Chapter 11 in Evaluation of Novel Approaches to Software Engineering,
Volume 1556 of Communications in Computer and Information Science, (c) Springer 2022

Addressing the influence of end user human aspects on software engineering

John Grundy[0000−0003−4928−7076], Ingo Mueller[0000−0003−2240−712X], Anuradha Madugalla[0000−0002−3813−8254], Hourieh Khalajzadeh[0000−0001−9958−0102], Humphrey O. Obie[0000−0002−6322−2984], Jennifer McIntosh[0000−0002−6655−0940], and Tanjila Kanij[0000−0002−5293−1718]

HumaniSE Lab, Faculty of IT, Monash University, Australia {john.grundy, ingo.mueller, anu.madugalla, hourieh.khalajzadeh, humphrey.obie, jenny.mcintosh, tanjila.kanij }@monash.edu
https://www.monash.edu/it/humanise-lab

Abstract. We outline some of the key challenges in supporting diverse human aspects of end users in software engineering. This includes examples of age, gender, physical and mental challenges, human values, personality, emotions, language and culture. We review key related work from a range of disciplines, and propose an initial taxonomy of end user human aspects that need careful consideration throughout the software engineering life cycle. Finally we outline a research road map of key areas requiring further investigation and work.

Keywords: Human Aspects of End Users · Taxonomy · Living Lab · Stakeholders · Personas · Modelling · Design · Defect Reporting

1 Introduction

Software is designed and built to solve human challenges in almost every domain [27]. However, we continually hear about many issues with current software relating to its poor fit with its target end users. This results in hard-to-use software that does not meet the user's needs and causes frustration, economic cost, inefficiencies, not fit-for-purpose solutions, and even dangerous and life-threatening situations. Many of these problems can be traced to a lack of understanding and incorporation during software engineering of end user human aspects. Humans are diverse and present software designers and builders with diverse challenges, including but not limited to different age, gender, culture, language, language proficiency, socio-economic status, physical and mental challenges, personality, emotional reaction to technology, engagement, and many others [30, 31].

In this paper we present an initial taxonomy of end user human aspects that impact software usage and hence design, implementation, testing, defect reporting and correction and ultimately its requirements and processes, methods and and tools needed for development. Providing a taxonomy of human centric aspects in software engineering helps not only to classify human-centric aspects,

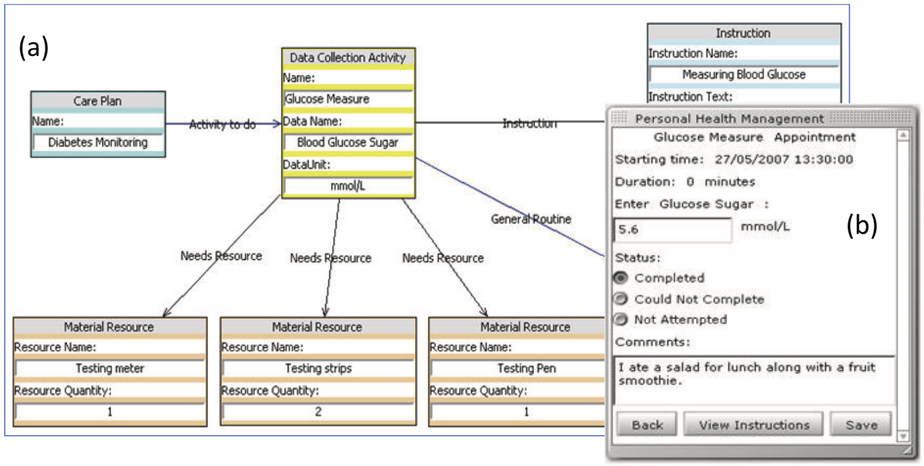

Fig. 1. A clinician-oriented Domain-specific Visual Language for care plan modelling and generating eHealth app (from [42])

but also provides a language to use when describing human aspects [64]. Taxonomies are used in science to define, group and rank by similar characteristics, and by doing this we will generate a scientific nomenclature that can be used to explore and address human centric aspects relevant to the industry.

Our aim is to better characterise these diverse end user aspects and their impact on software development, with some examples from our recent work addressing some of these. We begin with a motivating example in Section 2 and review of key related work in 3. We follow this by introducing our initial taxonomy of end user human aspects needing careful consideration during software engineering in Section 4. In Sections 5 to 10 we discuss a range of our recent work to try and address some of these human aspects during different software engineering tasks. In Section 11 we outline a research roadmap needed to continue to advance this area, and summarise this paper in Section 12.

2 Motivation

Motivating Example: Figure 1 shows an example model-driven engineering tool that takes a high level visual care plan model (a) and uses this to generate a fully functional eHealth app to implement the care plan (b) [42]. While in theory this is a good idea, the tool fails to account for the fact that end users of the app are generally elderly, many have English as a second language, they are unfamiliar with much of the health and technology terminology used, come from different cultures with differing concepts of and approaches to health and illness, many users have diverse physical and mental challenges that impact using the app, the app is boring for many to use, and many users worry about their data privacy and transparency of recommendations the app makes to them. The resultant app is unusable for many target end users, illustrating the problem of failing to consider diverse end user human aspects, individually and in combination, during development.

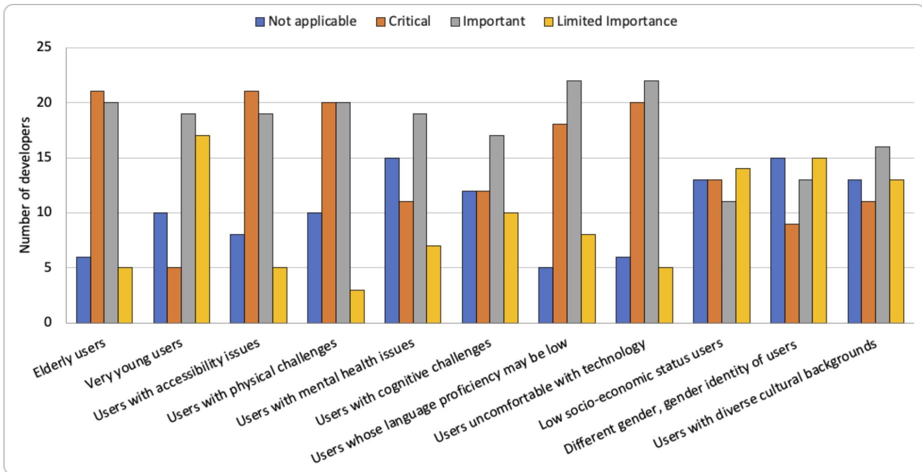

Fig. 2. End user human aspects survey respondents judge to be critical (or not) in their work (from [31])

Developer Survey and Interviews: We wanted to gain a better understanding of current developer approaches to addressing such diverse end users' human aspects, key open challenge areas in this domain, and determine key focus areas for researching new techniques and tools to address these, both in general and specifically for eHealth app development [31, 68]. We conducted a detailed survey, answered by 61 developers and managers, and then interviewed a further 12 developers in detail. We wanted to better understand how these diverse end user human aspect issues are understood and addressed from a software engineering perspective. Figure 2 summarises our respondents' ratings of some of the end user human aspects they encountered in their work.

Some of the key reasons given by respondents why they find these end user human aspects challenging to address included: the broad range of the end user human differences that exist and have to be catered for; the different languages and range of comfort with technology of different user groups; different problem solving styles of many end user groups; complexity of user interfaces in many application domains; and differences in terminology used, digital literacy and the need to carefully consider text and icon usage for many target end users.

We asked developers what would help them to improve development of their software to better address some of these diverse end user human aspects. Examples reported included: better requirements capture and human aspect modelling support; providing developers with better guidelines and practices to follow to address diverse end user human aspects; better design frameworks and tooling to address a greater range of end user human aspects; development tools that automatically prompt and advise on missing end user human aspect issues in designs and implementations; simpler interfaces in software for many end user groups; more live testing with representative end users, including better participant recruitment approaches ensuring more diverse end users are included; better defect reporting to enable end users to more easily identify, describe and report problems with their software; the need for better development processes to improve target end user involvement; a need for better education of software engineers about diverse end user human aspects and their impact on software usage; and more research into human aspects in software engineering.

eHealth App Guidelines Assessment: We assessed a number of eHealth app guidelines, and have been interviewing eHealth app developers and end users to determine how human aspects impact their development [68]. Many current guidelines are very general, not giving clear instructions for how to integrate different human aspects impacting health-oriented apps. Surprisingly, despite the range of challenges many eHealth app users have, many current guidelines do not take these into account when recommending approaches to build app features or overall eHealth app design.

App Review Analysis: We semi-automatically reviewed a very large app reviews for a large number of apps to determine (i) human values violations [59], and (ii) privacy concerns [33]. Similar to our findings from eHealth domain guidelines analysis and end user aspects developer survey and interviews, there are many issues in taking into account diverse end user values in app design. How to fix many of these human values issues is still unclear.

3 Related Work

Human aspects impacting software design, implementation and evaluation have been studied for many years. Work of particular note has occurred in the disciplines of business, ergonomics, and human-computer interaction (HCI). Several taxonomies and classifications have been created as part of these efforts [57, 4, 72, 67, 58]. However, we argue that these taxonomies have shortcomings that limit their applicability for practical software development activities:

1. they focus on problem-solving rather than comprehension of diverse human needs and empathising with the people behind these needs and expectations;
2. they predominantly centre on the formal representation and modelling of software features derived from human needs - often at a higher level of abstraction - thus bypassing the people and their natural characteristics, skills, abilities, strengths, weaknesses, etc; and
3. they lack comprehensiveness.

We argue that these shortcomings result in a lack of support that developers require for critical activities, not only during requirements elicitation and capture but also software design and testing.

Morris and Stauffer [57] developed a taxonomy of keywords that is aimed at helping with requirements elicitation activities by providing a systematic structure for user / customer interviews. It is deliberately kept short in order to allow its application under tight time-to-market deadlines. The elements of the taxonomy refer to features of computer systems rather than human aspects. Each keyword is presented at a higher level of abstractions (e.g. 'control', 'display', 'form'), thus potentially providing barriers between the developers and users who are not familiar with this terminology and way of thinking.

Saurin and Patriarca [4] specifically focus their taxonomy on interaction (human-to-human as well as human-to-machine) in socio-technical systems. The 'nature of the agent' taxonomy category is a generic placeholder for collecting information about human aspects but it does not provide support for systematically and comprehensively cover such aspects.

Longo et al. [48] propose a taxonomy of human factors to enable the systematic capture of cognitive and physical abilities and psychological attitudes of industrial workers. The taxonomy comprises very detailed sub categories such as attention, comprehension, knowledge, memory functions, musculoskeletal health, neurological health, motion, perception (including sight, sound, and touch) to name a few. The categories are certainly human-centric, however the entire taxonomy does not link them back to people. Moreover, the large number of 50 sub categories may potentially make this taxonomy hard to operationalise.

Singh et al. [72] discuss a taxonomy to capture usability requirements for telehealth systems. The scope of the taxonomy is tailored towards addressing concrete usability and accessibility needs. However, the categories of the taxonomy are structured based on abstract terms such as 'natural input' and 'guided instructions'. The taxonomy is designed from a system perspective and does not centre on people and their needs *per se*. Furthermore, the categories of the taxonomy are accompanied by long descriptions. This may indicate that additional training is required to work with this taxonomy and individual categories may be interpreted and applied differently by developers.

Seneler et al. [67] propose a technology adoption taxonomy aimed at supporting developers to identify key properties of user interfaces that promote the adoption of a technological solution. The motivations behind this taxonomy are usability and accessibility aspects and several categories relate to human aspects such as 'user characteristics & mental and emotional states'. Each category has a number of sub categories that are aimed at helping to systematically document information about human users. However, the set of sub categories is incomplete, for example they do not cover physical aspects and needs.

Mosqueira-Rey et al. [58] propose two taxonomies. Firstly, a usability taxonomy organised in abstract categories, e.g. knowability, operability, safety, efficiency. Secondly, a context-of-use taxonomy with three main categories named 'user', 'task' and 'environment'. Both taxonomies remain at an abstract level. Human values, needs and characteristics are not in focus.

Our approach does not overcome all shortcomings of these existing taxonomies. However, we believe that by making the end user human central and focusing on taxonomy categories that reflect human values, needs and characteristics, we provide developers with knowledge and tools to better understand the needs of diverse users and consequently to develop more inclusive software.

Table 1. Personal characteristics of end users and their impact on software usage and engineering

Human Aspect	**Impact / Issues to consider**
Age	Impact of age on usage and/or age bias in design of software systems. People of differing ages e.g. young vs older may have quite different expectations, challenges and reactions to software which need to be carefully designed into solutions[53]
Gender	Impact of gender bias towards software end users. Not just terminology/words used, but unconscious biases including treatment, assumptions about users and software usage [76]. Several prominent mainstream articles and books have highlighted the gender bias in e.g. apps and smart technologies [74].
Ethnicity	Ethnic bias against some end users, assumptions about users from different ethnic backgrounds, biased training sets etc [69, 78].
Personality	Researchers have studied personality impact on programming, testing, design, requirements engineering and maintenance, which has shown to have impact on developer performance [19]. Much less researched to date has been on impact of personality of different end users.
Emotions	Different people react differently to technology solutions from an emotional perspective [23]. Impact of emotional reactions, effective and cognitive states on use of software, perception of software. Some react positively, while others negatively, to the exact same solution, which can dramatically impact the acceptance and usage of the software.
Engagement	No one wants to use boring or disengaging software; this is especially important for software for behaviour change e.g. diet, fitness eHealth apps, finances etc, and also for games in general [44].
Physical or mental challenges	Impact of wide range of physical challenges on end users e.g. colour-blind, sight challenges, hearing, coordination, stroke, obesity, cardiac, infection, etc [39]. Impact of mental challenges e.g. due to injury or illness.
Cognitive style	Impact of different problem-solving approaches e.g. neuro-atypical users.
Preferences	The users own personal preferences, whatever their particular demographics - all else may be the same, but individual users might have different personal preferences for some aspects of their software.

4 A Preliminary Taxonomy – End User Human Aspects and Software Engineering

In previous work we have developed taxonomies to better structure and understand different domains in software engineering. This included a new synthesized taxonomy for usability defect report classification[80] and human aspects impacting software engineers involved in the requirements engineering process[34]. We want to develop a similar taxonomy of human aspects impacting end users of software, and/or how human aspects of end users impact the engineering of software[30]. We started by dividing human characteristics into three groups: 1. personal demographic characteristics (Table 1); 2. skill or expertise-based characteristics (Table 2); and 3. group-based characteristics (Table 3).

Table 1 summarises some key ***personal, demographic*** characteristics of end users that (i) may impact their usage of software and (ii) need to be taken into

Table 2. Skill, experiential or environmental-influenced characteristics of end users and their impact on software usage and engineering

Human Aspect	**Impact / Issues to consider**
Spoken Language	Impact of spoken and written language assumptions on software end users. Does the software support users who need different read/spoken languages? Are translations accurate? Are requirements multi-lingual [41].
Socio-economic status	Different end users may have very different access to technology, living and work environments etc [32, 75]. Can they afford the latest handset to run an app on? Does their internet connection support expectations of high bandwidth?
Language proficiency	Impact of language complexity, jargon, dialogue on users.
Education	Impact of different end user educational attainment, range of technology and domain-specific skills developed [10].
Comfort with technology	Many end user groups are much less comfortable using technology solutions than the software engineers that develop them [62].
Location	Where an end user is may impact their software usage, including the different environments the software is used in, rural vs urban living, different software regulations, etc [79].
Religious beliefs	End users have wide variety of different beliefs and practices, some greatly impacting their use of software in different parts of their living and work.
Human values	Set of human values important to individuals, teams, organisations, end users, societies [59]. Includes but not limited to values including inclusive, transparent, creative, authoritative, belonging, secure, security, family, tradition, devout, polite, open, obedient, loyal, forgiving, social justice, protecting environment, privacy, ...
Skill level	Personal (vs team) experience in a work domain that impacts software usage.

consideration by software engineers to ensure software meets that characteristics, different to other end user groups. It is understood that people can have multiple combinations of these characteristics.

Table 2 summarises some end user human aspects that are ***context driven***, due to external influences such as upbringing, training, experience or other influences. Many of these aspects change over time unlike characteristics in Table 1 which may also change e.g. due to ageing or illness, but generally remain more or less the same over one's lifetime.

Table 3 summarises end user human characteristics that result from ***social contexts and interactions***, including living and working with other people. Some may change over time as social interactions and contexts change, while some have a life-long influence.

In following sections we give some examples of our recent work addressing some of these human aspects of end users in the context of software engineering. For each we briefly describe the end user human aspect, discuss issues software engineers face in addressing the human aspect, and briefly describe the research we are undertaking to try and assist them. We then summarise key outstanding issues we have found and present the outline of a research roadmap to address these.

Table 3. Group or multiple human characteristics of end users and their impact on software usage and engineering

Human Aspect	**Impact / Issues to consider**
Culture	Different cultural practices, assumptions, behaviours, accepted and unaccepted practices, biases against particular users of software [5, 6, 78].
Geographic location	Where end users are located geographically may impact them and a range of software systems significantly [79].
Team climate	Working in a team environment means end users need to interact with others and may be impacted by other individuals, group, organisational and even societal differences [47].
Family environment and Martial status, Child status, Caring responsibilities	Impact of various living arrangements that differ from person to person and family unit to family unit. Potential for bias against different arrangements or simply not sufficiently taking into account differences [43].
Work status	Impact of having no work, being under-employed, working in preferred vs non-preferred job, income, and other work-related differences [51].
Collaboration, Communication style	Living and working with others requires communication and collaboration skills, many of which need to be adequately and appropriately supported in software systems.
Organisational or Societal values	Related to personal human values above, collective group / organisational / societal values differences that impact end users and their software usage.

5 Age

5.1 Design for Differently-aged Users

There is an increase in the average age of internet users, with 73% of United States adults over the age of 64 accessing the internet according to the internet usage report from Pew Research centre [63]. Web-browsing behaviour differs between various age groups according to a study by Joyce and Nielsen in 2019 [36]. Elderly users may face issues such as screen readability due to visual impairment associated with ageing [36]. Young people also report age-specific preferences. Despite developers and designers are mainly being younger[61], teenagers also complain about poor visual designs, such as font size, background colour and layout of certain websites.

Based on the findings by Masood et al., younger children also commonly have problems working with mobile applications, such as the system status not being apparent for them, having a hard time working out what to do next, and not remembering which page or button was accessed earlier [50]. They recommend that children-oriented software needs to more clearly show the current state of the page, and sometimes the child users may need some guidance to do the next step. Moreover, the buttons and menus need to be simple enough and buttons and menu links should be easily identified as being clickable while menu headings and titles as being not clickable. A Fingerprint app [60] is used to describe how to design the software user interface for the children. This work discusses four key points for the vision element regarding kids – integer vision effect, functional area design, icon and button design, and font design. The work of Michaels and Boyatzis et al. discusses colour preferences of children users as well as the effect of colour on children's emotions [55, 13].

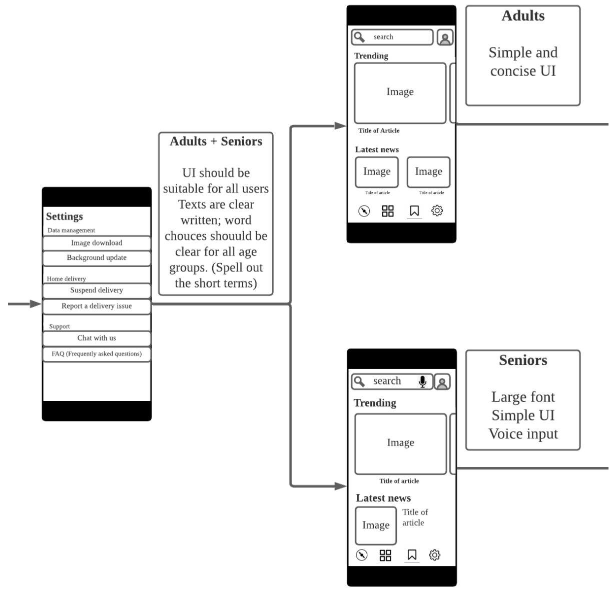

Fig. 3. Example of branching (from[40])

5.2 AgeMag: Countering Age Bias in eCommerce Software

Boll et al. provide a set of user interface design guidelines for people between 55 to 75 years old [12]. Based on their studies with older users, most reported that the icons were too small, and that double-clicking a small button was a problem for older people. Moreover, they found that the menu needed to be put in conventional positions to make consistent in the software.

We had similar findings in our study exploring age bias in eCommerce software. We found age bias in the design and presentation of eCommerce websites. Elderly people had a lot of difficulty navigating their way around eCommerce websites as the interface was difficult to see (small font) and visually confusing (too cluttered). Older people had trouble navigating back and forth through the website and could not easily locate what they were looking for. Often they had to ask for help, and were likely to give up rather than try to resolve a problem. Interestingly, there was a clear delineation between the elderly or 'Silent Generation' (born between 1928 and 1945), and the 'younger generations' (born before 1945) when using eCommerce, suggesting people in their 60s and even early 70s had less difficulty in using eCommerce.

From our results, we developed an 'AgeMag' to evaluate eCommerce applications for age bias. The AgeMag had two personas - an 'elderly user persona' and a 'general user persona' - which were used to conduct a cognitive walk-through of eCommerce websites. The cognitive walk-through involves using the personas as if they are real people, to identify where they might struggle to use the website, and how the design and interface of the website might be adapted to make it more usable. Given many people have been forced to rely on eCommerce for their necessities during the pandemic, this is a simple way to evaluate eCommerce for age bias and adapt the design of eCommerce websites for elderly people, many of whom are using eCommerce for the first time [53].

Designing emotion-oriented software has been the focus of some researchers [23]. A goal model for a smart home device was created by analysing the emotions of older people to helps developers to understand the expectations of an older adult. The model included different emotions for the elderly people to help get the elderly people to accept the device and feel like this is what they need. Curumsing et al. demonstrate a case study of an emergency alarm system for elderly people [24]. They suggested a few important factors in designing the framework and also keeping the interest of the elderly people. These include designing solutions in a way that is easy to use and cost-effective.

These studies reflect the fact that users' age should be taken into account when designing software applications. Age-related considerations need to be taken into the account from the early stages of software development, i.e. modelling and design of the software. There are many existing modelling frameworks, most do not currently support modelling the age of end users and providing different design solutions for different age groups and needs of end users [40]. We developed a set of extensions to the commonly used wire-frame modelling approach to incorporate different designs for child, adult and senior end users in [40]. The modelling approach was evaluated with developers and a prototype news app was developed using our approach with a range of differently aged end users. Further work includes incorporating other human-centric aspects into the extended wire-frame model e.g. gender, culture, language, and trying the same model extension approach in other modelling frameworks, such as user stories, use cases, and sequence diagrams. Figure 3 shows an example of such an extended wire-frame design. This shows multiple "pathways" suitable to different aged users planned for in the application under design.

6 Gender

Gender is another important human aspect of the end user. After a comprehensive review of literature Burnett et al. [16] concluded that there are five facets where people differ based on gender. Those include: motivation, information processing style, computer self-efficacy, risk aversion and tinkering. They incorporate the facets into persona descriptions of users and propose a cognitive walk-through approach of usability inspection on problem solving software. They developed a tool called GenderMag based on the approach [16]. They reported several application of GenderMag tool to successfully identify gender inclusiveness issues in problem solving software [15].

In our study of understanding the characteristic, challenges and goals of domestic workers in Bangladesh, in order to design digital solution for them, we found that there are 1.8 million domestic workers in Dhaka and around 90% of them are female. The research project is in progress and with the focus groups already conducted we have already identified some gender-based characteristics such as motivation behind using digital solutions. We plan to use these requirements to inform future domestic work recruitment, management and payment solutions with these requirements. In addition, many works have low educational attainment, come from rural areas, and often share smart phones among several, introducing several other important human aspects to the design process.

7 Physical Challenges

7.1 Adaptable User Interfaces for Diverse End Users

At a time where User Interfaces (UI) are becoming increasingly complex, it is no longer sufficient to develop a single UI for all users with a 'one size fits all' approach. As a solution, designers can aim to cater all diverse users with a single design, but this will lead to compromises due to conflicting user requirements. Therefore, in most software design, usability is generally designed for the majority of users with homogeneous characteristics, often neglecting those in need of special features and support. These users who need special features can be users with vision impairments, cognitive impairments, aging population as well as users from socially and culturally diverse backgrounds.

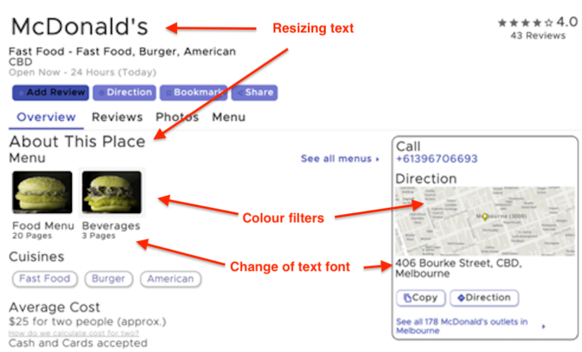

Fig. 4. Adaptive Zomato – text size increased, colour changed (from [49])

Our solution to this problem is to develop an *Adaptable UI*, which allows users to tailor the UI components to their individual needs [49]. To achieve this, we developed a framework with an adaptable user interface component library. We implemented it with the open-source web development platform, Flutter, which is increasingly becoming famous among web developers. Our framework supported the following three adaptable features:

1. adaptive colour themes: Pre-defined colour vision and custom colour themes;

2. adaptive image settings: Allowed selecting image colour filters to apply on all images; and
3. adaptive font settings: changeable text colour, size, font type.

We rebuilt the popular Zomato website using our framework and provided the adaptive feature via an accessibility menu. In this prototype, to represent the diverse users, we chose users with colour blindness, low vision and dyslexia. We evaluated the rebuilt Zomato in a user study, based on W3C guidelines for accessibility [77]. Due to COVID-19 restrictions, we adopted a Persona based evaluation method, where each participant was assigned a Persona with a vision impairment and this impairment was simulated via browser plugins during the study. Figure 4 shows an example of our adaptive Zomato app in use.

All participants found the functionality provided by the adaptive features helpful for their simulated vision deficiency, enabling them to view certain elements of the website easier. Our evaluation indicated that the adaptable version of Zomato outperformed the original Zomato website in all anticipated sections from the W3C guidelines. Therefore, we recommend web developers to explore the use of such adaptable widgets via a framework similar to ours as this would assist them to cater for the diverse user needs with a less workload.

7.2 Improving human-centric software defect evaluation, reporting and fixing

Customarily, *defect reporting* exists in most applications and web sites to enable end-users to report issues and for developers to receive actionable feedback. However, the impact of "human-centric" issues - such as age, gender, language, culture, physical and mental challenges, and socio-economic status - is often overlooked in defect reporting. Therefore, most defect reporting tools lack focus on human-centric features to enable a challenged user (eg: a vision impaired user) to adequately navigate and report defects. Furthermore, most defect reporting tools lack sufficient defect report structuring, reporting guidance, and do not emphasize the perceived severity of the defect to developers. Due to these usability issues, sometimes diverse end-users are unable to report defects effectively and thus developers find it difficult to understand and fix the reported defect.

In our research, we aimed to understand the issues faced by diverse end-users in reporting defects and developers issues in understanding them via a novel human-centred defect reporting tool prototype [37]. Our research contained two stages. In the first stage, we developed a simple prototype of a defect reporting tool and developed Personas to represent diverse end users with vision, hearing, motor and reading impairments. Using these Personas, we conducted cognitive walk-throughs on four chosen applications (Grab, a university Moodle, Snapchat and Skype) and requested participants to report any human centric issues via the prototype. Based on this, we identified a list of potential improvements that could be introduced to a human-centric defect reporting tool. In the next stage, we re-implemented the prototype tool by addressing most of the identified improvements and conducted a second cognitive walk-through. In this walk-through, we evaluated the end users' ability to report a defect as well as developers' ability to understand the reported defect, using Personas of both end users and developers. Figure 5 shows an example of some of the defect reporting app screens we prototyped.

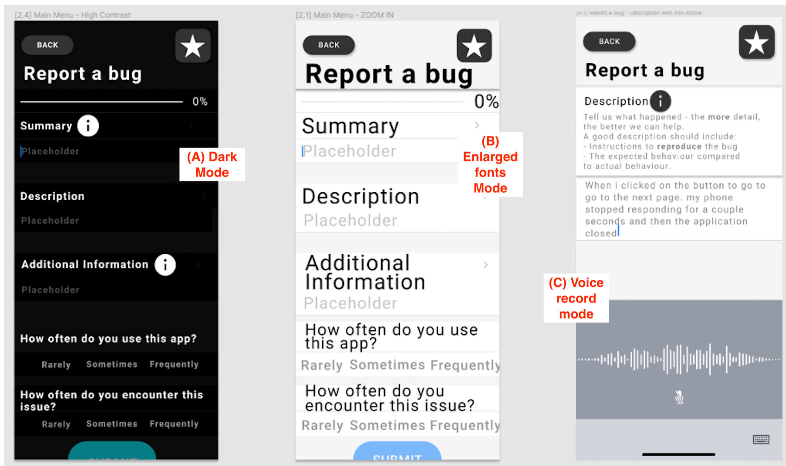

Fig. 5. Human-centric defect reporting prototype with Additional accessibility controls turned on (from [37])

Based on both these stages, we developed a set of guidelines to improve the usability of defect reporting for diverse end users and to increase the useful information provided to developers. Additionally, we identified three major factors that can assist developers in human-centric defect evaluation and resolution: 1) Educating users about defect reporting and educating developers about Personas of different users and their diverse challenges 2) Capturing the frequency of the application use and defect encountering frequency does not affect the developers perceived severity of the issue 3) Increasing the amount of extra information collected about a defect, while taking appropriate steps to prevent the over complication of defect reports, is effective. We recommend developers and defect reporting tool designers to adopt these guidelines and findings to generate more human-centric defect reporting tools.

8 Human Values

Human values such as tradition, helpfulness, freedom, creativity, etc. are a critical human aspect and should be considered in the design, development, and deployment of software systems. Human values are the guiding principles for what people consider to be important in life [22] and these affect the choices and decisions that they make including technological choices. Because human values serve as a vehicle for expressing need [26], they should be properly articulated and captured in the requirements gathering process. Moreover, human values also determine behaviour and attitude [66], thus users' and other stakeholders' values should be embodied in the interaction models of the software systems – showing how their interaction with the system reflects and supports their values (or at the least not violate their values).

Human values, if they are not properly captured and integrated throughout the entire software development process, can have deleterious consequences, not only on direct stakeholders such as companies and end-users but also on indirect stakeholders and society as a whole. Recent media has shown some of these violations of human values and their associated negative consequences, e.g., Robodebt - an inaccurate automated debt recovery tool that distressed thousands of Australians [52], Facebook Cambridge Analytica scandal - privacy and tool for social manipulation and undermining of democratic processes [17], Amazon algorithms terminating contracts of package delivery drivers [73].

We have carried out some work in mining values requirements and detecting violations of human values from app reviews [59]. App reviews provide useful information such as critiques, bug reports, feature requests from a user perspective, and have been mined to support change requirements for future software updates and evolution. We analysed 22,119 app reviews from the Google Play store using natural language processing (NLP) techniques to understand potential values violation as reported by users. We analysed reviews from 12 apps chosen to cover different audiences and age groups, with different expectations and interactions with technology. We based our NLP values violations detection approach on the widely accepted Schwartz theory of basic human values. The results of our analysis showed that 26.5% of the 22,119 app reviews contained user-perceived violations of human values. In terms of the broader values categories, *benevolence* and *self-direction* were the most violated values categories while *tradition* and *conformity* were the least violated values categories. Moreover, looking into the finer details of specific values items showed that *helpfulness, pleasure,* and *curiosity* ranked amongst the topmost violated value items while *obedience* and *influential* were the least violated values items.

Our results show that values requirements can be mined from app reviews. Nonetheless, we note that app reviews mining is still a reactive approach that happens post-factum, and we recommend proactive approaches such as participatory design and the direct engagement of and elicitation of values requirements from all stakeholders involved in the software project. Furthermore, careful consideration of stakeholders' specific domain context in the design of values elicitation protocols should be made - thus effectively capturing human values as an important human aspect.

9 Emotions

Different end users may react to the same software application in quite different ways, in terms of their "emotional response" [23]. For example, in an in-home aging user support solution, a "smart home" software solution can be perceived in quite different ways by different elderly end users and their carers. For example, one person may find the presence of sensors and voice-enabled interaction induces a feeling of safety, care, security, and reduces feelings of isolation. Another end user may find the exact same software solution overly controlling, inducing feelings of lack of control, invasive, and even threatening to their self-worth and liberty. This will lead to very different experiences, acceptance and take-up of the solution. Very different configuration of the solution – or adoption of a totally different solution – may be needed to satisfy the range of diverse end user emotional reactions.

We used extensions to a goal-directed requirements modelling language[56] in a case study of designing and building a smart home for elderly[23]. An example of an extended goal-directed requirements model is shown in Figure 6. This allows requirements engineers to reason about potential emotional reactions of stakeholders, positive and negative, look to address these in design solutions, and evaluate whether these are adequately addressed in testing.

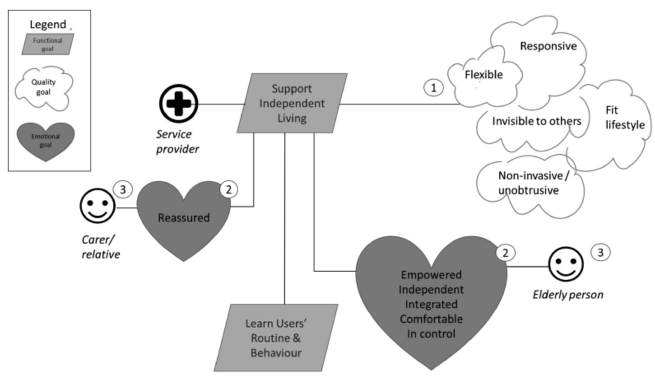

Fig. 6. An Emotion-oriented Domain-specific Visual Language (from [29]).

10 Language and Culture

Socio-cultural context and language preferences are important human aspects that influence end users' interaction with software. In our on going research project with domestic workers and fisher-folk in Bangladesh [2], we identified that user interface and voice instructions in the software developed for them needs to be in "Bangla". English language literacy of the target end users are low, therefor in order to make the software interface usable by them the communication language used in the software needs to be "Bangla" - only language they are proficient in. We have found that, due to socio-cultural context, mobile phones are often shared among family members, as such designing of personalized user interface needs to consider "plural" end users.

In another on going project of identifying interrelationship among different factors influencing online shopping, we have found the influencing factors are related very differently for Australian and Chinese online consumers. For Chinese consumers, their demographic characteristics influence how web aesthetics is perceived, on the other hand, for Australian online consumers demographic characteristics have connection with web aesthetics, however both demographic characteristics and web aesthetics influence their overall shopping experience. This can be related to "Indulgence Vs Reatraint" dimension of Hoftesde cross culture theory [35]. According to this dimension indulgence society tends to enjoy life and free gratification of human desires while restraint society tends to restrain one's desire to abide by social norm.

Fig. 7. Comparing Australia and China by Hofstede's cross-culture theory (from [1]).

In our smart parking development project [46], we identified several personas with a variety of differing human aspects impacting their usage of the smart parking app. Several of these factors we have highlighted in previous sections – age, gender, emotional frustrations with current solutions. Other key differences included language proficiency – both primary spoken language and also level of read language ability. We needed to provide users the ability to ask for the app to display text in their preferred spoken/read language, and to request simplified descriptions of tasks to perform. Similar to the human-centric defect reporting app and adaptive user interface widgets projects, we also encountered users with dyslexia who wanted different fonts be used and reduced text label usage.

11 Research Roadmap

To further investigate a range of end user human aspects impacting software engineering, we propose a number of research directions below. Some of these we are currently working on. Some extend previous work, and some we think are promising directions for the research and practice communities to explore. These fall into several research themes:

- End user human aspects – better understanding the nature of human aspects
- Engagement of stakeholders – better engaging diverse stakeholders throughout development
- Requirements capture – eliciting, modelling and reasoning about requirements relating to diverse end users of software
- Understanding end users – helping developers to better empathise with, appreciate challenges of and address diverse end user human characteristics and challenges using software
- Design support – providing developers with better techniques and tools for design and implementation of software to address diverse end user human characteristics and challenges
- End user empowerment – helping end users take greater control of their software solutions and engagement with developers
- Developer human aspects – understanding how developers own diverse human aspects impact different aspects of developing software

11.1 Address Diverse Human Aspects: End User Human Aspects to Further Investigate

A number of human aspects require further research to determine (i) how to describe and discuss them; (ii) how best to elicit software requirements from and regarding end users with these human aspects; and (iii) how to take better account of them during software design, implementation and defect fixing:

- personality – while a lot of SE research has investigated personality impact on software engineers, little to date has focused on how different end user personalities impact software usage;
- ethnicity – AI-based software biases have been highlighted in recent times. How to better address these has been subject to recent work, but more work is required.
- engagement – how to understand diverse end user preferences and needs around their engagement with their software requires more research.
- cognitive style – limited understanding of how different cognitive styles can and should be taken account of during software development exists. To date this has main focused on children or elderly with specific cognitive challenges.
- socio-economic status – and its impact on access to and use of software is still not well understood. Addressing the growing digital divide is critical.
- human values – and taking account of different end user values has become an area of increasing interest in recent years. The impact of different values on software design is still poorly understood.
- culture – understanding diverse end user culture, incorporating cultural values into software, and fixing culture-related defects is also poorly understood.
- family environment – diverse non-work living environments and its impact on software design and usage by end users also requires further study.

– organisational values – and their impact on teams and individuals complements individual human values research.

As noted in several of our projects, humans have multiple characteristics. It is largely unknown in many domains and in general how these interplay to influence software usage i.e. which is the most important characteristic to consider, or how different characteristics interact in different ways for different people and have a major influence on their software experience. Studying various combinations of human aspects is of course very difficult as there are an almost infinite set of combinations. In addition, one's own experiences, current state of physical or mental health, family, team or organisational usage context, and other variables might themselves have a major influence on what makes a "good" or "bad" software design for a particular user.

11.2 Address Engagement of Stakeholders: Co-creation Living Lab

We have begun to explore the "Living Lab" approach to co-creating software requirements and designs with end users [30]. The idea is to better involve end users, and indeed all stakeholders whether eventual end users or not, throughout the software development process as equals with software engineers. Living lab approaches have been used in a variety of domains but most particularly for eHealth software and smart living software domains, but not usually from a software engineering perspective. We would like to explore the use of such an approach on software engineering processes, techniques and tools. The attraction of this approach is the concept of co-creation with diverse stakeholders that fully takes their varied human characteristics into account during requirements and design, and treat them as equal co-creators of the solutions.

11.3 Address Engagement of Stakeholders and Better Requirements: Stakeholder Identification

Not all stakeholders of a software system are end users. How different stakeholders e.g. sponsors, managers, affected people who do not directly operate or interact with the software used by others, etc are impacted by their own human aspects is a potentially important area for further study and translation to practice. Using living lab and other techniques, we would like to develop better guidelines for identifying and having dialogue with diverse software stakeholders, taking better account of their human aspects as well as those of direct end users during its development.

11.4 Address Better Requirements and Understanding Diverse End Users: Diverse Personas

Software is increasingly used by more diverse users. How to convey this diversity, their differing needs, differing experiences with the same software, and how to ensure peoples' differing needs are met is still unclear. One approach we have used in several projects is developing a range of personas to represent diverse end user demographics, goals, and frustrations with current solutions. Some of these have used data-driven techniques to help construct the personas, such as app reviews highlighting key unmet needs [46]. We have been compiling persona examples for diverse end users, particularly the elderly and children, reported in a variety of literature to inform development for these user groups [45, 3, 8, 71].

11.5 Address Better Understanding Diverse End Users: Modelling Human Aspects

Human aspects of end users are sometimes captured during requirements engineering and design phases of software development. However, we currently lack suitable approaches to formally model these in models and then reason about completeness, correctness and whether addressed in software solutions [40]. We have been developing preliminary extensions to a number of models – including iStar, wire-frame designs, goal-based requirements, user stories and others – to try and support better capture, reasoning and use of differing end user characteristics [40, 23, 46].

11.6 Address Better Understanding Diverse End Users and Better Designs: Design Support for End User Human Aspects

As well as better identifying diverse end user needs during requirements engineering, developers need to be better supported to recognise, appreciate, understand and design for these diverse end users during their design and implementation tasks. Some end users have different problem solving styles, depending on gender, age, cognitive style, personality, etc [76, 19, 21]. Accessibility has been long studied and design approaches to address developed for many end user challenges. However, many of these approaches are as yet not well known or well supported in software design tools, platforms and APIs [7]. Adaptive interfaces have been trialled to address platform differences, but also to a lesser degree end user differences [49].

11.7 Address Better Designs and End User Empowerment: End User Development

End user development offers the promise of end users being able to develop, or at least reconfigure, their software solutions to suit their individual preferences and needs. Many approaches have been tried, from programming by example to low code/no code pre-packaged solutions to (more or less) configurable applications [11, 14]. While software is too complex in general for end users to develop solutions, in controlled settings giving end users the ability to (re)configure software solutions may greatly improve their experiences and efficiency and effectiveness of the software. This includes reconfiguring user interface layout and flow [49], specifying rules and constraints around data, data integration support from diverse sources [9], configuring domain-specific applications [28], and integration of multiple disparate solutions [18].

11.8 Address End User Empowerment and Better Understanding Diverse End Users: Human-centric Defect Reporting and Fixing

We carried out a preliminary investigation of improving "human-centric defect reporting", including developing a set of human-centric defect reporting interfaces and a set of personas to represent different defect reporters [37]. This highlighted a number of unsolved and under-researched to date issues in defect reporting, understanding, diagnosing and fixing. Examples of further research to carry out include understanding if personas representing end user defect reporters help developers to understand, diagnose and fix human-centric defects; how industry teams currently handle such defect reports; whether different users have different challenges reporting defects with their software; and whether software domain, company culture and developer human characteristics impact human-centric defect fixing.

11.9 Address Developer Human Aspects: Understand impact of developers' own diverse human aspects

While end users are naturally diverse in terms of their human aspects for many software domains – banking, education, home automation, business etc – software developers are far less diverse and often very different to their end users [70, 65]. Most software developers are relatively wealthy, highly educated, have high language proficiency, most are male, and most relatively young [70]. Many of the end users and stakeholders they develop software for are often quite different to this profile, particularly in software domains. Assisting developers to appreciate, understand, empathise with and ultimately design and build software more suited to people very different to them remains a challenge for the software engineering discipline. A greater diversity of software engineers and better education about the challenges of supporting diverse end users are claimed by various studies to improve this situation [20, 54, 65, 25]. However, how a variety of developer human aspect differences impact software engineering in general and particular phases of software development is still largely unclear[34, 38, 25].

12 Summary

End users of a software application have many different "human characteristics". Some have been studied extensively in psychology, human-computer interaction, management and other disciplines, if not Software Engineering researchers. We have presented a preliminary taxonomy of some of these human aspects and some key related work done to date, predominantly in other disciplines. Some of the areas that we have been studying how they impact software engineers and their work include age, gender, physical and mental challenges, emotions, culture and language. Many however are as yet unclear in terms of their impact on software engineering, how well they are accommodated by software teams, and how combinations of human differences impact software engineering and software products. Some areas we see as how potential for further research include some under-researched human aspects; improving stakeholder identification and engagement via living lab co-creation approaches; further use of diverse personas in software engineering; better modelling and design support for diverse human characteristics and challenges; end user development allowing end users to tailor their solutions to their own needs; and improved human-centric defect reporting. Understanding how developers own diverse human aspects impact software engineering and interplay with their end user human aspects is also a rich area for continued work.

Acknowledgements

Parts of this work has been supported by ARC Discovery Projects DP200100020, DP170101932 and DP140102185, Industry Transformation Research Hub IH170100013, and ARC Laureate Fellowship FL190100035.

References

1. Compare countries - hofstede insights. (2021). [online]. Available: https://www.hofstede-insights.com/product/compare-countries/
2. Participatory research and ownership with technology, information and change (protic) ii. [online]. Available: https://www.monash.edu/it/hcc/dedt/projects/participatory-research-and-ownership-with-technology,-information-and-change-protic-ii
3. Abd Malik, S., Azuddin, M.: Mobile technology for older people: Use of personas. In: 2013 International Conference on Research and Innovation in Information Systems (ICRIIS). pp. 97–101. IEEE (2013)
4. Abreu Saurin, T., Patriarca, R.: A taxonomy of interactions in socio-technical systems: A functional perspective **82**, 102980 (2020)
5. Alkaabi, A., Maple, C.: Cultural impact on user authentication systems **4**(4), 323–343 (2013)
6. Alsanoosy, T., Spichkova, M., Harland, J.: Cultural influence on requirements engineering activities: a systematic literature review and analysis. Requirements Engineering pp. 1–24 (2019)
7. Alshayban, A., Ahmed, I., Malek, S.: Accessibility issues in android apps: state of affairs, sentiments, and ways forward. In: 2020 IEEE/ACM 42nd Int. Conf. on Software Engineering (ICSE). pp. 1323–1334. IEEE (2020)
8. Antle, A.N.: Child-based personas: need, ability and experience. Cognition, Technology & Work **10**(2), 155–166 (2008)
9. Avazpour, I., Grundy, J., Zhu, L.: Engineering complex data integration, harmonization and visualization systems **16**, 100103 (2019)
10. Ball, K., Mouchacca, J., Jackson, M.: The feasibility and appeal of mobile 'apps' for supporting healthy food purchasing and consumption among socioeconomically disadvantaged women: a pilot study. Health promotion journal of Australia **25**(2), 79–82 (2014)

11. Barricelli, B.R., Cassano, F., Fogli, D., Piccinno, A.: End-user development, end-user programming and end-user software engineering: A systematic mapping study **149**, 101–137 (2019)
12. Boll, F., Brune, P.: User interfaces with a touch of grey?–towards a specific ui design for people in the transition age **63**, 511–516 (2015)
13. Boyatzis, C.J., Varghese, R.: Children's emotional associations with colors **155**(1), 77–85 (1994)
14. Brich, J., Walch, M., Rietzler, M., Weber, M., Schaub, F.: Exploring end user programming needs in home automation. ACM Transactions on Computer-Human Interaction (TOCHI) **24**(2), 1–35 (2017)
15. Burnett, M., Peters, A., Hill, C., Elarief, N.: Finding Gender-Inclusiveness Software Issues with GenderMag: A Field Investigation, p. 2586–2598. Association for Computing Machinery, New York, NY, USA (2016), https://doi.org/10.1145/2858036.2858274
16. Burnett, M., Stumpf, S., Macbeth, J., Makri, S., Beckwith, L., Kwan, I., Peters, A., Jernigan, W.: GenderMag: A Method for Evaluating Software's Gender Inclusiveness. Interacting with Computers **28**(6), 760–787 (10 2016). https://doi.org/10.1093/iwc/iwv046, https://doi.org/10.1093/iwc/iwv046
17. Cadwallader, C., Graham-Harrison, E.: Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach. [Online.] Available: https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election
18. Cappiello, C., Matera, M., Picozzi, M.: End-user development of mobile mashups. In: Int. Conf. of Design, User Experience, and Usability. pp. 641–650. Springer (2013)
19. Capretz, L.F., Ahmed, F.: Making sense of software development and personality types **12**(1), 6–13 (2010)
20. Capretz, L.F., Ahmed, F.: Why do we need personality diversity in software engineering? **35**(2), 1–11 (2010)
21. Charness, N.: Aging and problem-solving performance pp. 225–259 (1985)
22. Cheng, A.S., Fleischmann, K.R.: Developing a meta-inventory of human values. In: ASIS&T. vol. 47 (2010)
23. Curumsing, M.K., Fernando, N., Abdelrazek, M., Vasa, R., Mouzakis, K., Grundy, J.: Emotion-oriented requirements engineering: A case study in developing a smart home system for the elderly **147**, 215–229 (2019)
24. Curumsing, M.K., Lopez-Lorca, A., Miller, T., Sterling, L., Vasa, R.: Viewpoint modelling with emotions: a case study **4**(2), 25–53 (2015)
25. Gila, A.R., Jaafa, J., Omar, M., Tunio, M.Z.: Impact of personality and gender diversity on software development teams' performance. In: 2014 International Conference on Computer, Communications, and Control Technology (I4CT). pp. 261–265. IEEE (2014)
26. Gouveia, V.V., Milfont, T.L., Guerra, V.M.: Functional theory of human values: Testing its content and structure hypotheses **60** (2014)
27. Grundy, J.: Human-centric software engineering for next generation cloud-and edge-based smart living applications. In: 2020 20th IEEE/ACM Int. Symposium on Cluster, Cloud and Internet Computing (CCGRID). pp. 1–10. IEEE (2020)
28. Grundy, J., Abdelrazek, M., Curumsing, M.K.: Vision: Improved development of mobile ehealth applications. In: 2018 IEEE/ACM 5th Int. Conf. on Mobile Software Engineering and Systems (MOBILESoft). pp. 219–223. IEEE (2018)

29. Grundy, J., Khalajzadeh, H., McIntosh, J., Kanij, T., Mueller, I.: Humanise: Approaches to achieve more human-centric software engineering. In: International Conference on Evaluation of Novel Approaches to Software Engineering. pp. 444–468. Springer (2020)
30. Grundy, J., Khalajzadeh, H., McIntosh, J., Kanij, T., Mueller, I.: Humanise: Approaches to achieve more human-centric software engineering. In: 15th Int. Conf. Evaluation of Novel Approaches to Software Engineering, Prague, Czech Republic, May 5–6, 2020, Revised Selected Papers 15. pp. 444–468. Springer (2021)
31. Grundy, J.C.: Impact of end user human aspects on software engineering. In: 2021 Int. Conf. on Evaluation of Novel Approaches to Software Engineering (ENASE2021). pp. 9–20 (2021)
32. Grundy, J., Grundy, J.: A survey of australian human services agency software usage. Journal of technology in human services **31**(1), 84–94 (2013)
33. Haggag, O., Haggag, S., Grundy, J., Abdelrazek, M.: Covid-19 vs social media apps: Does privacy really matter? In: 2021 IEEE/ACM 43rd Int. Conf. on Software Engineering: Software Engineering in Society (ICSE-SEIS). pp. 48–57. IEEE (2021)
34. Hidellaarachchi, D., Grundy, J., Hoda, R., Madampe, K.: The effects of human aspects on the requirements engineering process: A systematic literature review (2021)
35. Hofstede, G., Hofstede, G.J., Minkov, M.: Cultures and Organizations - Software of the Mind: Intercultural Cooperation and its Importance for Survival (3. ed.). McGraw-Hill (2010)
36. Hussain, A., Abd Razak, M.N.F., Mkpojiogu, E.O., