] in their study report that existing tools perform poorly in feature extraction. Other accompanying data to feedback may be useful for determining the true nature or needs of users. This problem relates to validation, an important aspect of requirements engineering, which refers to whether a software organization is going to build the right thing for users. A recent study has also tried to match app reviews with issue trackers as developers struggle with guessing whether a reported app review has already been satisfied in the issue tracking system [65]. Organizations may further match issues from an app with how users are using the app. Similarly, one study explored matching app features with user interaction data and found that such an approach is promising for shedding light on user usage behaviors on mobile apps [66]. This could be useful for organizations to understand how users are utilizing their mobile applications.

More recently, Fazzi *et al.* [67] analyzed 2,611 app reviews from 57 COVID-19 apps and found nine categories of human aspect-related discussions that impact software usage. Obie *et al.* [68] analyzed 22,119 app reviews from the Google Play Store using natural language processing techniques, identifying that 26.5% of the reviews indicated user-perceived violations of human values, where benevolence and self-direction were the most frequently violated categories. Another study on 1,500 top free Android apps, more focused on accessibility issues, revealed that the majority of these apps contain significant problems that prevent individuals with disabilities from using the apps [69]. These studies demonstrated that various sub-categories of human aspects related requirements are identifiable from user feedback, in addition to functional related features.

Moreover, recent studies have used deep learning models such as BERT [70] to find feature requests and bug reports [71]. In an approach proposed by Mekala *et al.* [72], they developed a BERT-based sequence classifier that achieved 87% accuracy in analyzing user feedback. Das *et al.* [59] analyzed the comments generated on YouTube videos using natural language processing techniques and categorized comments on YouTube videos related to autonomous vehicles, further suggesting that YouTube can be a useful source of information for understanding consumer opinions and concerns. Additionally, the NoRBERT approach successfully classified non-functional requirements with limited training data [73].

This underscores the potential of utilizing state-of-the-art deep learning models in identifying requirement relevant feedback. These studies demonstrate that textual feedback is a viable input for the development of new requirements from user feedback as software organizations can gather insights about what users need regarding functional, non-functional, and any other human aspect-related concerns. This literature review lays one of the foundations for my overall research, which seeks to bridge various sources of user feedback with scalable, AI-assisted analysis to improve software products.

2.3 AI for Software Engineering and Requirements Engineering

The rapid evolution of Large Language Models (LLMs), such as GPT-3.5 and GPT-4, has opened new opportunities in software engineering, particularly within requirements engineering (RE).

Language models aim to model the generative likelihood of sequences of words. Given a token sequence, it predicts the probabilities of the future tokens or missing tokens in the sequence. With the availability of large-scale open-source datasets of source code, researchers and practitioners have reproduced such success on a series of software engineering activities. Researchers recently show that language models like ChatGPT (i.e., the one I leverage in this paper), produce outstanding performance in program repair, test case generation, and reproducing bug reports. Language models also demonstrate potential in education, for example, helping serve as an instructor in computer science courses [74].

Several studies explored the benefits of using Copilot in software engineering [75, 76, 77]. Yetistiren et al. [75] assess the quality of generated code provided by GitHub Copilot in terms of validity, correctness, and efficiency. They found that Copilot successfully generated valid code 91.5% of the time. Likewise, Nguyen and Nadi [76] and Dakhel et al. [77] evaluated the code generated by Copilot and found that the code quality varies depending on the programming language. These studies indicate that while AI tool(s) like GitHub Copilot is promising, there are still challenges of code quality and errors which engineers need to validate before using.

A couple of studies explore AI tools beyond just code generation [78, 79]. Jaber et al. [79] conducted a systematic literature review of 12 papers on the application of

ChatGPT in software development. The study indicates that ChatGPT has potential in various domains of software development, including automated program repair and bug fixing, programming numerical algorithms, software engineering decision-making, and more. Similarly, Khojah et al. [78] conducted an observational study of 24 professional software engineers who use ChatGPT over one week in their jobs. Through analysis of the dialogue between practitioners and ChatGPT, the authors were able to find guidance for how to solve their tasks.

AI has also been applied in requirement engineering [72, 80, 81, 82, 83]. Ronanki *et al.* [81] employ ChatGPT to evaluate the quality of user stories and show that ChatGPT demonstrates a consistent agreement rate throughout the evaluations.

A recent systematic literature review provides a broad overview of how LLMs are being applied across various RE activities. Khan et al. [82] reviewed 35 primary studies published between 2023 and 2024, identifying the use of LLMs in tasks ranging from requirements elicitation to modeling, validation, prioritization, and tracing. Their work highlights the increasing adoption of LLMs in RE but emphasizes the need for deeper understanding for measuring performance of LLMs for requirements engineering. Similarly, Hemmat et al. [83] conducted a thematic review of NLP and LLM applications in RE, noting that while automation is becoming more prevalent, future research should explore how to integrate LLMs with humans in the loop.

Many existing empirical studies assess LLMs based on the quality of their outputs when compared to human-generated artifacts [84, 85, 86]. For instance, Seifert et al. [84] replicated a human inspection experiment to evaluate LLMs (including GPT-4) on defect detection in requirements documents. While LLMs provided some useful results, they detected fewer issues than human reviewers and often missed defects.

Norheim et al. [85] offered a broader exploration, arguing that while LLMs show promise, there is a lack of systematic evaluation of what requirements engineering tasks are well matched based on the capabilities and limitations of LLMs. They emphasize that much of the existing literature evaluates LLMs as standalone systems rather than tools designed for interactive, human in the loop tasks. Moreover, they argue that future work should explore how RE tasks and use cases fit together in the context of a real product development process.

A handful of studies have investigated the use of LLMs for specific RE tasks beyond basic generation [87, 88, 89]. For example, Sami et al. [87] introduced a tool that uses LLM agents to perform automated prioritization of software requirements using common methods such as MoSCoW and the analytic hierarchy process (AHP). Simi-

larly, Oguz and Kuster [88] conducted a comparative analysis of use case descriptions generated by ChatGPT versus those written by human analysts, identifying quality differences and trying to offer an objective measure of ChatGPT generated use case descriptions. Santos et al. [89] conducted an empirical study with 30 participants where they compared human-authored user stories against those that are ChatGPT-generated. They found no significant difference in story quality with a simple prompt, and that a more sophisticated "meta-few-shot" prompt led ChatGPT to produce even higher quality stories.

Despite the increasing number of studies showcasing LLM applications in RE, a critical gap remains. The studies do not examine how users engage with the LLM during requirement analysis, including how users understand, adapt, challenge, or rely on the model's outputs. I focus on using LLMs and AI tools to generate requirements from a collection of user feedback from a wide variety of sources. By focusing my study on the overall interaction between practitioners and AI tools and by observing how users reason, adapt, and interact with the AI tool, the second-to-last chapter in this dissertation provides a human-centered perspective on the integration of AI tools into the development of new requirements from user feedback. This complements and extends prior work by bridging the current disconnect between automation and collaboration in requirements engineering research.

Chapter 3

Industry Perspective on Analyzing User Feedback

This chapter details my approach and exploration in addressing my **RQ1:** *How do software organizations manage user feedback to h improve existing products?*. Recent software development trends have encouraged rapid iterations and quick feedback loops [90]. Understanding user feedback is crucial for software organizations to respond to the evolving needs of customers.

User feedback analysis in requirements engineering has long followed an end-user-centric approach. Typically, organizations rely on interviews and surveys with stakeholders to elicit key and important requirements [91]. However, these user feedback sources have changed significantly over the years [11]. As per my former definition, CrowdRE is the term for research studying processes and tools for collecting and analyzing online user feedback (i.e., the crowd [5]). Specifically, social media analytics are frequently used as a key source of feedback for retail companies [42] and organizations can benefit from deeper analysis of user preferences [47].

To answer the RQ, I conducted an exploratory study following a Straussian Grounded Theory approach [92] through interviewing 40 industry practitioners from 32 organizations regarding their management of user feedback. The study participants came from 9 small, 9 medium, and 14 large organizations in domains ranging from e-commerce, analytics, and gaming, and played diverse organizational roles such as product manager, CEO, data scientist, data analyst, and customer success.

Table 3.1: Example Coding of Raw Quotes

Raw Quote	Code	Category	Concept
We do direct feedback from the customer via email, phone call, or chat support. So that can come in through our CSMs, or through our tech support channels. We also have, like an online community that our company built, it's kind of like open source feedback.	SocialMediaUtilization, InboundFeedback, OutboundFeedback, SourceOfUserFeedback	Collection	Life Cycle to Manage User Feedback
There's a lot of ownership among devs. If you're working on something, it's expected that you own that and see the potential risks and impact of you're doing and make design decisions	ProductManagement, SoftwareDevelopmentProcess	Active Employee Participation	Best Practices in Managing User Feedback

3.1 Research Methodology

To achieve broader understanding of the industry practices for managing user feedback, I adopted *Straussian Grounded Theory* [92] to collect and analyze data from industry practitioners. I conducted semi-structured interviews that constantly evolved during the study based on the information emerging from previous interviews. I then employed open coding, axial coding, selective coding and constant comparison with existing literature to constantly update my findings and questions throughout the study until reaching saturation [93].

3.1.1 Participant Recruitment and Selection

I invited practitioners from my personal contacts who worked at various positions in software organizations. To increase the pool of interviewees, I then extended to networking events and recommendations from other interviewees. I aimed to talk to practitioners from different sizes of organizations (i.e., a small business has less than

50 workers, a medium business between 50 and 249 workers, and a large businesses has 250 or more workers [94]), as the size could play a role in the practices and challenges experienced by an organization. I did not solely study large organizations as they are more likely to have more resources than smaller counterparts and likely to use the best practices for managing user feedback.

As small and medium sized organizations represent the vast majority of businesses [95], I aimed to uncover discrepancies between the organizations and identify potential areas of improvement. My participants came from 9 small, 9 medium and 14 large organizations. Additionally, I strived to engage practitioners from a wide range of industries to help fill the breadth understanding, and was able to engage with software organizations from industries ranging from e-commerce, analytics, and gaming.

As my study advanced, I enriched my strategy of engaging diverse organizational roles in my study; my interviews revealed various facets of user feedback, and helped me evolve my recruitment to fill the gaps of my understanding regarding the industry practices. My initial interviews included participants directly associated with product management, such as product managers, CTOs, CEOs, and customer success managers. I included senior management roles such as CTOs and CEOs, as in smaller organizations, they play a significant role in the requirements engineering process [96]. Upon reaching saturation from the interviews with the participants related to typical product roles, I learned about the benefit of involving other roles within organizations who are often assumed to be part of development rather than management or product process. These roles included: data engineer, requirements engineer, and QA engineer. This resulted in interviewing a wide range of roles, each of whom brought a diverse set of experiences in collecting or managing user feedback as part of the software development life cycle.

Thus, I ended up with a rich set of roles such as those (1) involved in user feedback collection (e.g., customer success agent who takes user phone calls) and (2) involved in feature development life cycle in some capacity (e.g., product manager who makes the final decision to add a new feature). Table 3.2 provides the demographic details of my 40 participants from 32 organizations. To ensure confidentiality, I have used P# (P1-P40) to indicate the participants and O# (O1-O32) to indicate the organizations they belong to.

3.1.2 Interview Design

I conducted semi-structured interviews lasting approximately 30-60 minutes and collected detailed notes and recordings. An initial set of 10 interview questions was prepared by the research team, following the general interview guide method [97], and based on my understanding of the existing work on user feedback [18, 19]. I provide my questions in the replication package, where the first 10 questions are indicated as base questions [98]. I followed the base interview transcript for each interview, which includes questions such as *"What are the sources of user feedback that your company typically uses? Do you feel your organization is effective in managing user feedback?"*

However, as the interviews progressed, I adapted my questions depending on the role of the participant to prioritize questions relevant to their role. For example, a product manager has a broad understanding of the function of the product team. Thus, I would ask the question: "Do you or your product team consider recent trends while identifying features?". In contrast, a developer may not have a comprehensive knowledge of the tasks carried out by the product team. As the interviews progressed, and in line with grounded theory practices [92], my questions evolved with the addition of questions relevant to specific roles or the situation so I improvised depending on a participant's response.

3.1.3 Data Analysis

The analysis process comprised of few different steps. As I recorded each interview, I transcribed them using an automated transcribing tool that converted audio to text. The data analysis involved coding process: open, axial and selective along with constant comparison during every step [92]. For open coding I broke down the interview transcripts into manageable dialogues and identified initial concepts based on my interpretation of the data. Two of the co-authors conducted open coding on the interview transcripts until all the transcripts were codified. After codifying each transcript, the two co-authors would discuss the definitions of each code and usage of the codes to increase shared understanding.

In the following steps, I aggregated the codes into broader categories based on different contexts and patterns using axial coding [92]. Throughout this process, I continued conducting interviews to gather more evidence on the developed concepts and categories. However, during the last four interviews, I did not uncover any

new insights, indicating that theoretical saturation has been reached [92]. I finally combined the broader categories into one core category through selective coding to reach my final theoretical conceptualization [93].

Table 3.1 includes an example of the steps taken during the analysis process. The two major concepts found from the analysis steps include: life cycle to managing user feedback and best practices organizations engage in for managing user feedback. A detailed codebook consisting of examples and code to concept generation has been provided in my replication package [98]. To ensure participant confidentiality I refrained from including the coded transcripts.

3.1.4 Member Checking

To ensure reliability and checking fit of my findings according to Strauss and Corbin [92], I conducted member-checking with ten of my interviewees. I presented my life cycle of managing user feedback and the best practices for managing user feedback to the participants of the member checking. While these member-checking participants agreed overall with the life cycle steps because they felt they captured the essence of managing user feedback. Several member-checking participants indicated that some of the steps in their organizations are often merged together (i.e., triangulation and prioritization). Several member-checking participants also pointed out that implementing all four best practices in an organization can be very difficult. Depending on the organization and with the limitations in the current available tooling, it may not be easy for an organization to collect all available user feedback from all the relevant sources.

3.2 Findings: Managing User Feedback

The forty interviewee participants in my study indicated a rich variety of sources of user feedback that are important to their business, ranging from app reviews, support channels, emails, phone calls, to usage metrics. I capture three broad categories of user feedback that emerged from my study in Table 3.3 and explain approaches for managing the information from these sources in detail in Section 3.2.2. The one emergent category of feedback that manifested throughout my interviews was the feedback stemming from social media. Interviewees highlighted to me that social media can have a significant impact on user perception, and misinterpretation can be

costly. Therefore, organizations that use social media user feedback should continuously monitor and respond to it. My study organizations were quite cognizant of the risks from misunderstanding user feedback, particularly as social media can amplify snowballing of user dissatisfaction, as I describe in Section 3.2.1.

My interviews showed a lack of a consistent, systematic approach to managing user feedback across the organizations. Yet, several activities emerged as across the organizations in how they consider user feedback and I describe them in a *life cycle of managing user feedback* and which encompasses the *collection*, *analysis*, and *validation* of user feedback, and its use in the *prioritization of features/bugs* in the software development process (Section 3.2.2). The variety and richness of information in the many user feedback channels, although offering unprecedented opportunities to listen to the voice of the users, creates significant challenges for organizations once the user feedback is collected, and there are subtle ways in which it is analyzed, checked for accuracy, and validated across multiple sources.

I also performed an in-depth analysis of interviews from practitioners who believed their organizations were effective in managing user feedback (i.e., in terms of understanding their user concerns and perceptions and avoiding user dissatisfaction). I refer to them as *high performance organizations* henceforth, with the view of distilling actionable insight for the smaller organizations that acknowledged their ongoing struggle and potential shortfalls in their approaches to user feedback. These practitioners came from typically larger organizations *(Orgs.: 1, 10, 19, 21, 22, 27, 28, 31)*. Four *best practices in managing user feedback* emerged from my data analysis, and I describe them in detail in Section 3.2.3.

3.2.1 Why User Feedback Matters

Misunderstanding User Feedback Can Have Major Consequences and Lead to Misguided Decisions and Wasted Resources

A common theme that emerged throughout my interviews with practitioners was the criticality of *accurately* understanding user feedback. P1 used a movie example as an analogy to illustrate the potential consequences of misinterpreting users. *"The movie Morbius became a meme... The meme was that no one [saw] this movie. [Discussion about the movie] became very popular. Sony extended the theatrical release of the movie because they saw a trend that the popularity of the movie was increasing. They did not test to determine that the [people] were having fun talking about how no one*

had [any] intention of seeing the movie." (P1) Consequently, Sony wasted money re-releasing the movie to no audiences [99].

This example touches on two crucial aspects regarding user feedback. First, user feedback snowballs and can rapidly increase in magnitude [23]. It would be in the interest of an organization to consider the discussion and potentially harness the discussion to its benefit. Second, the consequences of misunderstanding feedback are economic. If Sony had made an accurate assessment of user feedback, they would have realized that no one would spend money on the product, but their attention was just on the pure volume of mentions of the movie. Hence, my participants emphasized that it is not enough to just consider feedback; they also need to accurately analyze user feedback to understand what is influencing user perception regarding their product. This example not only demonstrates "why user feedback matters" but also demonstrates how the emerged user feedback category "social media" plays a significant role in effective feedback management.

Social Media Influences User Perception and Behavior

My participants emphasized that the increased importance of social media means placing more concern regarding the user perceptions that develop. *"One YouTuber [highlighted a big problem in our game]. In [our game]... for this year, the game team decided to revise how the system works... So there were a lot of negative feedback from the community through public channels" (P38)* The organization had to swiftly address this growing storm of discussion and assuage users about the problem.

To address the user perception, my participants also discussed its role in shaping user behavior. In a previous work that analyzed *narratives* in social media, the authors found that user perception has a major influence on user behavior. For example, where software products are perceived as suffering from a lack of privacy in turn, experienced users flock to competing platforms [100]. As P35 put it, *"user behavior [also] spurs research and insights and feedback to a large degree, you know, because you can take that and mold it and shape it and put it back into a product road map."*

User complaints are problematic because the organization will have to address them to diffuse negative feedback. P31 explained the measures that their organization took to increase the positivity from user perception on social media. *"[We] used to analyze the perception. So most of the times the feedback is negative on Twitter.*

[Redacted] made efforts to improvise that image in public by analyzing the tweets and then taking action." Upon releasing new features that addressed major concerns, the amount of negative feedback decreased. The reduction in negativity in user perception was a result of carefully curated efforts to deliver features that were well received by users.

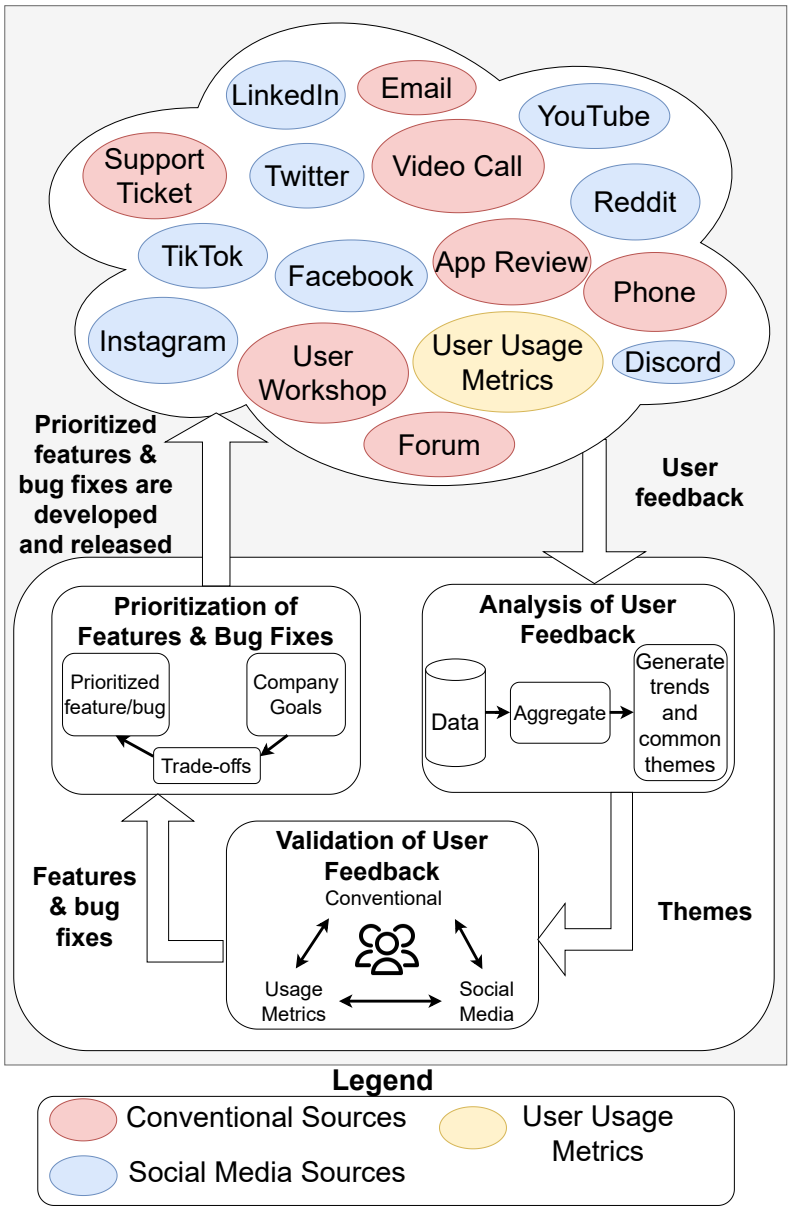


Figure 3.1: Life Cycle to Manage User Feedback

3.2.2 The Life Cycle of Managing User Feedback

The consequences of not acting or acting wrongly on user feedback and perception are both significant and best avoided. From my data, I synthesize a life cycle to manage user feedback (Figure 3.1) for software organizations based on my practitioner interviews and previous literature.

Collection of User Feedback:

The three broad categories that emerged from analyzing the transcripts of my interviews consisted of conventional, social media, and user usage metrics. I found that depending on the feedback category, the organizations manage them quite differently. This difference largely originates from the provenance of each user feedback as methods for collection, and the resulting data type is different for each feedback category.

For conventional user feedback, a user is providing direct user feedback to a particular software organization in the form of a phone call, video conferencing, or user workshop. Unlike social media sources where it is unclear whether a user actually uses a product, an organization receives feedback from someone who is likely using the product with conventional user feedback.

The conventional feedback will often be in the form of textual transcripts of user input from sources like support tickets and emails. These sources will also often include photos or screenshots of the software, which organizations will collect into issue tracker systems. Sources from conventional sources include the most popular source, email, as every organization listens to email feedback. Moreover, support channel communications and phone calls are also frequently leveraged by many organizations.

Similar to conventional sources, a vast majority of organizations *(Orgs.: 1-2, 5, 11-27, 29-31)* emphasized the importance of collecting user feedback from social media sources. I found that organizations are increasingly considering social media feedback, especially from newer types of social media such as TikTok *(Orgs.: 17, 25)*, Discord *(Orgs.: 1, 2, 5)*, and Instagram *(Orgs.: 1, 17, 20, 31)*. While previous studies on user feedback have tried to provide greater understanding on the types of feedback, they did not specify the details on how organizations actually collect and utilize the social media feedback [19].

I found that large organizations often paid close attention over social media, which is reasonable as they typically generate a lot of media attention. *"You can imagine this*

company is in the news all the time. It's definitely a part of our weekly life of what we look at." (P27) In contrast, many of my smaller organizations do not consider social media as they have not reached the scale to experience a large magnitude of feedback *(Orgs.: 3, 4, 6, 7, 9)*. My finding contradicts earlier research on the proliferation of user feedback in organizations, which indicated that small organizations consider social media, where as large ones do not [101].

However, the smaller organizations that do consider social media in my study have extensive experience and shared some of the pitfalls and mitigation strategies. For example, both O1 and O2 heavily rely on a combination of social media to build understanding of the user base and conceptualize new features in their software. *"Before we jump in and spend $40,000 plus on developing that functionality... let's get a sample of the social media posts, let's get a sample of the events in question." (P1)*

For social media feedback, a user typically provides feedback in the form of video (e.g., YouTube or TikTok) or text (e.g., Twitter or Facebook). Social media feedback is also often collated in the form of text or image, but social media requires a lot more active monitoring, and this process for monitoring is very much ad-hoc. Sometimes, social media feedback is more amplified into a trend because *"if someone starts talking about a particular feature, more people will start talking about that feature, that doesn't necessarily mean that is a valid signal, they're talking about it because of conversation is occurring" (P1)*.

Finally, I categorize data collected by tools such as Hotjar or Google Analytics in the category of user usage metrics. Tools like Hotjar provide comprehensive insights on the way that users use a product, ranging from button clicks to mouse hovers. These tools are sophisticated enough so that anyone at the software organization could replay a user usage session and visualize every action. *"Everything is tracked, every click is [tracked], every action is tracked. That user feedback is how users interact with the [product]." (P39)* With user usage metrics, the source of feedback more accurately represents user actions and does not depend on what users are telling the organization.

As shown by Table 3.3, different sources are used by different organizations. The sources used depend on several factors, including ease of access, cost, availability of tooling and data. For example, small organizations may not have reached critical mass to receive significant amounts of user feedback. Therefore, it is important for organizations to weigh their current needs and determine the critical sources.

Analysis of User Feedback:

My participants indicated the next step after collection is analyzing the trove of feedback. Key to the process is listening to the user feedback channels and group similar patterns. *"[We] went through all that feedback and grouped the similar themes about what were the common complaints and what were the common dream features that everybody wanted." (P4) "The number of times the question was raised was also important, because that means many [users] are having this confusion." (P5)*

For both conventional and social media user feedback, organizations tend to consider feedback as a "theme" when it gains traction. *"You got a trending amount of, let's say support tickets complaining about a specific issue" (P14)* A problem emerges as a theme when there is a growing or significant amount of feedback. However, organizations may view conventional sources as more credible, as social media feedback may be noisier. In the case of user usage metrics, organizations look for patterns through graphs that indicate a trend. *"Pendo [a kind of product analytics and feedback collection tool,] is kind of cool. Because it tracks every click, you can go into it on a user level and see exactly where someone clicked, at what time, and they generate these fantastic graphs for you." (P10)*

Once the common "themes" are generated, they are sent to the next phase through different methods such as Jira tickets. *"Once we have kind of enough information... who is it's going to affect? Is it affecting every user, how big of a deal is this? The product team creates a [Jira] ticket... and then we work on it and scope it out and do my research. [And] then it is forwarded from there." (P4)*

The perks of using social media user feedback includes rapid insights on bugs that occur in the product. *"During the testing process, we missed [including the menu items]. We woke up one day... All these messages on social media being, "what the heck, I just updated and none of the menu items [exist] anymore."" (P13)* Monitoring social media, if done effectively, can be powerful for quick feedback on bugs in the software. I see the importance of pure volume of feedback manifesting as a key factor of identifying themes. *"People will post snapshots of a Reddit post with a whole bunch of votes. And [users are] mad about the [feature], or on the new UI. And then a month later, there's a JIRA ticket and someone's working on it." (P22)*

A challenge with monitoring social media user feedback is the presence of noise, particularly in social media where spontaneous comments can quickly gain momentum. P38 emphasizes that it is essential to balance the signal-to-noise ratio and

consider which groups of users are over-represented or under-represented in this type of feedback. Some groups may be more likely to post on social media than others, which can skew the data [102]. While participants such as P38 mentioned that organizations should be cognizant of any bias that social media data may contain, they emphasized that they currently do not have any formal processes in place to eliminate the risk. Moreover, it is ideal to ensure that the feedback received represents a diverse range of users and is not biased towards any particular group. It is crucial to listen to all feedback, but it is equally important to recognize that not all feedback is equal, as feedback can suffer from poor quality [19].

The process of analyzing social media content is predominantly carried out manually by individuals or teams who actively monitor social media platforms during their regular activities. Many of the successful companies have teams that conduct this analysis. However, smaller companies struggle with the process and prefer to have a tool that would support their team. Moreover, the organizations heavily rely on manually looking through incoming user feedback to uncover underlying patterns and trends not just from social media, but also for conventional user feedback as well. Existing literature [19] also reported the reliance on manual method and the limited usage of automated feedback analysis among practitioners, despite a significant number of tools developed [103, 10, 104, 12, 105, 106, 107].

My participants reported similar limitations, but also specified that they do not use existing tools as they are too hard to use, expensive, or still not publicly available. For example, O4 attempted to use an existing tool to analyze their community forum in a better way. However, the tool was not as effective as expected and had an expensive yearly subscription cost. My findings indicate there is high emphasis on identifying reliable "themes", albeit through manual analysis, so there is strong demand for tools that can enable reliable and cost effective feedback analysis.

Validation of User Feedback:

Validating the emerging "themes" that arise from the analysis phase into a "list of actionable feature and bug fixes" for an organization to work on is the next important step. The validation process involves triangulation between the 3 categories of user feedback to reduce misinterpretations. *"In theory, if a [social media] trend is significant, you should be able to see the impact of that trend in your own user observation data as well. And if you can't, that's either you're not doing very good user*

observation data, or the trend is not impacting you yet." (P1) One can compare the themes from conventional and social media sources with user usage metrics and vice versa.

Another possible strategy is starting with social media to identify themes and then validating the themes using conventional user feedback such as video or phone calls with users to acquire direct feedback. *"Social media is a good place to get started. You find a trend. Next step is to collect a statistically relevant, smaller sample of genuine accounts that you care about." (P1)* However, similar to the analysis activity, triangulation has primarily relied on manual means in these organizations. After triangulation, an organization would have collated a list of feature and bug concepts that were validated and represent genuine issues.

By tracking user engagement with different tooling such as Hotjar or Google Analytics, one can see which features are being used effectively and which ones are being ignored. Social media, in particular, come with a lot of noise as discussed previously, but triangulating the themes from social media with user metrics can help understand the user behavior better. *"Feedback is a gift... The metrics don't lie. Metrics are metrics, they have no face... Sometimes it's about how a new feature or change you made, sometimes it's about a bug, you always look for what's the cause and effect of a change in a metric. So you have to kind of look at it from that aspect and realize that you can't always get [feedback] from direct comments from users." (P36)* A study suggests that industry practitioners rarely applied systematic validation of user feedback [18]. While existing literature adds various feedback collection, analysis, and prioritization processes in depth [19, 108, 109, 110, 18], there is a dearth of guidance on how to determine the validity of the emerging "themes."

One challenge with triangulation is the pure volume of data that is produced from the user feedback channels, particularly from social media. This high volume of data is difficult to validate as this process is often ad-hoc. Moreover, the volume of data may even require organizations to dedicate new roles and teams to manage this information. Therefore, the effort in establishing tools and processes to support this triangulation would seem worthwhile for organizations as it increases the likelihood of identifying which features and bugs are relevant and impactful for users before investing valuable engineering and product hours.

Prioritization of Features & Bug Fixes:

My interviewees highlighted that prioritization is a crucial step in the life cycle to manage user feedback, which involves making trade-offs between different bugs and features. A critical factor to prioritization is whether the "list of actionable feature and bug fixes" received from the validation stage match a company's vision and goals. Best described by P27, *"if we get feedback, that doesn't align to anything we're doing... we're not going to redesign it, we're not going to revamp it."*

The underlying driving force for major features is often the revenue that a feature may bring to the organization. *"If there is a feature that one customer wants and [has] a significant impact on revenue, we're going to implement [it]... If they ask to put a pink elephant on the homepage, we're not gonna do that. But [we will] if they [want] a feature that's gonna benefit [many] people." (P2)* Organizations place emphasis on activities that generate revenue or prevent financial losses, while also factoring in practical considerations. This is in accordance with literature [111, 112] where cost and risk have been indicated as major aspects of requirement prioritization.

The key to prioritization is balancing trade-offs so that there is mutual benefit for the users and organization. P6 astutely observes, "*if they [users] are giving us [feedback], it's likely that [something] would help them. [They] help us because they are going to use our system.*" *"People's perception of the company as a whole" (P28)* is also an important factor. If the feedback is related to something that is damaging the company's reputation, it will be given priority even if it doesn't have a direct financial impact. After all, a company's reputation is invaluable and can influence user behavior to adopt or drop a software product [100].

My participants also described that they most often prioritized bug fixes over features. Bugs are almost always prioritized over feature requests as bug indicate a blocker to current revenue, a top priority issue, where as new features is usually meant to obtain future revenue. *"New features are typically lower priority than bug fixes. So usually bug fixes are prioritized top of the list, because it's something that the users are paying for or having issues with as compared to something that just doesn't exist." (P17)*

The often ad-hoc process of prioritizing can result in the wrong decisions, which is particularly troublesome for smaller organizations with limited resources. *"Sometimes we end up choosing such a [feedback] to implement, which probably just takes a lot of time. [Alternative] task which [ended up] more important gets delayed. So we do*

make mistakes, but we try to avoid that by discussing and analyzing." (P6) Multiple approaches have been suggested to aid in determining the priority of requirements [113, 109]. A number of studies have focused on automating the end user feedback prioritization process [114, 115, 110, 116, 107], but as reported by my interviewees, industry practitioners rarely use the tools.

3.2.3 Best Practices in Managing User Feedback

Participants from *high performance organizations* shared with me one or more organizational practices in managing user feedback and which they described made them feel confident in achieving a good understanding of users, and in their ability to address their most pressing needs. I refer to these *best practices in managing user feedback* as the **4A**s: 1) Active employee participation, 2) Active feedback collection, 3) Active triangulation, 4) and Active response time. I outline them in Table 3.4, together with satisfying criteria. Many participants (11 out of 32 organizations) indicated the need for these practices, *"You know in startups, because the real struggle is we don't know the best practices out there." (P6)*

Active Employee Participation:

Organizations described from their experience that high performance in managing user feedback depends on an environment that encourages employees to thrive in user feedback management. When employees in different management and development roles actively participate in the feedback collection, analysis, validation, and prioritization, the team can effectively incorporate all three categories of user feedback.

I found that large organizations *(Orgs.: 19, 21, 22, 27, 28, 31)* often heavily encourage their employees to take the initiative, particularly for identifying issues in the software products. This observation aligns with prior research [117], which highlights how organizations structure initiatives to empower employees regardless of role or hierarchy to proactively contribute to innovation and problem identification. *"There's a lot of ownership among devs. If you're working on something, it's expected that you own that and see the potential risks and impact of you're doing and make design decisions." (P22)* Employees who are invested in the product may pick out themes in user complaints on social media while browsing, *"High majority of [our company] employees are also very interested in the product. It's very common that*

the employees will frequently visit [our company's] subreddits. Bug reports and feedback comes from internal employees being on [our] social media." (P22) Employees will frequently post screenshots from social media into internal communication channels regarding popular concerns. *"[We see] screenshots of a Facebook post saying "what does this alert mean?" Then someone's on it, trying to make the customer facing UI more understandable or adding something. This initiative is definitely led by employees that have other jobs that just happen to be also invested in these kinds of things." (P22)*

Participants from *(Orgs.: 19, 21, 22, 27, 28, 31)* suggested that organizations may benefit when they empower employees to speak up and advocate for certain features that they think are necessary. *"The philosophy at [our company] is there's a significant amount of personal ownership over everything you do... It's a little more effective to bring [decision making] closer to the source." (P28)* The participants explained that managers often relied on the tacit knowledge and collective wisdom from the team to share expertise about the software to make decisions. *"I [consider the] sources of feedback and then using my contextual knowledge about how important I think this is, we can choose to fix it before it becomes a problem... I [get my managers to] fight for this in the next meeting." (P28)*

Unlike larger organizations where active employee participation was more commonly reported, only one small and one medium sized organization *(Orgs.: 1, 10)* expressed utilizing this practice. The others instead relied on more top down approaches instead of taking advantage of the tacit knowledge and collective wisdom. *"The product owner would decide the prioritization of what the backlog looks like. [In] their sprint planning meetings, they would pull it into work to be done. I do not let the developers decide the priority." (P14). "I think [once] the product team [sic] has their vision as to what exactly it is that they need, then they'll start pulling the technical team because there's no point pulling us in any earlier [until] they know what they actually want." (P15)* However, organizations *(Orgs.: 19, 21, 22, 27, 28, 31)* that fostered individual empowerment in addressing user feedback indicated that they were able to reap benefits of achieving company goals and revenues.

Active Feedback Collection:

As described in the Section 3.2.2, all of my organizations actively collect feedback. However, for higher performance, the organizations suggest active feedback collection

from multiple user feedback categories, which 29 out of 32 organizations practice. To effectively gather and incorporate feedback from users, these organizations identify and utilize at minimum two sources that are representative of at least two of the feedback categories (social media, conventional feedback, and user usage metrics). To excel in active feedback collection, a participant would also regularly review and update its feedback channels pool to ensure it reflects the changing needs of the organization and its customers.

In addition, as highlighted by many participants, they benefited from having dedicated roles or teams to collate all the various user feedback sources. Despite data scientists and data analysts roles becoming ubiquitous in these software organizations, the volume of user feedback is quickly becoming vast. *"You have logs for everything. There's almost too much data. I think that one of the biggest issues is who's interpreting the data who's making the dashboard." (P22)* These organizations typically had to allocate resources to not only collecting user feedback, but also interpreting the data. All the large organizations with resources have dedicated teams and roles that serve as the first line to collect the different sources.

Sometimes these dedicated teams exist in the form of marketing or public relations teams. *"I think there are a couple teams that are dedicated to look into those. ... like what are the users talking about? What do they like? What do they don't like? ... I'm always getting those emails saying users really love this feature." (P26)* Additionally, organizations could also benefit from roles that are cross cutting in nature and who understand the various sources of user feedback such as a customer support agent who has knowledge about user usage metrics in addition to conventional user feedback sources.

As a whole, the participants reinforced the importance of resource allocation within an organization in the form of roles for successfully identifying the related sources. By utilizing multiple sources of feedback my high performance participants gained valuable insights and understood the needs and preferences of users.

Active Triangulation:

Triangulation emerged as a critical activity in the life cycle of managing user feedback *(Orgs.: 1, 4, 7, 10, 11, 13, 19, 20, 21, 22, 25, 27, 28, 29, 31)*. Considering the high number of participants that practice active triangulation of user feedback, my findings suggest that active triangulation is a best practice for validating *themes* in

user feedback.

In particular, I found 15 out of 32 organizations that conducted active triangulation, validated its *themes* from a feedback source with at least one other source from a different category of user feedback. By following steps described in section 3.2.2, organizations can verify whether a specific user feedback has validity or not. For example, if an organization notices that users are complaining about the UI of the landing page on Discord and determines that it is an emerging theme, the organization could triangulate this theme with user usage metrics to gauge whether or not users are spending less time on the landing page or bouncing from the landing page. Triangulating the two categories of feedback in this example, social media source and user usage metrics, would significantly improve the validity of the theme.

