This is a post-peer-review, pre-copyedit version of this book chapter.

Chapter 1

Elicitation Revisited for More Inclusive Requirements Engineering

James Tizard, Tim Rietz, and Kelly Blincoe

To create inclusive software, development teams need to consider how they identify inclusive requirements for a software product. Requirements elicitation is the first stage in the process of developing the requirements of a software system. Elicitation is about describing the functionality, reliability, efficiency, and usability of the system to be developed, so that it suits the end-users' needs [12].

Recent research has found that both traditional elicitation techniques (e.g. user interviews) and newer online crowd-based approaches may have challenges in gathering the views of a diverse set of users. In particular, there are significant challenges in eliciting requirements from users with cognitive disabilities, as well as ensuring that the full demographic spectrum (e.g. by gender, age, ethnicity) of a user-base is adequately represented.

This chapter discusses the motivations for more inclusive requirements elicitation, the challenges which need to be overcome, and finally makes recommendations for both requirements

engineering practitioners and researchers.

Motivation for Inclusive Elicitation

Understanding user needs and desires for a software product is a critical part of modern software development. In the modern software landscape, development teams must keep their users happy to remain competitive as, in many cases, the competition is just one click away. Elicitation of user needs is central in both the initial design of the software, and in it's ongoing maintenance and evolution. A 2021 survey of software developers found the vast majority of developers (97%) agreed that user feedback gives them a better understanding of user needs and makes them aware of usability issues [25]. Thus, user needs as described in feedback are often being used to drive product development decisions.

However, the user-base of software products can be extremely diverse, in terms of demographics (e.g. age, gender, geography, cultural background), as well as specialised needs related to physical and cognitive disabilities [4], [12], [22], [23]. If the diversity of the users being engaged through elicitation processes are not representative of the actual user-base, this introduces the possibility of developing biased software, that does not meet the needs of all users. A clear example of this comes from the broader field of engineering in the design of car safety devices. Women today are still significantly more likely to be seriously injured or killed in car accidents because car safety devices were designed and tested considering the size of the average man's body [7]. There are also many examples of software systems failing to consider the needs of all users. When YouTube first launched its mobile app, approximately 10% of videos were being uploaded upside-down because the software did not accommodate left-handed users[1]. More recently, various AI systems have been shown to be biased against some users. For example, Amazon's recruiting tool was found to be biased against women[2] and Twitter's image cropping tool had inbuilt racial biases[3].

Focusing on the needs of under-served people can make products better for everyone. Again, looking at an example in the broader field of engineering, curb cuts, which were original-

[1] https://www.cio.com/article/234087/consciously-overcoming-unconscious-bias.html
[2] https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G
[3] https://www.cbsnews.com/news/twitter-kills-its-automatic-cropping-feature-after-complaints/

ity designed to make city streets more accessible for wheel chair users, have improved the accessibility for many others, including people riding bikes or skateboards, pushing strollers, delivering packages, and pulling suitcases [3]. For a software-specific example, consider closed captioning of videos, which were originally designed to make videos accessible for people with hearing impairments [13]. Today, with the advent of social media, we see captions benefiting nearly everyone since they provide viewing flexibility; people can scroll through their social media feed and watch videos without the volume or consume videos in different languages[4].

Therefore, it is imperative that software requirements elicitation considers the needs of all of its users, to ensure the software is inclusive and fair for everyone. However, recent research has shown that both traditional requirements elicitation methods, as well as recent crowd-based requirement elicitation approaches, have representation challenges. These challenges are discussed in the next section. (See Chapter [PAPER-Grundy-InclusiveSoftware] for additional diversity-related challenges faced throughout the software development lifecyle.)

Challenges in Traditional Elicitation

Traditionally, software requirements have been elicited through methods such as interviews, focus groups, observations, and questionnaires. However, these approaches may miss segments of society. They need special focus to include diverse perspectives. Especially since many of these traditional methods can only engage with a limited number of users, due to time and resource constraints. For example you can't interview every user of your software product, and must instead engage with an extremely small sample, relative to the total user-base. Other techniques, such as design thinking as described in Chapter [PAPER-Devathasan-DevTeams], can provide rich insights into the needs of users, but also face similar scalability problems.

Traditional techniques can also have a lot of inherent bias, in both the selection of elicitation participants, and selective perception during elicitation. The requirement engineers' own perceptions can cloud how they understand requirements. This can lead to miscommunications and a lack of shared understanding, which may produce misunderstood or simply missed requirements.

