Accessibility in Software Practice: A Practitioner's Perspective

TINGTING BI, the Faculty of Information Technology, Monash University, Melbourne, Australia.
XIN XIA*, the Faculty of Information Technology, Monash University, Melbourne, Australia.
DAVID LO, the School of Information Systems, Singapore Management University, Singapore.
JOHN GRUNDY, the Faculty of Information Technology, Monash University, Melbourne, Australia.
THOMAS ZIMMERMANN, Microsoft Research, Redmond, WA, USA.
DENAE FORD, Microsoft Research, Redmond, WA, USA.

Being able to access software in daily life is vital for everyone, and thus accessibility is a fundamental challenge for software development. However, given the number of accessibility issues reported by many users, e.g., in app reviews, it is not clear if accessibility is widely integrated into current software projects and how software projects address accessibility issues. In this paper, we report a study of the critical challenges and benefits of incorporating accessibility into software development and design. We applied a mixed qualitative and quantitative approach for gathering data from 15 interviews and 365 survey respondents from 26 countries across five continents to understand how practitioners perceive accessibility development and design in practice. We got 44 statements grouped into eight topics on accessibility from practitioners' viewpoints and different software development stages. Our statistical analysis reveals substantial gaps between groups, e.g., practitioners have Direct v.s. Indirect accessibility relevant work experience when they reviewed the summarized statements. These gaps might hinder the quality of accessibility development and design, and we use our findings to establish a set of guidelines to help practitioners be aware of accessibility challenges and benefit factors. We suggest development teams put accessibility as a first-class consideration throughout the software development process, and we also propose some remedies to resolve the gaps between groups and to highlight key future research directions to incorporate accessibility into software design and development.

ACM Reference Format:
Tingting Bi, Xin Xia, David Lo, John Grundy, Thomas Zimmermann, and Denae Ford. 2021. Accessibility in Software Practice: A Practitioner's Perspective. *ACM Trans. Softw. Eng. Methodol.* 1, 1 (December 2021), 26 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Software systems are becoming increasingly complex, and the increasing need to use them in both work and daily life makes users ever more dependent on them. *Accessibility* is one of the prominent software qualities that determines usability and acceptance of a software product in

*Corresponding author: Xin Xia

Authors' addresses: Tingting Bi, the Faculty of Information Technology, Monash University, Melbourne, Australia. tingting.bi@monash.edu; Xin Xia, the Faculty of Information Technology, Monash University, Melbourne, Australia. xin.xia@monash.edu; David Lo, the School of Information Systems, Singapore Management University, Singapore. davidlo@smu.edu.sg; John Grundy, the Faculty of Information Technology, Monash University, Melbourne, Australia. john.grundy@monash.edu; Thomas Zimmermann, Microsoft Research, Redmond, WA, USA., tzimmer@microsoft.com; Denae Ford, Microsoft Research, Redmond, WA, USA., denae@microsoft.com.

1049-331X/2021/12-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

today's competitive market [22, 41]. An accessible information technology solution that is usable by people with a wide range of differences exhibits "universal design" [11]. Universal design is a collection of concepts and approaches that can be applied and incorporated into a product so that end-users can easily access software applications [25]. Most software systems assume that users can efficiently perform all the following tasks: read the text and images displayed on the screen and respond to them; use the keyboard for typing; use the mouse for selecting text or other information; hear the sounds and react to them, and manipulate items using mouse, finger or gesture [10]. However, many groups of people have difficulties in performing one or more of the above tasks due to physical and/or mental challenges (see Table 3 in Appendix) that prevents them from using many popular software applications [29, 31, 34].

The importance of accessibility is increasingly being recognized and many countries have mandated accessibility laws and policies for organizations. For example, W3[1] lists a set of governmental policies related to web accessibility (i.e., required Web accessibility goals or implementation). In addition, for an application to be successful, it is not enough to satisfy all functional requirements that are expected of it. Accessibility needs to be considered at all stages of the application development life cycle, for example, requirements specification, design, implementation, and testing [52]. Nevertheless, it is not clear how accessibility features are currently incorporated into the software engineering process and current software products [47]. The initial omission of accessibility features often results in drawbacks, such as poor usability, poor user feedback, and considerable re-engineering. Furthermore, accessibility requirements have evolved – accessibility is now more likely to ensure a wider range of end-users can use their software efficiently and effectively [45].

A variety of research focuses on understanding accessibility development and design for specific disabilities [3, 54]. For example, previous work has investigated accessibility issues in Android apps [2, 47]. Whilst these works are valuable for understanding accessibility issues, there is a paucity of empirical evidence on how practitioners actually perceive accessibility in practice, and how projects go about providing accessibility features for different end-users. A better understanding on the characteristics of accessibility and software practitioners' opinions would provide more tailored support for accessibility design and implementation. The following gaps illustrate that accessibility needs to be considered in software development and design:

- Accessibility issues are often addressed in the later development stages [15]: both the Human Computer Interaction (HCI) and Software Engineering (SE) communities report that accessibility plays a crucial role in development [14, 40]. Accessibility is however sometimes not a requirement in the first place when developing the projects.
- Difficulties in incorporating accessibility into projects: special knowledge on disabilities and standards about accessibility features is required [9]. Trade-offs of incorporating and managing accessibility during the software life cycle are hard to address [37].
- Accessibility is essential for both general and specific end-users. Accessibility should be considered throughout the whole software development life-cycle. Most importantly, accessibility that is not well managed (e.g., remaining implicit or becoming late requirements) can lead to severe software development issues [39].

Software engineering plays a fundamental role in developing accessible applications since it promotes the integration between methodologies and specific accessibility techniques and activities in the software development process [35]. Given the importance of accessibility, we aim to determine accessibility development and design in software practice. As much of this field has not been grounded in the practitioners' perspective, we conducted a mixed quantitative and qualitative study to gather data from their viewpoint. In this paper, we advocate that accessibility needs to be treated

[1]https://www.w3.org/WAI/policies/

as a first-class issue throughout software development, i.e., accessibility considerations are critical for improving end-users use of software products. This is also corroborated by other studies [2, 41]. The main contributions of this work include:

- We investigate how practitioners perceive accessibility design and development in practice. To the best of our knowledge, this is the first attempt to investigate how prevalent accessibility needs are incorporated into software projects from the participants' perspectives. We present results of 15 detailed interviews and an online survey with 365 responses from 28 countries.
- Our analysis of our interviews and surveys show that accessibility is a prevalent consideration in software practice. However, it is often addressed in a simple way and with short term goals. In addition, organizational factors strongly impact the success of accessibility for many projects.
- We highlight from this empirical evidence how accessibility fits into software project considerations, and identify a set of challenges that our participants report that their projects face when addressing these accessibility issues. We provide a set of guidelines based on this participant feedback to help practitioners be aware of these challenges. We also list suggestions for incorporating accessibility considerations into projects.
- We identify a set of gaps between work experiences of different participants, e.g., between Web and Mobile App development practitioners on how they perceive accessibility. We attempt to form a better understanding of accessibility challenges in different domains and contexts.

The remainder of this paper is structured as follows: Section 2 describes our study. Section 3 and Section 4 answers and discusses the results of our study, respectively. Section 5 presents the threats to validity. Section 6 introduces background and related works. Finally, Section 7 concludes this work with future directions.

2 STUDY DESIGN

2.1 Objective and research questions

In this work, we aim to understand and identify accessibility challenges and issues in practice. Following this aim, we guided our study with the following two research questions.

RQ1: What are practitioners' perceptions of accessibility development and design in practice?

Motivation: We wanted to explore how practitioners perceive accessibility development and design in their software development and design practices. We explore this question from the following perspectives.

- **Understanding accessibility**: What level do they consider the importance of accessibility and what motivates practitioners to incorporate and address accessibility in practice?
- **Work characteristics**: How do developers' work characteristics impact accessibility development and design in practice? For example, (1) *Skill variety*: Accessibility design and development intensively requires specific knowledge. (2) *Task complexity and problem-solving*: Accessibility design and development focuses on specific software aspects to enhance accessibility. (3) *Task identification*: Is it harder to make an accurate plan for accessibility-related design tasks? (4) *Interaction*: To incorporate accessibility considerations, development teams may have to communicate more frequently with their clients.
- **Organizational factors**: Whist software functional requirements and quality requirements are well defined in shaping software design, other factors, such as company culture, development size, and development platform, may impact the ability of developers to incorporate accessibility into a system [6].

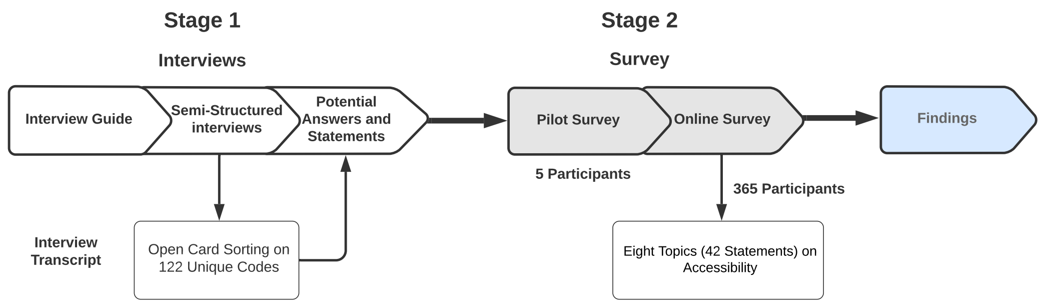

Fig. 1. Sequential mixed-methods approach includes semi-structured interviews and an online survey.

RQ2: How does addressing accessibility needs fit into the software development life cycle?

Motivation: We wanted to systemically investigate how accessibility fits into the software development life cycle from a practitioner's perspective, including requirements elicitation, software design, implementation, testing, and evaluation. Answers to this RQ can help us to characterize current accessibility design approaches, limitations, and highlight future work directions.

- **Requirements for accessibility**: Collecting requirements to incorporate accessibility involves more preliminary efforts. We plan to understand what challenges practitioners have met when they elicit accessibility requirements.
- **Design and implementation for accessibility**: Detailed design incorporating accessibility may be more time-consuming and tends to be conducted in an intensively iterative way. What issues developers meet when they are dealing with accessibility design and potential solutions that have been proposed?
- **Testing and quality assurance for accessibility**: Quality assurance is important in any project. However, good performance during testing can not guarantee accessibility of a system to a diverse range of users. Furthermore, it is hard to involve a sufficiently broad end-user base for conducting comprehensive accessibility testing.
- **Evaluation for accessibility**: Accessibility requirements drive relevant functionality implementation. They are also necessary to consider when making trade-offs to implement accessibility requirements. How do practitioners evaluate the process of software development to incorporate accessibility into the system?