Additionally, organizations actively practiced validation before making any decisions about the feedback. *"If someone says, this is a problem, we have a lot of data sources that we can then query to verify that is a problem. And like, hopefully, if someone's DMing, you put in some amount of work to justify this as a priority." (P22)* The process of triangulation would also require coordination amongst different roles and teams in the organizations. Data teams would communicate findings from user usage metrics with findings found from product, customer support, and customer success teams who are more closely in touch with conventional and social media sources. This aspect relates to the aforementioned factor in section 3.2.3 "active employee participation" where *every* employee is encouraged to be cognizant of the feedback and product. This may open new collaborations and communication channels with cross-functional teams as different teams share responsibility in validating user concerns. *"I have to bug the [product people] often [for] validating because there are customers that are louder than others, you have to factor in that they could just be one very squeaky wheel. If one customer complains about it, [other] customers behind the screen [may have same issue]." (P12)* Participants discuss facilitating a culture for coordination and collaboration to help support validation.

Active Response:

My participants who expressed confidence in managing user feedback also suggested active response *(Orgs.: 1, 2, 7, 10, 11, 12, 14, 16, 17, 19, 20, 21, 22, 25, 27, 28, 30, 31)* to user feedback as a best practice. Active response implies responding to a feedback as quickly as possible. The response can be either a message or marketing post

to the end users addressing the issues. It could also correspond to fast development and release of the feature. In both cases, the time of response varies depending on the complexity of the user feedback.

Participants explained it is insufficient to just consider important user feedback. They need to also respond to user feedback, particularly critical feedback, as soon as possible. I know from literature that user feedback if left unresolved, can lead to cascading consequences that impact an organization's reputation and even bottom line [100]. This sometimes means that an organization shifts priority due to a sudden emerging user concern. For example, P1 highlights a notable example where responding to important user feedback and shifting priorities in a timely manner is paramount. *"The [product] security updates that occurred following [major] criticism [on social media]. [It] very quickly shipped some very good community safety and encryption changes. It came down to a matter of prioritization... Priority changed, and its message to customers changed because it's quite a bit easier to change priority and messaging." (P1)*

This approach shares similarities to the principles of DevOps, which principles encourage an "organization's ability to deliver applications and services at high velocity." [118] In the case of DevOps, organizations are measured by their lead time for a change, among other metrics, to gauge an organization's ability to release code in small iterative batches. However, for active response, it does not necessarily always mean that an organization must "release" new code right away.

An organization's response to user feedback could be in the form of a new version of software, but it could also take form in its messaging or marketing. As explained by P13, customers value the prompt response even if the product roll out takes time, *"[Following up] on customers, they really appreciate that... I think, sometimes they paint these stories in their brain that [customer support is] going to take days to get back to me, there's going to be so much back and forth. I think something my customers are always a bit surprised by is: "you actually follow up with with me."" (P13)*

The perceived benefit of applying active response is that users are more satisfied when an organization displays care for user concerns. A previous study on user-developer interaction on app reviews reports that users tend to increase their ratings after they receive a response from the product page [119]. This is in line with my findings that user perception is positively influenced by proactive engagement.

3.3 Threats to Validity

I use the total quality framework of Roller [120] and its elements of credibility, analyzability, transparency, and usefulness, to assess the threats to validity and mitigation approaches in my study.

For *credibility*, my study may suffer from sampling bias as I could only talk to participants who agreed to interviews with me. However, my interviews indicated that all of my participants manage user feedback in some capacity, and a non-trivial number of participants believed that they were effective in managing user feedback. I also tried limiting bias by informing the interviewees at the beginning of interviews that participants would be anonymized and the study would not cause risk to them. For *analyzability*, I utilized tooling to assist with transcribing the audio into text and also manually verified the transcripts against the audio. Two co-authors followed the steps of grounded theory and conducted open, axial, and selective coding to analyze the transcripts. For *transparency*, I attempted to provide rich descriptions and quotes where possible, and make available a replication package containing my base interview questions and codebook. Due to confidentiality agreements, I cannot release the interview transcripts.

For *usefulness*, my study is intended to shed more insights on how software organizations manage user feedback. I provide more empirical results regarding the life cycle of activities and best practices that they utilize. I conducted member checking with 10 of the participants and presented all the findings that emerged from the study in checking that my research findings resonate with my study participants and their organizational practices. I do not expect my results to hold true for all software organizations, though I would expect organizations of similar demographics to those in my studies to share similarities.

3.4 Discussion

Prior work [19, 18] reported a high level overview of user feedback collection and analysis practices. However, there lacked a detailed and structured set of practices for organizations to follow. My study aimed to address this gap, via a grounded theory study with a diverse group of 40 practitioners from 32 organizations of various sizes and in several domains of software development. My results yielded a life cycle to manage user feedback and best practices in managing user feedback.

In addition to the life cycle and best practices, I found differences with prior works on how organizations collect feedback. Maalej et al. [54] categorized user feedback into *explicit* and *implicit* to account for the broad spectrum of feedback channels. My study refines this categorization into three main categories of feedback: social media, conventional, and user usage metrics, to better reflect the difference in the methods of managing each category.

Previously, the use of social media user feedback has been championed in other domains of research. For example, in the airline industry customers who leave complaints and find resolution on social media are more satisfied [121]. My interviewees also highlighted that social media is increasingly an important source of feedback, but they also warned that social media feedback may not represent verified users and noise can be difficult to filter out. Regardless of these limitations, social media may be a powerful source that can benefit an organization, if social media sources pertain to the organization.

One of the main contributions of our work is the *life cycle of managing user feedback*. Other domains such as pharmaceuticals [46], retail [42], entertainment [44], and hotels [43] have also reported on utilizing user feedback to assist in understanding user concerns and behavior. The pharmaceutical industry [46] uses social media to identify the most discussed topics from customers and identify potential areas for improvement based on negative complaints. Some studies from other domains provide suggestions of how to collect data from online sources and make actionable insights [46, 122], but these studies are more focused on the analysis activity as opposed covering the entire life cycle.

Additionally, organizations shared with me *best practices for managing user feedback*. Unlike prior literature [19, 18], where they describe some of the activities that organizations employ for feedback collection, I provide guidelines for four practices that may help in achieving a comprehensive understanding of users.

These practices were more common in large organizations in the study, suggesting a change in the state-of-practice from a decade ago that large companies only loosely tracked their social media presence [101]. Both of these contributions represent empirically developed actionable insights of user perception and behavior for better products and to reduce user attrition. Next, I further discuss these implications for both the practice and research of software engineering.

3.4.1 Implications for Practitioners

Social Media User Feedback: The unprecedented, unique opportunity afforded by social media is that organizations not only can catch user dissatisfaction right away, but it also supports organizations to address and respond to users in a open manner. For example, a recent Reddit post sparked a discussion on tenfold increase in prices for the Pro tier version of Google Colab without informing the users [123]. This conversation was promptly noticed by an employee from product at Google Colab during his regular online browsing as he mentions in response to the post *"I mostly lurk [on Reddit]"*. The employee immediately addressed this issue acknowledging that the price increase was a bug in their update, quickly rolled out the bug fix, and issued refunds to anyone who was charged incorrectly. The fast response was appreciated by users who admitted that it prevented a significant number of subscription cancellations. This example illustrates the power of leveraging social media user feedback and addressing and responding to the issue in an open and transparent manner.

Best Practices for Managing User Feedback: The example demonstrates characteristics of active employee participation that emerged out of my study. My findings indicated that all my participants manage user feedback in some capacity, but few organizations follow all the 4As of best practices. These best practices emerged from my participants that consider their companies high performing in managing user feedback, which suggests that other organizations may benefit from applying these practices.

3.4.2 Implications for Researchers

Impact of Organization Size and Maturity: Most of the larger organizations employed the best practices that I identified, whereas smaller organizations were often seeking the best practices. Further empirical research should investigate the impact of organization size and maturity on the use of these best practices and whether or how they can be refined for most effective implementations in organizations.

Life Cycle Activities: Although some prior works that exist pertain to the collection and analysis of user feedback from requirements engineering [12, 107], future work should explore utilizing the amalgamation of various sources. As emerged from my study, social media sources are also increasingly important, albeit the presence of noise and sheer volume of data. In addition feedback validation and prioritization should also be further studied in the context of the evolving nature of feedback sources.

More Tools are Needed: Practitioners in my study reported that they heavily relied on manual approaches for both user feedback collection, analysis, and triangulation, despite the significant number of tools that have been developed and proposed for user feedback analysis and prioritization, both commercially and research-driven. Future research could study improving these tools and making them more accessible for practitioners.

Definition and detection of Themes and Trends: My study highlighted the importance of user feedback themes and the trends that may quickly emerge from the various feedback channels. However, how these organizations *define* or *quantify* trends was quite ad-hoc. Pointing out a trend or theme often relied on an employee's conscience or feeling. Further research can investigate strategies to better define, quantify or automatically identify themes and trends so that a more systematic and consistent approach may be leveraged by organizations.

3.5 Conclusion

In this chapter, I provided empirical insights into how software organizations manage user feedback through a life cycle consisting of collection, analysis, validation, and prioritization. The study also revealed the best practices employed by high performing organizations, including active employee participation, active feedback collection, active triangulation, and active response. These findings provide a direction to software organizations that strive to improve their user feedback management processes. The study further implies that requirements engineering is still predominantly a manual process. However, this study emphasized that organizations are keen to automate and streamline the user feedback analysis but lack the tools and processes to achieve it. In the following chapter, I examine how the automated development of new requirements from user feedback can be conducted using textual feedback.

Table 3.2: Participants and Their Roles and Organizations

Org.	P#	Role	Org. Size	Industry
O1	P1	Co-Founder and COO	Small	Analytics
O2	P2	Founder and CEO	Small	Analytics
O3	P3	Founder and CEO	Small	Analytics
O4	P4	Product Manager	Small	Educational
O5	P5	Software Engineer	Small	Crypto
O6	P6	Co-Founder and CTO	Small	Food SaaS
O7	P7	Co-Founder and CEO	Small	Financial
O8	P8	Head of Engineering	Small	CMS
O9	P9	Senior Product Manager	Small	Healthcare
O10	P10 P11 P12	Cust. Success Manager Data Analyst Product Manager	Medium	E-Commerce
O11	P13	Cust. Success Manager	Medium	Content
O12	P14	Co-Founder and CTO	Medium	Food SaaS
O13	P15 P16	Technical Lead Co-Founder and CTO	Medium	Analytics
O14	P17	QA Engineer	Medium	Content
O15	P18	Software Engineer	Medium	Healthcare
O16	P19	Data Engineer	Medium	Music
O17	P20	Software Engineer	Medium	Dating
O18	P21	Software Analyst	Medium	Industrial SaaS
O19	P22	Senior Engineer	Large	Automobile
O20	P23	Product Manager	Large	Travel
O21	P24, P26 P25	Software Engineer Senior Product Manager	Large	Software
O22	P27 P28, P29	Product Manager Software Engineer	Large	E-Commerce
O23	P30	Software Engineer	Large	Security SaaS
O24	P31	Data Engineer	Large	Insurance
O25	P32	Software Engineer	Large	Gaming
O26	P33	Data Scientist	Large	Financial
O27	P34	Software Engineer	Large	MaaS
O28	P35 P36	Head of Consumer Product CTO	Large	Software
O29	P37	Requirements Engineer	Large	Technology
O30	P38	Technical Project Manager	Large	Gaming
O31	P39	Data Engineer	Large	Social Media
O32	P40	Senior Software Engineer	Large	Advertising

Table 3.3: User Feedback Sources

Category	Source	Organizations That Rely On This Source
Social Media	Reddit	1-2, 12, 16-17, 19, 21-22, 25, 30
	Twitter	1-2, 5, 11, 13, 17, 19, 22, 24-26, 29-30
	YouTube	1, 23, 30
	Discord	1, 2, 5
	LinkedIn	7, 12, 21, 23, 29
	TikTok	17, 25
	Instagram	1, 17, 20, 31
	Facebook	17, 19, 27, 31
	Community Forums	4, 6, 8-12, 18, 21, 22, 30, 31
Conventional	Email	1-32
	Phone Call	3, 4, 6, 7, 9-11, 16, 22, 25, 28, 30
	Video Conferencing	3, 4, 7, 20
	Support Channels	1, 4, 5, 9, 10, 11-25, 27, 28, 30-32
	App Reviews	17, 22, 25, 26, 27, 28, 30, 31
	User Workshops	11, 28, 29, 30
User Usage Metrics	Data from Tools (e.g. Hotjar, Pendo, Google Analytics, etc)	1-8, 10-11, 13-14, 16-17, 19-23, 25-31

Table 3.4: Best Practices for Managing User Feedback

Practices	**Criteria**
Active Employee Participation	Every employee is actively contributing to the life cycle to manage user feedback (i.e., collection, analysis, validation, and/or prioritization)
Active Feedback Collection	Proactive data collection from all the feedback sources related to the product. (i.e., Sources equal n where n is number of sources containing an organization's feedback and $n \geq 2$) Sources must come from at least 2 out of 3 feedback categories
Active Triangulation	Organization actively leverages 2 or 3 categories (i.e., social media, conventional, and/or user usage metrics) to validate the "collated list of bugs and features"
Active Response	Prompt response to the feedback. A response could be in terms of marketing, communication, and/or bug fix or feature roll-out.

Chapter 4

Experiments Towards Automating User Feedback from Text Based Sources

This chapter presents my investigation in addressing my **RQ2:** *How can we automate the development of new requirements from textual based user feedback?* Building on the findings about how organizations manage user feedback, I identified the need for automation in analyzing user feedback from online sources. To explore this, I conducted two studies using large language models (LLMs) to analyze user feedback, each focused on a specific category of requirements relevant to user feedback.

The first study focused on *privacy*, a critical type of non-functional requirement that has become increasingly important for any organization that develops software. The second study explored *inclusiveness*, which is an emerging requirement relevant issue that refers to the inclusion, exclusion, or discrimination of specific user groups. In both studies, I demonstrated the potential of LLMs to support automated analysis from textual user feedback and highlight practical ways software organizations can respond to evolving user needs around privacy and inclusiveness.

4.1 Exploration through Lens of Privacy Requirements

Privacy is becoming, arguably, more than ever, a critical type of non-functional requirement for any software organization that collects or processes user data. Its im-

portance is evident from the enactment of privacy regulations in jurisdictions worldwide [124, 125, 126, 127]. Non-compliance with these privacy regulations can result in heavy penalties for software organizations [124]. In addition, organizations are also amenable to users who are more vocal than ever with concerns related to privacy breaches and infringements that captured global attention [128, 129, 130]. User concerns about their privacy are warranted and understandable. After all, serious personal data such as personally identifiable information or credit card information can be exposed when software fails to adequately protect or purposefully misuses personal data.

Although seemingly separate, privacy regulations and user concerns are interrelated. For example, privacy regulations such as the General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA) are impacting user privacy concerns. For an organization that must comply with privacy regulations, it is important that it recognizes and complies with not only regulatory-mandated requirements, but also shared user concerns. This is, however, challenging for many organizations, particularly small resource-constrained ones, that struggle with privacy compliance and must strike a delicate balance between regulatory compliance and business requirements.

Consequently, user involvement and feedback are becoming an important avenue for organizations to identify areas of privacy concerns that users may have about their software, as well as privacy requirements that could be developed to address these concerns [25]. Determining when and what concerns users express about privacy can be a critical success factor for an organization's privacy measures. Requirements engineering research has recognized the value of monitoring user feedback for the development of product requirements, though advances have largely been made in identifying user feedback and functional requirements from app stores and forums [55, 11, 57].

To this end, I conducted an empirical study focusing on Reddit user feedback because it is one example of a social media feedback source gaining increasing popularity, and Reddit allows for rich discussion between users in online communities. In Reddit, each community is known as a "subreddit" and allows users to engage in online discourse about a specific topic. A software development organization can benefit from monitoring its associated subreddit and eliciting potential requirements from user feedback.

Posts in a Reddit software forum have a higher character limit for their discussion

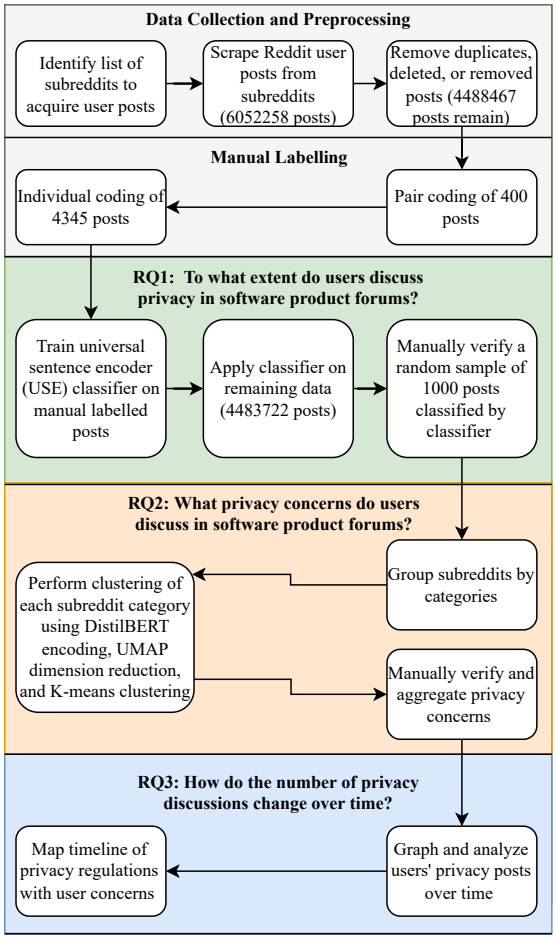


Figure 4.1: Research Process

as compared to app store reviews. This allows users to provide a more detailed review or longer discussion that covers multiple facets of a software, particularly non-functional requirements like privacy, which may be difficult to describe succinctly.

This study was guided by three research questions:

RQ2.1 How can we identify privacy-related user feedback from software product forums?

RQ2.2 What privacy concerns do users discuss in software product forums?

RQ2.3 How does the number of privacy discussions change over time?

4.1.1 Methodology

I conducted an exploratory study on user privacy concerns and changes in privacy discussion over time in software product related user communities using Reddit as a source. The methodology is summarized by Figure 4.1.

Data Collection

To answer the RQs I first collected data from 66 online communities (i.e., subreddits). 62 of the 66 are communities that are associated with popular software products such as WhatsApp, Telegram, AirBnb, or Instagram. I selected these software related subreddits because these subreddits should contain user discussions about the relevant software products. For example, if a user had a complaint or concern with a software like WhatsApp, they should in theory expound such issues in the WhatsApp subreddit. I selected subreddits of large, well-known software products. I did not select subreddits for smaller, niche-specific software and I note that these smaller software may be a source for a future study. I identified a further four privacy specific subreddits, as I thought they might contain many user discussions about privacy regulations that I could analyze for **RQ2.3**.

I collected each subreddit's posts via Pushshift [131], which is a large data store of all Reddit posts and comments and has a public facing API that supports ease of downloading for data analysis. In total, I collected 6,052,258 posts across the 66 subreddits. For the 62 software product related subreddits, I collected 5,890,182 posts, and for the 4 privacy related subreddits, I collected 162,076 posts. I looked only at the original (possibly edited) post, not followup comments. For a complete set of subreddits, please visit the replication package at `https://doi.org/10.5281/zenodo.6272629`.

After collection, I filtered the data to collect all posts between the creation of the subreddit and December 2021. I removed any duplicate posts and any post that was either deleted by the author or removed by a moderator of the subreddit. The data was left with 4,488,467 posts and I combined the title and text of each post to use for classification.

Manual Labelling

I collaborated with a colleague to label a ground truth dataset for training. I collated a list of privacy terms that I believe would be relevant such as "GDPR", "CCPA",

"privacy" ,"leak", "consent" and "unlawful", [124] and ran each post against the list of privacy terms. Based on the privacy term count, I randomly sampled 1721 0-privacy-term posts and 3024 1-or-more-privacy-term-posts. My colleague and I, with experience in privacy requirements and requirement elicitation in industry, performed four rounds of pair coding before individually coding the rest of the posts. The intent of pair coding was to establish a shared understanding of what it meant for a post to be privacy or non-privacy related. During pair coding, the two coders had the same code (i.e., either *privacy* or *non-privacy*) in 77%, 89%, 83%, and 85% of the posts. The Cohen's Kappa was 0.39, 0.53, 0.51, and 0.60, which our agreement levels hovered around moderate levels of agreement [132]. Cohen's Kappa is lower than the overall pair coding agreement percentage because it accounts for imbalance in the dataset. In my dataset, there were more non-privacy posts than privacy-related posts.

Subsequent to the pair coding, we individually coded a further 4345 posts. Of the manually labeled 4745 posts, 794 posts were labeled as privacy related and 3951 were labeled as non-privacy related. Since a user may write a post containing several critiques, including, but not limited to, privacy, we labeled a post as *privacy related* as long as part of the post referred to privacy. For example, *"I switched from my former browser to Firefox recently... mainly for the very useful add ons available. One set of add ons I added has privacy in mind... one of these privacy focused add ons... was Multi-Acc Containers..." (Firefox)* was labeled as a privacy post.

Training the Classifier to Identify if Users Discuss About Privacy in Software Product Forums

Model: I trained a classifier to help answer (RQ1) and to identify privacy related posts that I could analyze for user privacy concerns (RQ2) and change in number of privacy discussion over time (RQ3). Previous work using Reddit data [58] applied bag-of-words [133] and TF-IDF [134]. However, Devine et al. [135] compared text embedding techniques for analyzing user feedback and found that pre-trained deep learning models, particularly Universal Sentence Encoder (USE) [136], performed much better than word frequency models like TF-IDF. The model I trained using the base USE model with the transformer encoding mode [137] had a precision, recall, and AUC of 0.84, 0.8, and 0.91. Privacy is not a common topic in app reviews or users posts, so the data was imbalanced. Like previous work on Reddit [58], I used

the oversampling technique SMOTE [138] to address the imbalance.

Manual Verification of Classified Results: After running the USE model to identify whether users discuss privacy in software product forums via Reddit (RQ1), I randomly sampled 1000 posts from the data that contained 500 labeled privacy and 500 non-privacy. My colleague and I manually labeled the 1000 posts to check the performance of the USE model. I did not have access to the model's predictions until verifying the classified results was finished to reduce bias. The USE model achieved a precision, recall, and accuracy of 96.8%, 96.8%, and 96.7% on the random sample.

Identifying the Privacy Concerns Users Discuss

I clustered similar privacy posts to find the main areas of privacy concerns to answer (**RQ2.2**). I first grouped similar software into categories as shown by Table 4.1. For (**RQ2.2**), I implemented K-Means clustering on the posts from each of the categories to find out the primary privacy concerns. Next, I found the best number of clusters using the silhouette coefficient as the metric before combining similar clusters across different categories into fewer clusters. I used DistilBERT [139] for embedding the data and performed dimension reduction using UMAP [140] as part of clustering of the data. Moreover, UMAP is a general-purpose machine learning dimension reduction algorithm that is both fast and scalable [140].

After dimension reduction, I conducted clustering via K-means [141] and tried clustering with 2 to 10 clusters for each category. Clustering with more clusters (e.g. 20 clusters) can result in a larger number of privacy concerns, but the goal was to analyse the primary categories of privacy concerns, so I choose 2 - 10 clusters. I compared the results of different clustering and embedding using the silhouettes coefficient, which represents the distance of each point to the center of its cluster and with the closest neighboring cluster [142]. I randomly sampled a minimum of 25 posts from each cluster to compare with other clusters and merge similar clusters to collate common concerns.

Mapping User Privacy Posts Over Time

I mapped posts over time to observe the change in the number of privacy discussions over time (**RQ2.3**). I plotted the number of posts based on monthly time intervals. The rationale for plotting the number of posts over time was to observe if privacy

Table 4.1: Categories used to group similar subreddits

Category	Subreddit
General Privacy	degoogle, privacy, PrivacyGuides, privacytoolsIO,
Social Media	Bumble, discordapp, facebook, facebookmessenger, Fiverr, instagram, lineapp, linkedin, MicrosoftTeams, Pinterest, signal, SLACK, snapchat, Telegram, Tinder, Twitter, Upwork, WeChat, whatsapp
Tools	androidapps, chrome, duckduckgo, duolingo, firefox, Google, kahoot, miband, operabrowser, zoom
Platform	Android, chromeos, microsoft, ios, windows, windows8, windows10, Windows11,
Entertainment	DisneyPlus, HBOMAX, netflix, soundcloud, spotify, youtube, YoutubeMusic
Financial	CoinBase, CashApp, venmo
Shopping	Aliexpress, amazon, Ebay, Wish
Food and Drink	deliveroos, doordash, McDonalds, starbucks, UberEats
Voice Assistant	amazonecho, googlehome, Siri
Travel	AirBnb, GoogleMaps, Lyft

regulations like the GDPR had any noticeable impacts on the number of privacy posts. Specifically, I wanted to visualize if there was a substantial increase in privacy posts when privacy regulations like the GDPR became law. Finally, I normalized changes in privacy and non-privacy posts by their median as shown by Figure 4.2.

4.1.2 Findings

RQ2.1: How can we identify privacy-related user feedback from software product forums?

The results of applying the classifier are shown in Table 4.2: I identified 129,075 privacy-related posts in the entire dataset of 4,488,467 total posts. Privacy is represented by 2.9% of all posts and 1.8% if only considering software product subreddits. 1.8% is higher than the 0.12% found in prior work on Android app reviews [143], but this was anticipated because users have more space to comment on multiple concerns in Reddit posts, and the Android sample was not focused on privacy. Popular social media software such as WhatsApp and Telegram not only had high numbers of privacy posts, but also had a high proportion of privacy posts.

Table 4.2: Result from classifying posts from all subreddits

Subreddits	Classification	Count	Percentage
62 Software Product Subreddits	**Non-Privacy**	4274892	98.2%
	Privacy	78979	1.8%
4 General Privacy Subreddits	**Non-Privacy**	134596	96.3%
	Privacy	50096	3.7%
	Total Privacy	129075	2.9%
	Total Posts	4488467	100%

RQ2.2: What privacy concerns do users discuss in software product forums?

After embedding data using USE and then clustering the subreddit posts using K-means, I manually labeled those clusters and discovered that Reddit users primarily discuss 9 major privacy concerns.

Table 4.3 shows the total number and name of the subreddit categories representing these privacy concerns. The most common concerns among the categories were about software privacy policies and permissions, and user experienced privacy issues and recommendations for remedying the issues. For instance, the post *"Apps which request sensitive permissions must provide link to valid privacy policy in the app and Google Play Developer Console" (Android)* indicated the need of transparency for sensitive permissions.

The next most common concern among subreddit categories was regarding users sharing their experiences and stories of privacy compromises and scams describing the consequences of losing their privacy over the internet because of using a particular software product. To illustrate, *"I was hacked on my just eat account Saturday night someone ordered food and used my card details I had cleared my card details and sorting it out..." (Deliveroos).*

Unconfirmed privacy compromises were instance where users discussed times they were unsure and suspected being hacked or scammed. A similar frequency was noticed for concerns where users are distressed about revealing their personal or monetary information to the application and querying its safety on the forum. For instance, *"Quick question is it safe to put your debit/credit card info in? I have bought a few*

Table 4.3: Major privacy concerns and associated subreddit categories (from Table 4.1)

Privacy Concerns	**Post Count**	**Subreddit Categories**
Privacy Issues and Recommendations	39771 (31%)	(8) Entertainment, Tools, Food and Drink, Social Media, Travel, Voice Assistant, Platform, General Privacy
Privacy Policy and Permissions	27128 (21%)	(8) Tools, Financial, Shopping, Social Media, Travel, Voice Assistant, Platform, General Privacy
Privacy Compromise Experiences	7037 (5%)	(5) Entertainment, Tools, Food and Drink, Financial, Platform
Personal Information Exposures	23986 (19%)	(4) Food and Drink, Shopping, Social Media, General Privacy
Unconfirmed Privacy Compromises	10715 (8%)	(4) Financial, Shopping, Social Media, Travel
Bug and Suspicious Activity Complaints	15501 (12%)	(3) Entertainment, Financial, Shopping
Warnings and Advisories	3829 (3%)	(3) Entertainment, Food and Drink, Travel
Privacy Breach Effects	226 (0.2%)	(2) Food and Drink, Travel
Phishing Emails	885 (0.7%)	(1) Financial

things from [AliExpress] before but used paypal... the seller i want to buy from does not accept paypal so before I buy I want to make sure its going to be safe." (Aliexpress).

Furthermore, complaints about privacy related bugs and suspicious account activities were observed. Users draws attention to some issues they were facing that were neglected by respective companies. For example, *"I've been dealing with this for several months now... I [struggle with Amazon], calling Amazon's customer service multiple times and asking if there is a way to force every logged in device to sign out of my account...Amazon should be able to secure my account... I wanted to... see if other people are dealing with it too." (Amazon).* Next, I observed posts conveying warnings and advisories for other users to avoid scams and even recommendations to avoid particular software to be secure from privacy breaches. For instance, the post *"Beware: WhtsApp web violation the privacy of Firefox Users. Use Telegram instead and uninstall this garbage!" (WhatsApp)* specifically warned the other users of WhatsApp terms and suggested to others to use other apps.

The remaining concerns were about the effects of privacy breaches that represented

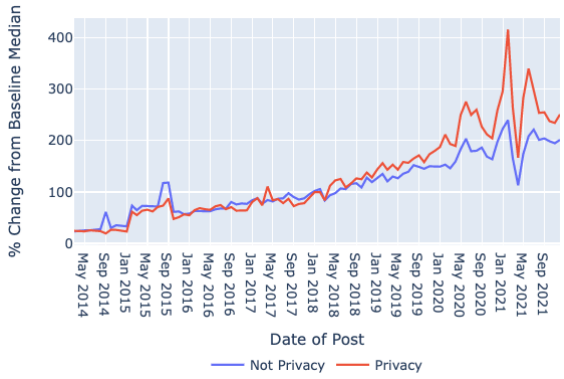


Figure 4.2: Change in all subreddits post count by month, normalized by the median post count for privacy and non-privacy.

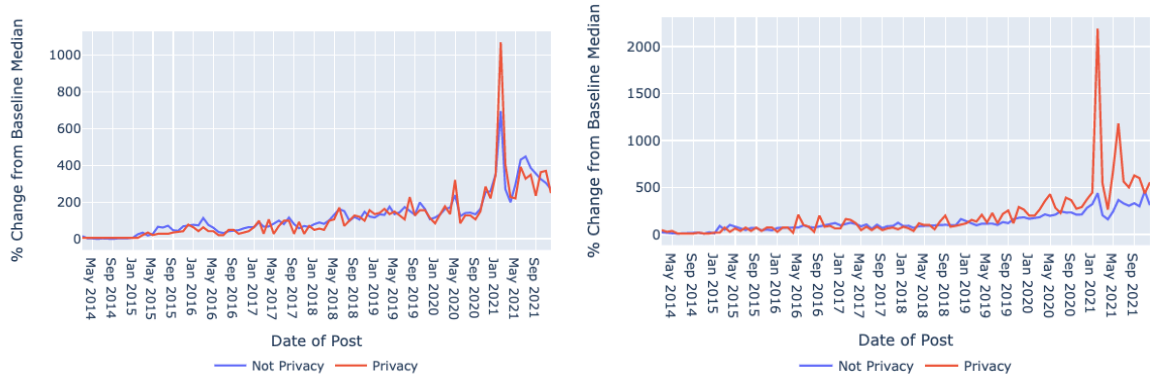
the consequence of the privacy issue and violations of company rules and regulations by other users (such as illegal AirBnB cameras), and user concerns about phishing emails and email related scams.

RQ2.3: How do the number of privacy discussions change over time?

To analyze how the number privacy discussions changed over time I mapped the posts over time and normalized the number of posts per month by the median post count starting from May 2014. Although, I initially plotted the posts from the beginning of the subreddits' creation time, the number of posts before May 2014 were not significant enough to be shown on the graph. Thus, I showed the plot from May 2014 to 2021. I can further see from Figure 4.2 that there was a steady increase in the number of privacy posts per month over time, reaching a peak in January 2021. In consideration of this peak, I observe a rise in the number of posts beginning in September 2020, so I calculate the growth rate for the 4 months leading up to January. Privacy posts showed a 119% increase per month, compared to a 34% increase per month for non-privacy posts, between September 2020 to January 2021.

Part of this increase can be explained by the overall increase in users posting on Reddit during this time, which aligns with reports that more users joined Reddit the last few years [144].

Starting in January 2020 and culminating in January 2021, privacy related posts increased significantly over the median. In January 2021, the number of privacy posts reached a new high of more than four times the median. While total posts on Reddit increased during this period, I can see from Figure 4.2 that non-privacy posts did not



(a) Telegram Subreddit (b) WhatsApp Subreddit

Figure 4.3: Change in the number of posts per month, normalized by the median post count for privacy and non-privacy posts.

increase in the same magnitude.

To better understand factors for the increases in the number of privacy discussions, I conducted a more thorough manual analysis of these posts. the preliminary analysis indicates that an important factor for stimulating user privacy discourse are privacy *narratives* that may trigger user behavior beyond the influence of privacy regulations. I describe this analysis next.

4.1.3 Narratives as Explanations of Variations in Privacy Discussions

Narratives are a story or way of understanding a series of events that fosters a point of view [145]. I borrow the term narratives from economics [146] where researchers try to understand "how stories go viral & drive major economic events". They expand on the dictionary definition of "narratives", a story involving humans, to include songs, jokes, and explanations. In economic narratives research, people analyze historical events and have hypothesis on narratives that caused economic events.

For example, one of the sharpest economic US contractions ever occurred between 1920 to 1921 when consumer prices fell 16%. One possible narrative explanation for the cause was that consumers were angry at supposed war profiteers in World War I and decided to boycott the profiteers by holding off buying necessary goods, thinking that they could get back at the profiteers [147]. However, what people did not realize is that they would help cause a short term depression in the process.

For understanding the reason behind the outlier behaviour in the privacy discussion, I analyzed the data to see whether narratives can explain trends in the user discussions about privacy. This is another area that a software organization can supplement as part of its requirements engineering processes. It provides more understanding about the user's perspective on privacy. Moreover, for an organization this provides a whole new dimension for risk analysis before the development of a software product.

I approached narrative analysis in two ways. One approach, top-down, looks for well-known events and identifies what narratives form in response. I chose the enactment of the GDPR as an example of this. The other approach is to examine the data bottom-up and identify popular narratives that emerge from user discussions. I highlight narratives around chat apps Telegram and WhatsApp and browsing apps Chrome and Firefox as two examples.

Introduction of the GDPR

Although the number of privacy regulation posts was only a small fraction of the entire data, I found trends at various points in time with similar patterns for both events.

I found a total of 1351 posts that are related to the GDPR. The first mention of GDPR was detected on 16 June 2015. However, the number of conversations about it was low before increasing in 2018. Since the GDPR was enacted (came into effect) on 25 May 2018 [124], I assume the GDPR enactment acted as a catalyst for the rise in posts during the deadline week (21-27 May 2018). 78 posts mentioning GDPR were posted that week, out of which 22 were on 25 May, which is the maximum number of daily discussions on GDPR from 2015 to 2021. Observing Figure 4.2, I see that there is little change in the number of privacy posts in the months around May 2018. The number of privacy posts is hovering around the median level, significantly lower than the level observed in 2020 or 2021.

These posts indicated user concerns over the use of their data by software like Google and Facebook. Posts like *"How can individuals get access to all their personal data being stored by companies like Google and Facebook... GDPR coming into force today entitles users access to their personal data... it would be fascinating to find out exactly what data is being stored and to what use it is being put..." (25 May 2018)* highlight such concerns.

However, I noticed a gradual drop in GDPR-related discussions as news surrounding the event wore down. Although the first GDPR related posts were identified in 2015, I found that the overwhelming majority of conversations regarding the GDPR occurred in the short window leading up to, and shortly after the privacy regulatory event. Moreover, most user discussions primarily focused on the privacy policies of big companies and user data concerns and not the regulation itself.

the analysis on posts after mapping them over time concerning privacy regulation indicates that **privacy regulatory events have a short term impact on people** (I report on the similar short-duration impact of the CCPA in the replication package). Users mainly post about privacy regulations in the short interval during the week that the regulations became law.

The enactment of the GDPR, while of short-term relevance, never acquired a viral status to persist beyond this period of time. I elaborate on the importance of the short-term narrative in the Discussion section.

Telegram vs WhatsApp

After Facebook announced plans to merge the infrastructure for WhatsApp, Instagram, Facebook, and Messenger [148], the Reddit post data showed a outlier behaviour in posts from users in those related subreddits criticizing the move and expressing concern about the repercussions for their privacy and also comparing it with telegram. I can see from Figure 4.3b and 4.3a that there were major outlier behaviour in the months near January 2021. In fact, the month of January 2021 represented a month of significant growth of the amount of privacy conversation from Whatsapp and Telegram. If software companies can identify the influence behind such increases, they can potentially link these concerns to privacy requirements and issues.

I thus manually analyzed 350 privacy posts from these five social media subreddits to get a deeper understanding. I found that there was a common theme between the privacy concerns for Telegram, WhatsApp, and to an extent other Facebook related software such as Instagram, Facebook, and Messenger and analysed these themes to understand the narratives behind.

Narrative 1 - "WhatsApp is sharing advertising data": Users voiced concerns regarding the implications from shared advertising data and merged infrastructure. For example, *"Now Facebook, Instagram and WhatsApp are all linked to pry on you...[they] are all now linked. How deep this could affect your privacy is not known"*

(Signal). The perception is that Facebook intends to integrate WhatsApp and other services, thereby increasing Facebook's advertising ability. Users expressed concern for sharing advertising data, *"We talked about sweatshirts [sic] at WhatsApp with my friend now [I] am seeing sweatshirts [sic] ads everywhere on Instagram, [I] swear [I] did not googled or something else sweatshirts [sic] word" (Facebook).*

Narrative 2 - "Telegram is a privacy-centric alternative": I contrast the previous narrative with the narrative around Telegram, a WhatsApp competitor, in which the perception is that Telegram promotes user privacy. The perception before and especially after Facebook's announcement is that a user could consider joining Telegram if they value their privacy, *"[Signal] and Telegram are better than whatsapp. Yes my privacy is important to me..." (WhatsApp on 15 Jan 2021).* Similarly, *"Well, I've been using WhatsApp since 2010 and I honestly kinda wanted to quit for years but everyone I know kept using so what could I do? But with this new privacy policy, many of my friends have made the switch and I followed." (WhatsApp on 11 Jan 2021).* In addition to merging infrastructure, Facebook also introduced a mandatory and controversial privacy policy for WhatsApp [149]. A user must accept it to continue using WhatsApp [149]. Narratives surrounding WhatsApp quickly became about WhatsApp's perceived privacy issues.

However, not all users were convinced about Telegram's privacy. For example, *"After WhatsApp's new service policy, I see majority of my friends are moving to Telegram. They still believes It's more secure than Signal. I don't know why they do so." (Signal on 23 Jan 2021)* and *"The mark of Telegram is privacy, why only the secret chat has a good encryption?... if they boast so much about privacy, well they should show it, I don't feel so safe using the app anymore." (Telegram on 20 June 2020)* Still, I see from Figure 4.3a that Telegram experienced about 1000% increase in privacy and 700% increase in non-privacy discussion from their median counts around the time of WhatsApp's controversial privacy policy.

Partially fueling this increase in user discussion was the massive increase in first time Telegram users. Similarly, I see from Figure 4.3b, WhatsApp users discussing privacy a lot more than normal during this time. Narratives about these software seem like a potential driving force for the large spikes in user privacy discussion during this time, at a time when there was no other privacy regulation to influence privacy discussion.

In economics narratives, Shiller often compares narrative hypothesis with GDP growth and other economic measures to verify if narratives appear reasonable [147].

I conducted a short comparison between the privacy policies between Telegram and WhatsApp to investigate whether the user concerns and privacy narratives regarding Telegram and WhatsApp are accurate representations of reality.

Firefox vs Chrome

Narrative 1 - "Choose Firefox for its privacy": While Chrome is the more popular browser [150], I found that users frequently express concerns about Chrome's privacy and praise Firefox for its perceived privacy-centric approach for handling user privacy. To this end, I found instances of users discussing their migration from Chrome to Firefox. For example, *"A friend had told me about the privacy issues involved with Google Chrome so I decided to make the switch to Firefox." (Firefox)* The analysis indicates that users encouraged others to switch, which suggests that narratives could influence user perception and potentially even human behaviour to an extent.

Privacy-centric browsing modes provide users with more options to address their privacy concerns, but I note that it is unclear the prevalence of users who truly understand the differences between the browsing modes and can decipher the use cases for when and what parts of their data may be shared. In 2019, Firefox began blocking third party trackers as a default setting as part of a major update [151]. Although the setting was originally introduced in 2017, a mere 3% of Firefox users applied the setting prior to the 2019 update, most likely because as suggested by a Firefox SVP, expecting users to alter their browser settings placed an "undue burden" on them [151]. Therefore, despite the narrative and perception that a user should switch to Firefox if they value their privacy, **it is unreasonable to assume that their privacy is secure without fully understanding the implications of each privacy setting.**

The preliminary analysis suggests that narratives surrounding software may play a profound role in shaping the perception and concerns about software products, and may even impact user behaviour. The privacy-centric narrative surrounding Telegram helped it increase its user count during the backlash against WhatsApp, at one point attracting 25 million new users in a span of 3 days and reaching 500 million global users overall [152]. Similarly, the privacy-centric narrative regarding Firefox may have convinced some users to switch from Chrome to Firefox.

Narratives can be identified top-down—using external events to search for narratives—

or bottom-up, browsing user discussions. Based on the preliminary study that I performed in this section, I conclude with this hypothesis for future studies:

> **Hypothesis**
>
> Privacy related narratives impact user privacy concerns about a software product.

4.1.4 Threats to Validity

External validity

The first threat is that generalizability of the results could be limited because we collected only 66 subreddits. However, to mitigate this we collected the subreddit data from a wide range of applications and services. the data is comprised with subreddits that are associated with popular software products such as AirBnb, or Instagram which have relatively higher user counts, but the data also contain popular software products with fewer number of user posts. Since my focus was on popular software products that generally have large numbers of users, it is possible that my approach for identifying user privacy concerns and narratives does not generalize well to software products that do not have a high number of user discussions.

Construct validity

The threat applies to the manual labeling of posts to privacy and non-privacy which is to prepare the ground-truth data. Manual labeling can cause experimental bias as humans tend to be subjective in their judgment which can be very difficult to eliminate, but text classification is generally done manually. We tried to address this problem by having two experts who are well versed, experienced, and understand privacy. I calculated the Cohen's kappa value and the inter-rater agreement levels which also reflects our labeling efficiency.

A similar issue of subjectivity also applies when it comes to analyzing and comparing privacy policies as lawyers have different judgments of privacy policy implications. For example, lawyers had conflicts in the interpretation of complaints in GDPR regulations [25]. The possibility of different interpretations cannot be eliminated. We tried to mitigate this threat by having a law scholar compare these policies.

Internal Validity

There are threats internal validity that relate to the understanding of the data. It was not possible for us to manually cluster each post to privacy concerns. To mitigate this threat we took the 9 main areas of privacy concerns and randomly sampled at least 25 posts from each area to check if they belong. Furthermore, there are limitations to the choice of time intervals for mapping user privacy posts over time. A researcher would need to reduce the time interval to daily or hourly if they want to visualize short-term narratives. Contrastingly, one may need to expand the time interval to yearly if they want to visualize the impact of long-term narratives.

We also acknowledge that finding all impacts of privacy regulations on user concerns is not possible. To address this we extracted all the combinations of terms for GDPR, collected the relevant posts, and manually identified if a post is talking about these regulations. One threat to the work is that we did not take advantage of addition attributes corresponding to a post such as number of comments or number of up votes. As the primary focus was answering the RQs we did not investigate the effect from the additional information, we leave that for future work.

However, we address that we could not explore other aspects of the privacy regulatory events such as considering the impact of changes made in the products due to the enactment of the policies that could potentially trigger the user discussion. Another potential limitation of the work is my interpretations of narratives. My study of narratives was exploratory and my definition originated from economics. I acknowledge that my approach of identifying narratives from observing multiple similar user posts and matching these with news stories from the same time frame may not be the best approach, but I believe my work still brings attention to this area of research. Future work may leverage tools or strategies from social network analysis [153] for further study of narratives.

4.2 Exploration through Lens of Inclusiveness Requirements

As software usage continues to grow worldwide, an increasingly diverse user base is engaging with the applications. The diverse group includes individuals from various genders, regions, cultures, socio-economic backgrounds, political beliefs, people with physical and cognitive abilities, values, and educational backgrounds, among many

others. However, software is often built for the "average user" [154] and fails to adhere to the diverse user needs. For instance, X (previously known as Twitter), a widely used social networking app with over 390 million global users [155], released an image cropping algorithm that automatically cropped images. It focused on important parts, such as faces and text, to optimize space on the main feed and allow multiple pictures in a single tweet. However, users soon identified that the algorithm could only detect white faces and cropped out faces of black people [156]. The topic soon became trending as thousands of users joined the discussion. Similarly, numerous other incidents have emerged from online user feedback [100], highlighting the lack of inclusiveness in software.

The feedback provided by users on online platforms (e.g. app reviews) has grown significantly in amount and has become important to software organizations. Software companies can identify areas of product improvement based on this feedback. In this space, Crowd Requirement Engineering (CrowdRE) has become a popular area of study for identifying product relevant information from large volumes of user feedback on various online platforms such as app stores, social media, and forums. A growing body of research in "end user human aspects" has attempted to address and understand aspects such as gender and accessibility using CrowdRE sources such as App reviews [157, 69].

Khalajzadeh *et al.* [158] studied user feedback from Google Play Store and developer discussion from GitHub to understand human aspects related conversations from 12 open source apps. The authors reported insights from discussions and concerns related to inclusiveness from both sources (31 from Google Play Store and 31 from Github). Although insightful, open-source applications represent only a small portion of the many applications used in our society and, therefore, can result in limited representation of the diverse user needs.

Hence, there is a need for a more extensive exploration of concerns about inclusiveness from a larger, more diverse user base. With the increasing number of user feedback platforms (e.g., social media), diverse users may prefer to use different mediums due to different levels of engagement with particular online platforms [38]. Thus, exploring a variety of sources of feedback can reveal more insights about inclusiveness. Finally, the growing amount of user feedback, while useful, represents a significant manual effort for software organizations, making the automation in identifying inclusiveness-related user concerns worth the effort.

To fill this gap, I collaborated with a colleague to analyze (using manual and large

language models) user feedback for 50 of the most popular mobile apps with millions of users, from Google Play Store, Reddit, and X. Building on top of my previous work, this research makes use of multiple "CrowdRE" user feedback sources and focuses on a broader non-functional requirement than privacy. Additionally, this work improves existing research by involving more data sources, different types of apps, and deeper qualitative analysis.

We borrow from Khalajzadeh *et al.*'s [158] definition of inclusiveness, namely *issues related to the inclusion, exclusion or discrimination toward* ***specific groups of users***. This study uses the definition and emphasizes **groups of users** when conducting the qualitative analysis and identifying instances of users expressing exclusion from using an app or feature due to a lack of support for their needs. Developing more insights regarding inclusiveness-related user feedback can assist software organizations in identifying critical issues that negatively impact diverse end users' ability to use an app whether it is due to age, location, or values.

Guided by the following research questions, my colleague and I first employed a Socio-technical grounded theory (STGT) approach [159] to manually analyze the user feedback. Then, I leveraged LLMs to automatically identify inclusiveness-related user feedback from the multiple sources and software apps.

RQ2.4 What are the different types of inclusiveness-related user feedback found on online sources?

RQ2.5 How does inclusiveness-related user feedback differ across different sources of feedback?

RQ2.6 How effective are large language models in automatically identifying inclusiveness-related user feedback?

RQ2.4 and **RQ2.5** were more closely related to my colleague's research work. Thus, for the purposes of this dissertation, I will briefly discuss those finding and focus more on **RQ2.6** which is related to my work.

We collected over 10 million posts and examined the inclusiveness-related user feedback both through qualitative analysis and by experimenting with large language models to automatically identify inclusiveness-related user feedback. The study provides the following contributions:

- A two-layer taxonomy of inclusiveness based on user feedback from 50 of the most popular apps in a variety of types of software. The taxonomy comprises of categories of inclusiveness concerns such as *algorithmic bias*, *technology*, *demography*, *accessibility* and *other human values*.
- Insights into the different inclusiveness concerns in different user feedback sources.
- Insights and empirical results from using large language models to automatically identify inclusiveness-related user feedback from multiple sources, which companies can leverage to address the inclusiveness concerns of their end users.
- A manually annotated dataset of inclusiveness-related user feedback that can facilitate future research and practice.

4.2.1 Motivation

Inclusive software is about designing with everyone in mind and considering the full range of human diversity [160]. Software systems are often built with a focus on functional or technical aspects as opposed to quality aspects [2], or requirements from diverse users [161]. Consider a situation where a user of a specific software application cannot access content simply due to the user's geographic location. This lack of consideration of requirements in software systems from diverse users negatively impacts end users and excludes them from adequately using applications [161]. Our study answers the call for further research into "inclusiveness" user feedback from previous literature [158]. In this study we borrow the previous literature's [158] definition of inclusiveness:

> "This category (inclusiveness) covers the issues related to the inclusion, exclusion or discrimination toward specific groups of users."

However, previous literature focused their analysis on app reviews and issue comments [161, 158], which missed out on other key sources of user feedback where end-users can voice their concerns regarding inclusiveness. In addition to app reviews, Reddit posts or X posts offer users a space to express their opinions. Reddit in particular offers end users an expanded commenting platform that is suitable for long-form discussions.

Figure 4.4 presents an example from Google Play Store where the user comments on the potential exclusion of the new dark mode feature and the limitations of it being a paid subscription. Similarly, Figure 4.5 shows an example from Reddit where a user describes their inability to chat with their friends on a popular communication app due to the lack of support for their cochlear implant. In Figure 4.6, a user comments that they are excluded from accessing Instagram music due to being in Singapore, a geographic location not supported by the application. In these examples, users describe potential perceptions of exclusions, ranging from location, accessibility, or device support.

The examples also exemplify how diverse end-users have inclusiveness issues that companies should take into account to ensure a more inclusive user experience. To further illustrate, consider an instance where an app does not support more than one language, in such case it would be excluding users who do not understand the provided language. Using the aforementioned inclusiveness definition from previous literature, we embarked on an STGT study that explored the inclusiveness user feedback for mobile applications from the perspective of end-users.

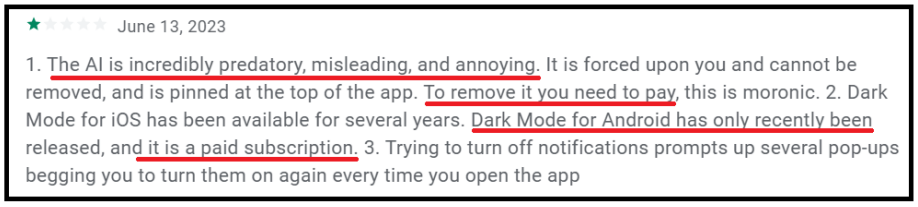


Figure 4.4: App review for Snapchat from Google Play Store. Underlined text indicates inclusiveness concern.

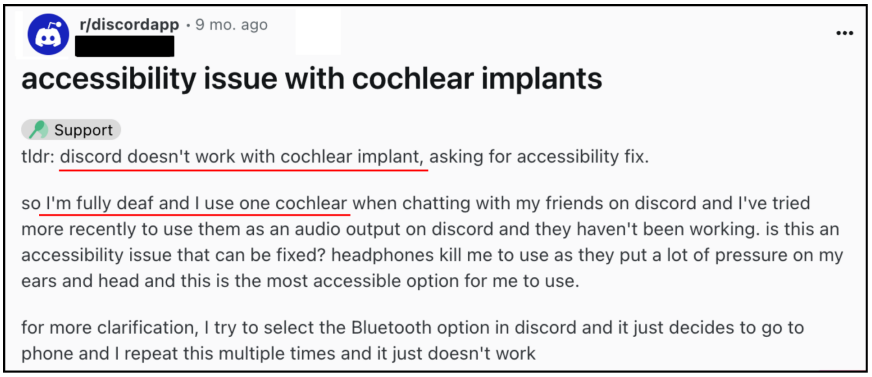


Figure 4.5: Reddit Post from Discord subreddit. Underlined text indicates inclusiveness concern.

Figure 4.6: Post on X from a user.

4.2.2 Methodology

We collected user feedback from three popular platforms and employed Socio-Technical Grounded Theory (STGT) [159] method. STGT facilitates software engineering research using modern research data, tools, and techniques such as mining publicly available software code repositories [159]. Our research methodology adheres to STGT data analysis steps including open coding, constant comparison, and basic memoing, which resulted in a taxonomy of inclusiveness-related user feedback.

Data Collection

We collected data from three popular online sources of user feedback: Reddit, Google Play Store, and X. We chose the three sources as prior studies have successfully found software relevant information, such as bugs and features from these channels [12, 14, 15]. As described in my previous study, Reddit is popular for having a high character limit that allows users to engage in elaborate discussions. Google Play Store offers app users to leave reviews about any app, which is useful for software organizations to elicit concerns regarding any particular app. In contrast, X is well known as one of the most popular social media platforms, which supports short form textual user discourse about any particular topic. X has been shown to provide requirement relevant information for organizations to analyze [63].

We collected a list of 50 of the most popular apps from Google Play Store. These apps have come from various domains and are actively installed by a diverse group of users from across the world. This list is used to scrape the data for Reddit and X as well, thereby giving us a unified range of apps. For Reddit, we use a publicly available dataset [162] and obtain over 380,000 Reddit posts. Next, we collect 9 million app reviews from Google Play using the library Google Play Scraper.[1] Lastly, we use the snscrape library[2] to scrape over 800,000 discussions from X.

Then we filtered the original data by removing empty posts and filtering out any

[1] `https://github.com/JoMingyu/google-play-scraper`
[2] `https://github.com/JustAnotherArchivist/snscrape`

post that has less than three words as posts that cannot satisfy this criterion most likely do not provide meaningful information. We were left with over 3.7 million app reviews, 824 thousand X posts, and 359 thousand Reddit posts.

Qualitative Analysis

Since the qualitative analysis part of the study was my colleagues research, I will briefly discuss it here. To analyze the data, my colleague and I followed the basic data analysis technique from Socio-Technical Grounded theory (STGT) [159], and which allows us to establish important categories or initial hypotheses from the user feedback. The STGT basic analysis step consist of open coding, constant comparison, and basic memoing. For the open coding, we randomly sampled feedback from the three sources for all 50 apps. To guide our open coding process, in line with STGT, we conducted a literature review to identify the existing understanding of inclusiveness in software engineering (as outlined in the Related Work section), and used the following definition of inclusiveness [158].

> "This category (inclusiveness) covers the issues related to the inclusion, exclusion or discrimination toward **specific groups** of users. It includes issues related to the age, gender, and socioeconomic status of the users."

As per the definition, we labeled any user feedback as inclusiveness when we found a user describing that they were unable to use an app or its features, and perceived that this was happening because they belonged to a specific group such as being a person with a disability, being from a specific location, or even having particular devices. My colleague and I analyzed the randomly sampled data and assigned an inclusiveness or misc (non-inclusiveness) label to each post.

When we label a post as inclusiveness, we then included a code based on the characteristics of the feedback. We assigned one code to each inclusiveness feedback because our goal was to identify the predominant inclusiveness concern in each post and understand the most pressing issues indicated by users. We employed the basic memoing technique to document the reflections on the emerging codes and categories. Therefore, the two human annotators continued the labeling process until no new categories emerged and code definitions became stable (i.e.saturation).

For Reddit and Google Play Store, we observed no additional insights emerging

from the last 200 posts. For X, due to the presence of irrelevant discussions in the data, saturation was reached after we coded 500 additional posts. In total, we labeled 4,420 Reddit posts, 4,962 from Google Play Store, and 13,500 from X. The analysis resulted in a two-layer *taxonomy of inclusiveness*, with five categories forming the primary layer and the 14 codes distributed within each category as a sub-category. The labeled data and memoing can be found in the replication package [163].

Automated Analysis

My work addressing **RQ2.6** encompassed analyzing the effectiveness of automatically identifying inclusiveness-related user feedback. Thus, I detail the methodology of the automated analysis in this section.

I experimented with GPT4o mini, which is one of the state of the art large language models. Using GPT4o mini,[3] we conducted binary classification: *inclusiveness* and *non-inclusiveness*. I selected GPT-4o Mini due to its balance between computational efficiency, cost, and performance. GPT-4o Mini is suitable for diverse data types and its cost-effectiveness and reduced computational requirements facilitate experimentation. Other studies have already explored leveraging GPT3.5 and GPT4 for similar text classification activities. For example, papers have investigated ChatGPT for stance detection of social media [164] and financial text classification [165]. Recall in Section 4.2.2 we collated a human annotated set of user feedback. This labeled set served as a ground truth for us to train and evaluate the model.

To assess the effectiveness of GPT4o mini, I implemented three different approaches.

- Zero shot Learning: This approach evaluates the model's ability to classify inclusiveness-related content without prior exposure to context specific examples.

- Few shot Learning: This approach is provided with 5 labeled examples, one from each category, to guide its classification process. This approach assesses how minimal contextual information can improve the model's understanding to identify inclusiveness issues.

- Fine tuning: This approach involves training GPT-4o Mini on a substantial subset of labeled data specific to our inclusiveness taxonomy. Fine-tuning aims

[3]https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

to adapt the model's parameters to our dataset, thereby improving its performance in detecting inclusiveness-related feedback.

I evaluated the three approaches on the same test dataset containing 1000 user feedback. The test dataset is imbalanced (5% inclusiveness, 95% non-inclusiveness). Since a total of 826 inclusiveness-related posts were identified among 22,882 manually labeled entries, comprising 3.6% of the dataset, the imbalanced test dataset aligns with the imbalance we identified in the manual label. For Zero-Shot learning and Few-Shot learning, I directly evaluated them with the test dataset. For Fine-Tuning, I first trained on a balanced dataset of 1,200 user feedback (50% inclusiveness, 50% non-inclusiveness) before evaluating on the test dataset. I prepared a balanced training dataset, as an imbalanced dataset can cause machine learning models to prioritize the major class and bias against the minor classes [166].

Due to space considerations, I did not include our GPT4o-mini prompts here in the dissertation, however, they are provided in the replication package [163]. I report the performance of the classifiers and use 3 widely used evaluation metrics: precision, recall, and F1-score. I compute the macro average score for all three metrics because the macro calculates scores independently for each class and then takes the average across all classes. Macro average treats all classes equally, regardless of their size. This makes it particularly useful for evaluating the model's performance on minority classes, as it does not favour the dominant or majority class.

In the sections that follow, I present the empirical results of our study.

4.2.3 RQ2.4: A Taxonomy of Inclusiveness

Answering the **RQ2.4** *What are the different types of inclusiveness-related user feedback found on online sources?*, our in-depth analysis identified a total of 1,211 inclusiveness-related posts: 712 from Reddit, 377 from Google Play Store, and 116 from X (Twitter). Using STGT we derived 5 categories and associated sub-categories of inclusiveness from the three sources. We integrated them into a *taxonomy of inclusiveness* from end-user feedback, as illustrated in Figure 4.7.

> **RQ2.4**: We find five major categories of inclusiveness, ranked in order of prevalence: *algorithmic bias*, *technology*, *demography*, *accessibility*, *other human values*, and which we present in the form of a taxonomy of inclusiveness.

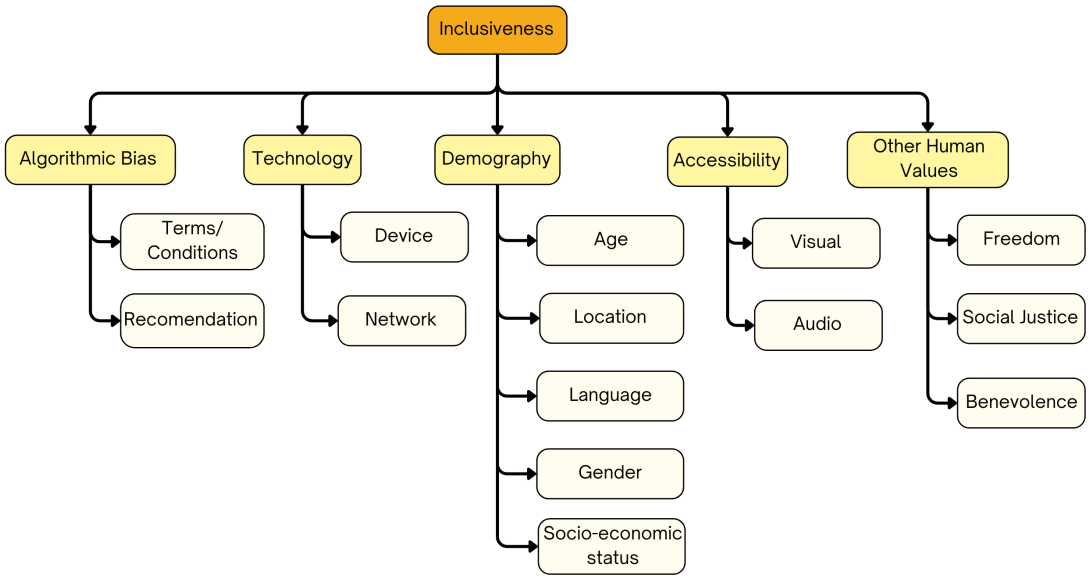


Figure 4.7: Taxonomy for inclusiveness-related user feedback from an analysis of Reddit, Google Play and X

4.2.4 RQ2.5: Inclusiveness across different Sources of User Feedback

To answer the **RQ2.5**, *"How does inclusiveness-related user feedback differ across different sources of feedback?"* we analyzed the distribution of the categories in our taxonomy across the three sources. Table 4.4 shows the distribution.

Table 4.4: Total number of inclusiveness-related user feedback in the 5 types of Apps from Reddit, Google Play Store, and X

Source	Technology	Algorithmic Bias	Demography	Other Human Values	Accessibility	Total
Reddit	124	102	92	50	111	479
Play Store	67	68	49	34	37	255
X	23	15	20	18	16	92
Total	214	185	161	102	164	826

> **RQ2.5**: Reddit contains more inclusiveness feedback in comparison to X and Google Play Store. Depending on the source of feedback, users express different

kinds of categories of inclusiveness feedback.

4.2.5 RQ2.6: Automated Identification of Inclusiveness user feedback

In answering the **RQ2.6**, *"How effective are large language models in automatically identifying inclusiveness-related user feedback?"* I assessed the effectiveness of using GPT4o-mini as the model was recently released by OpenAI and it supports fine-tuning, as detailed in Section 4.2.2. I used three different approaches on the top of the model for classification including zero shot, few shot, and fine-tuning, and measured the performance in terms of precision, recall, and F1-score. I use the macro version of the metrics as the data is imbalanced [167]. Since I focus on the class of *Inclusiveness*, hence, I consider the Recall of the *Inclusiveness* class as our main evaluation metric rather than the F1-score.

The results are outlined in Table 4.5. I found that the best overall macro average was for zero-shot, where the precision was 0.95, the recall was 0.80, and the F1-score was 0.86. Among the three approaches, zero-shot achieved the highest precision (0.92) for inclusiveness, however, this was achieved at the cost of a lowered recall (0.61) for inclusiveness. This result suggests that zero-shot learning has limited sensitivity to inclusiveness-related feedback. For few-shot learning, inclusiveness recalls improved to 0.72, indicating enhanced sensitivity to inclusiveness-related feedback, but at the trade-off of precision (0.45).

Finally, fine-tuning achieved the lowest macro average F1-score, however, it also achieved the highest inclusiveness-related recall at 0.78. This showed that the fine-tuned model could capture most inclusiveness-related feedback. The trade-off is that precision drops to 0.28, but I can identify more true positive inclusiveness feedback. Since the test dataset is heavily imbalanced towards non-inclusiveness feedback, the number of user feedback incorrectly labeled as inclusiveness by the model is low overall. One limitation of these results was that, given the imbalanced classes, the performance of the three models could be biased. However, the imbalanced nature of the test dataset was inspired by the imbalance observed from the manually labeled data, where inclusiveness-related posts accounted for only 3.6% of the data. Given that our purpose is to identify inclusiveness-related user feedback using large language models, our results indicate that the fine-tuning approach achieves the best

performance for this goal.

Table 4.5: Results of Classifying between Inclusiveness and Non-Inclusiveness using GPT4-o Mini

Source	Class	Precision	Recall	F1-Score
Zero shot	Inclusiveness	0.92	0.61	0.73
	Non-Inclusiveness	0.99	0.99	0.99
	Macro Avg.	0.95	0.80	0.86
Few shot	Inclusiveness	0.45	0.72	0.55
	Non-Inclusiveness	0.99	0.98	0.99
	Macro Avg.	0.72	0.85	0.77
Fine tuned	Inclusiveness	0.28	**0.78**	0.41
	Non-Inclusiveness	0.99	0.95	0.97
	Macro Avg.	0.64	0.86	0.69

> **RQ2.6**: Fine-tuning achieved the highest recall, which indicates it is the most proficient model for identifying inclusiveness-related user feedback, but at the trade-off of increased false positives. Whereas, zero shot and few shot learning have lower recall and are less proficient in identifying inclusiveness-related user feedback.

4.2.6 Threats to Validity

I describe several threats and mitigation strategies in our study using the total quality framework of Roller [120].

Credibility indicates "the completeness and accuracy associated with data gathering" [120]. This study may have the threat of sampling bias because we collected user feedback from 50 apps from the sources of feedback. However, we selected a diverse group of apps, and the feedback sources are also common platforms that users often use to discuss concerns. Our study also used standard web scraping libraries to collect the data. Additionally, we try limiting bias by creating a randomly sampled batch of user feedback to conduct manual annotation. We did not seek to give more weight to any particular app or source of feedback.

Analyzability refers to "completeness and accuracy related to the processing and verification of data" [120]. There is a potential limitation from the annotators misinterpreting implicit information from the data. To mitigate the threat, we analyzed

the data with two co-researchers who followed a social-technical grounded theory approach [159], where open coding, constant comparison, and memoing were used to analyze the feedback for inclusiveness. The co-authors were in constant dialogue during the coding process to ensure consistency and remove bias. Additionally, I only used GPT4o-mini for inclusive feedback identification. However, GPT4o-mini is one of the most recent models released by OpenAI.

Transparency refers to the "completeness of the final documents and the degree to which the research can be fully evaluated and its transferability" [120]. I provide extensive descriptions of the automated analysis methodology. I release the entirety of our data in our replication package, including our manually labeled dataset [163].

Usefulness specifies the "ability to do something of value with the research outcomes" [120]. Our study shed more insight into the role of inclusiveness in user feedback. More importantly, we advance the state of knowledge of inclusiveness by providing a taxonomy for the different types of inclusiveness-related discussions. In particular, our study encompasses a significant amount of user feedback and includes more empirical insights for organizations. We acknowledge that our results may not hold for every software app, depending on its functionality, but we believe organizations can benefit from the inclusiveness categories as they try to consider the concerns of diverse end users.

4.3 Conclusion

This chapter presented two studies that collectively show the potential of automated requirements analysis from textual user feedback. I explored how LLMs can support the detection and analysis of non-functional requirements, particularly privacy and inclusiveness.

In the first study, I analyzed over 4.5 million Reddit posts using machine learning techniques and classified privacy-related concerns. The second study shifted focus to inclusiveness. Using user feedback from Reddit, Google Play Store, and X, I demonstrated how LLMs such as GPT-4o Mini can be used to automatically identify inclusiveness concerns. Together, these studies emphasize that automated techniques can be used across platforms and different requirements, enabling broader organizational awareness of requirements relevant feedback.

The two studies provide empirical support for the claim that automated analysis of user feedback can provide organizations a usable and helpful way to analyze re-

quirements from online sources. Future work can build on these insights by refining classification models, expanding to less popular software products, and integrating additional metadata (e.g., upvotes, comments, user demographics). Ultimately, as the volume and complexity of user feedback continue to increase, automated tools will be paramount for practitioners and researchers. In the following section, I explore how automated development of requirements insights can be conducted from video-based user feedback.

Chapter 5

Experiments Towards Automating User Feedback from Video Based Sources

The prior chapter focused on analyzing textual user feedback, particularly from platforms like Reddit, to uncover requirements-related feedback. However, it is important to note that user discussions increasingly occur on video-based platforms such as TikTok and YouTube, where users post short-form product reviews to express their opinions and concerns. In this chapter, I examine whether requirements can be found in video-based sources with my **RQ3:** *How can we automate the development of new requirements from video-based user feedback?*.

Online videos are becoming more important for organizations to consider for user feedback, as videos provide an immersive experience for viewers. Videos are a very popular medium for social media and communication [20]. For example, TikTok is one of the world's most popular video-based social media platforms [20, 168] and YouTube has also grown to an astronomical magnitude [21].

Previous research on requirements and videos has been limited to investigating the comments section of videos while users engage in discussion [59, 60, 17]. However, videos are rich sources of data [169] with both audio and visual components and metadata (e.g., description, title, date created). In this study, I collaborated with two colleagues to look at all three sources: the audio track of the video (converted to a transcript), any text that appears in the video, such as captions and subtitles, and the metadata.

Paying attention to the direction of CrowdRE research is critical for companies to improve requirements elicitation [5, 8]. The ability to vastly increase the amount of feedback considered [9] is extremely valuable. The process proposed converts video content to requirements-relevant feedback, which can significantly impact companies' requirements and development activities.

I present a data-driven exploratory study on leveraging user-generated videos from TikTok and YouTube to identify requirements-related user feedback for 20 distinct products. This information can serve as a foundation for requirement elicitation, facilitating a more comprehensive understanding of consumer preferences and needs. My colleagues and I analyzed videos about products from a variety of industries, including software, consumer electronics, and automotive. My approach involves extracting textual data from audio and visual content from the videos and processing it using natural language processing (NLP) and machine learning (ML) techniques to uncover important user feedback that may not be captured through traditional elicitation methods.

The study contributes to the growing body of research on using social media as a data source for product development and user feedback analysis. It also provides insights into the strengths of using videos as a data source and the opportunities of applying NLP and ML techniques to analyze video data. The study was guided by two central research questions:

RQ3.1 How can video-based social media be used to identify requirements relevant user feedback?

RQ3.2 What are the main user feedback themes that we can identify?

For the purposes of this dissertation, I will elaborate **RQ3.1**, and will briefly describe the findings for **RQ3.2**, as **RQ3.2** was my colleague's research work.

5.1 Methodology

The study investigated the feasibility of using video-based social media platforms (i.e., TikTok and YouTube) for identifying requirements relevant user feedback themes. The methodology is summarized by Figure 5.1.

5.1.1 Data collection

My colleagues and I conducted extensive market research and analysis to identify the top-performing products in each industry to build a representative dataset of twenty different products from 4 common consumer categories: software, mobile phone, computer, and automotive. Table 5.1 shows the dataset characteristics. I chose the most widely used software across different domains, including browsers such as Chrome and Firefox, tutoring applications like Duolingo, networking platforms like Discord, and productivity software like Notion. For each of the other categories, I strived to pick products that were among the best selling flagship products in North America in 2022, with on focus videos that would be in English. I chose not to select products that may otherwise sell more units overall worldwide, such as Vivo versus Oneplus, but have a smaller English audience as they would impact the videos that I could collect. For example, for the products from the automotive industry, I chose 5 brands and their best selling model in North America in 2022. I applied the same approach to mobile phones and computers by selecting the most popular flagship product released in 2022.

To collect the data, I used the public facing APIs from TikTok and YouTube and scraped the videos by searching for each product using its name. The search term for each product is provided in the replication package [170]. I downloaded all the available videos from each search term and this process took about a day for each product. In total, I collected 11,341 videos, with 6,080 from TikTok and 5,261 from YouTube.

5.1.2 Preprocessing

The videos that I collected represented all the available videos according to my search term for each product. To ensure that my dataset is focused on user-generated content and not official promotional material, I implemented a two-level data filtration process. First, I filtered out any videos uploaded by official product accounts, as they are more likely to discuss product features in a promotional manner. The remaining videos in my dataset represented those that matched my search terms, but not from official product accounts such as Apple. Second, I filtered the videos to only include those in the English language. I used Spacy FastLang [171] to detect the language of the video description text, and OpenAI Whisper [172] to detect the language from the audio text, as described in detail in Section 5.1.3. I was left with 6276 videos

Table 5.1: Products used for the analysis

Category	**Products**	**TikTok Videos**	**YouTube Videos**
Software	Notion	280	232
	Duolingo	224	217
	Discord	94	103
	Chrome	82	105
	Firefox	50	189
Phone	Google Pixel 7	223	183
	Apple Iphone 14	178	142
	Samsung Galaxy S22	162	214
	Motorola Edge 30	76	92
	Oneplus 10	59	119
Computer	Microsoft Surface Pro 9	201	187
	Apple Macbook Air M2	193	161
	Asus Zenbook 14	119	132
	HP spectre x360 14	130	95
	Dell XPS 15	30	49
Automotive	Tesla Model 3	210	193
	BMW X5	190	197
	Ford F150	187	102
	Toyota Rav4	177	305
	Mercedes Benz GLC	154	239

after filtration.

5.1.3 Analysis of Videos

Videos contain audio tracks, metadata in the form of descriptions, and finally, text that appears in the video itself (such as a caption or subtitle). I used both the audio and visual text data extracted from the videos, as well as the descriptions provided by the content creators. To make sense of videos I worked with both the *visual* and *audio* elements of a video. First, I converted the audio of the video into text; secondly, I sampled visual frames and performed computer vision to collect any displayed text in a video (i.e., video subtitles). Out of the total of 6,276 videos in my dataset, 403 videos were found to have no audio content. Additionally, 101 videos did not contain any visual text. For each video I also paired this textual content with a video's metadata including video description and title where applicable. I then

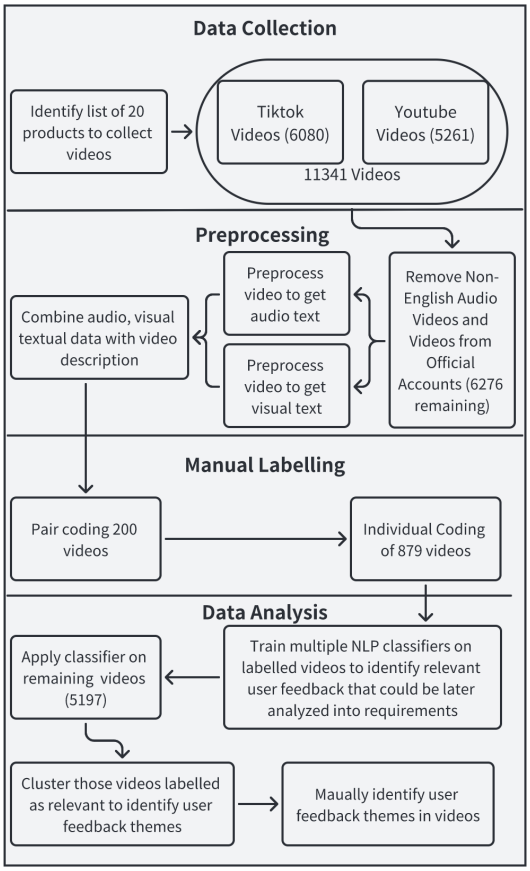


Figure 5.1: Research Process

classified the TikTok and YouTube videos using various state-of-the-art deep learning models as user feedback that could be later refined into requirements (referred to as "relevant") or user feedback that was not useful for later refinement into requirements ("irrelevant"). I describe these techniques below.

Extracting Text from the Audio

OpenAI's speech recognition model "Whisper" [172] was used to extract audio text from videos. The "Large" Whisper model used in this study is one of the most accurate models and is designed for high-quality transcription tasks.

The extracted audio from videos was run on the Whisper model to generate transcriptions of the audio content. On average each TikTok videos was less a minute in duration, where as the YouTube videos were roughly eight minutes in length. Hence, some YouTube videos were a little bit longer in length. To reduce computational

time, for the longer YouTube videos, I transcribe only the first thirty minutes of each audio and trim the rest. I assume that the premise of the video is conveyed in first 30 minutes, to minimize the processing time for longer videos. Whisper can detect the audio language being spoken during the transcription process and I utilized this to filter out videos that were not in English, as it is imperative to ensure the accuracy and relevance of the extracted text for my analysis.

Extracting Visual Text from the Video

In addition to audio, we also extracted visual text that may appear in a video as some content creators display subtitles or other important visual text. Since videos consist of many repetitive or similar frames, *motion-based video summarization* is used to select a small subset of all frames. For frames that are detected as having "text", we use an optical character recognition (OCR) system to extract the text; common spelling errors are corrected. The following section describes the video extraction pipeline.

This part of the study was conducted in collaboration with my colleague Dr. Amanda Dash. For candidate frame selection, we use a modified version of the algorithm proposed by Dash and Albu [173]. This algorithm was chosen because it is a heuristic and not a ML-based video summarization system, therefore it is independent of the video domain. Their approach integrates motion and saliency analysis with temporal slicing to extract features and unique candidate frames from the video.

We do not use their candidate frame summarization; instead, we leverage the information from the saliency energy map (instead of the slower background subtraction model) to find the probability divergence for the temporal slices. We take the Kullback-Leibler divergence, $D_{KL}(.)$, of each temporal slice, $k \in \{\text{vertical}, \text{horizonal}, \text{diagonal}\}$ at time $t-1$ and t, where t is defined as the current frame. We thus obtain a vector $s_t \in \mathbb{R}^3$ (Eqn. 5.1),

$$s_t^{(k)} = D_{KL}(p(k)_t || p(k)_{t-1}) \tag{5.1}$$

where $p(k)$ is the temporal slice k, normalized as a probability vector. The vector is then thresholded by values greater than $T_h \in \mathbb{R}^3$ to select the candidate frame (Eqn. 5.2); for this paper $T_h = [1e-4, 1e-4, 1e-4]$ is used.

$$\text{candidate frame} = \begin{cases} f_t & \forall k \in \{(s_t^{(k)} - s_{t-1}^{(k)}) > T_h\}, \\ \emptyset & \text{otherwise} \end{cases} \tag{5.2}$$

Intuitively, when the movement distribution changes significantly, a new candidate frame is selected. The candidate frames are analyzed for text using CentripetalText [174]. If no sufficient size text is detected, the candidate frame is discarded. In the next step, we consider two scenarios: (1) an audio track exists, and (2) no audio track exists.

When the primary content in a video is visual (i.e. no audio), we utilized Google's commercial state-of-the-art OCR system called "Google Cloud Vision"[1] to capture all text in the candidate frame. When audio is available, we assume the video content is primarily communicated by audio. Therefore, we supplement the audio by using HuggingFace's open-source OCR Tr-OCR [175] system with the "trocr-large-printed' pretrained weights to extract larger OCR text discovered by CentripetalText. Both OCR methods are not completely accurate, so we attempt to fix common spelling mistakes by processing the raw extract text with Peter Norvig's algorithm.[2] The choice to use multiple OCR algorithms was due to budget constraints.

5.1.4 Manual Labeling

To evaluate the accuracy and effectiveness of the classification models, we employed a manual labeling process to create a ground truth dataset for training. The dataset was randomly selected from my entire data pool, and we labeled a total of 1079 videos. My labeling process consisted of labeling videos as either "relevant" or "irrelevant".

My criteria to labeling a video as "relevant" include aspects such as problem reports, reviews of a product feature, comparison of features with competitors, feature requests, etc. Any time a video included content that could be used by a company to make informative decisions regarding changes (positive or negative) to their product, we labelled it as "relevant". For example, *"To find out what the safest browser to use in 2022 is based on empirical testing techniques So we re going to go through 200 of the latest malware links... Firefox only blocked 145... Chrome not quite as good as Edge it blocked 198 links out of 200..." (Firefox)* was labeled as relevant. In contrast, a video that do not describe a product in any meaningful way or in a superficial manner (i.e., "The new M2 MacBook Air is finally for sale. I'm not gonna buy one") was labelled as irrelevant. Exemplified by these quotes, the pair coders would label a video as "relevant" if the main point of the video or substantial part of the video

[1] https://cloud.google.com/vision
[2] https://norvig.com/spell-correct.html

content (i.e., several sentences with details) discusses the product in a way that a company could take actionable steps. In the case of short videos, where the total amount of content is just a few sentences, the threshold for labeling as "relevant" could be a single sentence, but the sentence would need to provide adequate details.

To prevent bias towards any specific product, we made sure to label videos from each product that we analyzed. Two of our authors with extensive experience in requirement analysis, pair coded a set of 200 videos for the manual labeling process. The pair coding process resulted an average Cohen's Kappa score of 87%, indicating high levels of agreement between the coders. This high inter rater reliability also indicated that the separation between "relevant" and "irrelevant" was quite clear. After the successful completion of the pair coding process, one author individually labeled the remaining 897 videos. Of the 1079 videos that were manually labeled, 601 were labeled as relevant and 478 were labeled as irrelevant.

5.1.5 Automated Analysis of User Feedback

Classification

I employed five state-of-the-art deep learning transformer-based models, namely GPT-2 (Generative Pre-trained Transformer 2) [172], BERT (Bidirectional Encoder Representations from Transformers) [70], RoBERTa (Robustly Optimized BERT Approach) [176], XLM-RoBERTa (Cross-lingual Language Model - Robustly Optimized BERT Approach) [177], and ALBERT (A Lite BERT) [178], to classify videos as either relevant or irrelevant. Fine-tuning these models allowed us to identify the most effective approach for video classification.

I evaluated the performance of these models using different combinations of data, including visual text, audio text, and both audio and visual text. Furthermore, I included title and description data for all combinations. By comparing the performance of these models, I aimed to identify the optimal approach for accurately classifying textual data from these popular video sharing platforms. For each of the five deep learning models (GPT-2, BERT, RoBERTa, XLM-RoBERTa, and ALBERT), I followed a similar training process. I used the pre-trained models and fine-tuned them on my dataset of labeled video text data, which included both the audio and visual text, as well as the video metadata (i.e., title and description). After training, I evaluated the performance of each model on a balanced test set of video text data. I measured the performance using accuracy and area under curve (AUC) metrics. I

repeated this process for each combination of data (visual text, audio text, and both audio and visual text) and for each platform (YouTube and TikTok) to compare the performance of the models on each type of data and platform.

Clustering

I clustered the data to learn user feedback themes. I used BERTopic [179] to infer document distribution over topics and then use BERTopic topic descriptions for clustering. BERTopic allows us to choose the cluster model. I selected K-means as my cluster model and ran the clustering process for 2 to 6 clusters. To determine the best cluster, I used the Silhouette Coefficient [142], a metric that measures how similar an object is to its cluster compared to other clusters.

Next, one of my colleagues conducted a manual analysis of the cluster for their research study which addressing **RQ3.2** The manual analysis resulted in the creation of themes for each cluster to represent their respective content.

Table 5.2: Results of Different Deep Learning Models on Classifying between Relevant vs Irrelevant. AUC is area under curve.

Dataset	Model	Accuracy	AUC
YouTube with only visual text	GPT-2	0.71	0.71
	BERT	0.76	0.76
	RoBERTa	0.74	0.74
	XLM-RoBERTa	0.67	0.67
	ALBERT	**0.79**	**0.79**
YouTube with only audio text	GPT-2	**0.94**	**0.94**
	BERT	0.86	0.86
	RoBERTa	0.86	0.86
	XLM-RoBERTa	0.83	0.83
	ALBERT	0.79	0.79
YouTube with both visual and audio text	GPT-2	**0.91**	**0.91**
	BERT	0.85	0.85
	RoBERTa	0.80	0.80
	XLM-RoBERTa	0.80	0.80
	ALBERT	0.79	0.79
TikTok with only visual text	GPT-2	**0.71**	**0.71**
	BERT	0.70	0.70
	RoBERTa	0.50	0.50
	XLM-RoBERTa	0.50	0.50
	ALBERT	0.70	0.70
TikTok with only audio text	GPT-2	0.92	0.92
	BERT 0.92	0.92	0.92
	RoBERTa	**0.93**	**0.93**
	XLM-RoBERTa	0.90	0.90
	ALBERT	0.90	0.90
TikTok with both visual and audio text	GPT-2	0.93	0.93
	BERT	0.95	0.95
	RoBERTa	**0.97**	**0.97**
	XLM-RoBERTa	0.90	0.90
	ALBERT	0.93	0.93

5.2 Findings

5.2.1 RQ3.1: How can video-based social media be used to identify requirements relevant user feedback?

Recall that for each video I 1) converted the audio track into text and 2) performed computer vision to collect any displayed text in a video (i.e., captions or video subtitles). I then classified the TikTok and YouTube videos using various state-of-the-art deep learning models as either "relevant" or "irrelevant"; The results of the classification using these techniques are summarized in Table 5.2. I observe that the datasets that leverage *audio* text paired with video metadata always performs extremely well. In particular, *audio* text paired with video metadata consistently performed better than *visual* text paired with video metadata. For YouTube videos and TikTok videos that utilize *audio* text paired with video metadata, accuracies of 94% and 93% were achieved.

I contrast these results with those datasets that leveraged visual text paired with video metadata. Table 5.2 shows that solely relying on text extracted from the video frames is not sufficient to identify requirements relevant user feedback. The most accurate model for classifying the dataset for YouTube's visual text was only able to achieve an accuracy equal to the worst performing model for the YouTube audio text dataset. TikTok videos using only video text and metadata is similar; 2 models had low accuracy of 50%, which for a balanced dataset means that it performs equal to a dummy model.

I surmise that the main reason for this is that audio extraction to text is quite accurate and as most videos include some host(s) speaking about the content, the audio text encapsulates the main idea of the video. In contrast, the visual text relies on sampling of visual frames to acquire the visual text, but this makes several assumptions 1) a video has clear subtitles that are easy to recognize 2) a video displays visual graphics of text that pertain to video's content. If a video's visual content had little visual text or did not include subtitles, a classifier had little to base decisions apart from accompanied metadata. While audio extraction to text suffers from potential limitations such as background music in place of a host's voice, the likelihood is lower that the existence of visual subtitles. Extracting text from audio also has less likelihood of encountering random audio that may confound the speech-to-text model.

Therefore, I found that datasets that utilized audio text performed better than datasets that utilized both audio and visual text. The only exception was "TikTok with both visual and audio text" where it actually performed 2% better than "TikTok with audio text." I believe the characteristics of TikTok videos (i.e., increased use of subtitles that complement the audio text of videos over YouTube) may be a factor for why the model could accurately classify "TikTok with both video and audio text". I expand on the effect of video content characteristics in Section **??**.

Findings 1: Text extraction from videos using audio was highly effective for classifying videos from YouTube and TikTok. Text extraction using video on its own was not effective. However, for TikTok, the combination of text extraction using both video and audio was the most accurate option.

GPT-2 and RoBERTa models were most accurate for classifying the videos. For YouTube videos with audio text, GPT-2 significantly outperformed the other four models by 8-15%. RoBERTa was the best performing classifier for two out of the three TikTok datasets. Roberta had the highest accuracy for "TikTok with audio text" and "TikTok with both audio and visual text" with respective accuracies of 93% and 97%. The 97% RoBERTa achieved for TikTok with both video and audio text was highest accuracy I obtained in all my tests in Table 5.2.

Findings 2: Deep learning models such as GPT-2 and RoBERTa can be utilized to perform classification of video content into relevant and non-relevant user feedback. GPT-2 was the most accurate model for classifying YouTube videos and RoBERTa was the most accurate model for classifying TikTok videos.

Table 5.3: Result from labeling and Classifying the Dataset

Dataset	Relevant	Irrelevant
YouTube Manual labeling	370	167
YouTube with audio text classification via GPT-2	1691	1029
TikTok Manual labeling	226	311
TikTok with both video and audio text classification via RoBERTa	810	1672
Total	3097	3179

After determining the most accurate models for TikTok and YouTube, I proceeded to classify the rest of the unlabelled dataset using these models. After classifying the rest, I also took a random sample of 50 videos with their classified labels and performed a round of manual annotation to determine the accuracy of the automated labeling.

I see consistency with the original experiments in Table 5.2 in the manual annotation. "YouTube with audio text" paired with GPT-2 achieved an accuracy of 98% and "TikTok with both video and audio text" paired with RoBERTa achieved 100%. I show in Table 5.3 the splits for *relevant* and *irrelevant* in the videos. In total, I found 3097 videos with relevant information and 3179 videos with non-relevant information for the 20 products in my study. YouTube videos (61%) had a higher concentration of requirements relevant videos compared to TikTok (34%).

Findings 3: Videos from YouTube and TikTok can be used to identify requirements relevant user feedback. Videos from YouTube (i.e., 61%) are more likely than videos from TikTok (i.e., 34%) to contain requirements relevant feedback.

5.2.2 RQ3.2: What are the main user feedback themes that we can identify?

As this section was part of my colleague's research, I will briefly highlight the findings. Table 5.4 presents the themes that were generated by reviewing the formed clusters and assigned theme names that represented each cluster accurately. These themes

can serve as a foundation for further refinement and analysis for a company to derive requirement statements.

Findings 4: User feedback themes can be generated from videos from YouTube and TikTok. These user feedback themes not only represent important aspects about products for companies to consider, but also represent relevant user feedback that companies can further refine into requirements in a subsequent step.

Table 5.4: Requirement Relevant Themes

Theme	Description	Number of Products (out of 20)
Feature Ratings	Praise/criticism of product features	20
Matching Competition	Comparison with other competitor products	13
Performance Ratings	Praise/criticism of performance of the products	8
Modifications Suggestions	Suggestions for tools/upgrade	5
Bug Report	Bugs and issues of products	4
Repair & Maintenance	Videos related to fixing and preserving	3
Design Ratings	Design evaluation	3
Affordability	Cost prospects of the products	3
Usage Tutorials	Tutorials for other user to help use the product	2

5.3 Discussion

My work indicates the potential to improve the practices of CrowdRE by utilizing valuable information present in video content.

5.3.1 Videos: A Source of Requirements Relevant User Feedback

TikTok and YouTube offer an interactive and immersive space for users to engage with the "crowd" through content creation. These platforms allow viewers to interact with the creator through likes, comments, shares, and reactions. In this work, I explore

how videos extracted from these two social media can be used to identify requirements relevant user feedback. For example, *"The new Duolingo update is seriously messing me up I can't even get back into the lessons I was actively working on. Please revert it... Goodbye Duo it was fun. ... Also note that this person has super Duolingo which means they pay for a subscription." (Duolingo on TikTok)* This exemplifies how a paying user is leaving the product due to a recently introduced update on Duolingo, which has resulted in a series of bugs on the platform. For Duolingo, if they considered such a video, the actionable requirement would entail reverting the update or fixing the bug to not impede the user's experience. An organization that seeks to reduce user attrition may benefit from these bug report insights, and utilize them to develop requirements that developers could implement.

Furthermore, people often discuss about the problems they encounter while using a certain product, *"Great phone some bugs Google Pixel 7 Pro... I've just had quite a few instances where things will just randomly freeze up like apps will just get stuck or I'll get stuck on just a black screen and can't get out of it... things won't always work all the time which is kind of frustrating..." (Google Pixel 7 on TikTok)* Although bug report are a common issue for products in textual feedback (e.g. forums, app reviews), videos from YouTube offer extremely rich details about issues [169].

The videos from YouTube and TikTok offer a level of detail into problems and issues that companies can reference to understand underlying problems. For example, an app review may just include textual description about an issue [12], a Reddit post may include textual description along with a screenshot, but a video may include a short clip about how a bug was triggered or the outcome of the bug that the company can re-watch when they are creating actionable requirements. Since these videos are quite popular on TikTok and YouTube, they may influence potential new and current users with the perceived honest and objective opinions. Therefore, my work highlights the greater importance that organizations should place on analyzing the video based content for requirements relevant user feedback.

Example use case: We present an example scenario regarding how organizations can leverage the approach proposed in this study. Duolingo is one of the premier language learning applications in the world. If they manage their online user feedback, they could search and download for Duolingo related videos from video based social media like YouTube and TikTok. Upon converting the audio text and visual text from each video, the organization would be left with a series of videos with their corresponding audio and visual text. The organization could then run a RoBERTa

model for all its TikTok video to identify the requirements relevant user feedback. Subsequently, the organization could use a topic modeling model, like BERTopic, to identify the various types of feedback.

If the organization wants to identify bugs, they could expect to find bug related topics from the topic modeling. For instance, there could be multiple videos covering the issues related to the Duolingo update. If the problematic update was covered by multiple videos, the organization would find a common theme that describes the issue. The next step for the organization would be eliciting actionable requirement(s), maybe in the form of user stories, to resolve the user concern. Eliciting the actionable requirement is outside the scope of this work, but an organization can extend my approach in a further step. The work left for a product person to create a user story for their issue tracker is quite straight forward, as they could already infer the type of issue (i.e., bug, feature request, etc) and the content from a video is generally quite explicit about the specific issue.

5.3.2 Implications for Practitioners

My findings suggest a number of implications for practitioners who can incorporate the user feedback from the video content as part of their requirement generation process. An organization may learn about how users are rating their products in terms of features, design, specifications, and performance through the video content itself. Software products like Chrome, Firefox, Notion, and Discord videos often contain feature and user experience-related discussion, which may be beneficial for organizations to consider before rolling out a new feature.

Products related to automotive companies have the potential to learn about user concerns related to repair, maintenance, and efficiency etc. Many of the videos further express the consumer's feedback about affordability, customization, and modification suggestions. One previous study has explored using YouTube videos for improving marketing and advertising purposes [180]; however, to the best of our knowledge, this is the first to explore the use of video-based user feedback to automate the development of new requirements.

The cost for an organization to adopt my approach is, for the most part, minimal. They would need to build a web scraping pipeline to download the videos from YouTube and TikTok, but once it is built, they can repeatedly use it. The main cost would likely come from the extraction of visual text as OCR extraction using third-

party subscriptions may be expensive, but there are alternative open-source tools that are available. Processing a software organization's sample batch of 100 TikTok videos would take approximately 15 minutes for audio text extraction and 1.5 hours for visual text extraction on 2 x Intel E5-2683 v4 Broadwell @ 2.1GHz and a P100 16G RAM.

One important note is the increased support for audio transcription these days from video-based platforms themselves and AI tools. Platforms like TikTok and YouTube support automatically generated captions [181]. Transcription tools like Otter.ai and Happyscribe also offer AI enhanced support for transcribing audio. Once an organization implements data analysis of user feedback similar to my approach, it can collect the main user feedback themes.

The final step would involve having an employee, such as a product manager or technical lead, to parse the user feedback themes into actionable requirements, but this should be a straightforward task. For example, in the sample content regarding Duolingo's flawed new update, a product manager can quickly see that, at minimum, the organization should roll back changes to a previous version. Otherwise, the organization should implement a bug fix to allow users to access the current lessons. Hence, the actual real dollars cost to an organization to use my approach is limited, and the organization could realize the benefits of obtaining user feedback themes based on insights from the crowd.

Users are engaging in video content creation on a regular basis, providing various feedback about the products. My study has the potential to influence industry requirements and product management practices, as practitioners can gain valuable insights about user behavior and concerns.

5.3.3 Implications for Research

I believe my research has several important implications for researchers and future studies. First, I identified the efficacy of leveraging videos from two large social media platforms for identifying product requirements relevant user feedback. Social media, particularly video-based social media platforms, has exploded in popularity in recent years, and their growth across different demographics provides new and important sources for CrowdRE. Other large platforms are joining this foray, with Meta's Instagram Posts and Reels and Twitter's Vertical being the significant alternatives. Future research should explore these alternative patterns and study whether these patterns

have characteristics that inhibit or enable relevant information for products.

Second, another area of future work is in the methodological domain. In my approach, I tried to focus on larger, more pronounced visual texts in videos as opposed to every possible text that may appear in a single frame, but future work should consider other approaches to analyzing the visual text. Other approaches may help to increase the usefulness of the visual text for identifying user feedback, especially on TikTok, where I noticed that there were more visual texts in general. Hence, there is also the potential for researchers to explore identifying other information from the visual aspect of videos. Potential areas may involve the automated interpretation of the visual content in a video and converting that into text.

While I tried to focus on user-generated content by filtering out official product account videos, it does not fully clear the dataset of potential sponsored or promotional types of videos. Future research could further separate this promotional content, perhaps through a classification filtering stage similar to the models used in my study, and explore the user-generated content in more detail. Additional work from research could include automatic detection of actual feature requests or bugs, as this could assist practitioners in identifying requirements from the videos.

Finally, future work can involve correlating video content with other accompanying characteristics in a video, such as the number of likes, the number of comments, as well as the content of those comments. Previous work has already explored the usefulness of video comments [17], but utilizing both the video itself and its accompanying data could prove even more effective for interpreting user feedback that may be refined into requirements through subsequent steps.

5.4 Threats to Validity

5.4.1 Construct Validity

Construct validity relates to whether I measured what I intended to measure. In my case, one threat relates to my manual labeling of “relevant” versus “irrelevant”. There could be subjective bias introduced in this labeling, but I tried to mitigate this through a definition of this concept from the literature and two coders who have in-depth experience in requirements concepts and pair coding. I used Cohen’s kappa and agreement levels as a measure of the reliability.

5.4.2 External Validity

In terms of external validity, there is the limitation that my study may not be generalized to other video platforms or other software products. However, I tried to mitigate this issue by studying two leading video based social media platforms and exploring 20 leading products from 4 major industries. Therefore, I anticipate that videos from other software products on TikTok or YouTube will produce similar results.

5.4.3 Internal Validity

My conclusions about the visual text extraction could be limited by my project budget in terms of optical character recognition (OCR) extraction. The "Google Vision" API had superior performance over the HuggingFace API, but the cost of the "Google Vision" API at $1.50 per 1000 frames was a constraining factor. Also, videos are high-redundancy media, which increases processing time, even with the algorithmic frame sampling I employed. Increasing the sampling rate will decrease the computation runtime, at the expense of the information-loss rate. I chose parameters to sample at a minimum rate of 1.5s/frame and 2.5s/frame for TikTok and YouTube, respectively. This may have caused missed frames containing pertinent information.

5.5 Conclusion

Video based social media platforms generate a wide range of discussions regarding different products, and analyzing these platforms have become a popular CrowdRE practice. In this study, I explored how I can leverage videos from two of the most popular video based social media platforms: TikTok and YouTube, for identifying requirements relevant to user feedback that may be refined later into requirements. As videos continue to gain popularity as a medium of communication and content creation, organizations can benefit from leveraging this data source to gain insights into user needs for their products.

While this chapter shows how online user feedback sources like videos can be used to analyze feedback, such approaches increasingly rely on automation tools. Specifically, since 2022, when novel generative AI tools such as ChatGPT were introduced, these tools have provided a new opportunity for practitioners to automate feedback analysis. These opportunities raise broader questions about how generative

AI technologies are being adopted and used in software organizations. In the following chapter, I examine how practitioners are adopting generative AI tools in software organizations.

Chapter 6

Understanding Adoption of LLM Based Tools

This chapter presents my investigation in addressing my **RQ4:** *What impacts generative AI adoption and use in software engineering?* This RQ helps to address the third research goal of my dissertation regarding how we can leverage generative AI tools to help with automating the development of new requirements from user feedback.

Recall from my earlier chapters about software organizations desiring more support for the automation of requirements analysis. In Chapter 3, one challenge identified by the software practitioners was that the tools available to them were difficult to use and unable to obtain their desired results for user feedback analysis. However, the research in Chapter 3 was conducted in 2021 to 2022 before the release of popular generative AI tools. The release of generative AI tools such as Copilot [182], ChatGPT [33], and Gemini [183] opened new opportunities that could revolutionize software engineering, ranging from tasks like code generation and user story writing. However, the sudden transition to leveraging generative AI tools had unknown ramifications, with one notable risk being human trust in generative AI output [30].

Thus, before exploring the use of generative AI tools for the development of new requirements from user feedback, which is my third and final research goal of this dissertation, it was important to first understand how generative AI tools such as ChatGPT are used and adopted in the broader software industry. This investigation can reveal common ways people use generative AI tools, along with the challenges they encounter, which can also apply to the development of new requirements from user feedback. As a foundation step, I explored how software practitioners make use

of generative AI tools using a socio-technical grounded theory [159] approach.

Early adoption numbers [184, 185] and empirical studies [29, 186, 27] have shown that generative AI tools are creating substantial value for software practitioners. Studies have focused on the usability aspects of AI assistance tools [187, 29]. Their primary emphasis is on how developers perceive and use AI programming assistants [187], as well as usability challenges that developers experience [29]. Industry surveys also tried to capture the nature of software developers' use, focusing on the specific tasks that developers most commonly rely on AI assistance [188, 189].

Despite these earlier works, there still lacks clarity on how practitioners are adopting and using AI tools from practitioners and their organizations. To fill this research gap, I was guided by an initial goal of gathering more understanding regarding: *What impacts generative AI adoption and use in SE?* In particular, I conducted 26 interviews with diverse software practitioners from a variety of development roles (e.g., software engineers, developers, DevOps, AI engineers, tech leads, etc.), organizational sizes, and geographic locations. I developed a theory of motives and challenges for adopting and using AI tools from both an individual and an organizational level, which provides insights that requirements practitioners can consider to improve generative AI tool use in practice.

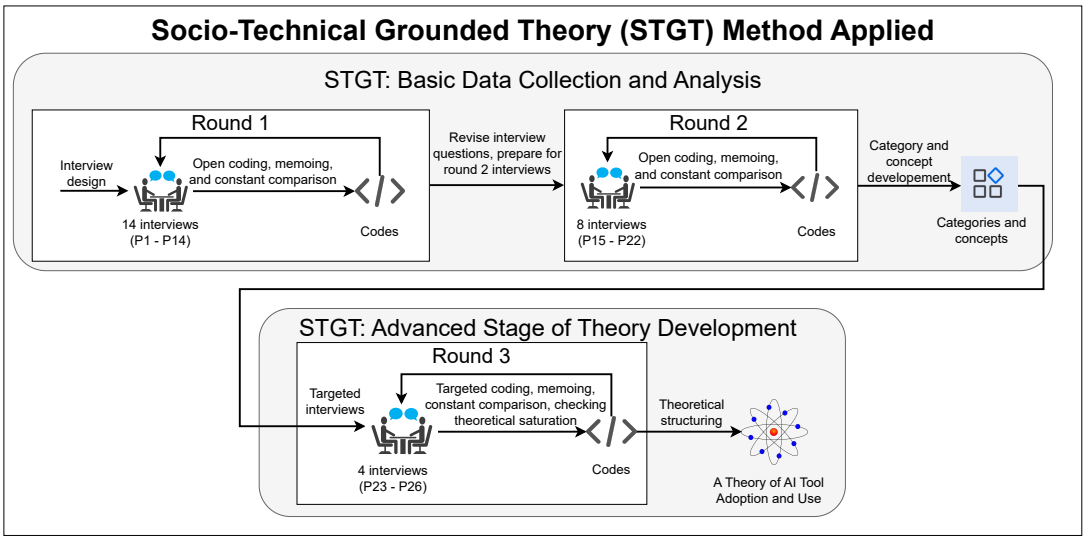


Figure 6.1: The Methodology of the Study

6.1 Methodology

Socio-Technical Grounded Theory (STGT) is ideal for studies like ours that involve data collected through interviews, theory development, practice, industry-relevant topics, and the human and social aspects of software engineering [159]. Since its inception, STGT has been a commonly used research method for software engineering research due to its suitability for technology intensive domains [190, 191, 192, 193, 194, 195]. Moreover, STGT is best suited for those studies that aim to develop deep understandings through qualitative data [190, 191, 195].

STGT is aptly suited for this research topic, since I aim to study the broad emerging area of AI tool usage and adoption by practitioners and reduce challenges in leveraging AI tools in their organizations.

Figure 6.1 illustrates the methodology of the study. I conducted 26 interviews in 3 rounds. Adhering to the STGT [159] process, the basic data collection involved interviews in two rounds: (1) round 1 with P1 to P14, (2) round 2 with P15 to P22. The subsequent advanced stage of theory development involved targeted interviews with 4 participants.

6.1.1 STGT: Basic Stage of Data Collection and Analysis

The basic stage starts with a lightweight review of the existing literature, known as a lean literature review in STGT [159]. I chose semi-structured interviews as the primary data collection approach. The first 14 interviews revealed new insights related to several aspects, such as company culture, data privacy, and company guidelines surrounding AI tool(s) usage. In the subsequent 8 interviews, I narrowed the questions to these aspects of AI tool use and adoption. Upon analyzing the data from the second round, the categories and relationships became more prominent, highlighting motives and challenges influencing the AI tool use and adoption. Then I embarked on the next stage, i.e., the Advanced stage.

Interview Design

My initial literature review helped to partially design the interview questions. In the literature review, I found survey studies related to the usability of the AI tools and industry-oriented surveys. For example, the study by Liang et al., [196] surveyed developers on topics such as *why and how often AI tool(s) are used, strategies used to*

make these tools work better, and why developers give up on using the AI-generated code.

I found another study conducted by Stack Overflow [188], where they surveyed practitioners about their perceptions of AI tool(s) and how these tools may or may not impact their workflows. The study revealed that practitioners use AI tool(s) throughout the various phases of Software Development Life Cycle (SDLC) for tasks like coding, debugging, documentation, learning about codebase, code commit and review, testing, as well as deployment and monitoring. Thus, I developed the semi-structured interview questions by building on top of these studies and following the general interview guide [97], as it is widely used by researchers [197, 198, 199, 200].

I prepared 40 broad base questions[1], including questions such as *1) How do you decide which AI tools to use? 2) How does using AI tools impact work assignments? 3) How do you apply these tools to your workflow?* I asked *how AI tools impact work assignments* because previous literature indicated that developers *saved time* from using AI tools. Hence, I was interested to see if these time improvements resulted in a difference in how practitioners assigned tasks and divided work.

Benefiting from the careful interview design, I was able to find new insights about potential influences on AI tool use and adoption, such as **culture of sharing insights with peers**. For example, from P7's interview I found that they share prompting techniques in their company Slack channel, *"I use a company Slack channel, so we have several channels,* ***sometimes we just share, how do you prompt*** *like best way to prompt some research papers? Like React prompting." (P7)* I also found discussions related to data privacy, peer judgment, lack of guidelines, and more.

These topics emerged as motives and challenges that influenced the AI tool use and adoption in the industry. The semi-structured nature of the interviews enabled me to unveil these findings, as it allowed me to ask probing questions and delve deeper into the emerging topics. Furthermore, the STGT methodology emphasizes iterative and incremental research, so I was able to adjust the interview questions in subsequent interviews and ask questions that are typically not supported in a survey-based approach. Hence, I updated the questions to the emerging topics and added 15 new questions. The new set included questions like *"Is there a culture of sharing practices or strategies for using AI tools within your company?"*, as I found many interviewees highlighting that talking to peers often taught new prompting methods and tips to try on the AI tools. A subset of the new questions also included: *1) Does*

[1] `https://zenodo.org/records/12165737`

your company provide any guidelines and restrictions on using AI tools? 2) Ever since the new AI tools have come out, how has the company culture changed? The full set of interview questions is included in the replication package.[2]

Sampling and Recruitment:

I began participant recruitment by using convenience sampling [201] and invited industry practitioners from personal contacts to participate in the semi-structured interviews. The initial selection criteria included any software practitioner who uses AI tool(s) for tasks related to software development. I expanded the reach through recommendations from previous interviewees.

To mitigate bias, I aimed to interview practitioners from diverse company sizes, years of experience, and countries to help me understand the overall worldwide usage and adoption of the AI tool(s). In total, I interviewed 26 participants from Asia, Europe, North America, Oceania, and South America, whose experience in the software industry varied from 1 to 15 years. Their companies also vary in size, with small, medium, large, and extra-large-sized. Table 6.1 shows the demographic information of the participants.

Collection and Analysis Procedure:

In the first round of interviews, I interviewed 14 participants (P1 - P14). Each interview was conducted via Zoom and lasted approximately 45-60 minutes. At the beginning of each interview, I informed the participants that their interview would be recorded for transcription purposes and would remain confidential. To analyze the transcripts three authors used open coding to inductively assign codes [92]. As new codes emerged I employed a constant comparison method and regularly met to discuss and compare the derived codes. I further prepared memos for each interview to reflect on the key points of the interview and draw connections between the codes. The codes from the 14 interviews pointed me toward new topics, e.g., discussion with peers, data privacy, peer judgment, lack of guidelines, and more. Hence, I updated the questions accordingly using these insights and narrowed the focus towards the aspects that influence AI tool use and adoption.

At this stage, I applied theoretical sampling which is the process of narrowing the focus of the data collection to the main findings that emerged from the analysis

[2] `https://zenodo.org/records/12165737`

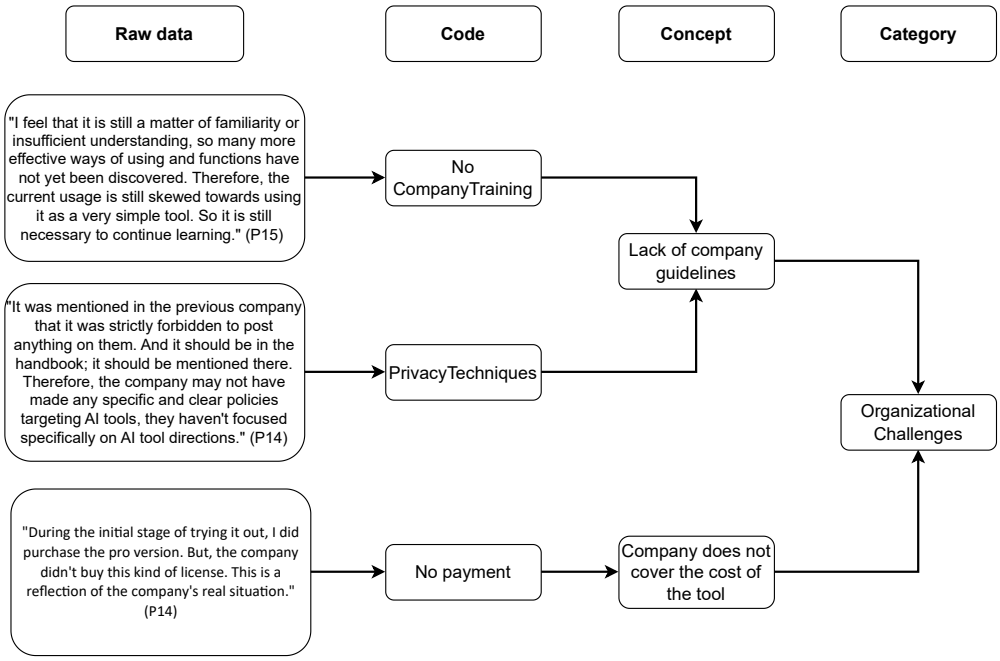


Figure 6.2: Example Coding of Raw Quotes

process. The process of theoretical sampling also involves updating the interview questions based on the emerging concepts, categories, and relationships, rather than keeping them broad as previous interviews [159]. Thus, I conduct round 2 of the interviews which involved 8 participants (P15 - P22) using the updated questions. I continued open coding and memoing the interviews. From the 22 interview transcripts, I first generated 48 codes. Then, I categorized the 48 codes into 12 concepts. I refer to these 12 concepts as motives and challenges *factors* that influenced the practitioners' AI tool use and adoption. For example, *discussion with peers* motivated a practitioner to use AI tool(s) during their work. Whereas, a *lack of guidelines* in the company made it challenging for them to use AI tools, as they were unsure about how much information they could put into the tool. I show an example of applying STGT analysis from raw quote to category in Figure 6.2.

During the last 3 interviews of round 2, no new insights emerged, and I found detailed and prominent concepts. Thus, I reached the end of the Basic Stage. However, by the end of the Basic Stage of Data Collection and Analysis, I still lacked a full theory. Therefore, in the next stage, I proceeded to the Advanced Stage of Theory Development with the "Emergent Mode" [159]. The STGT guidelines [159] defines the Emergent Mode as,

Table 6.1: Interviewee Demographics

P#	Role	Exp	Size	Continent
P1	DevOps	2	M	North America
P2	Senior Dev.	6	L	Asia
P3	Developer	3	S	Asia
P4	Senior Dev.	6	M	Asia
P5	Senior Fullstack	8	M	North America
P6	DevOps	3	M	North America
P7	Developer	3	L	Europe
P8	Sw. Engineer	4	L	North America
P9	Sw. Engineer	2	L	North America
P10	Sw. Enginerr	5	S	Oceania
P11	Applied Scientist	7	XL	North America
P12	CTO	5	S	North America
P13	AI Engineer	3	XL	Asia
P14	Tech Lead	15	XL	Asia
P15	Tech Lead	7	XL	Asia
P16	AI Engineer	7	L	Asia
P17	AI Engineer	10	XL	Asia
P18	Tech Lead	10	XL	Asia
P19	Sw. Architect	5	L	Europe
P20	Sw. Engineer	4	XL	North America
P21	Sw. Engineer	5	M	Europe
P22	Sw. Engineer	3	L	North America
P23	Sw. Engineer	3	L	North America
P24	Sw. Engineer	6	M	South America
P25	Sw. Engineer	5	L	South America
P26	AI Engineer	2	M	North America

Organizational size: S (fewer than 50 employees), M (between 50 and 249 employees), L (between 250 and 4999 employees), and XL (5000 employees and more).

"Enabling the emergence of theory through the iterative targeted data collection and analysis step which ends with theoretical saturation and results in a mature theory."

6.1.2 STGT: Advanced Stage of Theory Development

I conducted two steps in this advanced stage: 1) *targeted data collection and analysis*, and 2) *theoretical structuring*. Targeted data collection and analysis involves first targeting data sources (i.e., interviewees) to help strengthen the concepts. The analysis involves targeted coding and constant comparison, which refers to coding the most

important concepts from the Basic Stage.

Targeted Data Collection and Analysis:

To implement targeted data collection, I conducted a 3rd round of interview with 4 participants (P23 - P26). In round 3 of the interviews, I focused on strengthening the identified motives and challenges influencing AI tool use and adoption. I interviewed 4 practitioners, 3 software engineers and 1 AI engineer. These interviews lasted 30-40 minutes, slightly shorter than the previous round, as the questions at this stage were more targeted and focused.

Theoretical Structuring:

The last 4 interviews strengthened the 12 concepts, indicating that I reached theoretical saturation [159]. To structure the concepts, I decided to leverage visualization techniques using diagramming software. From the visualization, I found that the 12 concepts generally fall into four categories: individual motives, individual challenges, organizational motives, and organizational challenges. I further found a push-pull relationship among the concepts and categories which I denote using double sided arrows. I present the concepts, categories, and the push-pull relationship in Figure 6.3. The 4 motives and challenges and relationships together influence the use and adoption of AI tool(s) which I present as the "A Theory of AI Tool Use and Adoption for Software Development". In section 6.2, I provide an in-depth explanation of the theory with example quotes from the interviewees.

6.2 A Theory of AI Tool Use and Adoption for Software Development

This section details the theory (Figure 6.3) on the individual and organizational motives and challenges that impact the use and adoption AI tool(s). The theory encompasses the motives (i.e., pull) and challenges (i.e., push). In Table 6.2 and Table 6.3. For Tables 6.2 and 6.3, I also provide finer grained details for example *IM1 Learning and debugging* contain additional details: IM1.1 learning about new code, programming languages, or concepts easier, IM1.2 debugging errors or problems easier, IM1.3 searching new code, programming languages or concepts easier.

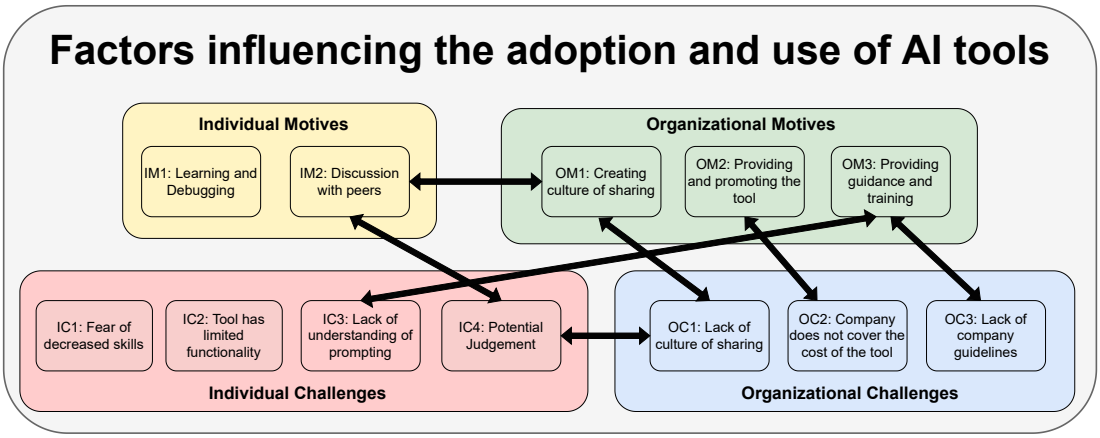


Figure 6.3: A Theory of AI Tool Use and Adoption in Software Engineering
I use double sided arrows to make connections between concepts to show there is a relationship between the two or more concepts.

6.2.1 Individual Motives

IM1: Learning and Debugging:

AI tools improving practitioners' ability to learn and debug were widely reported by the interviewees. Prior to AI tools, when writing new code, practitioners relied on a combination of personal knowledge, Stack Overflow, documentation, and Googling to reach their conclusion [202]. However, the participants reported that AI tools such as Copilot and ChatGPT lowered their cost of *time* to generate code, particularly boilerplate or repetitive code [29]. A previous survey study also briefly highlighted that practitioners use AI tools for learning, searching, and debugging [203]. However, the study failed to discuss *how the use* of AI tools impacts these tasks.

Through using AI tools, almost all of the interviewees reported perceived *time* savings in their work between 5% to 50%. Admittedly, since virtually none of the interviewees adopted a systematic metric to measure their work allocation and productivity, I relied on their self reported time savings. To my surprise, software developers reported low time savings with a range between 5%-25%. In contrast, people in more AI or research roles such as *P11, P17, P26* reported saving upwards of 50% in their code related work. My initial assumption was that since there was a plethora of training data on backend and frontend code that software engineer's work would be significantly simplified, but due to restrictions such as privacy, which I cover later, the time savings is less than I assumed.

Specifically, 13 interviewees *(P1, P2, P3, P4, P5, P10, P12, P15, P16, P20, P21,*

Table 6.2: Motives that increase AI tool use and adoption

ID	Motives
	Individual Motives
IM1	**Learning and debugging**
IM1.1	I realized AI tool(s) make learning about new code, programming languages, or concepts easier, my AI tool(s) usage has increased
IM1.2	I realized AI tool(s) make debugging my errors or problems easier
IM1.3	I realized AI tool(s) make searching about new code, programming languages, or concepts easier
IM1.4	I realized AI tool(s) reduce my use of tool(s) such as Google and Stack Overflow to help debug or learn
IM2	**Discussion with peers**
IM2.1	I started having regular discussions about our use of AI tool(s) (e.g., best practices for prompting in AI tool(s)) within the team and/or the company
IM2.2	The employee wants to improve their AI usage skill by sharing
	Organizational Motivates
OM1	**Creating culture of sharing**
OM1.1	As company created a space for employees to openly discuss about the use of AI tool(s) (e.g., events, slack channels)
OM1.2	The employee's team has a strong culture of sharing insights
OM1.3	The employee's company has a strong culture of sharing insights
OM2	**Providing and promoting the tool**
OM2.1	As company paid for the use of AI tool (e.g., provided subscription)
OM2.2	Because company provides AI tool plugin that is powered by external APIs (e.g., GPT4)
OM2.3	Company started notifying us on the availability of AI tool(s) (e.g., announced in a meeting)
OM2.4	Company started motivating us on the use of AI tool(s) (i.e., sending us email reminders)
OM2.5	Company mandated us to update our manager about our use of AI tool(s)
OM3	**Providing guidance and training**
OM3.1	Once company started providing training on prompting AI tool(s)
OM3.2	After the company provided a policy with rules and restrictions of AI tool(s) usage
OM3.3	Company provides an integration/tool that helps sanitize my prompt to protect company specific data

P22, P24) admitted to using AI tool(s) to help learn a new or existing piece of code. P12 recollects how they dealt with an assortment of unfamiliar technologies, such as dealing with cloud infrastructure while wanting to release a prototype as soon as possible. *"What's the first step to creating this service? I don't know how, so I ask ChatGPT, and it tells me I could use FastAPI. "But how do I get it online?" Then I asked, "how to do that", it says to use EC2, so I just hammered away at it, right? And then I got it up and running." (P12)* Before the introduction of AI tools, P12 had to expend considerable resources reading and studying manuals regarding these technologies.

Similarly, relying on AI tools to debug code errors was widely reported by the interviewees *(P1, P3, P5, P6, P9, P11-P15, P17, P20, P21, P23, P25, P26).* Before

Table 6.3: Challenges that limited AI tool use and adoption

ID	Challenges
	Individual Challenges
IC1	**Fear of decreased skills**
IC1.1	A fear of a decreased coding ability
IC1.2	A fear of losing learning ability
IC1.3	A fear of over relying on AI tool(s)
IC2	**Limited AI Capabilities for Software Development**
IC2.1	Tool does not have operation environment information (e.g., versioning, hardware)
IC2.2	Tool does not have access to codebase and is unable to generate the correct results
IC2.3	AI tool often refuses to provide a response for my question (e.g., "As a language model, I cannot answer your question")
IC3	**Lack of prompting skill**
IC3.1	Due to AI tool(s) as it needs a lot of experimentation before getting a desired result
IC3.2	Tool getting me into an infinite loop of prompting and diverging from the actual task
IC4	**Potential judgment**
IC4.1	Due to potential judgement from peers
	Organizational Challenges
OC1	**Lack of culture of sharing**
OC1.1	Everyone does different work, do not have similar project context
OC1.2	There is no culture of sharing within my team and/or company
OC1.3	Fear of potential judgment from others when sharing
OC2	**Company does not cover the cost of the tool**
OC2.1	Due the high prices and you have to pay for it
OC2.2	Number of API calls allowed in a time interval (e.g., 40 prompts/4 hours on ChatGPT)
OC3	**Lack of company guideline**
OC3.1	Due to company not providing any usage guideline
OC3.2	Due to company not providing training
OC3.3	Due to concerns about accidentally leaking company data

the use of AI tools, practitioners Googled stack traces or other messages to determine potential causes for issues [204, 205, 206]. With the ability of AI tools that allow uploading screenshots, interviewees explained that sifting through Stack Overflow pages is rather unnecessary. *"The advantage is that I don't have to spend time sifting through Stack Overflow to see which post actually resembles my problem. Often on Stack Overflow, even though many problems seem similar, they are not exactly the same." (P19)*

IM2: Discussion with peers:

Practitioners in the study described that a key motive for increasing the use of AI tools is taking on a personal initiative in sharing experiences and successes with AI

tools with others in their organization. 16 interviewees mentioned having experienced discussions with peers about using AI tool(s) agreed that their AI tool usage increased after having regular discussions about their use of tool(s) within the team and/or the organization.

One reason team discussions were helpful is that, although AI tools are popular, many colleagues were still unaware of or had not tried them. As described by Geoffrey Moore [207], early innovators and early adopters of emerging technologies and products represent only a segment of the overall population. Hence, the participants recounted examples of colleagues who emerged as strong proponents of AI tools as a result of discussions with colleagues. *"A teammate from our group could not figure out [a problem] after a day.* ***So I told him to try ChatGPT and it helped him write the function. He shared about [this incident] in our big group chat, saying, "Look at this, it's amazing, I was able to write such a function."****" (P15).* This exemplifies the organic spread of AI tool use and adoption driven by positive user experience and peer recommendations.

Another reason that discussion is beneficial is that it facilitates a debate of ideas and insights, and practitioners could improve their skills from this discussion. These discussion sessions allowed practitioners to *learn* about how others in the organization were using AI tools and tips to achieve the best prompts. *"I've done a few presentations ourselves everybody has tried and* ***we wanted to present what we have tried doing with AI*** *... I think everybody's curious. They want to try different things." (P23)*

The interviewees also told me about how tips are shared and others within the team benefit from the collective knowledge base. *"I talk about it during team meetings to figure out what has or has not been working for us. I do talk about what kind of tool(s) we are using, what kind of processes need to be we need to be using to help us do our work better." (P9)* These meetings also facilitate discussions about tips, such as *"ideas about the best way to prompt ChatGPT to get the kind of information you want." (P20)*

6.2.2 Individual Challenges

IC1: Fear of decreased skills:

A variety of fears related to the use of AI tool(s) were articulated by participants, including 9 interviewees *(P1, P2, P4, P11, P12, P14, P17, P19, and P20).* Although

AI tools assist in boosting learning and debugging, which lead to perceived speed ups in time, participants also reported fears that hinder the use and adoption of AI tools.

Despite the ability for AI tools to generate working code, P2 and P1 stress the importance of understanding the generated code *"If you're copying code from the ChatGPT, then you have to understand and you have to think about it.* ***I am using ChatGPT or other tool(s) for increasing our productivity, but not damaging our thinking, or our creativity****." (P2) "****I don't really like the idea of going to chatgpt, copying a bunch of code that somebody else implemented in a repository.*** *Even even with stuff prompts that I am entering, I try to think twice." (P1)*

Although AI tools offer access to rapid generation of potential solutions to various software development challenges, the interviewees described that over-reliance could lead to a knowledge gap, affecting the developers' ability to grow and adapt. *"If they are juniors, if they can explore the codebase, they will learn more. ...* ***there will be some knowledge gap about codebase for a developer who did not use ChatGPT and a developer who, who is using ChatGPT from the beginning of his software engineering career****." (P4)*

On a personal level, the participants described the risk of decreased creativity and critical thinking as a result of over-reliance. *"****When we use ChatGPT people don't want to think for themselves anymore at all****... **Because over time, if you're not engaging your mind, you're engaging yourself in thinking about things, you become less creative, and you become more dependable on these tool(s).**" (P20)*

IC2: Limited AI Capabilities for Software Development:

One of the major challenges of AI tools that impedes use and adoption is their limited capabilities in specific instances, as mentioned by the interviewees. Nearly all participants aspired for more functionalities, but 14 of them *(P1, P5, P9-P12, P14, P17, P20-P22, P23, P25, P26)* explicitly desired additional improvements such as access to the operational environment and codebase.

The interviewees highlighted that the lack of awareness of the operational environment, including specific software versions, hardware configurations, or the nuances of the programming languages in use often becomes a bottleneck. This leads to AI generated responses that may be incompatible with the developer's actual working

environment. P11 explains this issue, *" Current code can only understand the code, but it cannot comprehend the underlying running environment. I think this is the biggest problem. But if it can see what the* ***environment is like****, I think it could go to another level." (P11)* Prior to AI tools, developers would have to dig through Google or Stack Overflow and hope to find an answer with a compatible environment. It is easier now for AI to generate a "correct" sounding answer. As P1 described, *[AI response] is just sort of like a quality distillation of averages of what other people have already agreed upon in some sort of a public context.*

The participants also described that tool(s) most times do not have access to a user's codebase. This limitation hampers the ability to generate context-aware solutions or debug effectively. *"It doesn't have the amount domain knowledge that my team member would have. [It] just has general knowledge that the internet has." (P9)* The limitation also translates into debugging due to a lack of context. *"It's just really unreliable, you usually have to give it the full context. It is a lot of context you [have to] give it or it just doesn't know what to do." (P8)* To workaround this limitation requires detailing the codebase and context, but this may also lead to concerns about privacy. As I will describe in Subsection 6.2.4, data privacy could have significant revenue and legal ramifications.

The impact of limited AI capabilities is even more noticeable for software organizations working in less common domains or programming languages. *"As you get more and more into specific problems in your domain, it's less helpful." (P23) "I use a test framework called [removed], and it's a bit older, and very unintuitive and unfriendly... But [AI tools] hasn't really been that successful. I wonder if it's because [removed] is not that popular and it didn't really train on that data a lot." (P5)* For the interviewees who use less popular programming languages, they all indicated that their desired improvement is adding language support for their domain.

IC3: Lack of prompting skill:

Another notable challenge identified by 9 participants *(P1, P3, P5, P6, P14-P16, P19, P21)* revolves around the difficulty in crafting the right prompts. The process of finding the right prompt often involves trial and error, which can be time consuming.

As P21 described in their case, *"****I've still been struggling with what kind of prompts to use, how to do it more efficiently****... I feel like for different tasks, it's better to do different prompts." (P21)* Or, as P15 describes, they have a lack of

familiarity and understanding of using the AI tool(s). *"**I feel that it is still a matter of familiarity or insufficient understanding**, so many more effective ways of using and functions have not yet been discovered." (P15)*

The effectiveness of a prompt can depend on the user's experience and familiarity with the domain. *"When you have worked for a certain number of years and are in a certain field, I think you have that knowledge on what to write what the prompt should be." (P3)* These results indicated that senior practitioners are more proficient in crafting the correct prompt and asking the relevant questions to reach their desired solution.

IC4: Potential judgment:

I found potential judgment for using AI tool as a concern among some of the participants *(P4, P5, P14, P22, P24, P25).*

My interviewees indicated that a reliance on AI could be interpreted as a lack of skill, leading to a reluctance to discuss AI tool use openly, even when it might enhance productivity or creativity. The existence of *imposter syndrome* (i.e., "person's experience of feeling like an intellectual phony despite having outstanding academic and professional credentials" [208]) has been often experienced by women [209] and other minorities [208], but can also impact junior developers and other practitioners with less industry experience. *"**I almost don't want my coworkers to know.** And not wanting my coworkers to know I use it for weird reasons. Like **it makes me less of a programmer or something.** " (P5)* A fear of judgment exists in junior developers who may worry about being judged by their senior colleagues for not being proficient enough to code without AI assistance. There may be a perception that using AI indicates incompetence or laziness. *"I guess [me and my teammates], we are not seniors yet... With us, we don't have that [negative] perception. But with other teams and seniors, they have this kind of negative perception." (P25)*

The negative connotations lead to interviewees reporting that they are unwilling to share or admit to using AI tool(s) as they may feel embarrassed. *"It seems like it could really be possible [people are embarrassed to tell others they use AI tool(s)], which is why I've observed the proportion is so low..." (P14)*

6.2.3 Organizational Motives

OM1: Creating culture of sharing:

From the interviews, I found that core to organizational motives is fostering a culture of sharing experiences and practices for using AI tool(s). Participants *(P6, P7, P9, P10, P15, P18-P21, P23, P26)* described how their companies provided open spaces, such as Slack channels, to exchange knowledge about AI tool use. For example, *"we have a Teams channel and we have something called Tech Tips, where people just either talk about tool(s) or how to use the tool(s) and how to use better prompts." (P21)*

While organizational communication channels facilitate the quick sharing of tips, the participants mentioned that a more centralized document of best practices and knowledge would be ideal. *"It would be nice if there's like a more centralized repository where people can just share all their tips there." (P7)* Hence, as P7 alluded, a single, accessible, and searchable knowledge base would provide potential benefits to streamline the process of learning and leveraging AI tools.

Furthermore, practitioners described having a supportive environment where sharing about AI is welcomed and encouraged. *"Actually, nobody is feeling embarrassed or anything, it's more like everyone is finding this thing quite magical and wondering if it can play a significant role." (P15)* The narrative is less fearing judgment over using AI tools and more feeling excitement about their potential. These findings indicated that culture seems to influence whether organizations and team members help facilitate an environment that is conducive for practitioners to share insights [210].

OM2: Providing and promoting the tool:

I found 11 participants *(P5, P7, P8, P9, P11, P15, P17, P18, P19, P20, P25)* revealing that their companies either paid for or showed willingness to cover the cost of subscriptions.

Interviewees explained that the decision to invest in paid AI tools is often driven by the potential for growth and the relatively minor financial burden of these expenses. P10 highlights that, to many companies, the investment is not much. *"What is the subscription fee for GPT for two months, something like 20 USD? From the company's perspective, it's nothing." (P10).* P10 further adds that their company usually pays for tool if the employee shows that it is essential to learn and grow. *"When I can convince them that [AI tool] is necessary, and it will help us grow a little bit further,*

the company will definitely will pay for it" (P10) Interviewees whose organizations paid for their subscription argued that over the course of a month (i.e., 30 days), if developers could save at least a few hours of work, then the cost benefit of paying for a subscription would be positive.

In addition, organizations can enable employees to tailor models and versions for specific use cases and needs. There is also the ability to build add-ons and extensions from the tools. Prior to AI tools, this would be like an organization providing a custom Stack Overflow to its employees [211]. While previously difficult to achieve, this is now possible if organizations acquire subscriptions and API for its employees. *"You can choose including several other models that you can pick from. I get either 1,000 or 500 instances of free conversation." (P15)* Interviewees also mentioned that such enterprise versions of AI tools helps mitigate risks associated with data breaches and ensure a unified approach to tool usage. *"In the company, there are ready-made [tool(s)], and because you need to make extensive adjustments to your own account due to budget considerations, you will definitely use the company's unified planning." (P18)*

In addition, six interviewees *(P2, P10, P15, P17, P20, P25)* described how their organizations formally and informally promoted different tools. Due to constant notifications, it is possible for practitioners to miss new tools introduced by their organization. *"I are bombarded with a lot of information every day, some people might turn off certain notifications. They might only find out about [new tool] after they noticed a colleague using something, and then they became aware of it." (P17)*

OM3: Providing guidance and training:

Of the interviewees, 16 *(P2-P9, P13, P15, P17-P20, P23, P25)* described the existence of some form of guidance related to using AI tools in their organizations. Specifically, guidance refers to organizational policies and guidelines that govern what and how practitioners can use AI tools in their work.

The interviews *(P15, P19)* indicated one approach to provide clear guidelines was dedicating resources to training practitioners. Training covered best practices and techniques for prompting.

In P15's organization, senior management was heavily invested in pushing the entire company towards AI so training became mandatory. *"AI is essentially a strategy of the company, and requires everyone to get on board and take exams, earn credits,*

and then teaches you how to use these tool(s). Otherwise, it will be reported to the supervisor, and there will be some kind of reward and punishment measures in place." (P15) The organization not only provided training, but also required employees to demonstrate their knowledge through testing.

In addition to providing guidelines and training, one direct strategy to tackle privacy exposure risk mentioned by interviewees is to develop a company portal that sanitizes all prompts. A company created portal allows employees to deal with customer data with an AI tool and lower the potential risk of data disclosure to a third party provider.

Three interviewees *(P13, P15, P17)* indicated that their companies have implemented structured mechanisms (i.e., sanitization) to ensure the safe and responsible use of AI tool(s) by their employees. For instance, P15 highlighted *"When I visit ChatGPT, I definitely have to go through the company's portal, and then on that end, there will be some verification and filtering happening." (P15)* Similarly, for P13, their sanitization tool first filters out sensitive information and can directly reject requests that may compromise safety in the input. There is a specific team in charge of the prompt filtration service and they use a combination of rule based and model based approach across multiple layers to remove compromising content.

6.2.4 Organizational Challenges

OC1: Lack of culture of sharing:

While some organizations established a culture of sharing AI practices, the interviewees *(P5, P8, P11, P13, P14, P22, P24, P25)* also described how their organizations did not have a culture of sharing and collaboration. A lack of sharing is characterized by limited communication and interaction among colleagues about AI tool use and best practices. The interviewees highlighted that their discussion about AI tools was limited to individual use rather than summarizing lessons that could be disseminated company wide.

Emerging in the post-pandemic work environment, previous literature showed that organizations working in remote and hybrid work environments, which are increasingly common, have to watch for coordination, collaboration, and communication challenges [212]. Coordinating and communicating about software development can be challenging, without even considering **"What are the tools that my colleagues are using?"** I found a frequent narrative was that practitioners are either unaware

of their colleagues' use of tools or perceived the use of these tools as irrelevant to their field of work.

Another limiting factor is that participants may be dissuaded when they perceive an organization is against the sharing of experiences. *"Absolutely not. There's no culture there. I think like, I know that some people on my team... they don't know that it's out there... There's no culture of sharing or anything like that." (P5) "No, there are very few [sharing activities]. It might be that the whole company hasn't really promoted or initiated such activities. Someone at my level, a middle-tier leader, will relatively follow the company's policies in doing things, making plans." (P14)* In P14's case, even though they are a technical lead with dozens of employees working under their supervision, they still need to follow the company's lead. Therefore, if an organization demonstrates limited interest in promoting a culture of knowledge sharing, the effect seems to cascade down.

OC2: Company does not cover the cost of the tool:

A significant concern that emerged from the interviewees was the cost of the premium version of the AI tool(s). 8 interviewees (P5, P6, P11, P12, P13, P14, P22, P26) indicated that they bore the expense of these tool(s) out of their own pockets to have access to better features. For instance, P16's organization is based in Asia and has a large number of employees, but the organization was unable to cover the cost of the tool due to payment. *"The company is relatively poor. I don't have this paid version!" (P16)* Many of the largest AI tool providers are based in North America so it seems reasonable for organizations based there to establish partnerships with these providers [213].

Employees may purchase a subscription as a trial run and try it out for a period of time, but the high cost of subscriptions can lead people to just revert back to a standard free version. *"During the initial stage of trying it out, I did purchase the pro version. But, the company didn't buy this kind of license. This is a reflection of the company's real situation." (P14)* Not having access to the full paid version could have negative repercussions on the benefits a practitioners could realize given that paid tools usually provide bigger prompt limits, more recent knowledge base, and ability to more frequently query [214]. The paid version often offers better features which can help accelerate the development work.

Related to the paid versus free versions, is the number of API calls granted per

unit of time. As of this time of writing, a plus version of ChatGPT allows up to forty prompts per three hours. A developer can quickly reach the upper bound if they are conducting a lot of experimentation. In contrast, an organization such as P15's employer grants developers up to 500 or 1000 prompts per day. Juniors have less experience with software development so they may require more trial and error, leading to higher API use. Seniors, in contrast, have a wealth of experience and require less trial and error prompting so they are more likely to reach the solution using fewer prompts.

OC3: Lack of company guideline:

We found 9 interviewees *(P10, P11, P12, P14, P16, P21, P22, P24, P26)* who indicated that there is a lack of formal training and guidelines in their company, on the effective use of AI tools. Organizations implemented restrictions preventing the unauthorized sharing of sensitive information on platforms like GitHub, but no such guidelines have been prescribed to address the emerging challenges of AI tools. *"In the early days it was mentioned that it was strictly forbidden to post anything on [Github]. And it should be in the handbook. The company have not made any specific and clear policies targeting AI tool(s)." (P14)* These interviewees complained that their organizations adopted a laissez-faire attitude towards governing the use of AI tools.

Sometimes not publishing any guidelines is a purposeful decision, because a company may be unsure about how these tools handle data. P21 adds, *"I had a training about security, about what to do and how to not get your information leaked, the basic things. But for AI, it's still not that explicit." (P21)*

Moreover, I found 20 interviewees *(P5-P14, P16-P18, P20-P26)* indicating that there was a lack of formal training from their company on the effective use of AI tools. Participant P25 noted that AI tool(s) were introduced to their company's Slack with no training or guidance. *"It was dropped just like that." (P25)* The interviewees' perception was that training could enhance developers' proficiency in crafting prompts and leveraging AI tool(s)' capabilities. *"if there's a training, which is able to efficiently help you like software developers, specifically, and how to prompt, I think it will be very helpful." (P21)*

11 interviewees *(P1, P4, P10, P13, P14, P17, P20-P23, P26)* voiced concern regarding the exposure of company data privacy as a challenge that limits their adoption

of AI tool(s). Among the complex regulations to navigate are the GDPR [215], California Consumer Privacy Act [216], Personal Data Protection Act [217], Personal Information Protection Law [218], and Consumer Privacy Protection Act [219]. I found that one primary concern from participants is that AI tools are recording user prompts and then using the prompt contents as further training data. *"Data security is really concerning part for us, and that's why we can't depend on ChatGPT and we can't use in our IDE. What will be the use cases of the consumed data and whats the guarantee they will not leak it somehow and sell it to somewhere. Are they consuming my credentials? Are they consuming my personal data?" (P4)*

6.3 Push-Pull Relationships between the Motives and Challenges

This section describes relationships that I observed between some of the motives and challenges that have a significant impact on the use and adoption of AI tools. These relationships act in a push-pull manner, whereby motives and challenges are interconnected with each other. Motives *pull* practitioners and organizations to increase use and adoption of AI tools, whereas challenges act in the opposite way and *push* practitioners and organizations away from adopting and using AI tools. For example, *providing and promoting the tool* motivates the use and adoption of AI tools, but in contrast, *company does not cover the cost of the tool* inhibits use and adoption. I derived these relationships because they help indicate the factors that software organizations can consider to improve their use of AI tools.

6.3.1 Culture of sharing AI knowledge vs. Negative judgment and lack of support:

One of the most important relationships identified in the research was **establishing a culture of AI experience and knowledge sharing helps increase AI tool use and adoption, but a lack of support and potential judgments impede adoption and use.** This relationship intersects all four categories identified in the theory from individual motive, and organizational motive, to individual challenge, and organizational challenge.

From an individual perspective, I found that participants often had a natural

inclination to share their experiences and best practices from their use of AI tools. The willingness to share knowledge not only fosters a collaborative environment but also informs colleagues who may not be aware of the productivity boost. *"People give ideas about the best way to prompt ChatGPT. I think that was usually somebody will share cheat sheets. This was something that really helped them like over time. And people kind of like, keyed into that and started to use that." (P20)* This type of individual initiative was also on display from practitioners who actively encouraged others in their organization to start using AI tools. *"[Employees] encourage other people to use it. But it doesn't come from an organizational level where it doesn't come from the top down to say we should be using these to make us more efficient." (P9)*

However, this individual motive is limited by a counter factor, primarily fuelled by a fear of judgment from other team members. This fear stems from a perception that reliance on AI tools may be seen as a lack of capability or laziness. *"Because you're relying on it so much that you're not using your brain, which is that's a problem." (P6)* The criticism is typically directed towards junior developers who leverage AI tool(s) to help them learn, search, and debug code. *"I've seen that passing judgment. Some of them is still passing judgments when using the ChatGPT." (P4)*

Some of the participant organizations facilitated a culture of sharing, where team members were encouraged to discuss their experiences and share tips. This sharing of knowledge is often conducted through communication channels such as Microsoft Teams or Slack, which act as avenues for knowledge transfer, as described by P7 and P21 in Section 6.2.3.

6.3.2 Providing and promoting vs. Not paying the cost:

Another relationship identified in this work was **providing and promoting AI tools within an organization helps increase AI tool use and adoption, but adoption and use is impeded when organizations do not pay for the cost of AI tools** I found participants agreeing that organizations covered the cost of AI tools was a motivating factor for employees to use them. When companies do not cover the cost, fewer people use the tools because the cost becomes a hurdle. Depending on geographic location and available resources, the cost may be significant for a practitioner, especially as the number of AI tools increases every time a major AI company releases a new model or tool.

One trial and error approach seemed to work for some participants. An organization may not initially pay for a tool, **if an employee can show that the tool is necessary for their job and boosts productivity, the company might then agree to cover the cost.** *"If I can prove that without chatGPT or without the paid version, I cannot work or it will help a lot for me, as well as the company to work along the way. If I'm using the paid version of chatGPT, the company will definitely pay for it." (P10)*

6.3.3 Providing guidance and training vs. Lack of prompting skills and usage guidelines:

The final use and adoption relationship I found is **providing guidance and training for using AI tools to help increase AI tool use and adoption, but adoption and use are impeded when individuals lack prompting skills and organizations do not provide AI tool usage guidelines**. As described earlier in OM2 in Section 6.2.3, when organizations invest in training for developers about AI tool(s) and prescribe clear guidelines, it clarifies the limits of what employees can and cannot do. This can help reduce uncertainty where a developer may not know if their actions are legally permitted by their organization. P21 provides an example of how he believes training would help. *"I think if there is training, which is able to efficiently help you like software developers specifically, and how to prompt, I think it will be very helpful." (P21)*

AI is a rapidly changing field with a plethora of active research. Sometimes within the span of weeks, major updates are made to popular AI tools or a completely new AI model is released altogether. Consequently, it can be difficult for practitioners to keep up with the pace of AI change if they are not actively paying attention to all the changes [220]. Should an organization provide prompt engineering and other training, it would help guide practitioners to understand how to best use the available tools and which software development tasks work best for each tool or model.

In contrast, this work found challenges pertaining to when organizations failed to provide clear guidelines on the use of AI tool(s), as discussed in detail in Subsection 6.2.4. *"I think we should be more careful about compliance this and things like that, I think sometimes in software dev, you kind of forget, because you're more lenient on looking for things online." (P23)*

I found that when organizations implement measures to sanitize inputs and out-

puts of AI tool(s) described in section 6.2.3, developers feel more comfortable. These sanitization processes are particularly important for practitioners to be assured that sensitive information, such as API keys, user data, or proprietary code, is not inadvertently captured and misused by these tool(s). *"I also maintain partnerships with organizations like OpenAI... Could involve filtering out safety or safety issue-related inputs and outputs." (P13)* The absence of such measures leaves developers wary about using AI tool(s) with sensitive or confidential code.

The absence of such measures leaves developers wary about using AI tool(s) with sensitive or confidential code. *"I never use names or brands, especially in terms of my company, I never use any of the company names. I usually obscure even references to environment variables." (P22)* This fear often leads them to alter code or create a hypothetical situation in their prompts to avoid the risk of exposing sensitive data. *"I'll always have a sanity check is this. Will this be okay to put in ChatGPT? I don't just copy-paste and put it in there. I sometimes do like pseudocode." (P21)* Therefore, the participants suggest organizations implement a combination of clear policies governing tool use and providing means of filtering sensitive data directly inside prompts. Otherwise, the risk of privacy exposures can be costly [221, 222, 223, 224].

6.4 Discussion

Previous research studying AI tool(s) in software development has often focused on the tool(s) themselves and the usability of those tool(s) for software tasks [29, 188, 186, 189]. These prior works found that developers reportedly gain immense boosts in productivity when they leverage AI assistance to help with development [186, 29, 189]. As a result, AI tool(s) such as ChatGPT experienced monumental user growth, where in less than 1 year, it successfully attracted 100 million weekly users [184].

One previous study attempted to provide more contextualization regarding the adoption of generative AI [203]. Russo developed a preliminary theory for the adoption of generative AI; however, 3/7 of his hypotheses were rejected upon analysis [203]. In particular, he suggested that organizations play a limited role in impacting developer adoption of AI tools, that more future empirical work is needed to investigate this fast-changing landscape. Therefore, I still lacked clarity on the impact of organizations and individuals gravitating toward or away from adopting these tools.

In this work, I identified challenges and motives that push and pull practitioners

into using AI tool(s). Unlike individual motives, which may have a visible impact, the study also uncovered the importance that organizational culture serves in the adoption and increased use of tool(s). The findings on the impact of organizations contradict with previous literature [203], however, I surmise that this may be a result of the emphasis in my study about organizational role in the use and adoption of AI tool(s). Moreover, aspects such as guidelines and data privacy may be becoming more important due to ongoing lawsuits and regulatory concerns.

In this study, I aimed to fulfill this gap through a Socio-Technical Grounded Theory approach, which included 26 interviews with practitioners regarding their use and adoption of AI tool(s) for software development. The findings presented a list of 12 factors, including 7 challenges that influence the use and adoption of AI tool(s). I also presented 3 relationships between factors, which detailed how increasing motives for the use and adoption of AI tool(s) may help reduce its corresponding challenges.

In the following subsections, I discuss how to increase the use and adoption of AI tools in organizations and the importance of privacy concerns and ethics amidst the use and adoption of AI tools. Finally, I propose directions for future research.

6.4.1 Increasing adoption of AI tool(s) in organizations

The primary finding in this study is what the factors are that are motivating and challenging the adoption and usage of AI tool(s) for software development. Previous works have shown that developers who use tool(s) like Copilot and ChatGPT should reap significant productivity improvements and become more effective in their work [27]. While there is debate on the exact extent of improvement, the current literature and confirmed in this study supports the notion that AI tool(s) boost practitioner productivity.

However, despite these benefits, this study unveiled that not all companies or developers are embracing AI tool(s) to their fullest potential. Several limiting factors were identified, affecting both the extent of adoption and usage. At an individual level, knowledge gaps and uncertainty about the optimal use of AI tool(s) can hinder their adoption. I found that AI tool adoption in organizations is not as trivial as simply creating a ChatGPT account and using it to generate software code.

Previous works have stated productivity boosts [186, 29], materializing from auto completion, finding code solutions, and edge cases are motivations for developer use of AI tool(s). Whereas, motivations for not using AI tool(s) include lack of useful

or relevant output, particularly involving functional and non-functional requirements [29].

Additionally, with my overarching goal in this dissertation, this study shows the immense impact that AI tools are having on software practitioners. Albeit, prior studies have all primarily focused on coding related tasks such as code generation, requirements engineering also stands to benefit from increased adoption of AI tools. For instance, practitioners could benefit from the automated analysis of large quantities of user feedback collected from the "crowd" with a tool like ChatGPT. In theory, a product manager would receive measurable improvements to productivity if they give the raw user feedback about their software product to an AI tool and ask the tool to give the most important software requirements. Consequently, adopting AI tools for requirements engineering would also most likely mean that one should expect to see similar challenges to those I observed in this research. For example, is the parent organization going to pay for the AI tool for the product manager?

One organizational challenge that has a profound impact on software professionals is simply organizations not covering the cost of AI tool(s) such as ChatGPT or others. Although the cost of $20 USD may seem trivial to resource rich organizations, numerous participants in this study indicated that their organizations are not covering this cost. For participants who are in less resource rich situations, the price of "Plus" subscriptions may be cost prohibitive.

From an organizational perspective, the culture surrounding knowledge sharing plays a pivotal role. In environments where sharing and collaboration are encouraged, knowledge of AI tool(s) and their potential benefits disseminates more rapidly, fostering a more widespread and effective adoption. When employees share insights with others in the company or team, it can significantly improve the productivity of colleagues because some employees are not aware of the best practices. In P16's organization, colleagues share insights about how to use tool(s) to increase productivity. As an example, one colleague demonstrated how they were able to use AI tool(s) to develop a useful web scraper in 15 minutes, the previous human manual effort would have been a whole day.

In contrast, practitioners should be cautious of potential judgment and lack of culture of sharing manifesting in their organization as they may limit the gains an organization could realize. Organizations should avoid creating environments where employees are negatively judged and rated depending on their usage of AI. The findings from this study emphasize the importance of addressing both individual and

organizational challenges to maximize the benefits of AI tool(s) in software development. Software professionals can use this theory to identify areas to watch out and practices to adapt to increase motives for high use of AI tool(s).

6.4.2 Directions for Future Research

- **Prompt sanitization and privacy safeguards:** This research highlights the importance that many practitioners place on effectively protecting company and user privacy. Privacy and security of prompts when using AI tool(s) is hugely critical for more widespread adoption and use of these tool(s). Further research could investigate leveraging security and privacy mechanisms to protect confidential data as built-in plugins when using AI tool(s).

- **Education:** The participants of the study expressed how training and understanding prompting is crucial to achieving the most benefits out of the tool(s). While some organizations provide varying levels of guidance and basic training for producing more effective prompts, more research is needed to create more structured training. Future research could study the areas and content that should be part of manuals and training material to help practitioners become effective users of AI tool(s).

- **Requirements Engineering:** Finally, one of the most significant areas of future research, given the topic of this dissertation, is exploring the use of AI tools to automate requirements engineering. In this study, I shared insights on how practitioners gained several benefits from applying AI tools in their software tasks, namely increased productivity. However, my interviewees also shared with me the challenges they experienced from trying to apply AI tools in their work. I share in the next chapter a preliminary work on leveraging AI tools for the development of new requirements from user feedback. Since it is just a preliminary study, future work could explore the other areas of requirements engineering and strategies to minimize the challenges of adopting AI tools for requirements engineering.

6.4.3 Threats to Validity

Despite my best efforts in mitigating threats to the research, there are still some limitations. I use the total reliability framework from Roller [120] and the aspects of

credibility, analyzability, transparency, and usefulness to assess the threats to validity.

Credibility refers to the accuracy and truthfulness of the data. It is about ensuring that the research accurately represents the data collected and the perspectives of the study participants. For this research, my selection of interview participants is limited by convenience sampling, which means reaching out to practitioners from my personal contacts. However, I mitigated this threat by ensuring that each participant matched the selection criteria, which is aimed towards answering my research goal. I ensured that each participant was a practitioner who uses AI tool(s) in their software development work.

Analyzability refers to the ability to draw meaningful conclusions from the data. For my study, I relied on the transcription tool Otter.ai [225] to help transcribe each interview. Three co-authors followed the steps of STGT to collect and analyze the data, including conducting open coding, memoing, and constant comparisons until reaching saturation at which point the theory was developed.

Transparency refers to the clarity and completeness of documenting of the research process. For this study, I provided detailed descriptions of the entire research methodology. I also provided examples and used quotes wherever possible. I provided as many details as possible to show the relationships between the themes in the findings. In addition, I release a replication package containing the interview questions and code book. Unfortunately, I am bound by research ethics obtained from my institution and unable to release raw transcripts nor answers from interview participants.

Usefulness refers to the utility of my research and whether the findings have value to the relevant audience. AI assistance tool(s) for software is currently undergoing a rapid pace of change, which is challenging for my research. However, many of the findings transcend cultural aspects that are not easily *"solved"* by an update in an AI tool but otherwise require further study to mitigate any of the challenges limiting the adoption of AI tool(s). This is why I conducted interviews with diverse software professionals. Although I do not assume that every software organization will encounter the same challenges and motives identified in this research, I expect professionals and organizations that share characteristics as those in this study to experience some similar benefits and limitations.

6.5 Conclusion

When generative AI tools such as Copilot, ChatGPT, and Gemini were released to the public, software practitioners began experimenting with their potential to support many development tasks. Before investigating how these AI tools are applied in the development of new requirements from user feedback, it was essential to first understand how practitioners and organizations are adopting and using AI tools in general.

Practitioners and organizations that I interviewed are increasingly relying on generative AI assistance tools to help with software development, recognizing the importance these tools play in the present and future. I followed the steps of the Socio-Technical Grounded Theory to guide the research through 26 interviews with industrial practitioners. My findings show that practitioners are motivated to adopt and use AI tools based on 2 individual motives and 3 organizational motives, but their use is also challenged by 4 individual and 3 organizational factors.

My work in this chapter shows that while AI tools can create positive benefits, such as increased productivity for software practitioners, numerous challenges may impede the effective adoption of AI tools. In the next chapter, I describe my final empirical study that investigates how requirements practitioners use generative AI tools, specifically ChatGPT, to automate their development of new requirements from user feedback.

Chapter 7

Tool Based Automation of User Feedback

This chapter presents my final study that addresses **RQ5:** *How are requirements practitioners leveraging generative AI tools to conduct the development of new requirements from user feedback?*

In Chapter 3, I described that industry practitioners reported limitations from using existing user feedback automation tools because "they are hard to use, expensive, or still not publicly available." However, this perception was dramatically impacted due to the emergence of generative AI tools. At the forefront of the generative AI movement is ChatGPT, which has millions of monthly users, many of whom are software practitioners. My dissertation's goal of automating user feedback analysis therefore hinges on investigating the use and impact of generative AI on developing new requirements based on user feedback from online sources.

In Chapter 6, I made an initial attempt to develop more insights about how generative AI tools are used and adopted by software practitioners. Given the findings in Chapter 3 that software organizations prefer to have more automated approaches to process the growing amount of user feedback, the next logical step of my research was to explore generative AI with an emphasis on developing new requirements. Recently, AI tools have proven highly effective at parsing large volumes of natural language text [226]. I embarked on a study with the following research question:

How are requirements practitioners leveraging ChatGPT to conduct the development of new requirements?

Table 7.1: Study Participants. Product experience refers to the number of years that a participant has conducted requirements or product related work, such as a product manager, requirements engineer, or business analyst.

ID	Product Experience	Gender
P1	4 years	Male
P2	9 years	Female
P3	13 years	Male
P4	5 years	Female
P5	6 years	Male
P6	9 years	Male
P7	4 years	Male
P8	7 years	Female
P9	5 years	Female
P10	3 years	Male

7.1 Methodology

For this research, I opted for a think-aloud study following Fonteyn, Kuipers, and Grobe [227], to investigate how requirements engineering practitioners use AI tools, specifically focusing on ChatGPT[1]. In this section, I describe the study design, task design, participant recruitment, data collection, and data analysis. I employed a think-aloud protocol [227] to capture participants' thought processes during the task.

7.1.1 Study Design

The study design involved curating a dataset and developing the steps to conduct the study. Each study participant was given a dataset of 20 Reddit posts related to the popular application Zoom.[2] Recall from Chapter 4 that I explored Reddit data in several studies for developing new requirements. Reddit is different from other textual based user feedback because of the incredible depth and variety of product discussion that surpasses comparable user feedback channels.

It offers significant text length that is ten times greater than app reviews, allowing users to elucidate their thoughts and involve the community. Moreover, I selected the Zoom application for this task because of my prior research experience in Chapter 4. In that study, where I explored inclusiveness related user feedback, *Zoom* emerged

[1] `https://chat.openai.com`
[2] `https://www.zoom.com/`

as an application that has a wide variety of user discussions. Zoom feedback discussion often included user discourse from all requirements related subjects, including inclusiveness, other important non-functional requirements such as privacy, security, performance, and availability. End users often also chastised the company for not swiftly developing software features or fixing bugs that burdened the end users.

These reasons made Zoom an ideal sample as opposed to software applications that strictly had user complaints about content (e.g., video streaming applications) or gaming mechanisms (e.g., video game application). I wanted to select an application that had a broader range of user feedback categories. Moreover, given that a study participant would only analyze 20 posts in the think-aloud session, I chose to select a single application in order to minimize confusion and allow participants to focus on just one application. Selecting only one candidate application's user feedback should also create a more realistic scenario as a product practitioner in industry would most likely work on a single application.

Once I selected the source of user feedback and the relevant application, the next step was designing the think-aloud study subtasks that each participant followed. Each participant was informed that they were free to use ChatGPT at any stage of the task, without any restrictions on how or when it could be used. Since one of my recruitment criteria was that the participant uses generative AI for their day-to-day requirements development work, naturally, each participant ultimately used ChatGPT during the tasks.

I requested that each participant use the AI tool to the best of their ability and most closely resembles their day-to-day work. To ensure that each participant was receiving the same type of AI assistance, I confirmed that each participant was using ChatGPT and that their model was ChatGPT-4o, as it was free and allowed all participants to work with the same version of the model. Furthermore, before each participant started a new chat session with ChatGPT, I ensured that the participant switched off memory and custom settings so that their results were not biased by their prior chat history.

At the beginning of the study, I requested permission to record the meeting and explained that their identity would be kept anonymous as per the ethics approval from the IRB at UVic, who approved this protocol. Moreover, I explained to each participant that they could opt out of the study at any time.

Task Design: To simulate a realistic task in requirements development, I provided each participant with a dataset of 20 Reddit posts representing user feedback

from the Zoom app. Before curating the dataset, I first collected all the Reddit posts between January 1, 2024, to December 1, 2024, from the subreddit *Zoom*. I wanted to ensure that participants would face a variety of different themes of user feedback, so I conducted purposive sampling [201]. I worked with another colleague, who has industry experience in requirements development, to identify 20 posts. Table A.1 from the appendices section lists the user feedback that I curated for this study. I took a random sample of user feedback for Zoom and manually analyzed the posts to find actionable user feedback. A user feedback post is defined as actionable if it is specific, relevant, and detailed enough for a product manager or requirements practitioner to derive it into a user story for development for the software organization. The posts emerging from the manual process pertained to themes such as audio, accessibility, and privacy. Moreover, the list of user feedback was refined by conducting five pilots with experienced practitioners. They helped identify issues such as the difficulty of analyzing thirty posts within the given timeframe. The pilot participants also helped to improve the instruction manual to make the instructions clearer.

The think-aloud study was structured into three parts. I provided a detailed instruction document that explained each part of the task and outlined expectations for using ChatGPT. The three parts are:

1. **Classifying Feedback:** Each participant was to analyze and label the eighteen Reddit posts as either "actionable" or "not actionable." I informed each participant that "Actionable" feedback means specific, relevant, and detailed enough to be directly translated into a user story for software engineers to allocate time to develop.

2. **Creating Themes:** Each participant was then instructed to group actionable posts into themes that reflect common user concerns. They were to name each theme, explain the rationale behind their grouping, and identify the most important themes that emerged from the feedback. This is adopted in line with industry practices where user stories are grouped into higher-level themes, known as epics, which represent overarching goals.[3]

3. **Generating User Stories:** Each participant was to write user stories using the standard format: "As a [user], I want [feature] so that [benefit]." They were also to include at least one acceptance criterion per story. They were to ensure

[3]https://www.atlassian.com/agile/project-management/epics-stories-themes

traceability by linking stories back to the original Reddit posts and themes. Participants were to indicate the user stories that they perceived as the most important.

The participants, on average, took about 65 minutes to complete the task, with a minimum of 50 and a maximum of 80 minutes. While conducting the tasks, participants spoke aloud to explain their thinking, which supports tracing their decision-making in real time. As each participant conducted the task, they shared their screen with me, which showed their activities modifying and labeling the dataset and conversing with ChatGPT-4o. Each participant's think-aloud session was recorded, which included the audio and the screen share video.

After the task was completed, they submitted a completed worksheet with all the data and their chat conversation with ChatGPT. This setup gave me a detailed view of both the output of their requirements development, their thinking, and decision-making that led to their conclusion. After the task competition, I also spent 10-15 minutes asking each participant some follow-up interview questions about their experience with using AI tools for requirements development. In Table 7.2, I list the follow-up interview questions that were asked after each participant finished their session.

7.1.2 Participant Recruitment

Before recruiting the study participants, I first piloted the study with 5 participants. Each participant has industry experience with user feedback and requirements engineering. The range of industry experience for the pilot participants was 3 to 22 years. Piloting helped me refine my task design, including the user feedback I included in the dataset, and simplifying the task. For example, I reduced the number of posts in the dataset from 30 to 20. I also requested participants to follow the think-aloud protocol instead of writing down their justifications. The piloting experience demonstrated that the think-aloud approach supported me in analyzing participant decision-making.

For the main study, I recruited 10 participants through Upwork, an online freelancing platform. Previous literature has shown that Upwork [228] is an excellent platform to recruit experts for software engineering studies. Each participant was compensated with $30 USD for the project. In my recruitment post, I specified that I was seeking individuals with experience in software requirements development or

Table 7.2: Base follow-up interview questions asked after each session

Base Questions
How was the experience using ChatGPT for the experiment? How did you think it was beneficial?
What do you think using ChatGPT helped with?
Were there moments when you disagreed with the AI's suggestions or analysis?
How comfortable or uncomfortable were you relying on the AI assistant during the task? Can you explain why you felt that way?
Were there moments when you disagreed with the AI's suggestions or analysis? Could you describe one specific situation, and explain how you decided to respond?
Did you feel your own judgment was influenced by the suggestions of the AI assistant? Can you recall a specific instance where this happened clearly?
Would you have preferred completing this task without an AI assistant? Why or why not?
Can you elaborate on how the AI assistant specifically enhanced or hindered your productivity?
Do you have any final thoughts or reflections about this task, the process, or your interaction with the AI assistant you'd like to share?

product management.

Since the goal of this research was to uncover how requirements practitioners are using AI tools, I wanted to ensure that participants would be people with experience working in industry. I selected candidates with at least 3 years of experience in requirements work, including user feedback or writing user stories, and who were comfortable thinking aloud in fluent English. For their experience, I relied on their self-reported description and curriculum vitae that they submitted before starting the task. These participants worked in product and requirements-relevant roles, including product manager, business analyst, and product owner. Additionally, I wanted to gain a broader perspective of participant behavior. The recruited participants came from a wide range of professional backgrounds and geographic regions. Each participant was assigned a pseudonym (P1 through P10) to protect their identity. Table 7.1 details the years of experience that each participant has in product related work.

7.1.3 Data Collection

Each participant received the Reddit dataset of 20 posts, the instruction guide, and a blank worksheet at the beginning of the session. Every think-aloud session was conducted via Zoom as the software allows easy screen sharing and video recording features. I asked each participant to complete the study in a quiet environment where they could record their screen activity and audio without interruptions. Before beginning each session, I explained in detail the steps of the tasks and that each participant's data would be anonymized in any reporting. As per my data collection, I was interested in capturing participant interactions and decision-making with AI assistance. This resulted in four main forms of data:

- A *screen recording* of the entire session, including audio of their think-aloud narration and all on-screen activity (e.g., using ChatGPT, reading feedback, editing the worksheet).
- A *transcript* of the recorded session. I transcribed the audio recordings using Otter.AI[4]. I then reviewed and manually corrected each transcript to ensure accuracy. This helped ensure that the data I was working with accurately reflected participants' verbalized thoughts throughout the session.

[4] `https://otter.ai/`

- A *completed worksheet*, where a participant documented their feedback classification, main feedback themes, and user stories.
- A *ChatGPT chat log*, showing the complete transcript of a participant's interaction with the AI during the task.

Once they completed the task, I asked them to submit the worksheet and ChatGPT log.

7.1.4 Data Analysis

The reason I employed a think-aloud protocol is to better understand how participants reason through their decisions, how they navigate uncertainty, and most of all, how each participant engages with an AI tool during the requirements development task. I first watched each participant's screen recording in full, taking structured observation notes as I went. These notes focused on how participants moved through the task, the types of prompts they gave to ChatGPT, how they responded to the model's outputs, and any visible signs of uncertainty or confidence. I also noted instances where participants revisited earlier decisions, hesitated, or modified their strategy mid-task.

Both the audio transcripts and my observation notes formed the basis of the analysis. I conducted open coding [92] on the data. I assigned descriptive labels to passages in the transcripts and corresponding behaviors noted in the recordings. These codes captured actions such as *Adding detail or depth*, *Brainstorming missing features*, and *Polishing language*, reflecting how practitioners used ChatGPT's responses.

I opted for inductive coding, which allowed codes and patterns to emerge naturally from the data. After coding all sessions, I reviewed and grouped similar codes into broader conceptual themes. This iterative process enabled me to identify four major themes for practices and three themes for challenges. Table 7.3 illustrates an example of coding from raw quotes into themes.

7.2 Findings

Below, I describe the practices and challenges I observed from the think-aloud study. Tables 7.4 and 7.5 depict these themes that emerged from the data.

Table 7.3: Example Coding of Raw Quotes

Raw Quote	Code	Category	Concept
Example 1: ChatGPT sort of refines it in a way that comes out clearly ... makes it goal oriented and straight to the point, and sort of brings out the clarity that is needed.	Improving language	Human-Initiated Analysis Before Inviting AI	Practices
Example 2: So let me do a quick check, because this scene, this seems to be like an actionable thing for me, because normally, like, even if I'm a Product Manager from the computer, from the. [From] the same background, but these network related issues is always challenging for me to fix that. So I will be actually taking a little bit of help from ChatGPT [for this]	Adding detail/depth		
Example 3: I can refine them by giving different prompts to ChatGPT that it is not what I am wanting or it is not what the feedback meant.	Trial and error	Hallucinations and Maintaining Trust	Challenges
Example 4: ChatGPT may assist us, but we have to think from our own perspective as ChatGPT is a machine... Sometimes it gives a very ridiculous result, which may not be according to the feedback it the user I see in the first user story, his reply was not very [relevant] to the feedback I gave him.	Lack of trust		

Table 7.4: Practices for using AI Tools by Requirements Practitioners

Category	Code
Starting with Human-Initiated Analysis Before Inviting AI	Initial manual review Avoiding AI influence Manually drafting requirements Initial requirements Improving language Adding detail/depth Checking for completeness
Using AI as a Domain Expert & Research Assistant	Asking for justification support Getting expert advice Using AI with references
Providing All User Feedback Dataset to AI Before Conducting Manual Analysis	Automated categorization One-shot AI categorization Skepticism toward AI output Reactive interpretation of AI output Not blindly copying AI Correcting AI errors
Prompting the AI Tool and Brainstorming Features	Iterative prompting Specifying output format Rephrasing prompt Brainstorming missing features Completing requirements

7.2.1 How Requirements Practitioners Use ChatGPT for the Development of New Requirements from User Feedback

The requirements practitioners leveraged ChatGPT for analysis in several different ways. Through my analysis, I identified a set of emergent practices that reflect how participants integrated ChatGPT into their workflow. While participants varied in their approaches, common patterns emerged that suggest a set of generalizable practices rather than a single "best" method. Importantly, **no participant relied entirely on ChatGPT for the task**. Instead, they consistently combined AI assistance with manual labor at each step of the task.

For two out of four practices (i.e., "Starting with human-initiatedd analysis before inviting AI" and "Using AI as a domain expert & research assistant"), all ten par-

ticipants exhibited them in the think-aloud protocol. However, there was no "silver bullet" in their approach. The participants instead used an overlap of the four practices, whereby two participants (P3, P9) used all four practices, and the other eight all used at least three of the four practices. Overall, participants would interweave between AI use, where they applied one of the four practices, and manual analysis. For example, several participants (e.g., P4, P7, P10) began with human-initiated analysis (i.e., one of the four practices) and manually reviewed each feedback before involving ChatGPT. Later on, they, along with all the other participants, relied on ChatGPT to provide domain expertise (i.e., another one of the four practices) whenever they had questions or concerns that they did not have the answer to.

For requirements practitioners aiming to effectively use AI tools in similar contexts, these strategies offer practical guidance that can be adapted to suit different tasks, styles, and levels of domain expertise. Below, I report on the themes I identified related to different approaches for using AI for the development of requirements.

Starting with Human-Initiated Analysis Before Inviting AI:

All ten participants at some point during the think-aloud protocol immersed themselves in user feedback before turning to ChatGPT. One of the phrases that participants often mentioned when conducting the think-aloud study task was that they wanted to formulate their understanding. They wanted to get a holistic sense of the user feedback before trusting ChatGPT to refine or check their work.

For example, P1 preferred to draft user stories in their "own voice" first and then have ChatGPT polish the user stories. *"I want my initial thinking to be captured... GPT to sort of input more context or depth into it. The moment I prompt ChatGPT first, it kind of brings forth a lot of information that might not fully align with what I want" (P1)*

Most of the ten participants (P4, P7, P10) began with human-initiated analysis, manually reviewing and organizing feedback before involving ChatGPT. Others (P3, P6) adopted this approach later on in the session after experimenting with AI responses.

This human perception first approach was echoed by others. Several participants also noted that for a manageable set of feedback (e.g., 5 to 10 posts) and if they were not confined to the time allocated in the think-aloud study, they may have manually done all the analysis. P1 echoed this stance and despite assuring P1 that the data

was publicly sourced, they initially held off using ChatGPT until confirming the data was okay to share with ChatGPT. *"I think I should have asked first because I assumed it was sensitive" (P1)*

Additionally, practitioners often treated ChatGPT as a writing and formatting assistant to transform ideas into polished user stories.

For example, P4 first derived a user story that they thought was important from the data. P4 subsequently asked ChatGPT to refine the user story further. *"As a disabled Zoom User, I want to feel [included], when it comes to free calls and charges especially for us so that we do not feel left out compared to able users."* - (*ChatGPT Prompt*) In response, ChatGPT replied with a more precise and refined user story, *"As a disabled Zoom user, I want free access to essential features like phone dial-ins and captions, so that I can participate in meetings on an equal footing with non-disabled users and feel fully included without facing additional barriers or charges."* - (*ChatGPT Response*)

After identifying a need to make a refinement in wording, participants would ask ChatGPT to generate a better theme name, user story, or even acceptance criteria. This behavior shows requirements engineers using AI to raise the quality of written requirements. As one participant remarked, *"ChatGPT sort of refines it in a way that comes out clearly ... makes it goal oriented and straight to the point, and sort of brings out the clarity that is needed."*

However, participants were not inclined to follow the AI by default. P6 cautioned that while "ChatGPT does help us," one cannot "just [use] it all the time [and] copy-paste from it... we have to think from our perspective." In short, practitioners asserted their autonomy by deciding when and how ChatGPT would be involved, often doing an initial pass on their own before asking the AI to step in.

Using AI as a Domain Expert & Research Assistant:

Rather than treating ChatGPT as a tool that can solve all the requirements development tasks, all ten participants often used it as a sounding board to validate their initial analyses.

For example, P1 would first label a piece of user feedback as actionable or not and then prompt ChatGPT to validate and support their decision with evidence, effectively asking the AI to critique or justify the participant's decision. *"I am analyzing some feedback from reddit, I'd give you some cues into what I think each are (in terms*

of them being actionable or not), your job is to either agree with supporting logic or disagree" - (*ChatGPT Prompt*)

This prompting approach turned ChatGPT into a reviewer for the participant's work. It confirmed P1's labeling and generated a rationale for them: *"I agree with your assessment that this is a trust/data privacy issue rather than an actionable feature request or bug report. The user is expressing concern about a potential privacy vulnerability but hasn't identified a reproducible issue or a clear software malfunction. Instead, the concern stems from uncertainty about Zoom's behavior and settings."* - (*ChatGPT Response*) Participants found that this approach helped surface level reasoning they might have missed.

As P2 noted in their think-aloud session, *"At this point, I now need to consult ChatGPT, because this is not clear. I am confused by the user feedback"*, using the AI tool whenever they felt unsure how to interpret an unclear piece of feedback. Similarly, P3 copied complex user feedback into ChatGPT and asked domain specific questions like *"As a Zoom expert can you clarify... ?"* - (*ChatGPT Prompt*).

In response, ChatGPT provided detailed explanations of Zoom's expected behavior and possible causes (e.g., explaining how screen sharing works and why one participant might mistakenly think they saw the host's screen). This helped P3 quickly understand whether a reported issue was a known limitation, a user error, or a potential bug.

Practitioners treated ChatGPT as a junior assistant that could double check decisions and articulate justifications rather than as an omniscient source of knowledge. Using ChatGPT allowed participants to fill in gaps in their knowledge, which later in turn could help inform their requirements decisions. For example, confirming whether a piece of feedback was truly actionable or already addressed by existing features.

Providing All User Feedback Dataset to AI Before Conducting Manual Analysis:

In contrast to the refinement of requirements artifacts after a participant first manually analyzes the user feedback, four of the ten participants (i.e., P1, P3, P6, P8) decided to take an opposite strategy and apply ChatGPT first. For example, P6 loaded the entire dataset of 20 feedback posts to ChatGPT at once and instructed ChatGPT to perform an analysis. P6 wrote in a single prompt, *"Here's the feedback ... Add a theme column."* - (*ChatGPT Prompt*) ChatGPT generated a table listing

each feedback alongside a thematic category. The participant then iteratively asked for more columns, first "Effort" and "Risk" levels for each item, then "Actionability", and even a flag for whether the user's problem was well defined.

ChatGPT responded, augmenting the table iteratively, with the requested analysis (e.g., marking an accessibility request as high effort and high risk due to legal implications). My observation from the four participants who exemplified this practice suggests that some practitioners may be willing to offload some initial conceptualization work to ChatGPT. Future research studies would need to follow up and investigate the prevalence of this phenomenon and how must practitioners use ChatGPT as an automatic requirements analyzer, with humans in the loop.

Rather than helping with one step of the task, ChatGPT was performing a multi-dimensional analysis across the entire spectrum of tasks, both actionability and themes. The result was a quick categorization approach, where the participant could review each user feedback one at a time. Moreover, the positive benefit of analyzing the entire feedback dataset to the AI tool meant that a participant could get instant results, as opposed to analyzing the results manually and slowly over time.

Prompting the AI Tool and Brainstorming Features:

Another theme that I observed nine out of ten participants making was leveraging ChatGPT as a tutor before diving into the task. For example, P5 told ChatGPT to apply industry standards for making user stories, including the INVEST framework [229] for writing good user stories. The INVEST framework is made of six elements that form the acronym: I - independent, N – negotiable, V – valuable, E – estimable, S – small, T – testable. To reduce bias between each participant, I had asked each participant to use ChatGPT without the memory setting. However, by the time P5 started analyzing the Reddit feedback, they had conducted 'prompt priming"[5] of ChatGPT, whereby they influenced the direction of the tool. In this case, P5 prompt primed ChatGPT to write user stories adhering to the INVEST framework, which is one of the industry's best practices.

Apart from tweaking the AI tool, I noticed that participants would present a user's pain point and ask ChatGPT to brainstorm about the relevant missing feature. For example, after sharing user feedback about some screen sharing problems, P7 prompted ChatGPT to identify the missing feature. ChatGPT responded with a

[5]https://medium.com/aimonks/what-is-priming-the-prompt-1f12dcb855a8

list of feature gaps, such as a screen share indicator, screen sharing logs, and more fine grained sharing controls, each with accompanying rationale. The AI tool was helping the participant brainstorm the potential missing user story that captures what the user needs in the user feedback. From ChatGPT's suggestions, the participant subsequently had ChatGPT turn some of the answers into formal user stories with acceptance criteria. It exemplifies how human-AI collaboration can extend into using ChatGPT as a brainstorming buddy for requirements development.

Table 7.5: Challenges experienced when using AI Tools by Product Practitioners

Category	Code
Hallucinations and Maintaining Trust	Correct sounding output
	Trial and error
	Lack of trust
	Overreliance worry
Bias and Ethical Considerations	Bias reflection
	Privacy consideration
	Participant bias
	Prioritization bias
Limitations in Context and Understanding	Loss of nonverbal cues
	Lack of context
	Lack of clarification
	Lack of human experience

7.2.2 Challenges of Using AI Tools in Requirements Development

Despite their generally positive experiences, practitioners encountered several challenges and tensions when integrating ChatGPT into their RE analysis. Below, I report on the three main challenges that practitioners experienced when using AI tools for requirements development. Overall, almost all ten of my participants experienced all three challenges. In particular, all three challenges emerged while verifying ChatGPT's outputs against original user feedback. Often, a group of participants would identify them early and hence read the output attentively. Other participants would reflect on these challenges towards the end of the session when I ask them follow-up questions about their use of AI in the think-aloud protocol.

Hallucinations and Maintaining Trust:

A recurring issue I observed from all ten participant interactions was that ChatGPT's output sometimes misinterpreted the user feedback, requiring the practitioner to intervene. The idea of ChatGPT hallucinating has been documented in other industries, such as code generation [230], but this is the first observation of its impact on requirements development.

Participants experienced instances where the AI could produce answers that sounded confident but were *"not very valid to the feedback I gave him." (P5)* P5 described the situation as *"sometimes it gives a very ridiculous result"*. The tool had strayed from the actual problem described, essentially hallucinating a user story response that did not fit the user feedback. This forced the participant to revise the prompt, *"I can refine them by giving different prompts to ChatGPT that it is not what I am wanting or it is not what the feedback meant." (P5)*

Such iterations cost time and effort. More importantly, these scenarios reduce participant trust in ChatGPT. *"[ChatGPT] may assist us, but we have to think from our own perspective as ChatGPT is a machine... Sometimes it gives a very ridiculous result, which may not be according to the feedback it the user I see in the first user story, his reply was not very [relevant] to the feedback I gave him." (P5)*

Despite the heavy use of AI tools during the think-aloud study, practitioners were quick to point out the risk of over-reliance on AI tools. Several expressed caution about blindly accepting ChatGPT's suggestions. *"At the end, you need to read the response and then you need to decide that either I need to go with the result or not." (P1)* Participants frequently double-checked whether an AI-generated user story truly matched the intent of the original feedback. P3, for example, checked ChatGPT's answers about Zoom features against their knowledge to ensure no false information was missed. One participant noted that ChatGPT *"sometimes didn't understand the assignment and its answer is not valid to the question"*, attributing this to the limitations of the AI tool. Across the ten participants, I found that the tipping point typically occurred midway through the session, when hallucination issues and trust breakdown often occurred.

Bias and Ethical Considerations:

When it came to sensitive issues, all ten participants showed how ChatGPT could exhibit biases or omit ethical considerations unless prompted. P3 pointed out that

it is important to confirm certain user reports about accessibility issues, implicitly recognizing that the AI might not be able to identify the human impact or fairness aspects behind a request.

In another instance, participants were aware that ChatGPT, like any AI, has inherent biases learned from the training data. When I asked participants if they felt their decisions were biased by the answer that ChatGPT provides to them, participants often admitted that there was some influence. *"You are more inclined to pick whatever is generated."*

Participants are very aware that ChatGPT can miss key elements from the data, *"It was just summarizing and getting some keywords and really kind of missing the point of what they [the posts] were trying to mention. So I needed to iterate a lot, and most of the time I just ended it by reading it all myself, just because the quality was really bad." (P9)* Ultimately, practitioners do not want to be held accountable for missing features simply because they relied on ChatGPT and it overlooked important details.

This highlights a risk that requirements practitioners should be cognizant of, particularly when the results of an AI tool can bias practitioners about what is important. Since participants were selected based on their industry experience, each of them made a conscious effort to minimize bias and cross-checked the original feedback. As P9 mentions, *"Accessibility was a big one for me [when I manually read the posts], so it's weird that it doesn't come up [in ChatGPT's analysis]. So how do you [know] for sure, and then I would say also privacy. I'll just go back and have a look honestly, I don't trust [ChatGPT]."*

Limitations in Context and Understanding:

Finally, nine out of ten participants also bumped into the fundamental limitation that ChatGPT can only process the data that a user feeds it, and it only knows what it has seen in its training data. In traditional requirements elicitation, a product manager or requirements practitioner will typically have the opportunity to talk to the stakeholder and try to build an in-depth understanding of the problem. More often, the employee will have the opportunity to ask clarifying questions to the user or pick up on facial or tonal cues if they can see them.

Requirements engineering is ultimately a "co-creation process between analysts and stakeholders" and "eliciting goals for uncovering stakeholder's true needs" [231].

However, currently, it is challenging for an AI tool like ChatGPT to replicate these human interactions. Moreover, AI tools have been found to struggle with emotional qualities [232]. P6 also highlights, *"AI is not good at detecting if there are some lies hidden within [the user feedback]." (P6)*

According to the participant, the only input an AI tool is getting is text that the participant copies and pastes into the prompts. *"It cannot replace a face-to-face video interview where I'm seeing some some of your facial expression."* P3 also reflected on this, explaining that AI cannot capture expressions such as frustration.

As a result, if a user's post was sarcastic or emotionally charged, the AI might miss that context and produce a tone-deaf user story. ChatGPT does not understand real world perspectives, meaning it cannot determine why a post is important to a user. At the same time, ChatGPT does not know the context for the organization's product lines, business goals, or product history. These are all important information to make analytical decisions about the requirements.

Unlike ChatGPT, when a participant analyzes user feedback from their company, they rely on their past experiences within the company, including knowledge about compliance, the product line, and the company's overall goals. *"I would rely on my personal reasoning. For me, it's related to the context where I am, the knowledge about the system, the knowledge about the company, the knowledge about the kind of regulation or a framework that I need to be compliant to it, or that I should fit in." (P8)* While ChatGPT can help analyze text, a practitioner acknowledges that their contextual knowledge about their organization and rules pertinent to the software are still important for understanding user feedback.

7.3 Discussion

In a previous study, Ronanki et al. [86] compared the requirements generated by ChatGPT (GPT-3.5) with those written by five human experts. They measured the quality of the requirements across seven metrics, including abstraction, atomicity, consistency, correctness, ambiguity, understandability, and feasibility. The authors found that ChatGPT was able to generate comparable user stories of the human experts.

Nevertheless, these studies were structured experiments focusing on the quality of ChatGPT generated requirements rather than how requirements practitioners rely on and use these AI tools in practice. Therefore, there is a significant gap in our

understanding of how practitioners can use AI tools for the development of new requirements in industry. My work and findings fill this gap by leveraging a think-aloud method to observe and identify the practices that practitioners use when conducting requirements development. In the next section, I will explain some recommendations that could help practitioners navigate the challenging landscape of leveraging AI tools for requirements development.

7.3.1 Recommendations for Practitioners: Strategies for Leveraging AI Tools

"Human-in-the-Loop" Prompting

Many participants found it most effective to start the analysis themselves, then use ChatGPT to build upon or refine their work rather than the other way around. By giving the AI an initial frame, for example, a tentative decision or a draft user story, they communicated their intent to the AI. P1's approach of stating *"what I think"* - (*ChatGPT Prompt*) about each feedback before asking ChatGPT to respond is a prime example.

This strategy of prepping the AI tool ensured that the AI had a shared understanding with the participant. As P1 explained, this let them maintain their voice and direction *"I want my initial thinking to be captured."* In contrast, when participants tried prompting without setting the context, they sometimes got unrelated answers and had to iterate.

Additionally, every participant treated prompt writing as an interactive, iterative process. Instead of asking for everything at once, they broke down tasks. For instance, P6 first asked for common themes observed in the dataset, then followed up with additional instructions to add effort and risk to each user's feedback. This stepwise approach meant the AI's focus was narrow at each step, reducing confusion.

P5 recounts how they would *"give different prompts [telling ChatGPT] this is not what the feedback meant"* and gradually steer the AI tool to the correct interpretation. Similarly, another participant would push ChatGPT to clarify until the result was clear to the participant. Like working with a human assistant, the participants would assign a requirements subtask, review the result analysis (e.g., actionability, theme, or user story), and then ask for improvements if not satisfied.

Explicit Instruction and Constraints in Prompts

Participants explained that the way they phrased their prompts significantly affected the quality of ChatGPT's output, and they adjusted their prompting techniques accordingly. They frequently gave ChatGPT detailed instructions or format constraints to get the desired result.

For example, P7 requested user stories in a specific format: *"Given/When/Then format with Acceptance Criteria"*, which yielded structured acceptance tests that required less editing. Other participants would add a preamble to prompts such as *"refine"* or *"Write user story in the format of "As a user... I want this... to perform this action.""* In particular, one creative tactic was the use of pseudo-XML tags in the prompt. *"<instructions></instructions> <feedback></feedback>"*

This helps ChatGPT distinguish between a user's feedback and the practitioner's instructions, leading to a more structured and relevant answer. By crafting prompts with clear intents, practitioners avoid generic or misinterpreted answers and receive outputs that align with what they are looking for.

Knowing When Not to Use AI

Participants did not apply ChatGPT to every single subtask. Instead, they first assessed the nature of a subtask.

For straightforward, tedious tasks, such as formatting or refining the grammar of a user story neatly, they were content to delegate to ChatGPT. However, for tasks requiring deeper insight or company-specific knowledge, such as prioritizing themes, some elected to do the work with significant manual involvement.

P3 illustrated this trade-off. They used ChatGPT to suggest theme names, but when it came to consolidating those themes into final categories, P3 expressed that they preferred to allocate more cognitive effort to this manual labor.

Critical decisions were kept human-centric. This selective use is a strategic choice because it maximizes the benefits of efficiency, while reducing the risk on aspects where the AI may cause significant issues. Knowing when not to use AI is just as important as effective collaboration with AI.

7.3.2 Recommendations for Researchers

Based on the findings from this study, I provide the following recommendations for researchers.

Evaluate the AI interaction practices:

I synthesized four practices for how practitioners use AI for requirements development. Future work can empirically evaluate these practices to identify qualities, such as identifying the practice that results in the most efficient or effective outputs, or which practices can be applied in combination. Future researchers can create controlled experiments to test these practices.

Investigate prompting strategies for human-AI collaboration:

In this study, I discovered practitioners using different interaction approaches. Future research should systematically explore how different prompting styles and interaction paradigms (e.g., iterative refinement versus single shot prompting, human initiated versus AI initiated analysis) influence the accuracy, efficiency, and overall effectiveness of the requirements development process.

Ultimately, the goal of this study was not to determine a single best practice for practitioners to follow. Rather, the primary objective was to explore and characterize the ways in which requirements engineering practitioners engage with AI tools during analysis. The aim was to inform future tool design and practitioner support.

Additional studies could employ controlled experiments or industrial case studies. This would help evaluate both the quality of AI generated outputs and the cognitive load and decision making processes of practitioners when working collaboratively with AI.

Examine ethical implications and cognitive biases introduced by AI

Researchers should explicitly investigate cognitive biases such as anchoring and adjustment bias, and misleading information bias [233] that may be prevalent among practitioners for requirements development. Future work could involve studies explicitly designed to reveal biases inherent to AI generated content.

Studies can further conduct a qualitative investigation into practitioner awareness and mitigation strategies. Additionally, research could propose training programs,

prompting techniques, or AI explainability features to reduce cognitive biases.

Strategies to improve AI's contextual understanding

A limitation highlighted by this research was the AI's inability to accurately interpret contextual cues. Further research should aim to enhance AI's understanding of requirements development in context. This could include investigating novel prompting methods or fine-tuning large language models on specialized RE datasets to enrich AI understanding.

7.3.3 Threats to Validity

To ensure the rigor of this study, I use the total reliability framework from Roller [120] and the aspects of credibility, analyzability, transparency, and usefulness to assess the threats to validity.

Credibility

Credibility refers to the accuracy of the data and coverage. A potential credibility concern in this study stems from the use of the think-aloud protocol. Since participants verbalized their thought processes, this may have altered participants' natural workflows, making them more talkative or reflective than usual. Additionally, the presence of the researcher may have led participants to adjust their behavior due to perceived researcher expectations as part of the observer effect. To mitigate these threats, participants were explicitly informed that there were no predefined "correct" approaches and encouraged to interact with ChatGPT as naturally as possible.

One other credibility concern is the compensation payment to participants. Each participant was paid $30 for their time, which could result in participants joining the study solely for the money. However, I tried to mitigate this concern by finding participants who have extensive experience in the industry to ensure they would be a good fit for this study.

Another consideration is the variability in participants' prior experience with ChatGPT, potentially influence their comfort and effectiveness with prompting strategies. While I provided consistent onboarding and introduction, differences in comfort level and prompting expertise could have influenced individual performance. To strengthen the credibility of my interpretations, I used data triangulation, drawing

from screen recordings, think-aloud transcripts, and post-task interviews to cross-validate emerging insights.

Analyzability

Analyzability reflects the accuracy of the analysis and interpretation of the results. Each participant followed a standardized set of subtasks that I explained in detail before beginning the think-aloud study session.

One concern is that variation in AI prompting strategies and familiarity with ChatGPT may have introduced some inconsistency in how a participant accomplished a task. However, these inconsistencies are to be expected in an exploratory qualitative research and help me identify various real-world practitioner behaviors.

To improve analyzability, I ensured that each participant received the same task materials and was given the same instructions. Each participant also received the same dataset and used the same AI tool. To minimize risk, I grounded the analysis in multiple sources of data, including screen recordings, verbal protocols, and post-task interviews. Additionally, I used participant quotes and notes about their interactions to ensure that the synthesized themes were based on the collected data. While complete non-bias is unattainable in qualitative work, I did my best to ensure that my analysis was guided by the data rather than any assumptions.

Transparency

Transparency concerns the extent and completeness to which I disclose the study. To increase transparency, I provide details of the dataset that I used in the study. I described the methodology of my think-aloud study and provided participant quotes wherever relevant to the findings. I documented the data collection procedure and analysis steps to support future replication or comparison.

Usefulness

Usefulness concerns the extent to which the study's findings apply to other contexts. Due to the qualitative nature and limited sample size (i.e., ten participants), the generalizability of identified themes and insights could be limited. The research setting that I chose may differ from real-world requirements engineering practices. In practice, requirements elicitation often involves a longer time, negotiation with stakeholders, and handling of confidential or domain-specific data.

Nonetheless, I aimed to create a scenario that resembles common industry practices. I chose an application that is popular and picked data that is from Reddit, an extremely popular source of user feedback. The number of practitioners could limit the representativeness of the broader practitioner community, but I ensured that all participants analyzed the same publicly available user feedback.

Moreover, I studied only one type of generative AI tool (i.e., ChatGPT) as it is the most popular available. However, I expect other similar AI tools, such as Claude or Gemini, to provide similar results. The AI tool landscape is very rapidly evolving, and companies frequently release new models, but the industry leaders' models are roughly on par with each other at the moment. Future work could expand transferability by studying various industry settings, larger teams, different tools, and feedback sources across different organizational and domain contexts.

7.4 Conclusions

As described earlier in Chapter 3, analyzing user feedback at scale is far from trivial. Moreover, in Chapter 6, I showed how software practitioners and software organizations are heavily adopting AI tools to improve their productivity. In this chapter, my think-aloud study unveiled that practitioners can leverage an AI tool like ChatGPT to help with their requirements development for software products.

However, this study revealed that significant human oversight is still essential. Requirements engineering is fundamentally a socio-technical discipline. Creative collaboration between requirements practitioners and stakeholders is fundamental to conducting effective requirements engineering [231]. This research has shown several areas, including contextual understanding and bias issues, where AI tools still exhibit limitations.

Leaning on AI tools to assist with the development of new requirements from user feedback does not absolve practitioners from responsibility. Rather, practitioners are cognizant of the potential limitations that can arise from using AI tools, particularly overly relying on AI responses without cross-validation with practitioners' manual analysis and contextual knowledge.

In an ideal world, we can give a user feedback dataset to an AI tool where the tool can easily perform the end-to-end development of new requirements from user feedback, and we receive a perfectly curated set of requirements, but my study showed that we are not there yet. Practitioners in the study demonstrated a "Human-in-the-

Loop" approach to integrating AI tools involving iterative refinement of requirements artifacts and cautious validation of AI generated responses.

Chapter 8

Conclusion and Future Work

Through six empirical studies, this dissertation explored and detailed the automation of user feedback analysis for requirements engineering. From the aforementioned qualitative and quantitative studies, I have provided empirical evidence, advancing both theoretical understanding and practical implications for the industry.

My exploration provided significant insights into my overarching research goal: 1) how software organizations manage user feedback, 2) how to automate user feedback analysis, and 3) how we can leverage generative AI tools for the development of new requirements from user feedback.

My first and second research goals resulted in three pertinent RQs,

1. How do software organizations manage user feedback to improve existing products?

2. How can we automate the development of new requirements from textual-based user feedback?

3. How can we automate the development of new requirements from video-based user feedback?

These RQs resulted in Chapters 3-5. The first RQ was answered through a grounded theory study with 40 software practitioners about how they managed their user feedback. RQ2 involved analyzing user generated textual data from platforms such as Reddit, identifying privacy-related discussions.

I leveraged unsupervised clustering techniques and language models to derive requirements related themes from the large dataset, demonstrating that such automated methods can identify user concerns. I also demonstrated the effectiveness of applying

narrative analysis to explain trends in the user feedback. Further extending this approach, I explored video-based feedback from TikTok and YouTube, uncovering a rich trove of user-generated content previously unexplored for requirements engineering.

I addressed third RQ by leveraging multimodal analysis integrating audio, visual text, and metadata, my colleagues and I successfully classified video content into requirements-relevant categories. I demonstrated the viability and usefulness of our automated approach in handling video-based feedback formats.

For my third research goal of exploring how we can leverage generative AI tools to help with the development of new requirements from user feedback, I elicited two additional RQs: RQ4 and RQ5.

1. What impacts AI adoption and use in software engineering?

2. How are requirements practitioners leveraging generative AI tools to conduct the development of new requirements from user feedback?

As part of RQ4, my research explored general AI tool adoption in software organizations through a socio-technical grounded theory approach. The study elucidated organizational and individual factors influencing AI adoption, such as productivity gains, organizational culture shifts, and potential barriers, including privacy concerns and ethical considerations. The resultant theoretical framework provides insights for both researchers and practitioners.

Addressing RQ5, my sixth empirical study explored it from a practitioner's perspective, specifically examining how requirements practitioners use AI tools in practice. The study resulted in practices, challenges, and recommendations for integrating AI during user feedback analysis and requirements generation. These insights not only aid practitioners in effectively using AI tools but also guide future AI tool development.

8.1 Contributions

My research has delivered multiple contributions, starting with a detailed investigation into industry practices for managing user feedback. I have iteratively identified and synthesized the challenges organizations face, particularly regarding the difficulties and opportunities of social media as a feedback source.

To address these challenges, I developed a structured life cycle of managing user feedback, encompassing stages including collection, analysis, validation, and prioritization. Moreover, I bring forth insights on the best practices, as self-described by software organizations, that enable organizations to effectively manage user feedback. The life cycle and best practices serve as practical guidance for software organizations to facilitate the widespread analysis of meaningful requirements and relevant information from diverse user feedback.

Additionally, this dissertation contributes in terms of empirical evidence for leveraging LLMs to process textual and video-based user feedback. I also provide empirical insights about the challenges and motives for practitioners to adopt and use AI tools for software practitioners. Finally, this dissertation contributes insights about using AI tools to process user feedback from the crowd for requirements development. In particular, Figure 8.1 shows how my research fits into the requirements development process with AI.

Figure 8.1: Requirements practitioners use of AI for the development of new requirements

8.2 How can a Requirements Practitioner Use Machine Learning and AI Tools to develop new requirements from CrowdRE User Feedback

The studies I described in this dissertation have spanned a wide range of empirical investigations about automating the requirements development process. Reflecting on these studies, I synthesize a **CrowdRE-AI process** that integrates CrowdRE, machine learning techniques, and AI tools.

This process relates to my research goal and attempts to answer how we can automate and leverage AI tools to process user feedback for the development of new requirements. My process explains how practitioners can approach the management

and analysis of large volumes of user feedback in the future.

The process comprises three steps:

1. **Collect User Feedback Sources**: At the core of this process is first recognizing that a software organization has important user feedback that it should and must consider. As described by the seminal CrowdRE work [5], an organization benefits from obtaining and analyzing relevant user feedback from the "crowd."

   My study in Chapter 3 on how organizations are managing user feedback sheds important light on the wide variety of feedback that organizations give significant resources. More importantly, those organizations highlight how video and text based feedback from online platforms is growing in consequence. Therefore, the first important step is for practitioners to recognize the diverse feedback channels, whether textual or video-based, and identify the relevant sources critical for the organization's software evolution. For example, for a business to business (B2B) company, they may find LinkedIn as the most appropriate source of feedback regarding what corporate clients are looking for in updates and what competitors are hoping to release.

   Moreover, recognizing these newer online-based user feedback is numerically significant, and traditional approaches for manual analysis can no longer suffice. To assist in identification and management, my work in Chapter 3 provides a life cycle that documents the activities for managing user feedback, including collection, analysis, validation, and prioritization.

2. **Use Large Language Models**: Once an organization identifies and collects the relevant user feedback from the "crowd," the next step is leveraging large language models to transform the vast, user-generated feedback into more structured insights. In Chapter 4 and 5 of my dissertation, I detailed several empirical studies that I conducted concerning textual-based and video-based user feedback.

   I showed how it is possible to conduct large-scale analysis to identify which feedback is actionable for requirements (i.e., feature requests, bug reports) and what kind of themes are users complaining about. Through automated approaches such as clustering, classification, and multimodal analysis, this second step can help filter and group raw user feedback, which may reduce human effort and support more scalable development of requirements.

3. **Include AI Tool as an Assistant**: Finally, the third component of my process is the most crucial step as it is the step that involves the AI tool to help primarily automate the manual effort. Specifically, this step emphasizes the interactive relationship between requirements practitioners and AI tools (i.e., ChatGPT or other comparable tools).

   Requirements engineering, at its core, is inherently complex and ill-structured, involving conflicting stakeholder goals and evolving requirements [234]. As an example of a "wicked problem", requirements engineering is difficult as there is no single correct answer, and each software project is unique.

   While it would be ideal for an AI tool like ChatGPT to solve the problem with manual requirements development, particularly for large quantities of online-based user feedback, AI tools do not (yet) offer a silver bullet. As highlighted in my Chapter 7, AI tools are not yet prepared to receive an Excel file and produce perfectly accurate and reliable requirements that organizations can use.

   Instead of handing over all user feedback analysis tasks to an AI tool, practitioners need to adopt a more iterative and collaborative interaction style. My empirical study in Chapter 7 sheds insights that the use of AI tools still demands significant human oversight and considerations concerning privacy and ethics.

   However, practitioners can achieve meaningful benefits from using AI tools if they provide explicit instructions and constraints in their prompts, such as the XML tags approach that my participant demonstrated.

   As coined by Andrej Karpathy, "vibe coding" is a new development phenomenon gaining popularity among developers where developers code iteratively with the help of an AI tool [235]. Similarly, for automating the development of new requirements from user feedback, my study participants in Chapter 7 showed how it is possible to conduct human-in-the-loop prompting with AI tools, where an AI tool is used to build upon or refine their analysis work.

   Acknowledging that at this moment, a generative AI tool is perhaps not yet capable of completely conducting creative, intensive analysis of user feedback. People still need to prepare the AI tool to ensure that humans control the voice and direction of the analysis. Ultimately, practitioners can benefit from the

use of AI tools for automating the development of new requirements from user feedback as long as they are aware of the faults and challenges and apply the tool as a form of assistance, not a replacement.

8.3 Final Conclusion

While the field of requirements engineering has existed for decades and is paramount for building the right software product, requirements engineering is still a complex aspect of software engineering. The motivation behind my work is developing more understanding in the problem space of analyzing user feedback at scale from the "crowd" with the assistance of first LLMs and later on AI tools.

My work in this dissertation has exemplified how both practitioners in the industry are transitioning towards AI tools and are negatively impacted by the ongoing challenges of trying to automate the development of new requirements from user feedback. Yet, the silver lining of my insights is the significant strides that we made toward leveraging AI tools and LLMs to make sense of the bulwark of user feedback collected from different online-based sources.

Ultimately, the adoption and integration of AI tools for automating the development of new requirements from user feedback is not without challenges. Unfortunately, analyzing user feedback from the crowd at scale is not as trivial as simply feeding a spreadsheet full of feedback into ChatGPT and expecting an instant answer. Therefore, it was important in my work to conduct empirical studies to understand how product managers, product owners, and other requirements practitioners can use these AI tools. It was paramount to understand, as a first step, where practitioners are even looking for relevant feedback, before conducting the important analysis.

In my empirical work, I unveiled that requirements practitioners are indeed increasingly using AI tools such as ChatGPT for automating the development of new requirements from user feedback. Many of them follow ad-hoc practices, such as first *manually analyzing the feedback before using AI tools* to help refine the requirements artifacts. Meanwhile, practitioners also have to be continuously cognizant of the *bias and ethical risk* created by AI tools. To the best of

my knowledge, my work is among the first to explore how we can better use AI tools for automating the development of new requirements from user feedback. Based on my research findings, several promising directions emerge from this research, offering significant opportunities for future exploration.

First, future work should explore additional approaches to automated requirements engineering. The entire requirements engineering process is quite complex, with several different activities. Ongoing challenges include requirements tracing and managing the complexities of growing software projects and their related issue tracking items.

Second, large language models and AI tools are known for propagating biases due to training data or other related subjectivity. However, if these biases are not considered, leveraging such LLMs or AI tools could manifest in biases for requirements engineering tasks. Additional research and study are necessary to understand the extent of the impact of bias in LLMs and AI tools and their impact on requirements engineering. Moreover, future research should consider exploring mitigating the negative impacts of such biases to foster more ethical AI tool and LLM use.

Third, my research highlighted the ongoing challenges and potential of AI adoption. Future research could investigate long term and longitudinal case studies with software organizations regarding the adoption and integration of AI tools. Such research would provide deeper insights into the evolving impacts of AI tools on organizational productivity, culture, and practitioner workflows.

Fourth, despite the proliferation of AI tools, my final study identified that it is still difficult to fully automate the requirements engineering process. In particular, I identified their practical approaches, challenges, and strategies for using AI during user feedback analysis and requirements generation. These insights provide practical implications for both practitioners and researchers seeking to leverage AI in requirements engineering.

Finally, this dissertation has improved the understanding and capabilities regarding the automation of user feedback analysis for requirements engineering through large language models and AI tools. By addressing the gaps and providing theoretical and practical insights, I paved the way for further innovations in software engineering.

Appendix A

Tool Based Automation of User Feedback Data

Table A.1: Reddit User Feedback about Zoom

ID	Title	Body Text
UF1	Is it possible to accidentally share your screen with a single Zoom participant from those in attendance?	I was hosting a Zoom meeting today with about 40 participants. I didn't think I was sharing my screen but one person said they could see it (made a joke alluding to what it was). I asked other participants and they said they couldn't and they would have told me if they could. Is it possible that one person had access to my screen? Is there a setting that allows that or prevents it? Thankfully it was completely innocuous but it did get me thinking. I would like to be sure about privacy because some of my work contains personal information. Thanks in advance!

Continued on next page

ID	Title	Body Text
UF2	Light reflecting from my monitor	This may sound really goofy, but it's true and I could use help from people smarter than me. I'm legally blind so I have a giant monitor, 43 in to be exact. I've noticed that light from the monitor is reflecting into my face during Zoom meetings and I look really terrible, way too bright with lots of highlights. Changing the settings on my Logitech software or the few settings available in Windows doesn't help at all. The bottom line is I look horrible and I don't know if there's anything I can do to look better on my multiple Zoom calls everyday. I would be up for any ideas that might be out there to help me either cut down the blue light coming off the monitor or otherwise make Zoom work better. Thanks in advance for anything you can share!
UF3	Is Zoom safe to download?	I've been using the browser based version for online language lessons and I hate that it crops off my background. I would definitely prefer a wider view and also I would like to set my camera to be mirrored. Just read on here so many not so good things about Zoom desktop app. Particularly in terms of privacy and safety. Has things improved since those security scares happened? Thoughts?

Continued on next page

ID	Title	Body Text
UF4	No audio. No "audio" tab. Tests work but can't hear others and they can't hear me. Volume is up.	Hello, I am not new to Zoom and all was working just fine the last time I used the app, but tonight for a meeting, the "test your microphone" and "test your speakers" worked just fine, but then I could not hear anyone in the meeting, and the "join audio" icon at the bottom of the screen only had two options: test the microphone or speakers and link to audio settings. Audio settings also only had two options related to phone numbers. I was on a desktop. When I went to the web page for Zoom help, it said to click on my profile pic (did that), then click settings (did that), then click the "audio" tab - there was no audio tab. Only an "audio conferencing tab" with the two options for phone numbers (show international numbers was one). Assuming "audio conferencing" was correct (because there were no other options), I followed the instructions and clicked "advanced" and again, no settings for audio. How can I adjust audio settings when there are none? Thank you!

Continued on next page

ID	Title	Body Text
UF5	Can Zoom detect me trying to record a shared recording of a meeting?	Hello everyone, recently I wanted to download recordings of a lot of meetings that had important things but the meetings couldn't be downloaded due to the host probably restricting it. The download option doesn't come up. I was wondering if I could record these recordings with third party recording software. They are shared recordings with links. However I don't know if Zoom or the host of these meetings or anything else could detect this recording software (like OBS which I use). This recording would be only for me. I don't want to post it online. Just for personal use.
UF6	Black Screen when signed in	Hello—For some reason every time I login to my Zoom account on any device (Mac or PC) my screen is black. When using just the desktop client without logging in it's fine. I have checked all permissions, AV and access on the settings portal on each device. Anyone else ever have this issue? Thanks in advance.

Continued on next page

ID	Title	Body Text
UF7	Captions on Recording Not Available When Downloaded???	For some background I go to a tech school that has a big deaf student population. I get that it's not difficult to add transcripts to a video, but I find it ridiculous that a massive company like Zoom doesn't have the ability to just keep the captions on the video when it's downloaded. It downloads the transcript alongside the video, but it's still insane that it downloads the transcript separately. Captions should be incredibly basic at this point, it's adding words to the video. It's an accessibility issue and should be very easy to fix. Am I missing something? Thoughts? Edit: Download needed to submit assignment.

Continued on next page

ID	Title	Body Text
UF8	Black screen when I screen share	I host virtual workshops a lot for work (through a paid Zoom account) and I only have 1 monitor. So to be able to see my notes but also have the participants see the slides, I use the "share portion of screen" option on Zoom. Then I resize it to the slide but cut out my notes. This has worked well for 2 years. On Friday I was pre-recording a presentation (something I've also done before using this method), and when I watched my presentation back, the screen shared was completely black. Didn't show my slides at all. I've been trying to troubleshoot it since then and nothing works. When I screen share my whole screen, it's normal and works well. When I screen share a portion of my screen, it's black and doesn't show anything. There are troubleshooting tips online but my Zoom doesn't have the settings mentioned in the steps online. If anyone can help me it would be sooo appreciated!
UF9	Change Font Size of Zoom Itself, Not Chat/Caption	I can barely read any of the menus in the settings or when I'm on a meeting. I only see the option to change chat and caption size, but not the actual app itself. Even with my font size larger in Windows 11, the menus in Zoom are like 8 point font.
UF10	Changing font size in polls and Question and Answer? Already have chat	Hi all, I've found the settings/accessibility option to increase the font size in the webinar chat window, but that doesn't scale up the text used in the Question and Answer or Polls pop-out boxes. Can anyone tell me how to increase those, as I've had no luck finding any such option. Thanks for any help!

Continued on next page

ID	Title	Body Text
UF11	zoom making my pc audio weird	So I've just joined a Zoom call and I have music playing and whenever someone in the meeting is talking my whole PC audio gets quieter and it cuts in and out of being quieter, and it's really irritating! Also I use Opera GX browser and I have the keyboard sounds on there and they sound like they go pop when people aren't talking and it's very loud! Does anyone know why this is happening/how to fix it?
UF12	Unique zoom issue... maybe?	Having an issue with Zoom where the virtual background will work for a time, then, if I move at all, the edge outline will ghost in place and that little part will just be un-blurred and show the background. This does not fix itself. Just stays unblurred in the shape of me until restarted. Anyone know what causes that? It happens randomly at times during the call.
UF13	I can turn myself into a cat but I can't add a logo as an overlay?	Zoom now has some pretty extravagant functions for the camera: virtual backgrounds, face filters, 3d avatars. I can change the color of my lips etc... but how is it I can still not simply add a logo to the top left of my screen (not a virtual background) without installing 3rd party software? I use OBS for all such things in my own studio but I wish I could easily install a simple logo on the computers of my clients.

Continued on next page

ID	Title	Body Text
UF14	Zoom settings/ preferences close out when a meeting starts or breakout room opens/ends	I am using Zoom desktop app on a Mac. When I open the settings to check my video or audio, they automatically close when the meeting starts. Let's say I join a meeting and I am waiting for the host to start it, so I check my settings, the settings/preferences window closes out automatically when the meeting actually starts. Same for breakout rooms—if I have my preferences open while in a room, they exit on their own when the breakout room closes. Is there any way to change or stop this behavior??? I find it ridiculously annoying to have to re-open my video settings or whatever I'm doing, especially because I like changing my virtual background. Thanks.
UF15	How to lower volume of app through screen share?	I use an app called iReal Pro on my tablet while on Zoom and it was too loud today. I went to the mixer in iReal Pro and lowered the volume, but the volume did not change for the person on the other end. I am not sure if this is an issue with iReal Pro or Zoom, but is there a way to lower the volume of an app through Zoom?
UF16	Dear Zoom Accessibility Team,	I hope this message finds you well. My name is X, and I am writing to bring to your attention some significant accessibility barriers that I have encountered while using Zoom. As a user with disabilities, I have found the current limitations in Zoom's free services to be not just frustrating, but discriminatory... [Shortened for brevity]

Continued on next page

ID	Title	Body Text
UF17	Audio Options Missing for Phone Dial-In. Any Solutions?	My Zoom settings have previously only allowed computer audio. Recently, I have had a need for an option for attendees to dial in via phone. However, that option is no longer available within my account under options. Is that hidden somewhere or is there something I should change or look for elsewhere?
UF18	Audio quality decreases during zoom meetings	Whenever I join a Zoom meeting the audio quality from all sources decreases. I am using headphones and it's like this even when I take it off. Not only does the audio quality from the meeting sound a little muffled, other apps such as Spotify and Chrome are also affected. Please help.
UF19	Recording call for quality control	Hey everyone, I run a small company, and since we went remote, I've been struggling with keeping track of how employees interact with clients. I want a way to automatically record client meetings for quality control so that I can see how my employees are handling conversations while maintaining. (maybe via a third-party integration). I would like use this feature to review client calls.

Continued on next page

ID	Title	Body Text
UF20	Silent Zoom Monitoring for Parents	I've been using Zoom to keep track of my kid's online classes, but I wish there was a better way to monitor without being disruptive. Right now, if I join a call, he sees my name on the participant list, and I don't want him to feel self-conscious. As a parent, I want an option to passively observe my child's online class in a way that does not disrupt their learning but still allows me to ensure they are staying engaged. It would be great if there was a parental monitoring mode where we could silently observe the session. Schools already record some classes, so this wouldn't be much different, but it would help parents stay informed without making kids feel watched. I think this could be useful for younger kids or students who struggle with focus. Has anyone else thought about this?

Figure A.1: User Instructions for AI Assisted Labelling

Study Instructions

Please read all the instructions first before starting the task!!!

You will need to make use of ChatGPT for this exercise

Task Overview:

You will be provided with 18 Reddit posts for one software app. You will act as a product manager/owner/requirements engineer who uses an AI tool (**ChatGPT**) to analyze user feedback and write requirement tickets (user stories) for the developers to work on. You will first use the AI tool to analyze the 20 Reddit posts and categorize them into different themes. Then you will use the AI tool to write user stories based on the feedback. You will have to record your screen while doing the task and submit the screen recording. You may use the **AI tool** to assist with **every step** of this assignment. Your primary goal is to make proper themes and user stories that will help developers implement the product. We will be following **think-out-loud** approach during the study, so please remember to **speak** whatever you are thinking as you are doing the analysis. We want to understand your thought process for labeling the user feedback.

Detailed Steps

1. **Understand the Dataset**:
    a. You will receive a set of 20 Reddit posts from one software app.
2. **Categorization**:
    a. Read each Reddit post carefully.
    b. For each post, indicate if the post is actionable or not actionable. Actionable refers to feedback that is specific, relevant, and detailed enough to be directly translated into a user story for development. Please use column D in the Excel sheet **"Posts"** to indicate whether a post is actionable or not. If a post is actionable, write "yes" otherwise "no".
    c. Next, determine the common recurring themes that come up from the feedback. A feedback post (i.e., a reddit post) could have one or more themes if the feedback discusses multiple different topics (e.g., “There is no data transparency” could be under a theme named “Privacy Challenges”). Please provide a list of all the relevant themes to the dataset in the Excel sheet “***List of Themes***”. You should have at least 2 themes in total.
    d. Please describe and justify why you create **each** feedback theme (e.g., “Privacy Data Challenges”) for each theme in the Excel sheet “***List of Themes”.***. Please describe why you came up with the theme. Your justification should answer these questions:
        - How did you develop the theme? Please describe all the steps and thoughts you had to develop the theme.
        - What steps did you take to reach the conclusion that this theme captures the feedback posts?
        - What did you think when creating a theme, was it a result of specific user feedback? Was it a result of a group of user feedback?
        - Did you rename any theme, why? How did you reach your decision in the end?
        - Did you merge different themes together, why?

3. **User Story Generation**:
    a. Please create **at least 5 user stories** that you think would solve the issues indicated by the user feedback. Each theme should have at least one user story under it. Use your knowledge and best judgment to make user stories. For those that you make, please provide justification for why you did. If there is any user feedback that you didn't use to make a user story, please explain why you think it is not needed. You can follow the **Atlassian's guidelines**. Please list each user story in the sheet "***User Story".***

    Note: Multiple user stories can be made from a single user feedback, and vice versa, a single user story could cover the topics raised in several user feedback. 5 user feedback does not necessarily mean you have to create 5 user stories.

    b. Each story should:
        i. Be actionable and relevant
        ii. Avoid duplication to minimize repetitive work
        iii. Address the primary concern of the feedback
        iv. Example Format: *As a [user type], I want [action or feature] so that [goal or outcome]*
    c. Provide **at least 1 acceptance criteria** for each user story. Please write this in the "acceptance criteria" column in the excel sheet "***User Story***".
    d. **Justify** and **explain** why you decided to write each user story.

4. **Traceability**:
    a. Ensure traceability for the user stories from Reddit posts and themes.
        i. In other words, a user story should indicate
            - which theme it belongs to. Please state the relevant theme(s) in the columns "Which theme does this belong to" in the Excel sheet "***User Story".***
            - which posts it is based on. Please state the relevant feedback in the column "Which user feedback does this belong to?" in the excel sheet "***User Story".***

5. **Submission Requirements**
    a. Complete the Google Spreadsheet with your results
    b. Submit link(s) to your chat with the AI assistant that included your prompts with AI to help you with your analysis work
6. **Estimated Time Commitment**
    The task will take approximately **60–80 minutes** to complete, but feel free to take more or less time as needed.
7. **Important Notes**
    a. Focus on creating unique and actionable insights
    b. Avoid overgeneralizing or merging unrelated feedback into a single theme

Bibliography

[1] F. P. Brooks and N. S. Bullet, "Essence and accidents of software engineering," *IEEE computer*, vol. 20, no. 4, pp. 10–19, 1987.

[2] C. Werner, Z. S. Li, D. Lowlind, O. Elazhary, N. Ernst, and D. Damian, "Continuously managing nfrs: Opportunities and challenges in practice," *IEEE Transactions on Software Engineering*, vol. 48, no. 7, pp. 2629–2642, 2021.

[3] A. Aurum and C. Wohlin, *Engineering and managing software requirements*. Springer, 2005, vol. 1.

[4] J. Tizard, T. Rietz, X. Liu, and K. Blincoe, "Voice of the users: A study of software feedback differences between germany and china," in *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*. IEEE, 2021, pp. 328–335.

[5] E. C. Groen, J. Doerr, and S. Adam, "Towards crowd-based requirements engineering a research preview," in *Requirements Engineering: Foundation for Software Quality: 21st International Working Conference, REFSQ 2015, Essen, Germany, March 23-26, 2015. Proceedings 21*. Springer, 2015, pp. 247–253.

[6] C. Gralha, D. Damian, A. I. Wasserman, M. Goulão, and J. Araújo, "The evolution of requirements practices in software startups," in *Proceedings of the 40th International Conference on Software Engineering*, 2018, pp. 823–833.

[7] F. Dalpiaz, "On the value of crowdre in research and practice," in *5th International Workshop on Crowd-Based Requirements Engineering (CrowdRE'21)*, 2021, discusses the importance of user feedback in developing Crowd-Based Requirements Engineering to address the challenges of traditional requirements engineering methods. [Online]. Available: https://crowdre.github.io/ws-2021/

[8] E. C. Groen, N. Seyff, R. Ali, F. Dalpiaz, J. Doerr, E. Guzman, M. Hosseini, J. Marco, M. Oriol, A. Perini *et al.*, "The crowd in requirements engineering: The landscape and challenges," *IEEE software*, vol. 34, no. 2, pp. 44–52, 2017.

[9] E. C. Groen and M. Koch, "How requirements engineering can benefit from crowds," *Requirements Engineering Magazine*, vol. 8, p. 10, 2016.

[10] E. Guzman and W. Maalej, "How do users like this feature? a fine grained sentiment analysis of app reviews," in *2014 IEEE 22nd international requirements engineering conference (RE)*. Ieee, 2014, pp. 153–162.

[11] D. Pagano and W. Maalej, "User feedback in the appstore: An empirical study," in *2013 21st IEEE International Requirements Engineering Conference (RE)*, pp. 125–134, ISSN: 2332-6441.

[12] W. Maalej and H. Nabil, "Bug report, feature request, or simply praise? on automatically classifying app reviews," in *2015 IEEE 23rd international requirements engineering conference (RE)*. IEEE, 2015, pp. 116–125.

[13] J. Tizard, H. Wang, L. Yohannes, and K. Blincoe, "Can a conversation paint a picture? mining requirements in software forums," in *2019 IEEE 27th International Requirements Engineering Conference (RE)*. IEEE, 2019, pp. 17–27.

[14] G. Williams and A. Mahmoud, "Mining twitter feeds for software user requirements," in *2017 IEEE 25th International Requirements Engineering Conference (RE)*. IEEE, 2017, pp. 1–10.

[15] T. Iqbal, M. Khan, K. Taveter, and N. Seyff, "Mining reddit as a new source for software requirements," in *2021 IEEE 29th International Requirements Engineering Conference (RE)*. IEEE, 2021, pp. 128–138.

[16] N. Kengphanphanit and P. Muenchaisri, "Automatic requirements elicitation from social media (aresm)," in *Proceedings of the 2020 International Conference on Computer Communication and Information Systems*, 2020, pp. 57–62.

[17] O. Karras, E. Kristo, and J. Klünder, "The potential of using vision videos for crowdre: Video comments as a source of feedback," in *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*. IEEE, 2021, pp. 298–305.

[18] J. O. Johanssen, A. Kleebaum, B. Bruegge, and B. Paech, "How do practitioners capture and utilize user feedback during continuous software engineering?" in *2019 IEEE 27th International Requirements Engineering Conference (RE)*, 2019, pp. 153–164.

[19] S. Van Oordt and E. Guzman, "On the role of user feedback in software evolution: a practitioners' perspective," in *2021 IEEE 29th International Requirements Engineering Conference (RE)*. IEEE, 2021, pp. 221–232.

[20] L. Xu, X. Yan, and Z. Zhang, "Research on the causes of the "tiktok" app becoming popular and the existing problems," *Journal of advanced management science*, vol. 7, no. 2, 2019.

[21] D. L. Hoffman and T. P. Novak, "Toward a deeper understanding of social media," pp. 69–70, 2012.

[22] E. Knauss, D. Lübke, and S. Meyer, "Feedback-driven requirements engineering: The heuristic requirements assistant," in *Proceedings of the 31st International Conference on Software Engineering*. IEEE, 2009, pp. 587–590.

[23] A. Arif, K. Shanahan, F.-J. Chou, Y. Dosouto, K. Starbird, and E. S. Spiro, "How information snowballs: Exploring the role of exposure in online rumor propagation," in *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, 2016, pp. 466–477.

[24] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Roziere, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. Kloumann, I. Misra, I. Evtimov, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. van der Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu,

J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnstun, J. Saxe, J. Jia, K. V. Alwala, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, K. El-Arini, K. Iyer, K. Malik, K. Chiu, K. Bhalla, L. Rantala-Yeary, L. van der Maaten, L. Chen, L. Tan, L. Jenkins, L. Martin, L. Madaan, L. Malo, L. Blecher, L. Landzaat, L. de Oliveira, M. Muzzi, M. Pasupuleti, M. Singh, M. Paluri, M. Kardas, M. Oldham, M. Rita, M. Pavlova, M. Kambadur, M. Lewis, M. Si, M. K. Singh, M. Hassan, N. Goyal, N. Torabi, N. Bashlykov, N. Bogoychev, N. Chatterji, O. Duchenne, O. Çelebi, P. Alrassy, P. Zhang, P. Li, P. Vasic, P. Weng, P. Bhargava, P. Dubal, P. Krishnan, P. S. Koura, P. Xu, Q. He, Q. Dong, R. Srinivasan, R. Ganapathy, R. Calderer, R. S. Cabral, R. Stojnic, R. Raileanu, R. Girdhar, R. Patel, R. Sauvestre, R. Polidoro, R. Sumbaly, R. Taylor, R. Silva, R. Hou, R. Wang, S. Hosseini, S. Chennabasappa, S. Singh, S. Bell, S. S. Kim, S. Edunov, S. Nie, S. Narang, S. Raparthy, S. Shen, S. Wan, S. Bhosale, S. Zhang, S. Vandenhende, S. Batra, S. Whitman, S. Sootla, S. Collot, S. Gururangan, S. Borodinsky, T. Herman, T. Fowler, T. Sheasha, T. Georgiou, T. Scialom, T. Speckbacher, T. Mihaylov, T. Xiao, U. Karn, V. Goswami, V. Gupta, V. Ramanathan, V. Kerkez, V. Gonguet, V. Do, V. Vogeti, V. Petrovic, W. Chu, W. Xiong, W. Fu, W. Meers, X. Martinet, X. Wang, X. E. Tan, X. Xie, X. Jia, X. Wang, Y. Goldschlag, Y. Gaur, Y. Babaei, Y. Wen, Y. Song, Y. Zhang, Y. Li, Y. Mao, Z. D. Coudert, Z. Yan, Z. Chen, Z. Papakipos, A. Singh, A. Grattafiori, A. Jain, A. Kelsey, A. Shajnfeld, A. Gangidi, A. Victoria, A. Goldstand, A. Menon, A. Sharma, A. Boesenberg, A. Vaughan, A. Baevski, A. Feinstein, A. Kallet, A. Sangani, A. Yunus, A. Lupu, A. Alvarado, A. Caples, A. Gu, A. Ho, A. Poulton, A. Ryan, A. Ramchandani, A. Franco, A. Saraf, A. Chowdhury, A. Gabriel, A. Bharambe, A. Eisenman, A. Yazdan, B. James, B. Maurer, B. Leonhardi, B. Huang, B. Loyd, B. D. Paola, B. Paranjape, B. Liu, B. Wu, B. Ni, B. Hancock, B. Wasti, B. Spence, B. Stojkovic, B. Gamido, B. Montalvo, C. Parker, C. Burton, C. Mejia, C. Wang, C. Kim, C. Zhou, C. Hu, C.-H. Chu, C. Cai, C. Tindal, C. Feichtenhofer, D. Civin, D. Beaty, D. Kreymer, D. Li, D. Wyatt, D. Adkins, D. Xu, D. Testuggine, D. David, D. Parikh, D. Liskovich, D. Foss, D. Wang, D. Le, D. Holland, E. Dowling, E. Jamil, E. Montgomery, E. Presani, E. Hahn, E. Wood, E. Brinkman, E. Arcaute, E. Dunbar, E. Smothers, F. Sun, F. Kreuk, F. Tian, F. Ozgenel, F. Caggioni, F. Guzmán, F. Kanayet, F. Seide, G. M. Florez, G. Schwarz,

G. Badeer, G. Swee, G. Halpern, G. Thattai, G. Herman, G. Sizov, Guangyi, Zhang, G. Lakshminarayanan, H. Shojanazeri, H. Zou, H. Wang, H. Zha, H. Habeeb, H. Rudolph, H. Suk, H. Aspegren, H. Goldman, I. Molybog, I. Tufanov, I.-E. Veliche, I. Gat, J. Weissman, J. Geboski, J. Kohli, J. Asher, J.-B. Gaya, J. Marcus, J. Tang, J. Chan, J. Zhen, J. Reizenstein, J. Teboul, J. Zhong, J. Jin, J. Yang, J. Cummings, J. Carvill, J. Shepard, J. McPhie, J. Torres, J. Ginsburg, J. Wang, K. Wu, K. H. U, K. Saxena, K. Prasad, K. Khandelwal, K. Zand, K. Matosich, K. Veeraraghavan, K. Michelena, K. Li, K. Huang, K. Chawla, K. Lakhotia, K. Huang, L. Chen, L. Garg, L. A, L. Silva, L. Bell, L. Zhang, L. Guo, L. Yu, L. Moshkovich, L. Wehrstedt, M. Khabsa, M. Avalani, M. Bhatt, M. Tsimpoukelli, M. Mankus, M. Hasson, M. Lennie, M. Reso, M. Groshev, M. Naumov, M. Lathi, M. Keneally, M. L. Seltzer, M. Valko, M. Restrepo, M. Patel, M. Vyatskov, M. Samvelyan, M. Clark, M. Macey, M. Wang, M. J. Hermoso, M. Metanat, M. Rastegari, M. Bansal, N. Santhanam, N. Parks, N. White, N. Bawa, N. Singhal, N. Egebo, N. Usunier, N. P. Laptev, N. Dong, N. Zhang, N. Cheng, O. Chernoguz, O. Hart, O. Salpekar, O. Kalinli, P. Kent, P. Parekh, P. Saab, P. Balaji, P. Rittner, P. Bontrager, P. Roux, P. Dollar, P. Zvyagina, P. Ratanchandani, P. Yuvraj, Q. Liang, R. Alao, R. Rodriguez, R. Ayub, R. Murthy, R. Nayani, R. Mitra, R. Li, R. Hogan, R. Battey, R. Wang, R. Maheswari, R. Howes, R. Rinott, S. J. Bondu, S. Datta, S. Chugh, S. Hunt, S. Dhillon, S. Sidorov, S. Pan, S. Verma, S. Yamamoto, S. Ramaswamy, S. Lindsay, S. Lindsay, S. Feng, S. Lin, S. C. Zha, S. Shankar, S. Zhang, S. Zhang, S. Wang, S. Agarwal, S. Sajuyigbe, S. Chintala, S. Max, S. Chen, S. Kehoe, S. Satterfield, S. Govindaprasad, S. Gupta, S. Cho, S. Virk, S. Subramanian, S. Choudhury, S. Goldman, T. Remez, T. Glaser, T. Best, T. Kohler, T. Robinson, T. Li, T. Zhang, T. Matthews, T. Chou, T. Shaked, V. Vontimitta, V. Ajayi, V. Montanez, V. Mohan, V. S. Kumar, V. Mangla, V. Ionescu, V. Poenaru, V. T. Mihailescu, V. Ivanov, W. Li, W. Wang, W. Jiang, W. Bouaziz, W. Constable, X. Tang, X. Wang, X. Wu, X. Wang, X. Xia, X. Wu, X. Gao, Y. Chen, Y. Hu, Y. Jia, Y. Qi, Y. Li, Y. Zhang, Y. Zhang, Y. Adi, Y. Nam, Yu, Wang, Y. Hao, Y. Qian, Y. He, Z. Rait, Z. DeVito, Z. Rosnbrick, Z. Wen, Z. Yang, and Z. Zhao, "The llama 3 herd of models," 2024. [Online]. Available: https://arxiv.org/abs/2407.21783

[25] Z. S. Li, C. Werner, N. Ernst, and D. Damian, "Towards privacy compliance: A design science study in a small organization," *Information and Software Technology*, p. 106868, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584922000362

[26] World Wide Web Consortium (W3C), "Accessibility, usability, and inclusion," 2016, accessed: 2025-05-30. [Online]. Available: https://www.w3.org/WAI/fundamentals/accessibility-usability-inclusion/

[27] E. Kalliamvakou, "Research: quantifying GitHub Copilot's impact on developer productivity and happiness," Sep. 2022. [Online]. Available: https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happ

[28] S. Barke, M. B. James, and N. Polikarpova, "Grounded copilot: How programmers interact with code-generating models," *Proceedings of the ACM on Programming Languages*, vol. 7, no. OOPSLA1, pp. 85–111, 2023.

[29] J. T. Liang, C. Yang, and B. A. Myers, "A large-scale survey on the usability of ai programming assistants: Successes and challenges," in *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*, 2024, pp. 1–13.

[30] Y. Karlson, "Why don't people trust AI? | Kin," Jan. 2025. [Online]. Available: https://mykin.ai/resources/why-dont-people-trust-ai

[31] N. N. Arony, Z. S. Li, D. Damian, and B. Xu, "Unveiling inclusiveness-related user feedback in mobile applications," 2024. [Online]. Available: https://arxiv.org/abs/2311.00984

[32] M. Sihag, Z. S. Li, A. Dash, N. N. Arony, K. Devathasan, N. Ernst, A. B. Albu, and D. Damian, "A data-driven approach for finding requirements relevant feedback from tiktok and youtube," in *2023 IEEE 31st International Requirements Engineering Conference (RE)*, 2023, pp. 111–122.

[33] OpenAI, "ChatGPT: Optimizing Language Models for Dialogue," https://www.openai.com, 2022, accessed: 12-Nov-2023.

[34] Z. S. Li, N. N. Arony, A. M. Awon, D. Damian, and B. Xu, "Ai tool use and adoption in software development by individuals and organizations: A grounded theory study," 2024. [Online]. Available: https://arxiv.org/abs/2406.17325

[35] D. Zowghi and C. Coulin, "Requirements elicitation: A survey of techniques, approaches, and tools," in *Engineering and managing software requirements*. Springer, 2005, pp. 19–46.

[36] C. Potts, K. Takahashi, and A. I. Anton, "Inquiry-based requirements analysis," *IEEE software*, vol. 11, no. 2, pp. 21–32, 2002.

[37] A. Bennaceur, T. T. Tun, Y. Yu, and B. Nuseibeh, "Requirements engineering," *Handbook of software engineering*, pp. 51–92, 2019.

[38] J. Tizard, T. Rietz, X. Liu, and K. Blincoe, "Voice of the users: an extended study of software feedback engagement," *Requirements Engineering*, vol. 27, no. 3, pp. 293–315, 2022.

[39] J. Gebauer, Y. Tang, and C. Baimai, "User requirements of mobile technology: results from a content analysis of user reviews," *Information Systems and e-Business Management*, vol. 6, pp. 361–384, 2008.

[40] W. Jiang, H. Ruan, L. Zhang, P. Lew, and J. Jiang, "For user-driven software evolution: Requirements elicitation derived from mining online reviews," in *Advances in Knowledge Discovery and Data Mining: 18th Pacific-Asia Conference, PAKDD 2014, Tainan, Taiwan, May 13-16, 2014. Proceedings, Part II 18*. Springer, 2014, pp. 584–595.

[41] C. Wang, M. Daneva, M. van Sinderen, and P. Liang, "A systematic mapping study on crowdsourced requirements engineering using user feedback," *Journal of software: Evolution and Process*, vol. 31, no. 10, p. e2199, 2019.

[42] O. G. Ayodeji and V. Kumar, "Social media analytics: a tool for the success of online retail industry," *International Journal of Services Operations and Informatics*, vol. 10, no. 1, pp. 79–95, 2019.

[43] E. N. Torres, H. Adler, and C. Behnke, "Stars, diamonds, and other shiny things: The use of expert and consumer feedback in the hotel industry," *Journal of Hospitality and Tourism Management*, vol. 21, pp. 34–43, 2014.

[44] A. Bhave, H. Kulkarni, V. Biramane, and P. Kosamkar, "Role of different factors in predicting movie success," in *2015 International Conference on Pervasive Computing (ICPC)*. IEEE, 2015, pp. 1–4.

[45] M. Silva, E. Vieira, G. Signoretti, I. Silva, D. Silva, and P. Ferrari, "A customer feedback platform for vehicle manufacturing compliant with industry 4.0 vision," *Sensors*, vol. 18, no. 10, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/10/3298

[46] Y. Zhan, R. Han, M. Tse, M. H. Ali, and J. Hu, "A social media analytic framework for improving operations and service management: A study of the retail pharmacy industry," *Technological Forecasting and Social Change*, vol. 163, p. 120504, 2021.

[47] N. R. A. Hamid and U. T. A. Razak, "Social media: An emerging dimension of marketing communication," 2013.

[48] A. Tkalich, E. Klotins, T. Sporsem, V. Stray, N. B. Moe, and A. Barbala, "User feedback in continuous software engineering: revealing the state-of-practice," *Empirical Software Engineering*, vol. 30, no. 3, p. 79, 2025.

[49] N. AlAmoudi, M. Baslyman, and M. Ahmed, "Extracting attractive app aspects from app reviews using clustering techniques based on kano model," in *2022 IEEE 30th International Requirements Engineering Conference Workshops (REW)*, 2022, pp. 123–129.

[50] H. Wang, P. Devine, J. Tizard, S. R. Shahamiri, and K. Blincoe, "The use of sub-forums in software product forums," in *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, 2021, pp. 293–297.

[51] J. Tizard, T. Rietz, and K. Blincoe, "Voice of the users: A demographic study of software feedback behaviour," in *2020 IEEE 28th International Requirements Engineering Conference (RE)*. IEEE, 2020, pp. 55–65.

[52] T. Ogink and J. Q. Dong, "Stimulating innovation by user feedback on social media: The case of an online user innovation community," *Technological Forecasting and Social Change*, vol. 144, pp. 295–302, 2019.

[53] J. Tizard, T. Rietz, and K. Blincoe, "Voice of the users: A demographic study of software feedback behaviour," in *2020 IEEE 28th International Requirements Engineering Conference (RE)*, 2020, pp. 55–65.

[54] W. Maalej, H.-J. Happel, and A. Rashid, "When users become collaborators: towards continuous and context-aware user input," in *Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*, 2009, pp. 981–990.

[55] J. Tizard, H. Wang, L. Yohannes, and K. Blincoe, "Can a conversation paint a picture? mining requirements in software forums," in *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pp. 17–27, ISSN: 2332-6441.

[56] G. M. Kanchev and A. K. Chopra, "Social media through the requirements lens: A case study of google maps," in *2015 IEEE 1st International Workshop on Crowd-Based Requirements Engineering (CrowdRE)*, pp. 7–12.

[57] G. Williams and A. Mahmoud, "Mining twitter feeds for software user requirements," in *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pp. 1–10, ISSN: 2332-6441.

[58] T. Iqbal, M. Khan, K. Taveter, and N. Seyff, "Mining Reddit as a New Source for Software Requirements," in *2021 IEEE 29th International Requirements Engineering Conference (RE)*, Sep. 2021, pp. 128–138, iSSN: 2332-6441.

[59] S. Das, A. Dutta, T. Lindheimer, M. Jalayer, and Z. Elgart, "Youtube as a source of information in understanding autonomous vehicle consumers: natural language processing study," *Transportation research record*, vol. 2673, no. 8, pp. 242–253, 2019.

[60] A. Madden, I. Ruthven, and D. McMenemy, "A classification scheme for content analyses of youtube video comments," *Journal of documentation*, vol. 69, no. 5, pp. 693–714, 2013.

[61] P. Vistisen and S. B. Poulsen, "Return of the vision video: Can corporate vision videos serve as setting for participation?" 2017.

[62] M. Harman, Y. Jia, and Y. Zhang, "App store mining and analysis: MSR for app stores," in *2012 9th IEEE Working Conference on Mining Software Repositories (MSR)*, pp. 108–111, ISSN: 2160-1860.

[63] E. Guzman, M. Ibrahim, and M. Glinz, "A little bird told me: Mining tweets for requirements and software evolution," in *2017 IEEE 25th International Requirements Engineering Conference (RE)*. IEEE, 2017, pp. 11–20.

[64] J. Dabrowski, E. Letier, A. Perini, and A. Susi, "Mining user opinions to support requirement engineering: an empirical study," in *International Conference on Advanced Information Systems Engineering*. Springer, 2020, pp. 401–416.

[65] M. Haering, C. Stanik, and W. Maalej, "Automatically Matching Bug Reports With Related App Reviews," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, May 2021, pp. 970–981, iSSN: 1558-1225.

[66] C. Stanik, M. Haering, C. Jesdabodi, and W. Maalej, "Which App Features Are Being Used? Learning App Feature Usages from Interaction Data," in *2020 IEEE 28th International Requirements Engineering Conference (RE)*, Aug. 2020, pp. 66–77, iSSN: 2332-6441.

[67] M. Fazzini, H. Khalajzadeh, O. Haggag, Z. Li, H. Obie, C. Arora, W. Hussain, and J. Grundy, "Characterizing human aspects in reviews of covid-19 apps," in *Proceedings of the 9th IEEE/ACM International Conference on Mobile Software Engineering and Systems*, 2022, pp. 38–49.

[68] H. O. Obie, W. Hussain, X. Xia, J. Grundy, L. Li, B. Turhan, J. Whittle, and M. Shahin, "A first look at human values-violation in app reviews," in *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE, 2021, pp. 29–38.

[69] A. Alshayban, I. Ahmed, and S. Malek, "Accessibility issues in android apps: state of affairs, sentiments, and ways forward," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, 2020, pp. 1323–1334.

[70] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[71] P. R. Henao, J. Fischbach, D. Spies, J. Frattini, and A. Vogelsang, "Transfer learning for mining feature requests and bug reports from tweets and app store reviews," in *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*. IEEE, 2021, pp. 80–86.

[72] R. R. Mekala, A. Irfan, E. C. Groen, A. Porter, and M. Lindvall, "Classifying user requirements from online feedback in small dataset environments using deep learning," in *2021 IEEE 29th International Requirements Engineering Conference (RE)*. IEEE, 2021, pp. 139–149.

[73] T. Hey, J. Keim, A. Koziolek, and W. F. Tichy, "Norbert: Transfer learning for requirements classification," in *2020 IEEE 28th International Requirements Engineering Conference (RE)*. IEEE, 2020, pp. 169–179.

[74] L. Coffey, "Harvard Taps AI to Help Teach Computer Science Course." [Online]. Available: https://www.insidehighered.com/news/tech-innovation/artificial-intelligence/2023/06/30/harvard-rolls-out-ai-help-free-tas-time

[75] B. Yetistiren, I. Ozsoy, and E. Tuzun, "Assessing the quality of github copilot's code generation," in *Proceedings of the 18th international conference on predictive models and data analytics in software engineering*, 2022, pp. 62–71.

[76] N. Nguyen and S. Nadi, "An empirical evaluation of github copilot's code suggestions," in *Proceedings of the 19th International Conference on Mining Software Repositories*, 2022, pp. 1–5.

[77] A. M. Dakhel, V. Majdinasab, A. Nikanjam, F. Khomh, M. C. Desmarais, and Z. M. J. Jiang, "Github copilot ai pair programmer: Asset or liability?" *Journal of Systems and Software*, vol. 203, p. 111734, 2023.

[78] R. Khojah, M. Mohamad, P. Leitner, and F. G. de Oliveira Neto, "Beyond code generation: An observational study of chatgpt usage in software engineering practice," *Proceedings of the ACM on Software Engineering*, vol. 1, no. FSE, pp. 1819–1840, 2024.

[79] A. Beganovic, M. A. Jaber, and A. Abd Almisreb, "Methods and applications of chatgpt in software development: A literature review," *Southeast Europe Journal of Soft Computing*, vol. 12, no. 1, pp. 08–12, 2023.

[80] H. Luo, P. Liu, and S. Esping, "Exploring small language models with prompt-learning paradigm for efficient domain-specific text classification," *arXiv preprint arXiv:2309.14779*, 2023.

[81] K. Ronanki, B. Cabrero-Daniel, and C. Berger, "Chatgpt as a tool for user story quality evaluation: Trustworthy out of the box?" 2023.

[82] J. A. Khan, S. Qayyum, and H. S. Dar, "Large language model for requirements engineering: A systematic literature review," *Research Square*, 2025.

[83] A. Hemmat, M. Sharbaf, S. Kolahdouz-Rahimi, K. Lano, and S. Y. Tehrani, "Research directions for using llm in software requirement engineering: A systematic review," *Frontiers in Computer Science*, vol. 7, 2025.

[84] D. Seifert, L. Jöckel, A. Trendowicz, M. Ciolkowski, T. Honroth, and A. Jedlitschka, "Can large language models (llms) compete with human requirement reviewers? – replication of an inspection experiment on requirements documents," in *Product-Focused Software Process Improvement (PROFES 2024)*, ser. Lecture Notes in Computer Science. Springer, 2024.

[85] J. J. Norheim, E. Rebentisch, D. Xiao, L. Draeger, A. Kerbrat, and O. L. de Weck, "Challenges in applying large language models to requirements engineering tasks," *Design Science*, vol. 10, no. e16, 2024.

[86] K. Ronanki, C. Berger, and J. Horkoff, "Investigating chatgpt's potential to assist in requirements elicitation processes," in *2023 49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 2023, pp. 354–361.

[87] M. A. Sami, Z. Rasheed, M. Waseem, Z. Zhang, T. Herda, and P. Abrahamsson, "Prioritizing software requirements using large language models," *arXiv preprint*, 2024.

[88] E. A. Oğuz and J. Küster, "A comparative analysis of chatgpt-generated and human-written use case descriptions," in *Companion Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems (MODELS 2024)*, 2024.

[89] R. Santos, G. Freitas, I. Steinmacher, T. Conte, A. C. Oran, and B. Gadelha, "User stories: Does chatgpt do it better?"

[90] M. Nath, J. Muralikrishnan, K. Sundarrajan, and M. Varadarajanna, "Continuous integration, delivery, and deployment: a revolutionary approach in software development," *International Journal of Research and Scientific Innovation (IJRSI)*, vol. 5, no. 7, pp. 185–190, 2018.

[91] C. Pacheco, I. García, and M. Reyes, "Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques," *IET Software*, vol. 12, no. 4, pp. 365–378, 2018.

[92] A. Strauss and J. Corbin, *Basics of qualitative research*. Sage publications, 1990.

[93] K. Sebastian, "Distinguishing between the strains grounded theory: Classical, interpretive and constructivist," *Journal for Social Thought*, vol. 3, no. 1, 2019.

[94] OECD, "Enterprises by business size (indicator)," 2023. [Online]. Available: http://data.oecd.org/entrepreneur/enterprises-by-business-size.htm

[95] W. Bank, "Small and medium enterprises (smes) finance," 2019. [Online]. Available: https://www.worldbank.org/en/topic/smefinance

[96] J. Aranda, S. Easterbrook, and G. Wilson, "Requirements in the wild: How small companies do it," in *15th IEEE International Requirements Engineering Conference (RE 2007)*. IEEE, 2007, pp. 39–48.

[97] M. Q. Patton, *Qualitative research & evaluation methods*. sage, 2002.

[98] Anonymous, "Unveiling the Life Cycle of User Feedback: Best Practices from Software Practitioners," Mar. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.8118389

[99] S. Mendelson, "Box Office: 'Morbius' Bombs Again With $85,000 Friday," 2022, section: Hollywood & Entertainment. [Online]. Available: https://www.forbes.com/sites/scottmendelson/2022/06/04/box-office-jared-leto-morbius-bombs-again-with-85000-friday/

[100] Z. S. Li, M. Sihag, N. N. Arony, J. B. Junior, T. Phan, N. Ernst, and D. Damian, "Narratives: the unforeseen influencer of privacy concerns," in *2022 IEEE 30th International Requirements Engineering Conference (RE)*. IEEE, 2022, pp. 127–139.

[101] D. Bajic and K. Lyons, "Leveraging social media to gather user feedback for software development," in *Proceedings of the 2nd International Workshop on Web 2.0 for Software Engineering*, ser. Web2SE '11. New York, NY,

USA: Association for Computing Machinery, 2011, p. 1–6. [Online]. Available: https://doi.org/10.1145/1984701.1984702

[102] Z. Tufekci, "Big questions for social media big data: Representativeness, validity and other methodological pitfalls," in *Proceedings of the international AAAI conference on web and social media*, vol. 8, no. 1, 2014, pp. 505–514.

[103] L. V. G. Carreño and K. Winbladh, "Analysis of user comments: an approach for software requirements evolution," in *2013 35th international conference on software engineering (ICSE)*. IEEE, 2013, pp. 582–591.

[104] C. Iacob and R. Harrison, "Retrieving and analyzing mobile apps feature requests from online reviews," in *2013 10th working conference on mining software repositories (MSR)*. IEEE, 2013, pp. 41–44.

[105] E. Oehri and E. Guzman, "Same same but different: Finding similar user feedback across multiple platforms and languages," in *2020 IEEE 28th International Requirements Engineering Conference (RE)*. IEEE, 2020, pp. 44–54.

[106] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall, "How can i improve my app? classifying user reviews for software maintenance and evolution," in *2015 IEEE international conference on software maintenance and evolution (ICSME)*. IEEE, 2015, pp. 281–290.

[107] C. Gao, Y. Li, S. Qi, Y. Liu, X. Wang, Z. Zheng, and Q. Liao, "Listening to users' voice: Automatic summarization of helpful app reviews," *IEEE Transactions on Reliability*, 2022.

[108] N. Seyff, M. Stade, F. Fotrousi, and M. Oriol Hilari, "End-user driven feedback prioritization," in *REFSQ 2017 Joint Proceedings of the Co-Located Events: Joint Proceedings of REFSQ-2017 Workshops, Doctoral Symposium, Research Method Track, and Poster Track: co-located with the 22nd International Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2017): Essen, Germany, February 27, 2017*. CEUR-WS. org, 2017, pp. 1–7.

[109] D. Firesmith, "Prioritizing requirements." *J. Object Technol.*, vol. 3, no. 8, pp. 35–48, 2004.

[110] S. Gärtner and K. Schneider, "A method for prioritizing end-user feedback for requirements engineering," in *2012 5th International Workshop on Co-operative and Human Aspects of Software Engineering (CHASE)*. IEEE, 2012, pp. 47–49.

[111] P. Berander and A. Andrews, "Requirements prioritization," *Engineering and managing software requirements*, pp. 69–94, 2005.

[112] S. Sivzattian and B. Nuseibeh, "Linking the selection of requirements to market value: A portfolio-based approach," in *Proceedings of 7th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ 2001)*, 2001.

[113] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. Mahrin, "A systematic literature review of software requirements prioritization research," *Information and software technology*, vol. 56, no. 6, pp. 568–585, 2014.

[114] S. A. Licorish, B. T. R. Savarimuthu, and S. Keertipati, "Attributes that predict which features to fix: Lessons for app store mining," in *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, 2017, pp. 108–117.

[115] F. M. Kifetew, A. Perini, A. Susi, A. Siena, D. Muñante, and I. Morales-Ramirez, "Automating user-feedback driven requirements prioritization," *Information and Software Technology*, vol. 138, p. 106635, 2021.

[116] S. Malgaonkar, S. A. Licorish, and B. T. R. Savarimuthu, "Prioritizing user concerns in app reviews–a study of requests for new features, enhancements and bug fixes," *Information and Software Technology*, vol. 144, p. 106798, 2022.

[117] N. Flocco, F. Canterino, and R. Cagliano, "To control or not to control: How to organize employee-driven innovation," *Creativity and Innovation Management*, vol. 31, no. 3, pp. 396–409, 2022.

[118] AWS, "What is DevOps?" 2023. [Online]. Available: https://aws.amazon.com/devops/what-is-devops/

[119] K. Srisopha, D. Link, and B. Boehm, "How should developers respond to app reviews? features predicting the success of developer responses," in *Evaluation and Assessment in Software Engineering*, 2021, pp. 119–128.

[120] M. R. Roller and P. J. Lavrakas, *Applied qualitative research design: A total quality framework approach*, ser. Applied qualitative research design: A total quality framework approach. New York, NY, US: The Guilford Press, 2015, pages: xviii, 398.

[121] P. Gunarathne, H. Rui, and A. Seidmann, "Whose and what social media complaints have happier resolutions? evidence from twitter," *Journal of Management Information Systems*, vol. 34, no. 2, pp. 314–340, 2017. [Online]. Available: https://doi.org/10.1080/07421222.2017.1334465

[122] Z. Xiang, Q. Du, Y. Ma, and W. Fan, "A comparative analysis of major online review platforms: Implications for social media analytics in hospitality and tourism," *Tourism Management*, vol. 58, pp. 51–65, 2017.

[123] "[N] Google is increasing the price of every Colab Pro tier by 10X! Pro is 95 Euro and Pro+ is 433 Euro per month! Without notifying users!" 2023. [Online]. Available: https://www.reddit.com/r/MachineLearning/comments/114hphp/n_google_is_increasing_the_price_of_every_colab/

[124] O. Radley-Gardner, H. Beale, and R. Zimmermann, Eds., *Fundamental Texts On European Private Law*. Hart Publishing, 2016. [Online]. Available: http://www.bloomsburycollections.com/book/fundamental-texts-on-european-private-law-1

[125] "California Consumer Privacy Act (CCPA)," Oct. 2018. [Online]. Available: https://oag.ca.gov/privacy/ccpa

[126] E. Harding and L. H. Ji-Otto, "Five Immediate Steps to Take in Preparation for China's New Comprehensive Privacy Law," Nov. 2021. [Online]. Available: https://www.natlawreview.com/article/five-immediate-steps-to-take-preparation-china-s-new-comprehensive-privacy-law

[127] J. K. Kadish and L. Thomas, "Brazil's Comprehensive Privacy Law Now in Effect," Sep. 2020. [Online]. Available: https://www.natlawreview.com/article/brazil-s-comprehensive-privacy-law-now-effect

[128] A. Ng and S. Musil, "Equifax data leak may affect nearly half the US population," Sep. 2017. [Online]. Available: https://www.cnet.com/tech/services-and-software/equifax-data-leak-hits-nearly-half-of-the-us-population/

[129] J. Silverstein, "Hundreds of millions of Facebook user records were exposed on Amazon cloud server," Apr. 2019. [Online]. Available: https://www.cbsnews.com/news/millions-facebook-user-records-exposed-amazon-cloud-server/

[130] D. Winder, "235 Million Instagram, TikTok And YouTube User Profiles Exposed In Massive Data Leak," Aug. 2020, section: Cybersecurity. [Online]. Available: https://www.forbes.com/sites/daveywinder/2020/08/19/massive-data-leak235-million-instagram-tiktok-and-youtube-user-profiles-exposed/

[131] "Pushshift github url." [Online]. Available: https://github.com/pushshift/api

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

[133] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1, pp. 43–52, 2010.

[134] T. Joachims, "A probabilistic analysis of the rocchio algorithm with tfidf for text categorization." Carnegie-mellon univ pittsburgh pa dept of computer science, Tech. Rep., 1996.

[135] P. Devine, Y. S. Koh, and K. Blincoe, "Evaluating Unsupervised Text Embeddings on Software User Feedback," in *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, Sep. 2021, pp. 87–95.

[136] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar *et al.*, "Universal sentence encoder," *arXiv preprint arXiv:1803.11175*, 2018.

[137] "Universal sentence encoder url." [Online]. Available: https://tfhub.dev/google/universal-sentence-encoder-large/5

[138] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, p. 321–357, Jun 2002. [Online]. Available: http://dx.doi.org/10.1613/jair.953

[139] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2020.

[140] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," *arXiv:1802.03426 [cs, stat]*, Sep. 2020, arXiv: 1802.03426. [Online]. Available: http://arxiv.org/abs/1802.03426

[141] "K-means clustering." [Online]. Available: https://tfhub.dev/google/universal-sentence-encoder-large/5

[142] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0377042787901257

[143] D. C. Nguyen, E. Derr, M. Backes, and S. Bugiel, "Short text, large effect: Measuring the impact of user reviews on android app security amp; privacy," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 555–569.

[144] J. Kastrenakes, "Reddit reveals daily active user count for the first time: 52 million," Dec. 2020. [Online]. Available: https://www.theverge.com/2020/12/1/21754984/reddit-dau-daily-users-revealed

[145] C. K. Riessman, "Narrative analysis," in *Narrative, Memory & Everyday Life*, N. Kelly, C. Horrocks, K. Milnes, B. Roberts, and D. Robinson, Eds. Huddersfield: University of Huddersfield, April 2005, pp. 1–7. [Online]. Available: http://eprints.hud.ac.uk/id/eprint/4920/

[146] R. J. Shiller, *Narrative Economics: How Stories Go Viral and Drive Major Economic Events*. Princeton University Press, Sep. 2020, google-Books-ID: YlDVDwAAQBAJ.

[147] ——, "Narrative Economics," *The American Economic Review*, vol. 107, no. 4, pp. 967–1004, 2017, publisher: American Economic Association. [Online]. Available: https://www.jstor.org/stable/44251584

[148] U. Perano, "Facebook begins merging WhatsApp, Instagram and Messenger infrastructures," Aug. 2015. [Online]. Available: https://www.axios.com/facebook-whatsapp-instagram-messenger-a4ba5ff0-7399-4d7c-bec6-446068a87480.html

[149] D'Amore, Rachael, "WhatsApp's new privacy policy sparks outcry. Here's what you need to know - National | Globalnews.ca," Jan. 2021. [Online]. Available: https://globalnews.ca/news/7570323/whatsapp-facebook-privacy-policy-explained/

[150] "Stat counter url." [Online]. Available: https://gs.statcounter.com/

[151] K. O'Flaherty, "Firefox Takes Aim At Google With A Bunch Of New Security Features," Jun. 2019, section: Cybersecurity. [Online]. Available: https://www.forbes.com/sites/kateoflahertyuk/2019/06/04/firefox-confirms-new-security-features-heres-how-to-enable-them/

[152] N. Dailey, "Telegram hits 500 million active users following backlash over WhatsApp's changing privacy policy," Jan. 2021. [Online]. Available: https://www.businessinsider.com/telegram-hits-500-million-users-after-whatsapp-backlash-2021-1

[153] J. Golbeck, *Analyzing the Social Web*. Elsevier Science & Technology, 2013. [Online]. Available: https://ebookcentral-proquest-com.ezproxy.library.uvic.ca/lib/uvic/detail.action?docID=1152671

[154] A. Savidis and C. Stephanidis, "Inclusive development: Software engineering requirements for universally accessible interactions," *Interacting with Computers*, vol. 18, no. 1, pp. 71–116, 2006.

[155] J. Shepherd, "23 essential twitter statistics you need to know in 2023," https://thesocialshepherd.com/blog/twitter-statistics, May 2023.

[156] T. Gaurdian, "Twitter apologises for 'racist' image-cropping algorithm," https://www.theguardian.com/technology/2020/sep/21/twitter-apologises-for-racist-image-cropping-algorithm, 2020.

[157] M. Shahin, M. Zahedi, H. Khalajzadeh, and A. R. Nasab, "A study of gender discussions in mobile apps," *arXiv preprint arXiv:2303.09808*, 2023.

[158] H. Khalajzadeh, M. Shahin, H. O. Obie, P. Agrawal, and J. Grundy, "Supporting developers in addressing human-centric issues in mobile apps," *IEEE Transactions on Software Engineering*, 2022.

[159] R. Hoda, "Socio-technical grounded theory for software engineering," *IEEE Transactions on Software Engineering*, vol. 48, no. 10, pp. 3808–3832, 2021.

[160] K. B. M. , "Designing inclusive software in Windows - Windows apps," https://learn.microsoft.com/en-us/windows/apps/design/accessibility/designing-inclusive-software, may 13 2022.

[161] H. Khalajzadeh, M. Shahin, H. O. Obie, and J. Grundy, "How are diverse end-user human-centric issues discussed on github?" in *Proceedings of the 2022 ACM/IEEE 44th International Conference on Software Engineering: Software Engineering in Society*, 2022, pp. 79–89.

[162] Watchful1, "Subreddit comments/submissions 2005-06 to 2022-12." [Online]. Available: https://www.reddit.com/r/pushshift/comments/11ef9if/separate_dump_files_for_the_top_20k_subreddits/

[163] N. N. Arony, "Unveiling Inclusiveness-Related User Feedback in Mobile Applications," Nov. 2024. [Online]. Available: https://zenodo.org/records/14232484

[164] B. Zhang, X. Fu, D. Ding, H. Huang, G. Dai, N. Yin, Y. Li, and L. Jing, "Investigating chain-of-thought with chatgpt for stance detection on social media," *arXiv preprint arXiv:2304.03087*, 2023.

[165] X. Li, S. Chan, X. Zhu, Y. Pei, Z. Ma, X. Liu, and S. Shah, "Are chatgpt and gpt-4 general-purpose solvers for financial text analytics? a study on several typical tasks," *arXiv preprint arXiv:2305.05862*, 2023.

[166] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Machine Learning: ECML 2004: 15th European Conference on Machine Learning, Pisa, Italy, September 20-24, 2004. Proceedings 15*. Springer, 2004, pp. 39–50.

[167] Y. Li, G. Sun, and Y. Zhu, "Data imbalance problem in text classification," in *2010 Third international symposium on information processing*. IEEE, 2010, pp. 301–305.

[168] S. Clarissa, J. Lobo *et al.*, "The rising popularity of tiktok during the pandemic: Utilization of the application vis-à-vis students' engagement," *American Journal of Interdisciplinary Research and Innovation*, vol. 1, no. 2, pp. 43–48, 2022.

[169] M. L. Khan and A. Malik, "Researching youtube: Methods, tools, and analytics."

[170] M. Sihag, Z. S. Li, A. Dash, N. N. Arony, K. Devathasan, N. Ernst, A. B. Albu, and D. Damian, "A Data-Driven Approach for Finding Requirements Relevant Feedback from TikTok and YouTube," Jun. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.8088427

[171] "Spacy fastlang." [Online]. Available: https://spacy.io/universe/project/spacy_fastlang

[172] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision."

[173] A. Dash and A. B. Albu, "A domain independent approach to video summarization," in *Advanced Concepts for Intelligent Vision Systems: 18th International Conference, ACIVS 2017, Antwerp, Belgium, September 18-21, 2017, Proceedings 18*. Springer, 2017, pp. 431–442.

[174] T. Sheng, J. Chen, and Z. Lian, "Centripetaltext: An efficient text instance representation for scene text detection," *Advances in Neural Information Processing Systems*, vol. 34, pp. 335–346, 2021.

[175] M. Li, T. Lv, J. Chen, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, and F. Wei, "Trocr: Transformer-based optical character recognition with pre-trained models," *arXiv preprint arXiv:2109.10282*, 2021.

[176] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," number: arXiv:1907.11692. [Online]. Available: http://arxiv.org/abs/1907.11692

[177] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," number: arXiv:1911.02116. [Online]. Available: http://arxiv.org/abs/1911.02116

[178] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," number: arXiv:1909.11942. [Online]. Available: http://arxiv.org/abs/1909.11942

[179] M. Grootendorst, "BERTopic: Neural topic modeling with a class-based TF-IDF procedure," number: arXiv:2203.05794. [Online]. Available: http://arxiv.org/abs/2203.05794

[180] K. Kousha, M. Thelwall, and M. Abdoli, "The role of online videos in research communication: A content analysis of youtube videos cited in academic publications," *Journal of the American Society for information Science and Technology*, vol. 63, no. 9, pp. 1710–1727, 2012.

[181] YouTube Help, "Use automatic captioning," n.d., accessed: 2025-05-21. [Online]. Available: https://support.google.com/youtube/answer/6373554?hl=en

[182] GitHub, "Github copilot- your ai pair programmer," https://copilot.github.com, 2023, accessed: 12-Nov-2023.

[183] "Gemini," 2024. [Online]. Available: https://gemini.google.com/

[184] J. Porter, "Chatgpt continues to be one of the fastest-growing services ever," *Retrieved December*, vol. 3, 2023.

[185] "Microsoft Q2 2024 Earnings Call Transcript." [Online]. Available: https://www.marketbeat.com/earnings/transcripts/101360/

[186] S. Peng, E. Kalliamvakou, P. Cihon, and M. Demirer, "The impact of ai on developer productivity: Evidence from github copilot," *arXiv preprint arXiv:2302.06590*, 2023.

[187] P. Vaithilingam, T. Zhang, and E. L. Glassman, "Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models," in *Chi conference on human factors in computing systems extended abstracts*, 2022, pp. 1–7.

[188] "Stack overflow developer survey 2023: Ai developer tools," https://survey.stackoverflow.co/2023/#ai-developer-tools, 2023, accessed: 12-Nov-2023.

[189] GitLab, "2023 Global DevSecOps Report Series," https://about.gitlab.com/developer-survey/, GitLab, Mar. 2023, accessed: 2024-03-20.

[190] K. Madampe, R. Hoda, and J. Grundy, "The Emotional Roller Coaster of Responding to Requirements Changes in Software Engineering," *IEEE Transactions on Software Engineering*, vol. 49, no. 3, pp. 1171–1187, Mar. 2023, conference Name: IEEE Transactions on Software Engineering. [Online]. Available: https://ieeexplore.ieee.org/document/9769966

[191] ——, "A Framework for Emotion-Oriented Requirements Change Handling in Agile Software Engineering," *IEEE Transactions on Software Engineering*, vol. 49, no. 5, pp. 3325–3343, May 2023, conference Name: IEEE Transactions on Software Engineering. [Online]. Available: https://ieeexplore.ieee.org/document/10061282

[192] H. Gunatilake, J. Grundy, R. Hoda, and I. Mueller, "Enablers and Barriers of Empathy in Software Developer and User Interactions: A Mixed Methods Case Study," *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 4, pp. 1–41, May 2024. [Online]. Available: https://dl.acm.org/doi/10.1145/3641849

[193] U. M. Graetsch, H. Khalajzadeh, M. Shahin, R. Hoda, and J. Grundy, "Dealing with Data Challenges when Delivering Data-Intensive Software Solutions," Mar. 2023, arXiv:2209.14055 [cs]. [Online]. Available: http://arxiv.org/abs/2209.14055

[194] K. Gama and A. Lacerda, "Understanding and supporting neurodiverse software developers in agile teams," in *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*, 2023, pp. 497–502.

[195] D. Hidellaarachchi, J. Grundy, R. Hoda, and I. Mueller, "Understanding the influence of motivation on requirements engineering-related activities," *arXiv preprint arXiv:2304.08074*, 2023.

[196] J. T. Liang, T. Zimmermann, and D. Ford, "Understanding skills for oss communities on github," in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2022, pp. 170–182.

[197] H. Alahyari, R. B. Svensson, and T. Gorschek, "A study of value in agile software development organizations," *Journal of Systems and Software*, vol. 125, pp. 271–288, 2017.

[198] A. Alami, O. Krancher, and M. Paasivaara, "The journey to technical excellence in agile software development," *Information and Software Technology*, vol. 150, p. 106959, 2022.

[199] J. Itkonen, M. V. Mäntylä, and C. Lassenius, "The role of the tester's knowledge in exploratory software testing," *IEEE Transactions on Software Engineering*, vol. 39, no. 5, pp. 707–724, 2012.

[200] S. Raghavan R, J. KR, and R. V. Nargundkar, "Impact of software as a service (saas) on software acquisition process," *Journal of Business & Industrial Marketing*, vol. 35, no. 4, pp. 757–770, 2020.

[201] S. Baltes and P. Ralph, "Sampling in software engineering research: A critical review and guidelines," *Empirical Software Engineering*, vol. 27, no. 4, p. 94, 2022.

[202] Y. Wu, S. Wang, C.-P. Bezemer, and K. Inoue, "How do developers utilize source code from stack overflow?" *Empirical Software Engineering*, vol. 24, pp. 637–673, 2019.

[203] D. Russo, "Navigating the complexity of generative ai adoption in software engineering," *ACM Transactions on Software Engineering and Methodology*, 2023.

[204] A. Li, M. Endres, and W. Weimer, "Debugging with stack overflow: Web search behavior in novice and expert programmers," in *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Software Engineering Education and Training*, 2022, pp. 69–81.

[205] X. Xia, L. Bao, D. Lo, P. S. Kochhar, A. E. Hassan, and Z. Xing, "What do developers search for on the web?" *Empirical Software Engineering*, vol. 22, pp. 3149–3185, 2017.

[206] C. Sadowski, K. T. Stolee, and S. Elbaum, "How developers search for code: a case study," in *Proceedings of the 2015 10th joint meeting on foundations of software engineering*, 2015, pp. 191–201.

[207] G. A. Moore and R. McKenna, "Crossing the chasm," 1999.

[208] K. Albusays, P. Bjorn, L. Dabbish, D. Ford, E. Murphy-Hill, A. Serebrenik, and M.-A. Storey, "The diversity crisis in software development," *IEEE Software*, vol. 38, no. 2, pp. 19–25, 2021.

[209] P. R. Clance and S. A. Imes, "The imposter phenomenon in high achieving women: Dynamics and therapeutic intervention." *Psychotherapy: Theory, research & practice*, vol. 15, no. 3, p. 241, 1978.

[210] S. Michailova and K. Hutchings, "National cultural influences on knowledge sharing: A comparison of china and russia," *Journal of management studies*, vol. 43, no. 3, pp. 383–405, 2006.

[211] "Trusted Knowledge Sharing Platform for Technologists: Stack Overflow for Teams – Stack Overflow for Teams." [Online]. Available: https://stackoverflow.co/teams/

[212] R. E. de Souza Santos and P. Ralph, "A grounded theory of coordination in remote-first and hybrid software teams," in *Proceedings of the 44th International Conference on Software Engineering*, 2022, pp. 25–35.

[213] Harvard Business Review, "How global companies use ai to prevent supply chain disruptions," *Harvard Business Review*, November 2023, accessed: 25 May 2024. [Online]. Available: https://hbr.org/2023/11/how-global-companies-use-ai-to-prevent-supply-chain-disruptions

[214] M. Diaz, "ChatGPT vs. ChatGPT Plus: Is a paid subscription still worth it?" [Online]. Available: https://www.zdnet.com/article/chatgpt-vs-chatgpt-plus-is-a-paid-subscription-still-worth-it/

[215] "Data protection in the EU," 2019. [Online]. Available: https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en

[216] "California Consumer Privacy Act Regulations."

[217] "PDPC | PDPA Overview." [Online]. Available: https://www.pdpc.gov.sg/overview-of-pdpa/the-legislation/personal-data-protection-act

[218] "Personal Information Protection Law of the People's Republic of China." [Online]. Available: http://en.npc.gov.cn.cdurl.cn/2021-12/29/c_694559.htm

[219] "Consumer Privacy Protection Act." [Online]. Available: https://ised-isde.canada.ca/site/innovation-better-canada/en/consumer-privacy-protection-act

[220] Yahoo Finance, "Is ai moving too fast to keep up with?" *Yahoo Finance*, May 2024, accessed: 25 May 2024. [Online]. Available: https://finance.yahoo.com/news/ai-moving-too-fast-keep-190958680.html?

[221] DPM, "Luxembourg DPA issues €746 Million GDPR Fine to Amazon," Jul. 2021. [Online]. Available: https://dataprivacymanager.net/luxembourg-dpa-issues-e746-million-gdpr-fine-to-amazon/

[222] "Binding decision 1/2023 on the dispute submitted by the irish sa on data transfers by meta platforms ireland limited for its facebook service art. 65 gdpr european data protection board," May 2023. [Online]. Available: https://www.edpb.europa.eu/our-work-tools/our-documents/binding-decision-board-art-65/binding-decision-12023-dispute-submitted_en

[223] "Data Protection Commission." [Online]. Available: https://www.dataprotection.ie/news-media/press-releases/data-protection-commission-announces-decision-instagram-inquiry

[224] "Gdpre enforcement tracker - list of gdpr fines," Jan 2024. [Online]. Available: https://www.enforcementtracker.com

[225] "Otter.ai," 2023. [Online]. Available: https://otter.ai/

[226] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, and D. Yang, "Is chatgpt a general-purpose natural language processing task solver?" *arXiv preprint arXiv:2302.06476*, 2023.

[227] M. E. Fonteyn, B. Kuipers, and S. J. Grobe, "A description of think aloud method and protocol analysis," *Qualitative health research*, vol. 3, no. 4, pp. 430–441, 1993.

[228] M. Gutfleisch, J. H. Klemmer, Y. Acar, S. Fahl, and M. A. Sasse, "Recruiting software professionals for research studies: Lessons learned with the freelancer platform upwork," *ROPES-ICSE 2022*, 2022.

[229] B. Wake, "INVEST in Good Stories, and SMART Tasks - XP123," Aug. 2003. [Online]. Available: https://xp123.com/invest-in-good-stories-and-smart-tasks/

[230] M. L. Siddiq, L. Roney, J. Zhang, and J. C. D. S. Santos, "Quality assessment of chatgpt generated code and their use by developers," in *Proceedings of the 21st international conference on mining software repositories*, 2024, pp. 152–156.

[231] M. Biekart and F. Dalpiaz, "Toward threshold concepts for teaching requirements engineering in higher education," 2025.

[232] N. Wake, A. Kanehira, K. Sasabuchi, J. Takamatsu, and K. Ikeuchi, "Bias in emotion recognition with chatgpt," 2023. [Online]. Available: https://arxiv.org/abs/2310.11753

[233] R. Mohanani, I. Salman, B. Turhan, P. Rodríguez, and P. Ralph, "Cognitive biases in software engineering: A systematic mapping study," *IEEE Transactions on Software Engineering*, vol. 46, no. 12, pp. 1318–1339, 2018.

[234] B. Nuseibeh and S. Easterbrook, "Requirements engineering: a roadmap," in *Proceedings of the Conference on The Future of Software Engineering*, ser. ICSE '00. New York, NY, USA: Association for Computing Machinery, 2000, p. 35–46. [Online]. Available: https://doi.org/10.1145/336512.336523

[235] H. C. Mann, Jyoti, "Silicon Valley's next act: bringing 'vibe coding' to the world," Feb 2025. [Online]. Available: https://www.businessinsider.com/vibe-coding-ai-silicon-valley-andrej-karpathy-2025-2