Inclusivity in traditional requirements elicitation requires intensive communication between

[4] https://www.3playmedia.com/blog/benefits-captioned-social-media-videos/

all participating stakeholders, especially when engaging users with cognitive disabilities [12]. In a recent study, researchers recommended an approach based in User Centred Design (UCD) [16], to engage with users with cognitive disabilities [12]. They found requirement development with those with cognitive disabilities was feasible, with the participants proving to be reliable interview partners, which were quite capable of expressing their needs for a software product. Through this process the development team gained a deeper insight into the requirements of their end users, which led to new interaction and information presentation concepts.

Collecting requirements from a diverse set of users may require a diverse set of traditional elicitation techniques, as not everyone would be comfortable, or able, to participate using the same methods. Therefore, an inclusive approach to traditional elicitation can be time consuming and expensive. For it to be done well, a development team needs a lot of motivation and drive to focus on inclusion. This can be a challenge in many software projects, where time-to-market, or other business factors, may also be an important consideration. These challenges were emphasised in a recent study, that found software companies often do not prioritize accessibility needs in practice [17]. They cited various reasons for this, including that there are no methods or tools available to help the teams with this process, and a general lack of training on how to consider accessibility needs.

Newer crowd-based elicitation approaches, can give developers access to large volumes of diverse user perspectives, through mining online channels such as app stores, social media, and support forums. However, recent research has highlighted representation challenges here also, which are discussed in the next section.

Challenges in Crowd-Based Elicitation

Online crowd-based elicitation is a modern approach that has promise for eliciting requirements from a diverse set of users. There are large volumes of user feedback on online channels, such as app stores, social media (e.g. Twitter), and user support forums. Recent research has identified significant amounts of requirement relevant information in each of these channels, including bug reports and feature requests [22]. Through mining user opinions online, requirements engineers are no longer limited by time, and other resource limitations that constrain

the number of users who can be involved in more traditional elicitation techniques, such as interviews, or focus groups.

Recent research has found a diverse set of users give feedback on these channels, with respect to traditional demographic categories (e.g. age, gender), geographic location, and accessibility needs [10], [11], [22], [23]. However, this research also suggests the representation across these groups in online feedback may not be in proportion to the actual user-bases of software products. Without considered attention, requirements generated from online feedback will disproportionately represent the loudest voices online, and miss groups that are underrepresented.

This problem was emphasised in Tizard et al.'s 2020 survey of software users, where women reported to give significantly less online feedback than men, across all the studied channels (app stores, forums, social media) [23]. This was in-line with Guzman et al.'s earlier gender study of feedback on the Apple app store [11]. With age, the 2020 survey found that software users between 35 and 44 years reported to give the most feedback on all channels, with younger and older respondents reporting to give significantly less feedback. Research has also found that feedback behavior varies significantly between different countries, and may be impacted by cultural factors [6], [10], [22].

Due to the large volume of online feedback, it is often necessary to prioritise user requests for development attention. One popular approach to prioritisation is to find requests that are made frequently [5], [9], [14], [15]. However, this may exacerbate the issue of considering the views of underrepresented groups. An additional challenge is that online feedback often doesn't contain much demographic information about feedback givers, meaning directly identifying requests from underrepresented groups is difficult, or perhaps impossible [23].

Those with physical or cognitive disabilities may also be missed in requirements generated from online feedback. A recent study of user reviews on the Google Play store identified requests related to accessibility needs, including vision, hearing, and cognitive impairment [4]. However, all the accessibility requests combined made up just 1.2% of the sampled app reviews. Therefore, these accessibility requests would certainly be missed by prioritization techniques based on frequency.

While mining user opinions online is a promising approach to source valuable requirements information, there remain challenges in ensuring the generated requirements are representative of the underlying user-base of a software product. In the final section below, we make rec-

ommendations for practitioners on how to elicit the most representative user views for software products, with the goal of producing products that meet the needs of the broadest possible set of users. We also make recommendations for researchers, identifying several promising paths forward to better understand representation issues in requirements elicitation, and develop new approaches to address these challenges.

Recommendations for Inclusive Requirement Elicitation

Recommendations for Practitioners

Where users are being directly engaged through more traditional elicitation techniques (e.g. interviews), requirements engineers must take initiative to understand the diversity within their user-base, and engage with them. Collecting requirements from a diverse set of users may require a diverse set of approaches, as not everyone will be comfortable, or able, to participate using the same methods [12]. In the case of users with cognitive disabilities Heumader et al. recommends an approach based on User Centered Design, finding that their process produced meaningful insights into the user needs.

Another possibility is for software teams to utilize the method described in Chapter [PAPER-Hamid-Methodology] to measure diversity gaps in their requirements elicitation process using a GenderMag survey. By employing this survey, teams could gain insights into the cognitive styles of those participating in the requirements elicitation process, allowing them to identify who is missing from the process from a cognitive style perspective. As described in Chapter [PAPER-Hamid-Methodology], cognitive styles can give insight into how users interact with software systems. Therefore, this survey can help teams identify whose voices are missing.

Crowd-based elicitation, where user opinions are mined from online feedback channels can overcome many of the time and resource constraints associated with traditional elicitation approaches. Online user feedback has been found to contain much requirement relevant information, from a diverse set of users [19] [23]. Analysis tools are available to help automatically

extract relevant information from the large volumes of feedback, which have shown promising performance in research settings [8], [20], [24].

As discussed, there are still challenges in ensuring user views mined online are representative. In their 2020 user study, Tizard et al. recommends that to elicit the most representative requirement information, development teams should consider feedback from multiple feedback channels. Their study found that different demographics were more likely to engage with different feedback channels. For example, younger software users reported to be more likely to engage with app stores, whereas older users may prefer support forums. They also found that a majority of feedback givers reported only engaging with one online feedback channel, therefore focusing on a single channel will certainly miss some users.

While the lack of demographic information, such as age and gender, continues to be a practical problem for mining the views of underrepresented groups in online feedback. Being aware that women and certain age groups may be underrepresented, gives requirements engineers the option to directly engage with those groups to supplement online feedback mining. Traditional elicitation techniques such as interviews or questionnaires, will be effective tools to target underrepresented demographics.

Mining user opinions from different geographic locations is more achievable in the current online landscape, as country-level location data is often available. For example the apple app store divides itself by country, and location data is often available for feedback givers on social media (e.g. Twitter). Requirements engineers can therefore sample user opinions to closely match a geographically diverse users-base. In doing so, the diverse views of users from different backgrounds and cultures can be uncovered, and help broaden the appeal of a software product.

Finally, there are smart analysis tools available to help extract accessibility requests from app store reviews. As previously mentioned, recent research found accessibility requests in app reviews related to vision, hearing, and cognitive impairment, among others [4]. These reviews were identified with keyword searches, followed by manual analysis. Subsequent research then applied the identified reviews to build smart analysis tools to automatically extract accessibility requests with promising accuracy, which have been made available [1].

CHAPTER 1

Recommendations for Researchers

For traditional elicitation techniques, there are several promising avenues of research to improve inclusivity. Heumader et al.'s work [12], points to a path forward in eliciting requirements from those with cognitive disabilities. They suggest the investigation of approaches that combine two existing design methods: Inclusive Participatory Action Research (IPAR) [18], and User Centred Design (UCD) [16], showing promising early results. Similarly, Chapter [PAPER-Devathasan-DevTeams] described success using design thinking techniques to elicit requirements from diverse users. However, such techniques are time intensive and difficult to scale to a large number of users. Another direction for research, is to address the challenges of scale facing traditional elicitation techniques, such as interviews and focus groups. For example automated conversational agents, such as LadderBot [21], hold promise for overcoming the time and location constraints of person-to-person elicitation. A conversational agent could enable end users to articulate needs and requirements, by mimicing a human (expert) interviewer. By automating the interview process, a significantly larger sample of a user-base could be engaged. Combined with the ability to target potentially underrepresented groups, automated user interviews hold significant potential to support inclusive requirements elicitation. Future work can evaluated the effectiveness of new conversational agents (e.g. LadderBot) against traditional person-person interviews, and digital questionnaires. Additionally, these chat bot approaches would be well suited to evaluation in lab based experiments.

Crowd-sourcing software requirements has been a significant focus for requirements engineering researchers in recent years. Traditionally, the Crowd has been conceptualised from a high-level, taking an aggregated view of their needs. However, as discussed, a growing number of studies suggest feedback habits and attitudes vary significantly between user groups (*e.g. with gender, age, country*). In the interest of more representative requirements engineering, researchers should continue to follow current trends, and investigate a more fine grained view of the Crowd. With this goal in mind, we see three key area's for research: 1) Continue to investigate the representativeness of online feedback, and so identify areas where there are representation issues; 2) Investigate the causes of representation issues, such as feedback channel design, and the impact of culture; 3) Investigate new approaches to encourage more representative feedback. These research directions are discussed below.

Researchers should continue to investigate the representativeness of online feedback. Per-

haps the primary challenge in understanding who gives online feedback, is that feedback channels give very little information about their users. On some feedback channels, such as the Google Play Store, even the full name of the person providing the feedback is often unavailable. Recent research has made progress through indirect analysis techniques, such as user surveys, inferring gender through usernames, and comparing the content of feedback across regions in the apple app store [6], [10], [11]. Looking forward, these research approaches can continue to be leveraged, in particular directly engaging software users (e.g. user surveys) continues to hold promise for gaining meaningful insights.

One avenue open for new research, is the study of additional feedback channels, beyond the existing studies. The gender and regional analysis studies from Guzman and Fisher ([6], [10], [11]) focused on the apple app store. While, Tizard et al.'s user survey studies ([22], [23]) focused on app stores, social media, and product forums. Extending these studies to additional feedback channels (e.g. issue trackers) would likely provide additional insights into online feedback behaviour.

Future work should also endeavour to understand additional demographic and minority groups within the crowd, and could also be extended to include intersectionality between groups [23]. For example, little is known about the ethnicity of feedback givers, or differences across the economic spectrum. With gender, current work has been limited to only the differences between participants who identified as men and women. This can be extended to understand the feedback behaviour of non-binary software users. There is also significant room to continue to investigate differences in feedback behaviour between countries [6], [10], [22].

The second main research direction we see, is to investigate the causes of representation issues in online feedback. Previous work found underrepresented groups were more likely to cite several key reasons not to give online feedback. For example, both women and those under 25 years old, more frequently (than their counterparts) reported that they found app stores confusing or hard to use, felt a resolution to their problem would take too long, and to not be aware feedback could influence software improvements. Research has found that most software has gender inclusivity issues [2], so it's possible that similar inclusivity issues exist in the software that collects online feedback.

A recent study also found software users in China and Germany reported significantly diverging reasons not to give feedback, and suggested underlying cultural factors, such as collectivism and power distance [22]. Similar to other underrepresented groups, Chinese respon-

dents were more likely (than Germans) to find online channels confusing or hard to use, and were less likely to be aware they could influence software improvements through their feedback. Future research should investigate why certain groups are disproportionately impacted by these factors. There is also significant room to investigate differences in the motivations to give feedback between counties, and the possible impact of culture.

Finally, researchers should investigate new approaches to encourage more representative online feedback. One promising direction for investigation is to directly address the factors underrepresented groups identify for not giving feedback, as discussed above. Methods proposed by software users in previous work, hold promise for addressing these challenges, and should be investigated [23]. For example, giving a quick response to online feedback could be used to emphasise the connection to software improvement, and help address the perception that a resolution will take too long. Clearly showing a track record of addressing feedback could also promote awareness of the process, and help motivate user input. Future work should also investigate feedback interfaces that underrepresented groups find encouraging and easy to use. Lab trials could be carried out to evaluate if the approaches identified above encourage feedback in a practical context.

Summary

Understanding and addressing user needs through diligent requirement elicitation is critical to success in the modern software landscape. In this chapter, we described the challenges in gathering views from a diverse set of users. Traditional elicitation methods (e.g interviews) can exclude a significant proportion of the user-base due to practical constraints, such as limited time. They can also suffer from bias and misunderstandings. For crowd-based elicitation, certain demographic groups can be significantly underrepresented in the online feedback it leverages. This issue is exacerbated by the lack of demographic information available about the online feedback givers, meaning it's difficult (or impossible) to target feedback from specific groups. We make several recommendations to help practitioners overcome these challenges, including using various elicitation techniques to accommodate diverse users, employing user centered design practices, and various strategies to increase the diversity of those participat-

ing in the requirement elicitation process. Finally, we outlined several promising directions for requirement engineering researchers to advance the literature on inclusive requirement elicitation.

Bibliography

[1] Eman Abdullah AlOmar, Wajdi Aljedaani, Murtaza Tamjeed, Mohamed Wiem Mkaouer, and Yasmine N El-Glaly. Finding the needle in a haystack: On the automatic identification of accessibility user reviews. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pages 1–15, 2021.

[2] Margaret Burnett, Simone Stumpf, Jamie Macbeth, Stephann Makri, Laura Beckwith, Irwin Kwan, Anicia Peters, and William Jernigan. Gendermag: A method for evaluating software's gender inclusiveness. *Interacting with Computers*, 28(6):760–787, 2016.

[3] Elizabeth F Churchill. Putting accessibility first. *Interactions*, 25(5):24–25, 2018.

[4] Marcelo Medeiros Eler, Leandro Orlandin, and Alberto Dumont Alves Oliveira. Do android app users care about accessibility? an analysis of user reviews on the google play store. In *Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems*, pages 1–11, 2019.

[5] Layan Etaiwi, Sylvie Hamel, Yann-Gaël Guéhéneuc, William Flageol, and Rodrigo Morales. Order in chaos: prioritizing mobile app reviews using consensus algorithms. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 912–920. IEEE, 2020.

[6] Ricarda Anna-Lena Fischer, Rita Walczuch, and Emitza Guzman. Does culture matter? impact of individualism and uncertainty avoidance on app reviews. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, pages 67–76. IEEE, 2021.

[7] Jason Forman, Gerald S Poplin, C Greg Shaw, Timothy L McMurry, Kristin Schmidt, Joseph Ash, and Cecilia Sunnevang. Automobile injury trends in the contemporary fleet: Belted occupants in frontal collisions. *Traffic injury prevention*, 20(6):607–612, 2019.

[8] E. Guzman, R. Alkadhi, and N. Seyff. A needle in a haystack: What do twitter users say about software? In *2016 IEEE 24th International Requirements Engineering Conference (RE)*, pages 96–105, Sep. 2016.

[9] Emitza Guzman, Mohamed Ibrahim, and Martin Glinz. Prioritizing user feedback from twitter: A survey report. In *2017 IEEE/ACM 4th International Workshop on CrowdSourcing in Software Engineering (CSI-SE)*, pages 21–24. IEEE, 2017.

[10] Emitza Guzman, Luís Oliveira, Yves Steiner, Laura C Wagner, and Martin Glinz. User feedback in the app store: a cross-cultural study. In *2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, pages 13–22. IEEE, 2018.

[11] Emitza Guzman and Andres Paredes Rojas. Gender and user feedback: An exploratory study. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pages 381–385. IEEE, 2019.

[12] Peter Heumader, Cordula Edler, Klaus Miesenberger, and Sylvia Wolkerstorfer. Requirements engineering for people with cognitive disabilities–exploring new ways for peer-researchers and developers to cooperate. In *International Conference on Computers Helping People with Special Needs*, pages 439–445. Springer, 2018.

[13] Richang Hong, Meng Wang, Mengdi Xu, Shuicheng Yan, and Tat-Seng Chua. Dynamic captioning: video accessibility enhancement for hearing impairment. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 421–430, 2010.

[14] Swetha Keertipati, Bastin Tony Roy Savarimuthu, and Sherlock A Licorish. Approaches for prioritizing feature improvements extracted from app reviews. In *Proceedings of the 20th international conference on evaluation and assessment in software engineering*, pages 1–6, 2016.

[15] Fitsum Meshesha Kifetew, Anna Perini, Angelo Susi, Aberto Siena, Denisse Muñante, and Itzel Morales-Ramirez. Automating user-feedback driven requirements prioritization. *Information and Software Technology*, 138:106635, 2021.

[16] Travis Lowdermilk. *User-centered design: a developer's guide to building user-friendly applications*. " O'Reilly Media, Inc.", 2013.

[17] Darliane Miranda and João Araujo. Studying industry practices of accessibility requirements in agile development. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 1309–1317, 2022.

[18] Janice Ollerton. Ipar, an inclusive disability research methodology with accessible analytical tools. *International Practice Development Journal*, 2(2), 2012.

[19] D. Pagano and W. Maalej. User feedback in the appstore: An empirical study. In *2013 21st IEEE International Requirements Engineering Conference (RE)*, pages 125–134, July 2013.

[20] Sebastiano Panichella, Andrea Di Sorbo, Emitza Guzman, Corrado A. Visaggio, Gerardo Canfora, and Harald C. Gall. Ardoc: App reviews development oriented classifier. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, FSE 2016, pages 1023–1027, New York, NY, USA, 2016. ACM.

[21] Tim Rietz and Alexander Maedche. Ladderbot: A requirements self-elicitation system. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pages 357–362. IEEE, 2019.

[22] James Tizard, Tim Rietz, Xuanhui Liu, and Kelly 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)*, pages 328–335. IEEE, 2021.

[23] James Tizard, Tim Rietz, Xuanhui Liu, and Kelly Blincoe. Voice of the users: an extended study of software feedback engagement. *Requirements Engineering*, 27(3):293–315, 2022.

[24] James Tizard, Hechen Wang, Lydia Yohannes, and Kelly Blincoe. Can a conversation paint a picture? mining requirements in software forums. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pages 17–27. IEEE, 2019.

[25] Simon van Oordt and Emitza Guzman. On the role of user feedback in software evolution: a practitioners' perspective. In *2021 IEEE 29th International Requirements Engineering Conference (RE)*, pages 221–232, 2021.