2.2 Methodology

Our research methodology, followed a mixed qualitative and quantitative approach [17], as depicted in Fig. 1. In the first stage, we conducted open-ended and detailed interviews with 15 software practitioners to get their opinions on software development and design about accessibility in practice. In the second stage, we conducted an online survey, which received 365 validated responses, to confirm or refute the results of the interviews. We then analyzed data from open-ended interviews and a validation online survey[2]. We presented detailed procedures of each stage in Section 2.2.1 and Section 2.2.2, respectively.

2.2.1 Interviews. **Participant selection**. We recruited full-time software practitioners from three IT companies in China and the U.S., namely, Alibaba, Hengtian, and Microsoft. Interviewees were

[2]The number of our Human Research Ethics Committee approval from Monash University is 24733.

recruited by emailing our personal contacts in each company. Volunteers would inform us if they were willing to participate in the study with no compensation. With this approach, we included 15 interviewees in this study. The 15 interviewees specialize in accessibility design, and they have varied job roles and experience with 9 years of professional experience on average (min 4 years and max 15 years). In the remainder of this paper, we refer to these 15 interviewees as I1 to I15. To preserve the anonymity of participants, we anonymized all items that constitute Personally Identifiable Information (PII) before analyzing the data [30]. In addition, based on the groups of disabilities and their potential difficulties that are summarized in Table 3, we asked our interviews which areas (i.e., types of disability groups) they focus on when they conduct accessibility design and development in Table 4 of Appendix.

Interviewing Process. The first author conducted a series of interviews with the 15 interviewees, and each interview was completed within 30 minutes. The interviews were semi-structured and divided into three parts.

Part 1: We asked some demographic questions, such as the interviewees' experience in software development /testing /project management.

Part 2: We asked open-ended questions to understand their opinions on software design and development for accessibility. The questions include: (1) their general opinions on accessibility (e.g., concepts of accessibility in practice); (2) accessibility use cases; (3) requirements elicitation used to inform accessibility design; (4) software design and implementation for accessibility; (5) testing and evaluation for accessibility; (6) tools and standards they have applied or followed for accessibility.

Part 3: We prepared candidate topics by carefully reading the contents of representative textbooks. We picked a list of topics that have not been explicitly mentioned in the open discussion and asked the interviewees to discuss those topics further. At the end of each interview, we thanked the interviewees and briefly informed them what we plan to do with his/her response.

Transcribing and card sorting. We used a commercial transcription service provided by a third-party company to transcribe recordings to transcripts. We then read the transcripts and conducted a thematic coding analysis of the transcripts [12]. We dropped sentences during the coding process that are not related to "software design and development for accessibility in practice". We followed the card sorting approach [42] to analyze and categorize the interview textual data.

Step 1: The first two authors read the transcripts and coded the contents of the interviews using the MAXQDA[3] tool for analyzing the qualitative data. We coded sentences into cards, and a total of 288 cards that contain the codes were generated. After merging the cards with the same words or meaning, we had a total of 122 unique cards. We noticed that when our interview transcripts reached saturation and new cards did not appear anymore. Our list of codes was then considered stable.

Step 2: We randomly chose 50% of the cards (i.e., 61 cards), and first two authors read the information of the card to understand the accessibility discussions. Then, the first two authors summarized the high-level themes of the discussion. After this iteration, all the topics (T1-T8) are listed in Table 1. To ensure the quality of coding, we invited a Ph.D. candidate to verify the first two authors' initial topics and provided suggestions for improvement.

Step 3: The first two authors independently categorized the remaining 50% of the cards into initial categories by following the same method, described in step 2. During this step, we did not find any new topic. The first two authors compared their results and discussed any differences. Any differences were discussed among the authors and reached an agreement. We then used Cohen's Kappa [13] measure to examine the agreement between the annotators, and the overall Kappa value is 0.79, which indicates substantial agreement between the two annotators. To reduce bias from

[3]https://www.maxqda.com/

two annotators sorting the cards to form initial themes, they reviewed and agreed on the final set of themes. Finally, we summarized 42 statements grouped into eight topics about accessibility from a variety of perspectives and across different development phases.

2.2.2 Survey. We then conducted an online survey aiming at confirming, refuting, and extending our results from the 15 interviews. We followed Kitchenham and Pfleeger's guideline for personal opinion surveys [26] and used an anonymous survey to increase the response rate [32, 46].

Survey design. Our practitioner survey includes different types of questions, i.e., multiple choice and free-text answer questions. We first piloted the preliminary survey with a small set of practitioners (i.e., 5 participants) who came from Alibaba. We obtained the feedback on (1) whether the length of the survey was appropriate, and (2) the clarity and understandability of the terms. We made minor modifications to the draft of the survey based on the received feedback and produced a final version. Note that the collected responses from the pilot survey are excluded from the presented results in this paper. The first author translated the survey into Chinese and this was checked for accuracy by the second author. Our questionnaire can be found in the following links[4].

The survey consists of three sections: (1) Demographic information; (2) Statement scoring; (3) and Rationale and Suggestions on "software design for accessibility".

- Demographics. We collected geographical location, the respondent's role, size of the team the respondent is in or manages, number of years the respondent has been in their current role, and accessibility work experience, and type of applications that they have developed. We designed two questions about their accessibility working experience. **Question 1**: "*Have you ever done any software activities (e.g., design, coding, and testing) supporting software accessibility*?"; "*Has your development team carried out any accessibility tasks?*", and "*Please briefly describe how familiar are you with software accessibility*?". **Question 2**: "*In which domain, have you done development and design for accessibility? Can you please briefly describe the project?*". The results show that around 56% of participants focus on Auditory disabilities, 37% of participants focus on Blindness, Color blindness and Low Vision, and 7% of participants focus on Cognitive disabilities.
- Statement scoring. In this section, based on the interviews, we summarized 42 statements grouped into eight topics, and we provided online survey participants with those statements. Participants were asked to score the statements according to their work experience and opinions. They assessed the importance of each statement on a 5 point Likert scale and one more extra option (Strongly Disagree, Disagree, Neutral, Agree, Strongly Agree, and I Don't Know). The "I Don't Know" option was provided in case some statements are not applicable to the experience of respondents, or for respondents who had a poor understanding of the statement.
- Rationale and suggestions. Apart from scoring the statements, they also can provide the rationale and suggestions for each statement and topic.

Recruitment of Respondents. To get a sufficient number of respondents from diverse background, we followed a multi-pronged strategy to recruit respondents. No identifying information was required or gathered from our respondents.

- We contacted professionals from a set of companies worldwide and asked them to disseminate our survey. Specifically, we sent emails to our contacts at Alibaba, Baidu, Hengtian, IBM, Microsoft, and other companies. We adopted a snowballing method to encourage them to

[4]English version: https://tinyurl.com/yymexzo9
Chinese version: https://www.wjx.cn/jq/93215056.aspx

disseminate our survey to some of their colleagues willing to participate in our survey. We received 194 responses out of 1,000 sent emails (i.e., 19.4%).
- We sent an email with a link to the survey to 3,987 developers from GitHub projects. We aimed to recruit open-source developers working in the software industry. We obtained 237 responses out of 3,987 sent emails (i.e., 5.9%).

Data analysis of the survey. In total, we obtained 431 responses, and we excluded responses from participants who had no direct or indirect (definitions are given below) accessibility-related work experience as their survey answers were not completed (65% of the survey questions were skipped).

Finally, we included 365 responses for analysis. The 365 respondents reside in 26 countries across five continents, and the top countries in which the respondents reside are China (177 respondents) and the United State (105 respondents). The majority of our responses came from participants falling into two groups: Web App development (181 respondents) and Mobile App development (172 respondents). We analyzed the survey results based on the question types. For multiple-choice questions, we report the percentage of each option selected. To identify the agreement of each statement, we analyzed the Likert-scale ratings. Furthermore, we extracted comments that our survey respondents explain why they give such a score of a particular statement. In our data sample, about 30% of participants have direct accessibility-related work experience. In fact, the actual number of the developers who have accessibility relevant working experience would be lower as it is possible that participants who have such working experience were willing to participate in our survey.

Comparison. To better understand participants' perspectives of accessibility, we divided all survey respondents into different demographic groups, and compared their scoring results towards various challenges and desired improvements mentioned by interviewees. Note that for the size of the teams, we followed the previous work [27], which considers a group with less than 20 developers as a relatively small team:

- Respondents with direct accessibility work experience (108 respondents).
- Respondents with indirect accessibility work experience (i.e., their groups have done accessibility-related work, but the respondents have not been directly involved in relevant tasks) (257 respondents).
- Respondents who are working in a relatively big team (≥ 20) (103 responses).
- Respondents who are working in a relatively small team (<20) (262 responses).
- Respondents who mainly develop Web applications (181 responses).
- Respondents who mainly develop Mobile applications (172 responses).

The statements and the results of the online survey are summarized in Table 1. We analyzed the Likert scale "Agreement" to each statement, and P-value is applied to test whether the discrepancies in the "Agreement" for each statement are statistically significant differences between the two groups at a 95% confidence level. The results show that 15 statements with statistically significant discrepancies between the groups are highlighted with different color cells (the detailed colors are explained below).

In statistics, the Effect Size is a concept that quantifies the difference between variables on a numeric scale (i.e., two groups) [8, 33, 43]. In this paper, we applied Absolute Effect Size, which is to calculate the difference between the average or mean in two different groups. For two independent groups. Absolute effect size can be measured by the standardized difference between groups means or mean (group1)−mean (group 2) / standard deviation (i.e., Cohen's D metric). The Absolute Effect Size values between groups are shown in Table 1. For example, the mean score of S1 for is 4.12, whereas the mean source for practitioners from Direct accessibility related working experience is 3.99 and from Indirect accessibility relevant working experience group is 4.37, and P-value is <0.001; as a consequence, the effect size is 3.99 - 4.37 = - 0.38, which means participant group with **Direct** accessibility relevant work experience agrees this statement more. We use the Dark Grey to indicate the former group is more likely to agree with the statement, and the Light Grey to indicate the latter group is more likely to agree with the statement. We numbered statements in the order (i.e., S1-S44 in Table 1), depending on the topics that we summarized, in which they appeared in the survey.

3 STUDY RESULTS

We report our findings for RQ1 (Section 3.1) and RQ2 (Section 3.2) as identified by using open card sorting on the interview contents. We summarize several interview statements about each topic, and we provide the percentage of agreements and disagreements for each statement from the online survey. We also selected some of most meaningful comments and highlighted some relevant statistics that we derived from our survey responses. We report and focus on the differences across different groups in term of accessibility development and design in Section 3.3. We applied [S1]-[S44] indicate the 44 statements when reporting the results.

3.1 RQ1: What are practitioners' perceptions of accessibility development and design in practice?

We answer RQ1 from three aspects, i.e., understanding of accessibility (see Section 3.1.1), how work characteristics impact accessibility (see Section 3.1.2), and how organizational factors impact accessibility design and development in practice (see Section 3.1.3).

3.1.1 Understanding accessibility. We explored how practitioners understand accessibility, for example, what motivates them to address accessibility issues, and what deters them from incorporating accessibility into their projects.

Table 1. Interview and survey results on accessibility statements. We used bold to highlight the P-values to indicate statistically significant different between groups. Dark grey cells indicate where former agrees more. Light grey cells indicate where the latter group agree more. The number in the Likert Distribution column indicates the size of each group. The bars in the Likert distributions from left to right are: Strongly Disagree (1 score), Disagree (2 scores), Neutral (3 scores), Agree (4 scores), Strongly Agree (5 scores), and I Don't Know option.

Statement		Likert Distribution		Indirect v.s. Direct				Big v.s. Small				Web v.s. App			
		In total	Overall score	Score	Score	P-value	Effect Size	Score	Score	P-value	Effect Size	Score	Score	P-value	Effect Size
T1. General Considerations of Accessibility															
Accessibility needs to be incorporated into all software projects.	S1		4.12	3.99	4.37	**< 0.001**	-0.38	4.15	4.13	0.947	0.02	4.14	3.99	0.306	0.15
Accessibility is not only for people who are unable to use standard software.	S2		4.07	4.01	4.19	0.215	-0.18	4.11	3.98	0.144	0.13	4.13	4.00	0.290	0.13
Accessibility should be integrated with all software activities.	S3		4.05	4.08	4.36	0.345	-0.28	4.15	3.82	0.244	0.33	4.06	3.83	0.890	0.23
Accessibility needs evolve during software development and design.	S4		4.02	4.14	4.25	1.000	-0.09	4.01	3.79	0.806	0.22	3.77	4.15	**<0.001**	-0.38
Goals for accessibility: easy to read, easy to operate, and simple to use.	S5		4.01	4.02	4.14	1.000	-0.08	4.06	3.83	0.066	0.23	4.07	3.89	0.079	-0.18
Accessibility design drives innovation and often solves unanticipated problems.	S6		3.91	3.94	4.11	1.000	-0.17	3.98	3.82	0.265	0.16	4.00	3.62	**<0.001**	0.38
Accessibility is a widely considered concept in software development.	S7		3.75	3.68	4.01	0.244	-0.33	3.83	3.74	1.000	0.09	3.82	3.62	1.000	0.20
T2. Characteristics of Accessibility															
Accessibility is intertwined with multiple activities.	S8		4.04	3.78	4.21	**<0.001**	-0.43	4.18	4.00	0.793	0.18	4.11	3.99	0.435	0.12
Accessibility is dynamic in nature.	S9		4.01	4.17	3.83	**<0.001**	-0.37	4.10	3.91	0.631	0.19	4.09	3.98	0.283	0.11
T3. Work Characteristics															
Accessibility development requires specific knowledge and information.	S10		4.12	4.18	4.14	1.000	0.04	4.21	4.04	1.000	0.17	3.95	4.19	0.922	-0.24
Interaction with outside organizations is needed.	S11		4.07	4.01	4.19	0.148	-0.18	3.88	4.12	0.280	-0.14	4.21	4.00	0.255	0.21
Accessibility task identification is time-consuming.	S12		4.01	4.01	3.88	1.000	- 0.13	4.03	4.08	1.000	- 0.05	3.89	4.28	**0.002**	-0.39
T4. Organizational Factors															
Accessibility is a marketing strategy.	S13		4.18	4.21	4.28	0.979	-0.07	4.12	4.01	0.897	0.11	4.28	4.22	0.265	0.06
Most commercial and mature projects abide by accessibility design principles.	S14		4.13	4.11	4.23	0.063	-0.12	4.07	4.06	0.894	0.01	4.12	4.08	0.564	0.04
For some big companies, accessibility is not optional but a key task.	S15		4.10	4.07	4.14	0.152	0.07	4.15	4.04	0.151	0.11	4.04	4.09	0.854	-0.05
Accessibility has lack of demand in the industry.	S16		4.05	4.15	3.88	**<0.001**	0.29	3.92	4.22	**<0.001**	-0.30	4.21	3.86	**0.002**	0.35
Accessibility design lacks financial and organizational support.	S17		4.02	4.12	4.02	0.107	0.10	4.09	4.00	0.069	0.09	3.95	4.03	0.068	-0.08
Accessibility is context-dependent.	S18		4.01	4.06	4.14	1.000	-0.09	4.11	3.89	0.806	0.22	3.99	3.81	0.079	0.18
Small companies (teams) do not consider accessibility design.	S19		3.82	3.97	3.82	0.063	0.15	3.81	3.77	0.995	0.04	3.85	3.70	0.166	0.15
There are limited relevant guidelines for accessibility design.	S20		3.70	3.73	3.70	0.947	0.02	3.62	3.78	0.282	-0.16	3.83	3.62	0.108	0.21
Supporting accessibility minimizes legal risks.	S21		3.37	3.48	3.35	0.056	0.13	3.42	3.23	0.320	0.19	3.47	3.29	0.271	0.18
T5. Accessibility Requirement															
Accessibility requirements are an expensive addition for the system.	S22		4.10	4.11	4.00	0.233	0.11	4.15	4.14	0.100	0.251	3.98	4.25	0.057	-0.27
Accessibility requirements are more focused on FR design.	S23		4.07	3.99	4.13	0.121	-0.14	4.05	4.20	1.000	-0.15	4.05	3.96	1.000	0.11
Accessibility requirements are not clearly documented.	S24		4.04	3.99	3.81	0.148	0.18	4.12	4.26	0.280	-0.14	4.09	3.98	0.255	0.11
No resources from companies for accessibility requirements elicitation.	S25		4.04	4.15	4.01	0.186	0.14	3.91	4.24	**0.004**	-0.33	3.99	3.95	1.000	0.04
Not a core requirement for the project.	S26		4.01	4.13	3.79	**0.002**	0.34	3.98	4.11	0.013	-0.13	3.99	4.11	0.241	- 0.12
ML and AL can be applied to collect requirements regarding accessibility.	S27		3.78	3.71	3.75	1.000	-0.04	3.88	3.72	0.016	0.16	3.81	3.83	1.000	-0.12
Difficult to understand the technologies behind the requirements.	S28		3.57	3.38	3.52	0.186	-0.14	3.50	3.60	0.394	-0.10	3.76	3.64	0.435	0.12
T6. Accessibility Design															
Provide multiple views to address trade-offs between different types of user groups.	S29		4.02	3.99	4.05	1.000	-0.06	4.01	4.00	1.000	0.01	4.02	4.05	1.000	-0.03
Front-end design is often the major focus to make sure projects are accessible.	S30		3.93	4.01	3.90	**0.003**	0.11	4.01	3.94	0.077	0.07	3.88	3.84	0.540	0.04
ML and AI technologies could be applied to help accessibility design.	S31		3.89	3.79	3.81	0.235	-0.02	4.03	3.91	1.000	0.12	3.96	3.86	0.280	0.10
Front-end design is often the major responsibility to make sure projects are accessible.	S32		3.77	3.71	3.54	0.299	0.17	3.94	3.83	0.251	0.11	3.80	3.82	0.079	0.08
Detailed design is time-consuming and conducted in an iterative way.	S33		3.71	3.70	3.66	0.138	0.04	3.64	4.00	**<0.001**	-0.34	3.68	3.61	0.266	0.07
Hard to make accessibility design decisions (like design patterns or tactics).	S34		3.70	3.70	3.60	0.122	-0.10	3.80	3.74	0.388	0.16	3.52	3.85	**<0.001**	-0.33
Some accessibility standards are out of date.	S35		3.69	3.67	3.79	0.897	-0.12	3.64	3.83	0.674	-0.19	3.70	3.53	1.000	0.17
T7. Accessibility Testing															
A long list for accessibility testing (FRs and NFRs).	S36		4.19	4.21	4.13	0.387	0.08	4.23	4.18	1.000	-0.05	4.12	4.24	0.281	-0.12
It is hard to engage with end-users (get feedback).	S37		4.18	4.29	4.23	1.000	0.06	4.18	4.14	1.000	0.04	4.18	4.08	0.079	0.10
Automatic testing tools are useful.	S38		4.09	4.01	4.14	0.215	-0.13	4.21	4.02	0.871	0.19	4.02	4.17	**<0.001**	-0.15
T8. Accessibility Evaluation															
No single tool can determine if a site or project meets accessibility guidelines.	S39		4.13	4.11	4.25	1.000	-0.14	4.13	3.98	1.000	0.17	4.19	4.09	1.000	0.10
Human evaluation is always required.	S40		4.10	4.07	4.10	0.266	-0.03	4.21	4.10	0.498	0.11	4.06	4.08	0.719	-0.02
A lot of extra effort is needed for accessibility evaluation.	S41		4.06	4.12	4.06	0.719	0.06	4.10	4.02	0.665	0.08	4.01	4.06	0.651	-0.05
It is difficult to get feedback from end-users for accessibility evaluation.	S42		4.03	4.14	3.94	0.320	0.10	4.02	3.96	0.275	0.06	3.99	4.11	0.182	-0.12
A more comprehensive evaluation for accessibility is necessary.	S43		4.02	4.05	4.00	0.797	0.05	4.14	3.80	**<0.001**	0.34	4.09	4.05	0.546	0.04
Standards (e.g., WCAG) are helpful for accessibility evaluation.	S44		3.99	3.90	3.84	1.000	0.06	4.13	4.02	1.000	0.11	3.88	4.21	**<0.001**	-0.33

General consideration of accessibility. Regarding the general considerations of accessibility, we summarized seven statements in Table 1 from our interviews.

Our interviewees' opinions on understanding accessibility are consistent with the ISO 9241 definition [24], which emphasizes that the goals of accessibility design are maximizing the number of users and striving to increase the level of usability [**S5**], and 25% and 14% respondents *agree* and *strongly agree* with this statement. The concrete implementations of accessibility are slightly different based on different requirements of software applications. Accessibility can be provided by a combination of both software and hardware controlled by projects. Software examples include on-screen keyboards and voice and gesture recognition that replace physical keyboards and mice. One interview also mentioned that they have developed AI-driven features that reads text written on images to help sight-impaired users shop. Hardware examples include head-mounted pointing devices instead of mice and Braille output devices instead of a video displays. However, eight interviewees stated that accessibility now needs to be shifted targeting the **general** end-users and is important for every project [**S2**][**S4**] (mean scores of the both statements are over 4.00). They also claimed that accessibility could be integrated into current projects as a competitive functionality [**S1**][**S6**]. Around 27% (i.e., 174 out of 365) and 28% (i.e., 102 out of 365) of our online survey respondents agree and strongly agree with [**S1**]. Five interviewees mentioned that even though accessibility is a key task and timing consideration, in current projects it is not well designed, especially in relatively small-sized teams and companies and less developed countries [**S7**]. Around 21% (i.e., 78 out of 365) and 8% (i.e., 29 out of 365) of respondents agree and strongly agree with this statement.

> "*Our shopping app helps the visually impaired stay connected with the world. We have introduced Braille Button functionalities, which help users access the interface faster and more accurately. In addition, Smart Touch also supports an "ear touch" feature, which gives blind and visually impaired users a simple way to listen to text clearly and privately in public, without the need for headphones. It uses a shape-finding algorithm to sense when users hold the phone to their ear and automatically routes the sound output from the loudspeaker to the earpiece speaker. " [I7-Manager]*
>
> "*Not everyone is familiar with accessibility and not everyone understands why it is important. While the principles of accessibility become known by more people every day, the concept is not on everyone's radar yet.*" [*I1- Programmer*]
>
> "*Accessibility should be becoming competitive functionalities in our development, as it's now not only for disabilities. For example, voice assistant (voice search functionalities) of Apps that we have noticed is widely used.*" [*I5-Designer*]
>
> "*Accessibility is poor addressed by current products. We should have accessibility design elements for marketing; however, I was wondering whether actually it is used by people.*" [*I2-Programmer*]
>
> "*Current companies, especially small-size companies, their products address accessibility focusing on a lightweight interface design way, for example, navigation and visual hierarchy.*" [*I10-Designer and Manager*]

Characteristics of accessibility development and design in practice: We summarized two main characteristics of accessibility development and design in practice, as reported by our interviewees.

1. Intertwined and iterative with certain types of activities. 72% of the survey respondents agree or strongly agree with the statement that accessibility design should not be an independent activity but intertwined with many software artifacts and activities (e.g., from the software requirement elicitation to software testing activities). Considering the difficulties inherent in developing accessible software projects and investigating how accessibility fits into the development life cycle is vital (also the goal of this paper) [**S7**], we expanded this statement in RQ2 and provided detailed results in Section 3.2.

> "*In building an accessible website, accessibility needs to be considered at all key stages of the web development lifecycle. Issues on accessibility could creep in whole life cycle of a project.*" [*I10-Designer and Manager*]

2. Dynamic. Accessibility has a dynamic nature. A few years ago, accessibility was primarily targeted to people with disabilities, but now it is often designed to better support all users. Besides, during the development life cycle, accessibility also has a dynamic nature; a valid requirement about accessibility can turn into a set of essential functionalities [**S8**]. Dynamic diversity provides an important step towards developing more effective interfaces for a specific group of people and the more general problem of "Universal interface design" [20].

> "*If you are a developer, our goal is to provide you with some idea about potential problems and issues to keep in mind and to get you started on finding ways to solve the problems. The accessibility requirement is dynamic content, which has changed over the past several years. There are many ways to meet accessibility requirements too.*" [*I2-Programmer and Tester*]

Regarding the results of RQ1, we found that five out of nine statements (i.e., S1, S4, S6, S8, and S9) have significant differences depending on participants' experience and abilities. For example, practitioners who have direct accessibility work experience agree more that accessibility needs to be incorporated into all the projects; however, practitioners who have indirect accessibility relevant work experience perceive accessibility is time-consuming and not a core task (i.e., S6). We discuss the difference between the groups in detail in Section 3.3.

3.1.2 How do work characteristics impact accessibility development and design in practice. Almost all of our interviewees mentioned that work characteristics impact accessibility development and design in practice. In this section, we discuss what work characteristics practitioners are concerned most regarding accessibility. We summarized three statements under this topic (Table 1).

Skill variety: Interviewees identified two main differences regarding skills if they consider incorporating accessibility into their software systems. Firstly, interviewees noted that developing an application incorporating a range of accessibility presents distinct technical challenges [**S10**]. Secondly, interviewees suggested that a wider variety of skills is required for incorporating sufficient accessibility support features into a system. From the interface to architecture design, accessibility is more complicated than many think – numerous obstacles hinder more practical accessibility design and implementation. This can make accessibility design more challenging if a developer lacks such skills [**S10**]. Around 26% of online survey respondents agree and strongly agree this statement. Working practices with accessibility indicate the degree to which the projects would be successfully integrated with accessibility design. Nine interviewees mentioned that they had not learned accessibility knowledge systematically at their universities. ACM/ABET recommendations do not require universities to teach students accessibility relevant knowledge systematically. However, we would suggest introducing such a training to improve students' awareness on accessibility.

> "*Developing a system with accessibility, developers can write the code for a particular business requirements once they gave learning a programming language, but for the accessibility design, they should learn the standards and other specific skills.* " [*I9-Programmer and Tester*]

> "*It's unlikely that you'd be expected to know the ins and outs of all accessibility best practices if you aren't also playing a technical role in the project, but having a working knowledge is key.*" [*I12-Tester*]

"*In the context of accessibility, even small changes in the interface will hinder users to access the system. Development teams and developers should be able to handle the changes and make the accessibility truly usefully for users. This is not easy, we don't know what exactly users wants. It seems interface development takes the majority responsibilities for accessibility, which is not true and enough.* " [*I2-Programmer and Tester*]

Task identification: Interviewees reported a few differences when incorporating accessibility in terms of task identity. Two interviewees suggested one difference: it is harder to make an accurate plan for incorporating accessibility into projects [**S12**], and 34% and 17% of online survey respondents agree and strongly agree with this statement. Three interviewees noted that incorporating accessibility means they often have less control over their progress towards target completion. Once starting accessibility-related design and development, they might require (much) more effort in eliciting requirements, consulting, or end-user testing [**S12**], and 35% and 15% of online survey respondents agree and strongly agree with this statement. Three interviewees proposed the second difference: it is challenging to guarantee achievement of other software quality attribute tasks, such as performance [**S12**].

"*Companies are failing to consider accessibility as an on-going initiative. If you don't have the tools and processes in place for ensuring accessibility as you continuously improve your digital experience, accessibility issues will persist, forcing you to backtrack when push comes to shove. This retroactive approach impedes digital innovation, increases development costs and extends delivery timelines.*" [*I5-Designer*]

Interaction outside the organization: Interviewees reported that incorporating accessibility into a system means they have more challenges when communicating with their clients. Development teams are not often adequately trained on appropriate techniques required to address accessibility in their roles [**S11**]. Around 30% of the online respondents agree and strongly agree that interacting with outside stakeholders regarding accessibility is difficult.

"*Integrating accessibility into every aspect of your system is not very hard, but it takes commitment, constant communication, and a sound strategy. In this case, you need to consult experts and communicate with you clients.*" [*I7-Manager*]

3.1.3 How do organizational factors impact accessibility design and development in practice? Organizational factors include developmental, technological, business, operational, and social factors that impact a software system in many ways [6] [28]. We discussed with interviewees the organizational factors that they found to have an impact on accessibility. Company value, culture, and project leadership are factors interviewees mentioned that strongly impact accessibility development and design in practice, and nine statements are summarized on this topic ([**S12**] - [**S21**] in Table 1). The results of the online survey also confirmed that organizational factors will impact accessibility development and design (i.e., six out of nine statements are scored over 4).

Ten interviewees identified that accessibility development and design is context-dependent [**S18**]. 34% of the survey respondents identified this statement as important (i.e., agree and strongly agree). For example, the same requirement could be implemented in one project, but it is difficult to be implemented in another project because of context changes (e.g., Web App development v.s. Mobile App development). 12 interviewees noted that big companies, such as Microsoft and IBM, have more resources and experience to focus on accessibility development and design, and those companies put accessibility as a high priority [**S14**][**S15**][**S21**]. Five interviewees mentioned that some companies consider accessibility as a vital factor to extend their target market [**S13**]. In contrast, small companies focus more on functional requirements implementation and often neglect accessibility or other quality attributes [**S19**]. Previous studies also confirmed that large and well-known software projects have better accessibility design than small companies' products [2]. Various practical working constraints of practitioners and projects, such as tight time-frames, lack of tools and design guidelines, lack of knowledge and experience, lack of support, and limited budgets, will impact the incorporation of accessibility success [**S16**][**S19**][**S20**].

> "*Supports from companies are vital for accessibility design, unfortunately, sometimes, our company does not have much focus and budgets to support such functionalities.*" [*I3-Designer*]

3.2 RQ2: How does addressing accessibility needs fit into the software development life cycle?

To answer RQ2, we investigated how accessibility fits into four stages and what challenges practitioners have in each i.e., software requirement elicitation stage (Section 3.2.1), software design stage (Section 3.2.2), software testing stage (Section 3.2.3), and evaluation stage (Section 3.2.4).

3.2.1 Software requirements for accessibility. **Effort towards accessibility requirements collection**. We summarized seven statements (see Table 1) on this topic. The main challenges reported by interviewees included the time and effort needed for accessibility-related issue collection, challenges of aligning with diverse software end-user characteristics, and involving the right set of people at the start of a project. Nearly every interviewee mentioned that incorporating accessibility requirements into a system requires significant effort [**S22**]. Interviewees also mentioned that accessibility requirements are more uncertain than detailed functional descriptions. Another accessibility requirement challenge mentioned by three interviewees is that accessibility requirements are usually documented ambiguously [**S24**]. 34% of our survey respondents agree and strongly agree with this statement, and 12 survey comments confirmed that accessibility requirement elicitation is hard due to the vague requests from the clients or vague documentation.

According to our interviewees, new techniques for eliciting accessibility requirements have been developed. Three interviewees mentioned that Machine Learning (ML) and AI techniques have been adopted in their companies to assist with accessibility requirement elicitation [**S27**]. For example, one interviewee mentioned they used mouse tracking techniques to elicit requirements, i.e., they will record where the users give up clicking and exit the application, and those places could be difficult for users to proceed. 23% of the survey respondents agree and strongly agree with this statement. Two interviewees mentioned that ontology was used to model the knowledge of accessibility guidelines. However, they noted that some companies are struggling to identify accessibility requirements due to tight marketing and release schedules, lack of resources, or that they are seen as not a core task [**S25**]**S26**][**S28**]. The survey results also confirmed those three statements, which scored 4.04, 4.04, and 3.78, respectively.

> "*There's a problem in the design world about accessibility. Accessibility requirements are often not documented clearly, consistently, or in a way for professionals to easily follow and implement.*" [*I5-Designer*]

> "*By beginning to be more mindful about documenting accessibility requirements, it can truly change the culture of design.*" [*I7-Manager*]

> "*The time allotted on computers could be stressful to the body. Those with physical impairments have more difficulty utilizing the web compared to normal users. People with mobility issues, for example, may find it difficult to move the mouse to the desired target audience. Collecting requirements, designing, therefore, for low physical efforts is important to take into account.*" [*I7-Manager*]

Strategies to address accessibility requirements. Three interviewees mentioned that analysis activities (e.g., user and task analysis) play an essential role. The interviewees stated that some developers simply thought of software requirements for accessibility as being only front-end design [**S23**]. However, many accessibility requirements are far more complicated than that, which practitioners should be more aware of. Around 34% of the responses tend to agree and strongly agree, and this statement scored 4.10 from the online survey.

> "*As a project manager, our ability and willingness to appropriately plan for accessibility, engage the right people at the right time, communicate the need and value of accessibility, and hold teams accountable for meeting accessibility requirements will ultimately play a major role in delivering accessible websites and apps.*" [*I10-Designer and Manager*]

> "*One major issue is that we and our clients have faced to ensure accessibility throughout the process. Some of the key integration points for accessibility in Agile development include iterative testing, design and architecture, and incorporation of accessibility in the development team's"definition of done." However, as with development for any system, ensuring accessibility begins with requirements..*" [*I12- Designer and Programmer*]

3.2.2 Software design and implementation for accessibility. **Architectural design**. Interviewees stated that there are differences in design and implementation if the software incorporates accessibility. Firstly, interviewees mentioned that the high-level architecture design for incorporating accessibility into a system typically contains richer hardware design, feature engineering, and model design [**S34**] [5]. However, in many situations, a lot of companies will not consider putting additional effort into such design. As a consequence of less-design-planning, interviewees reported very little up-front thought into the software architecture to enable developers to achieve suitable accessibility solutions [7].

> "*Companies have embraced accessibility as a component of their design, development and quality testing processes, such as American Airlines and Chase Bank, continue to win awards for innovation and customer experience, while providing digital access to all users, regardless of abilities. Integrating accessibility principles into every aspect of your digital operations is not hard, but takes commitment, constant communication and a sound strategy.*" [*I10-Manager*]

Secondly, accessibility needs to be separately designed for low coupling and re-use [**S34**]. For instance, one interviewee mentioned that "*I divide the accessibility functionalities into several steps for re-use and also may use existing libraries for each step.*" Thirdly, refactoring is a downside when needing to incorporate accessibility into a system. As many systems do not consider accessibility at the beginning, but then want or need to incorporate accessibility into the system later [**S32**], many problems present. Similarly, such a concern was observed in a study by Horton and Sloan [23]. The authors point out that accessibility audits are typically performed during quality assurance phases, and user acceptance testing and solutions for the issues identified usually take place in code.

> "*When companies consider accessibility at this stage, it sets a strong foundation for their overall strategy. If website or app updates are not designed with inclusive design principles in mind, it will undoubtedly create more obstacles later on. The following actions will set you up for a successful approach to accessibility.*" [*I5-Designer*]

Domain-dependent design: Five interviewees mentioned that accessibility development and design based on platforms and domains [**S29**]. We found accessibility is most likely incorporated into domains such as banking, education, government, voice recognition application, transportation system, and web services domains (see Fig 2 of Appendix).

Regarding software platforms, the majority of survey respondents focused on developing Web and Mobile applications. We also noted a trend in adopting other process-oriented frameworks to approach accessibility in software engineering. Despite the substantial uptake of mobile applications, software practitioners concerning proposals for supporting accessibility in mobile applications are still more limited in comparison to Web accessibility [**S33**]. However, it is noteworthy that many software practitioners mentioned that techniques to support the development of accessible mobile applications had started to receive recent attention from the research community.

Models and standards: Related to other kinds of support provided by the architectures, there were recurrent subjects like seeking the improvement of usability or information fusion from different resources [**S33**]. In addition, there is a trend in interfaces moving towards more multi-model interfaces and systems. Regarding the communication between components/layers and other systems, we would like to highlight that most systems used typical data structures encapsulated in files, such as microservices as architecture patterns.

Regarding the use of accessibility standards, almost every interviewee mentioned that they applied and followed a set of standards. We noted the predominance of WCAG standard, which has been established as the primary reference concerning accessibility guidelines and design issues for web applications. However, five interviewees mentioned that they have worked as web practitioners for over a decade, and they found some of the standards (e.g., WCAG and Section 508[5]) are hard to follow for practitioners who have less accessibility work experience. [**S35**].

Challenges for accessibility implementation. Implementation of accessibility-supporting solutions for software was reported to be an iterative process [**S33**]. There are Functional Requirement (FR) and Non-functional Requirement (NFR) categories under the first layer. There are interface, control, and entity components in the third layer under the Type-F requirements. The change and addition of requirements are tracked in this implementation. Six interviewees mentioned that accessibility often falls upon an organization's practitioners to take it upon themselves to decide whether or not to implement accessibility and how to do it. However, business pressures were reported by some respondents to motivate practitioners to achieve short-term goals rather than the longer-term or indirectly profitable work of accessibility [**S30**] [**S34**].

> *Refactoring is used for improving external attributes of, like, Web applications. Those refactoring consist of small navigation or interface transformations, like user interaction and presentation of content.* [*I6-Programmer and Tester*]

> "*Developers typically have the hardest time with accessibility. They need specific expertise to know how to resolve accessibility issues within code that is resulting in inaccessible page elements. This requires extensive accessibility training on the techniques for following the WCAG 2.0 AA or leveraging accessibility experts for guidance on coding options.*" [*I8-Tester*]

> "*When ARIA is implemented correctly, it can be a great way to make web content accessible to users with disabilities. On the other hand, if it is implemented badly, it can create further accessibility issues to users with disabilities.*" [*I6- Programmer and Tester*]

3.2.3 Software testing and quality assurance for accessibility. **Quality assurance deterrents**. We summarized three statements from the interviews of this topic, and all statements scored by the survey respondents are over 4.00. Around 61% of the respondents *agree* and *strongly agree* quality assurance is a vital step for accessibility development and design. As we mentioned earlier, accessibility is a quality that makes sure that end-users can access the system successfully. There are also other quality attributes that overlap with accessibility, such as security, privacy, and usability. Seven interviewees mentioned that other quality attributes (e.g., security and performance) are difficult to guarantee when needing to incorporate accessibility into a system [**S36**].

> "*We have been developing the online shopping for visual impaired people, and it is going well. They can shop using our voice assistant functionalities, and it will read everything shown in*

[5]Section 508 requires federal agencies to make their ICT such as technology, online training and websites accessible for everyone. https://www.epa.gov/accessibility/what-section-508

the screen. However, how to guarantee security, performance, and privacy is a headache. " [*I6-Programmer and Tester*]

"*Many developers are using free online testing tools, but find themselves putting in double the work and still struggling to maintain compliance. These tools are inconvenient and often provide false positives, which is why they ultimately are not worth it. For agile development cycles, developers need sufficient tools and streamlined technical support to efficiently maintain accessibility.*" [*I10-Designer and Manager*]

Seven interviewees also mentioned that accessibility testing is vital for different platforms, such as web and mobile app testing. There are some challenges including that developers are unlikely to write dedicated accessibility test suites, and existing test suites tend to be weak in finding many accessibility problems. A set of tools that participants mentioned can significantly reduce the manual efforts of accessibility testing [**S38**]. These included Anywhere [6] (a screen reader tool) and Hera [7] (used to check the software application style). Seven interviewees mentioned that standards and checklists are helpful for accessibility testing. However, the current checklists and standards need to be updated for a more comprehensive understanding and to address new interaction technologies [51]. Four interviewees mentioned the importance of manual accessibility testing, such as hiring disability end-users and getting feedback [**S35**].

"*Accessibility testing is no different, to me, from the other types of testing required for quality assurance — testing a digital experience to check it is fit for a certain purpose and that the developers followed the proper guidelines and requirements.*" [*I11-Manager*]

"*In terms of tools, your QA team should have access to the same accessibility testing tools as your developers. It's important for your QA team to easily show your developers where and what issues were identified, so creating an avenue of communication between these two teams is essential for maintaining delivery timelines.*" [*I2-Programmer and Tester*]

3.2.4 Software evaluation for accessibility. **Evaluation across stages**. We summarized six statements from the interviews of this topic (see Table 1). Five out of six statements are scored over 4. Almost all interviewees mentioned that evaluation is a **key** step for achieving required software accessibility. They gave a range of evaluation methods that assist developers in creating interactive electronic products, services, and environments that are both easy and pleasant to use by the target audience. It encompasses technical aspects such as tools and standards and guidelines [**S39**][**S44**], as well as non-technical aspects such as involvement of end-users during the evaluation process [**S40**][**S41**][**S42**].

Tool usage for accessibility evaluation. Five interviewees mentioned that automated and semi-automated accessibility evaluation tools are essential to streamline the process of accessibility assessment, and ultimately ensure that software projects, contents, and services meet accessibility requirements. Different evaluation tools may better fit different needs and concerns. 34% of respondents (agree and strongly agree) indicated that they used a set of tools and checklists to evaluate accessibility. However, four interviewees mentioned that no single tool could determine if a project meets accessibility guidelines [**S39**]. Thus, an development teams typically uses accessibility evaluation tools to deal with requirements (e.g., specification and testing).

"*Some designers need to meet U.S. Section 508 standards chose to provide alternative modes of operation and information retrieval. However, in some cases where the standard was technically met by providing an alternative, the products were awkward to use or were totally unusable by*

[6] http://webinsight.cs.washington.edu/wa/content.php
[7] http://www.hera.flexit.fr/

some people with disabilities. These cases illustrate the importance of going beyond just meeting a minimum accessibility standard without sufficient evaluation." [*I14-Programmer*]

3.3 Gaps differ across demographic groups

We analyzed the statements across three pairs of demographic groups in Table 1, i.e., **Direct** and **Indirect** groups, **Big** and **Small** groups, and **Web** and **Mobile** App development groups. As we presented in Section 2.2.2, we used P-value to indicate whether the differences in the agreement for each statement, and the *Absolute Effect Size* is used to indicates the magnitude of difference between groups. In this section, we discussed more details about the difference regarding perceiving accessibility statements across the groups.

Direct v.s. Indirect accessibility-related work experience groups. There are six statements with statistically significant differences between these two groups. It is worth noting that the Direct group is more likely to agree with these four statements more, i.e., [**S1**][**S8**][**S9**][**S26**]. For example, they agree that accessibility needs to be addressed in all software projects and agree with the characteristics of accessibility given by interviewees. On the other hand, Indirect group agree more that their projects lack of demands on accessibility and time-consuming, and they also agree that accessibility is limited to front-facing design in a light-weight way (i.e., [**S16**][**S30**]).

Big-size v.s. Small-size groups. There are four statements with statistically significant differences between these two groups. The small size group agrees more that they lack of resources and support needed to achieve accessible software products (i.e., [**S16**][**S25**][**S33**]). On the other hand, the big-size group agrees that a more comprehensive evaluation for accessibility is needed (i.e., [**S43**]).

Web v.s. Mobile development groups. Seven statements are with statistically significantly different answers between these two groups. It is worth noting that Mobile App practitioners are more likely to agree that the standards for accessibility in Mobile App development are unclear (i.e., [**S42**]). In addition, Mobile App practitioners agree more that accessibility development and design is time-consuming and not a core task (i.e., [**S6**][**S16**]). In contrast, Web App practitioners are more focused on the accessibility design and testing tasks (i.e., [**S34**][**S38**]).

Overall, the results of our survey confirm some differences in interviewees' claims. Being aware on the gaps (i.e., differences) between groups would help practitioners incorporate accessibility properly into their projects in practice.

4 DISCUSSION

The goal of this study was to understand how accessibility is currently incorporated into software development and how practitioners perceive accessibility issues and challenges in their projects. We discuss below our key findings and their possible implications for researchers and practitioners.

Accessibility awareness and challenges of accessibility development. Our participants reported that accessibility closely fits into different stages of the software life cycle, and developers need to be aware of them when they make design decisions. Despite the vast majority of our online survey respondents acknowledging the importance of accessibility, 45% of our survey respondents indicated that their projects are suffering from issues that effect accessibility development and design, such as inadequate resources and experts [**S17**][**S19**][**S28**]. The frequent occurrence of accessibility issues reported by our participants in their software development may result in software refactoring requiring more efforts and released code with severe accessibility issues. As such, we advocate that accessibility could be treated as an Functional Requirement essential to software projects. Such "accessibility-driven design" might drive innovation in products for all users, and it can ensure more accessible software for physically and mentally challenged users as well. This could motivate practitioners to spend extra efforts on accessibility management, design, and development. Our results also suggest that accessibility needs to be included in all phases of software development but achieving good accessibility solutions takes a lot of effort and time. Practitioners suggest that software development could be integrated with accessibility design elements, which can make the software to a wide range of average users, especially users with some disabilities. Future research may focus on better standardizing the process of incorporating accessibility in software development.

In addition, from the results of our survey, a considerable number of participants do not have any accessibility work experience. In addition, two interviewees mentioned that in their companies, accessibility is usually considered only at a late stage of development or during system refactoring and maintenance. Given the increasing demand for more accessible software for all users [21, 39, 47], this suggests that practitioners need to put more emphasis on accessibility understanding and the ability to appreciate trade-offs.

The benefit-cost was repeatedly mentioned by many of our interviewees. Some companies consider accessibility would be an extra cost for them to incorporate into development. However, they often find that they have to refactor/modify their systems to add accessibility functionalities afterward during the system evolution, ending up costing them more time and resources. As above, we encourage developers to put accessibility as a first-class, upfront requirement from the very beginning of software development. We advocate that accessibility can also be considered as a critical feature not only for the specifically challenged groups of people, but potentially all users who want different ways to interact and use software to suit their own preferences.

Accessibility-driven development. One of the main goals in this study is to investigate to what extent accessibility is integrated into existing software projects. Almost every interviewee mentioned that accessibility in today's projects has a second meaning to traditional support for challenged users – "*to make the projects more easy to be used*" [**S2**][**S5**][**S6**]. Some functionalities were developed to help those with disabilities, for example, voice assistant functionalities and a more natural interface. However, more and more users expect these to be available in many scenarios. This indicates that participants need to think more about addressing "accessibility" in their software now than in the past.

Education for accessibility knowledge. Our work highlights the accessibility relevant skills and knowledge that are perceived important by participants **[S10]**. As we discussed above, industry demands for software developers with knowledge of accessibility has increased substantially in recent years. However, respondents said that there is little knowledge about the prevalence of higher education teaching accessibility relevant knowledge [38]. None of our interviewees said that they had learned software accessibility design and implementation approaches at universities. Even though ABET/ACM do not suggest universities to teach or train students accessibility systematically, software has to be designed to work with new accessibility-supporting technologies.

Given this finding, universities would seem to need to increase SE students' knowledge of how accessibility fits within the development life cycle. In addition, as we presented in Section 3, accessibility-driven design could help drive software innovation in terms of helping all users use software products more efficiently and smartly. This also requires encouraging students to focus more on accessibility design. For example, general accessibility knowledge and standards. Educators should emphasize on ability to incorporate accessibility into systems, document well [**S24**], and test and evaluate efficiently [**S36**].

General recommendations for accessibility development and design in practice. Based on our survey and interview results, below we provide a set of general recommendations for incorporating accessibility into the projects in practice from two aspects (see Table 2).

1. Organizational factors and People - RQ1. Before making any decisions about "Accessibility": stakeholders (e.g., designers, architects, developers, testers, and clients) in a project should

Table 2. Challenges and recommendations for incorporating accessibility in practice.

Challenges	Non-technical	Technical	Recommendations
Organizational			
Lack of executive sponsorship.	✓		Strong executive support.
Lack of management commitment.	✓		Committed sponsor or leader.
Organizational culture factors.	✓		Cooperative organizational culture.
Organizational size too small.	✓		Face-to-Face communication with customers and collocation of the teams.
People			
Lack of necessary skills and knowledge.	✓		Include team members with high competence and expertise.
Lack of project management competence.	✓		Include team members with motivation.
Lack of team work.	✓		Knowledge in the accessibility development process.
Customer relationship.	✓		Self-organizing teamwork.
Process			
Unclear requirements.	✓	✓	Accessibility-oriented requirement identification.
Unclear project planning.	✓	✓	Accessibility oriented interactive development process.
Unclear project scope.	✓	✓	Commutation with end-users.
Lack of customer role.	✓		Regular working schedule.
Lack of customer presence.	✓		Increase customer presence.
Unclear standard and principle.	✓	✓	Unclear standard and principle.
Inappropriateness of testing suits.		✓	Include appropriateness of testing suits.
Include appropriateness of evaluation process.		✓	Introduce appropriateness of evaluation process.
Practice			
Lack of complete set of accessibility practices.		✓	Rigorous refactoring activities.
Inappropriateness of technology and tools.		✓	Well-designed documentation.
			Delivering most important feature first.
			Correct integration testing.
			Appropriate technical training to team.

reach a consensus on accessibility development and design. Stakeholders need to know what the exact accessibility requirements are and how to address these accessibility-related requirements. Another recommendation is to put accessibility requirements as a high priority to address. Furthermore, when making decisions about accessibility, some design decisions may be invalid in the first place, and some trade-offs might exist between design decisions. This moderates efforts since not every accessibility requirement might be met. Lastly, accessibility requirements need to be well-documented.

2. Process and Practices - RQ2. For better meeting accessibility needs in the software life cycle, we listed several challenges and recommendations in Table 2. For example, maintaining key accessibility requirements is recommended, and this needs better support in eliciting and modeling these accessibility requirements in software modeling notations. Our second recommendation is to transform accessibility requirements to other activities and artifacts(e.g., design decisions and source code). The relationship between accessibility and other types of activities can be unidirectional or bi-directional. In addition, practitioners need to be aware of the relationships between accessibility requirements and target users. Accessibility needs to be addressed right from the early phases of software developments (e.g., requirement engineering or software architecture design). Furthermore, having an iterative process to fit accessibility into the whole life cycle is vital. Using a dedicated approach and supportive tools (e.g., the accessibility framework) for accessibility, for example, considering machine learning and AI-based techniques to support accessibility design and evaluation [1, 19]. Lastly, we recommend generalizing accessibility from dedicated design to a universal design. We this we mean all software should consider diverse end user accessibility needs, including all users who may have the same characteristics but differing software usage preferences.

5 THREATS TO VALIDITY

We use the guidelines in [49] to discuss key threats to the validity in this work.

Construct validity reflects to what extent the research questions and methodology are appropriately used in a study. A threat in Stage 1 (i.e., interviews) is whether or not our interviewees are representative. To reduce this threat, we invited 15 practitioners who come from three companies in two countries. In addition, the projects they have worked on cover a wide range of domains. However, they may not be representative of all practitioners. To mitigate this potential bias, we have carefully chosen questions and topics for the interviews and performed a survey that involves a large population of practitioners from various companies around the world. Our survey respondents completed the survey based on their opinions and perception. It is possible that they conflate the skills that are very important and the skills that are very relevant to their projects or industrial contexts. To mitigate this threat, we have tried to survey a large number of practitioners. A total of 365 practitioners from various companies in 28 countries across five continents participated.

Internal validity focuses on factors that may influence the validity of the results. The main threat in our study is whether the data we analyzed and coded can answer our research questions. It is possible that some of our survey respondents did not understand some of the statements well. To reduce this threat to validity, we provide an "I Don't Know" option in our survey, and we find that the number of respondents who choose this option to be small (i.e., 5.3%). We also translated our survey to Chinese to ensure that respondents from China can understand our survey well. It is also possible that we draw the wrong conclusions about respondents' perceptions from their comments. To minimize this threat, we read transcripts many times and checked the survey results and the corresponding comments several times.

The selection of statements produced at the end of the interview may not be comprehensive and may be biased to the background of experts – who may not be able to articulate their own opinions. To mitigate this bias, we have taken the following steps:

- Aside from asking direct questions of what opinions about accessibility they deem important, we also asked them to discuss topics that they not explicitly mentioned. The topics were selected from accessibility text books and online resources; they include concepts, comprehension, programming language, requirements, design implementation, testing, tool usage, standards, and others.
- We have performed a survey to check whether the interviewee' opinions are perceived to be correct by a large number of practitioners.

External validity concerns the generality of our study results to other settings. Our results and summaries are based on the interview and survey participants' opinions instead of a rigorous analysis of the claims participants make. It is possible that the opinions in terms of accessibility differ from participant and participant. To improve the generalizability of our results and summaries, we have interviewed 15 respondents from 3 companies and surveyed 365 respondents from 26 countries working for various companies (including Google, Huawei, Alibaba, Microsoft, Hengtian, and other various small to large companies) or contributing to open source projects hosted on GitHub. Still, our findings may not generalize or represent the perception of all software engineers. For example, most of the respondents are from Asia and the U.S. It would be interesting to perform another study to investigate more software engineers to perceive accessibility in the future.

Another threat related to the completeness of our 42 statements about accessibility in general software projects. In this paper, we finalized these topics and statements based on the open-ended interviews of 15 interviewees. At the end of our survey, we also asked the respondents to provide additional opinions about accessibility. Among the 365 responses, 176 respondents provided comments for additional opinions. We also manually analyze these additional considerations of accessibility. Next, we applied closed card sorting to categorize the comments, i.e., we tried to categorize them into the eight topics, and we left the comments which belong to none of these topics. We notice most of the comments provided a supplementary explanation to our 42 statements. Moreover, we only found two additional statements that are not covered by our survey. This threat could be removed by including their two statements and inviting more developers to participants.

6 RELATED WORK

We review a range of key related works from two perspectives: software design and development for accessibility in practice (see Section 6.1). We then review key related works investigating practitioners' opinions for specific topics in the software engineering field (see Section 6.2).

6.1 Software design and development for accessibility

The accessible design ensures content is faithfully rendered and can be interacted across a broad spectrum of devices, platforms, assisting technologies, and operating systems by the widest possible range of end-users [18, 36]. In physical environments, everyone benefits from lower curbs, automatic door openers, ramps, and other features provided for disability access. For example, a part of multimedia that allows voice narration with captioning or transcription is inaccessible to students with hearing issues. There have been some attempts to investigate accessibility in software development. The closest work to ours is Paive *et al.*'s study [35]. The authors conducted a literature review on accessibility and how it fits into software engineering processes. They investigated guidelines, techniques, and methods that have been presented in the literature. The authors also provided designers and developers with updated methods that contribute to process enrichment, valuing accessibility, and shows the gaps and challenges that deserve to be investigated. This work motivated us to investigate how accessibility fits into the development life cycle. However, we aim to gain *participants*' feedback and opinions about accessibility design in practice in this work. We also identify key gaps across demographic groups that highlight accessibility development and design challenges in specific circumstances.

Alshayban *et al.* conducted an empirical study aiming at understanding the accessibility of Android apps [2]. The authors reported a large-scale analysis on the prevalence of a wide variety of accessibility issues in over 1,000 Android apps across 33 different application categories, and 11 types of accessibility issues were identified, for example, TextContrast and SpeakableText. The authors also presented the findings of a survey involving 66 practitioners that reveal the current practices and challenges in Android apps regarding accessibility. The results of the survey show that developers are generally unaware of the accessibility principles and the existing analysis tools are not sufficiently sophisticated to be used. In our work, we interviewed 15 practitioners and got 365 survey responses about accessibility development and design in practice. However, we focus on Mobile app development and many other general software development scenarios in terms of accessibility. Bai *et al.* presented an evaluation of nine accessibility testing methods that fit in the agile software process, and during the evaluation, the authors have investigated different accessibility testing and discussed the benefit and cost of the software development process and recommended methods from the center and apply more testing method [4]. Nganji *et al.* proposed a disability-aware software engineering process model that considers the needs of people with disabilities, hence improving accessibility and usability of the designed system. Shinohara *et al.* [41] conducted a survey to investigate the prevalence of higher education teaching about accessibility and faculty's perceived barriers to teaching accessibility. The results show that teaching accessibility is prevalent but shallow among U.S faculty as broad. The authors also reported that the most critical barriers to clear and discipline-specific accessibility learning objectives are the lack of faculty knowledge about accessibility. Some other works focus on the human-centered computing aspect in terms of accessibility design; for example, Zhao *et al.* presented SeeingVR, which is a set of tools to enhance a VR application for people with low vision by providing visual and audio augmentations. A user can select, adjust, and combine different tools based on their performances. The authors evaluated their tool with 11 participants with low vision. The results show that SeeingVR enabled users to better enjoy VR and complete tasks more quickly and accurately. In addition, developers found the toolkit easy and convenient to use [53].

6.2 Studies on developer expertise

Prior studies explored developers' opinions about various software activities. For example, Xia *et al.* conducted an empirical study to understand coding proficiency, and the authors interviewed and surveyed software practitioners. The authors got 38 coding proficiency skills, which can be grouped into nine categories. The authors also highlighted 21 critical skills, along with the rationale given by participants. The finding could help practitioners being aware of acquired coding skills [50]. Murphy-Hill *et al.* presented a study with 14 interviewees and 364 survey responses for understanding substantial differences between video game development and other software development. The authors contributed a new empirical foundation on which to understand those differences [33]. Wan *et al.* performed a mixture of quality and quantitative studies with 14 interviews and 342 survey responses from 26 countries to elicit significant differences between the development of machine learning systems and the development of non-machine learning systems [48]. In our previous work [8], we investigated participants' opinions on release note production and usage in practice and what gaps exist between release note producers and users.

Similarly, in this work, we followed the same approach (i.e., a mixed-methods approach approach) to investigate accessibility development and design challenges, issues, and benefits from practitioners' opinions, which help development teams incorporate accessibility in practice by highlighting essential skills to acquire.

7 CONCLUSION

We investigated how accessibility fits into the development life cycle of existing software projects, and how practitioners perceive accessibility in their projects. We have found empirical evidence from participants' viewpoints highlighting the key obstacles in accessibility design and implementation that they currently face (see Table 1). This can help both researchers and practitioners better understand what challenges and benefits to incorporate accessibility into projects. We advocate that accessibility should be treated as a first class consideration throughout software development. Through our analysis, we grouped practitioner statements about accessibility issues into SE topics throughout the software development life cycle. Base on the results of interviews and the online survey, we outline a set of challenges of accessibility and recommendations on how to support practitioners to consider accessibility in software development and design (see Table 2). With the findings, we conjecture that accessibility still needs considerable efforts to be further explored from various perspectives (e.g., the end-users, participants with no working experience, app reviews [16, 44], and low-level software application design and coding). Our findings also highlight opportunities for researchers and participants to address the existing issues.

ACKNOWLEDGMENT

The authors would like to thank all the interview participants and survey respondents; without them, this work will never be accomplished.

This research was partially supported by the National Research Foundation of China (No.U20A20173) and the National Research Foundation, Singapore under its Industry Alignment Fund - Pre-positioning (IAF-PP) Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. Grundy is supported by Laureate Fellowship FL190100035.

REFERENCES

[1] Shadi Abou-Zahra, Judy Brewer, and Michael Cooper. 2018. Artificial Intelligence (AI) for Web Accessibility: Is Conformance Evaluation a Way Forward?. In *Proceedings of the 15th International Web for All Conference*. 1–4.

[2] Abdulaziz Alshayban, Iftekhar Ahmed, and Sam Malek. 2020. Accessibility issues in Android apps: state of affairs, sentiments, and ways forward. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*. 1323–1334.

[3] Dominique Archambault, Thomas Gaudy, Klaus Miesenberger, Stéphane Natkin, and Rolland Ossmann. 2008. Towards generalised accessibility of computer games. In *International Conference on Technologies for E-Learning and Digital Entertainment*. Springer, 518–527.

[4] Aleksander Bai, Heidi Camilla Mork, and Viktoria Stray. 2017. A cost-benefit analysis of accessibility testing in agile software development: results from a multiple case study. *International Journal on Advances in Software* 10, 1&2 (2017), 96–107.

[5] Len Bass, Paul Clements, and Rick Kazman. 2003. *Software architecture in practice*. Addison-Wesley Professional.

[6] Tingting Bi, Peng Liang, and Antony Tang. 2018. Architecture Patterns, Quality Attributes, and Design Contexts: How Developers Design with Them. In *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*. IEEE, 49–58.

[7] Tingting Bi, Peng Liang, Antony Tang, and Chen Yang. 2018. A systematic mapping study on text analysis techniques in software architecture. *Journal of Systems and Software* 144 (2018), 533–558.

[8] T. Bi, X. Xia, D. Lo, J. Grundy, and T. Zimmermann. 2020. An Empirical Study of Release Note Production and Usage in Practice. *IEEE Transactions on Software Engineering* (2020), 1–1. https://doi.org/10.1109/TSE.2020.3038881

[9] Talita Britto and E Pizzolato. 2016. Towards web accessibility guidelines of interaction and interface design for people with autism spectrum disorder. In *ACHI 2016: the ninth international conference on advances in computer-human interactions*. 1–7.

[10] Peter Brunet, Barry Alan Feigenbaum, Kip Harris, Catherine Laws, R Schwerdtfeger, and Lawrence Weiss. 2005. Accessibility requirements for systems design to accommodate users with vision impairments. *IBM Systems Journal* 44, 3 (2005), 445–466.

[11] Sheryl Burgstahler. 2011. Universal design: Implications for computing education. *ACM Transactions on Computing Education (TOCE)* 11, 3 (2011), 1–17.

[12] Victoria Clarke, Virginia Braun, and Nikki Hayfield. 2015. Thematic analysis. *Qualitative psychology: A practical guide to research methods* (2015), 222–248.

[13] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, 1 (1960), 37–46.

[14] Nelly Condori-Fernandez and Patricia Lago. 2018. Characterizing the contribution of quality requirements to software sustainability. *Journal of systems and software* 137 (2018), 289–305.

[15] Carey Curtis. 2008. Planning for sustainable accessibility: The implementation challenge. *Transport Policy* 15, 2 (2008), 104–112.

[16] Andrea Di Sorbo, Sebastiano Panichella, Carol V Alexandru, Junji Shimagaki, Corrado A Visaggio, Gerardo Canfora, and Harald C Gall. 2016. What would users change in my app? summarizing app reviews for recommending software changes. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. 499–510.

[17] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. 2008. Selecting empirical methods for software engineering research. In *Guide to advanced empirical software engineering*. Springer, 285–311.

[18] Robert F Erlandson. 2007. *Universal and accessible design for products, services, and processes*. CRC Press.

[19] Sergio Firmenich, Alejandra Garrido, Fabio Paternò, and Gustavo Rossi. 2019. User interface adaptation for accessibility. In *Web Accessibility*. Springer, 547–568.

[20] Peter Gregor and Alan F Newell. 2001. Designing for dynamic diversity: making accessible interfaces for older people. In *Proceedings of the 2001 EC/NSF workshop on Universal accessibility of ubiquitous computing: providing for the elderly*. 90–92.

[21] John Grundy. 2020. Human-centric Software Engineering for Next Generation Cloud-and Edge-based Smart Living Applications. In *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*. IEEE, 1–10.

[22] Stefan Henß, Martin Monperrus, and Mira Mezini. 2012. Semi-automatically extracting FAQs to improve accessibility of software development knowledge. In *2012 34th International Conference on Software Engineering (ICSE)*. IEEE, 793–803.

[23] S Horton and D Sloan. 2014. Accessibility in practice: a process-driven approach to accessibility. In *Inclusive Designing*. Springer, 105–115.

[24] Timo Jokela, Netta Iivari, Juha Matero, and Minna Karukka. 2003. The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11. In *Proceedings of the Latin American conference on Human-computer interaction*. 53–60.

[25] Margaret King-Sears. 2009. Universal design for learning: Technology and pedagogy. *Learning Disability Quarterly* 32, 4 (2009), 199–201.

[26] Barbara A Kitchenham and Shari L Pfleeger. 2008. Personal opinion surveys. In *Guide to advanced empirical software engineering*. Springer, 63–92.

[27] Pavneet Singh Kochhar, Xin Xia, David Lo, and Shanping Li. 2016. Practitioners' expectations on automated fault localization. In *Proceedings of the 25th International Symposium on Software Testing and Analysis*. 165–176.

[28] Mathieu Lavallée and Pierre N Robillard. 2015. Why good developers write bad code: An observational case study of the impacts of organizational factors on software quality. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 1. IEEE, 677–687.

[29] Eleanor Lisney, Jonathan P Bowen, Kirsten Hearn, and Maria Zedda. 2013. Museums and technology: Being inclusive helps accessibility for all. *Curator: The Museum Journal* 56, 3 (2013), 353–361.

[30] Erika McCallister, Tim Grance, and Karen Scarfone. 2010. *Guide to protecting the confidentiality of personally identifiable information*. NIST Special Publication 800-122. National Institute of Standards and Technology.

[31] Nathan W Moon, Paul MA Baker, and Kenneth Goughnour. 2019. Designing wearable technologies for users with disabilities: Accessibility, usability, and connectivity factors. *Journal of Rehabilitation and Assistive Technologies Engineering* 6 (2019), 2055668319862137.

[32] Palmer Morrel-Samuels. 2002. Getting the truth into workplace surveys. *Harvard business review* 80, 2 (2002), 111–118.

[33] Emerson Murphy-Hill, Thomas Zimmermann, and Nachiappan Nagappan. 2014. Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development?. In *Proceedings of the 36th International Conference on Software Engineering*. 1–11.

[34] Mike Paciello. 2000. *Web accessibility for people with disabilities*. CRC Press.

[35] Débora Maria Barroso Paiva, André Pimenta Freire, and Renata Pontin de Mattos Fortes. 2020. Accessibility and Software Engineering Processes: A Systematic Literature Review. *Journal of Systems and Software* (2020), 110819.

[36] Hans Persson, Henrik Åhman, Alexander Arvei Yngling, and Jan Gulliksen. 2015. Universal design, inclusive design, accessible design, design for all: different concepts—one goal? On the concept of accessibility—historical, methodological and philosophical aspects. *Universal Access in the Information Society* 14, 4 (2015), 505–526.

[37] Woori Roh and Seok-Won Lee. 2017. An ontological approach to predict trade-offs between security and usability for mobile application requirements engineering. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*. IEEE, 69–75.

[38] Mary-Luz Sánchez-Gordón and Lourdes Moreno. 2014. Toward an integration of Web accessibility into testing processes. *Procedia Computer Science* 27 (2014), 281–291.

[39] Sandra Sanchez-Gordon and Sergio Luján-Mora. 2017. A method for accessibility testing of web applications in agile environments. In *Proceedings of the 7th World Congress for Software Quality (WCSQ). Em processo de publicação.(citado na página 13, 15, 85)*.

[40] Andrew Sears and Vicki Hanson. 2011. Representing users in accessibility research. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2235–2238.

[41] Kristen Shinohara, Saba Kawas, Andrew J Ko, and Richard E Ladner. 2018. Who teaches accessibility? A survey of US computing faculty. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. 197–202.

[42] Donna Spencer. 2009. *Card sorting: Designing usable categories*. Rosenfeld Media.

[43] Gail M Sullivan and Richard Feinn. 2012. Using effect size—or why the P value is not enough. *Journal of graduate medical education* 4, 3 (2012), 279.

[44] Yuan Tian, Meiyappan Nagappan, David Lo, and Ahmed E Hassan. 2015. What are the characteristics of high-rated apps? a case study on free android applications. In *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 301–310.
[45] Jenifer Tidwell. 2010. *Designing interfaces: Patterns for effective interaction design.* " O'Reilly Media, Inc.".
[46] Pradeep K Tyagi. 1989. The effects of appeals, anonymity, and feedback on mail survey response patterns from salespeople. *Journal of the Academy of Marketing Science* 17, 3 (1989), 235–241.
[47] Christopher Vendome, Diana Solano, Santiago Liñán, and Mario Linares-Vásquez. 2019. Can everyone use my app? An Empirical Study on Accessibility in Android Apps. In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 41–52.
[48] Zhiyuan Wan, Xin Xia, David Lo, and Gail C Murphy. 2019. How does machine learning change software development practices? *IEEE Transactions on Software Engineering* (2019).
[49] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering.* Springer Science & Business Media.
[50] Xin Xia, Zhiyuan Wan, Pavneet Singh Kochhar, and David Lo. 2019. How practitioners perceive coding proficiency. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE, 924–935.
[51] Nor Shahida Mohamad Yusop, John Grundy, Jean-Guy Schneider, and Rajesh Vasa. 2020. A revised open source usability defect classification taxonomy. *Information and software technology* 128 (2020), 106396.
[52] Nor Shahida Mohamad Yusop, John Grundy, and Rajesh Vasa. 2016. Reporting usability defects: do reporters report what software developers need?. In *Proceedings of the 20th international conference on evaluation and assessment in software engineering*. 1–10.
[53] Yuhang Zhao, Edward Cutrell, Christian Holz, Meredith Ringel Morris, Eyal Ofek, and Andrew D Wilson. 2019. Seeingvr: A set of tools to make virtual reality more accessible to people with low vision. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–14.
[54] Leming Zhou, Andi Saptono, I Made Agus Setiawan, and Bambang Parmanto. 2020. Making Self-Management Mobile Health Apps Accessible to People With Disabilities: Qualitative Single-Subject Study. *JMIR mHealth and uHealth* 8, 1 (2020), e15060.

8 APPENDIX

Table 3. Groups of disabilities and their potential difficulties to use software applications.

Disability group	Potential difficulties using software applications
Auditory disabilities	People have difficulties in hearing information presented as video or audio.
Blindness	People who are blind is a considerable challenge based on the obvious fact that the application is a visual interface.
Color blindness	People have difficulties in distinguishing between combinations and colors of software application.
Low vision	People with low vision use specialized monitors or software that increases the size of text or images large enough for the individual to see software applications.
Cognitive disabilities	People who have difficulties with understanding content, remembering how to complete tasks, and confusion caused by inconsistent or non-traditional application page layouts.

Table 4. Interviewees and their focus disability groups.

Main Role	Interviewee Codes	Focusing Disability Groups
Programming	I1, I2, I6, I9, I12, I13, and I14	Auditory disabilities, Color blindness, and Low vision
Design	I3, I4, I5, and I10	Auditory disabilities, Color blindness, and Low vision
Management	I7, I10, and I11	Auditory disabilities, Color blindness, Low vision, and Cognitive disability
Testing	I2, I6, I8, and I9	Auditory disabilities, Blindness, Color blindness, and Low vision

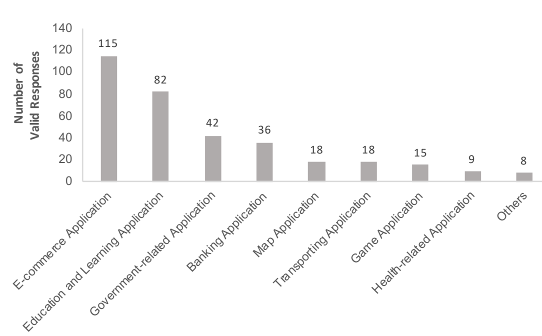

Fig. 2. Project domains of incorporating accessibility that identified by practitioners of the online survey.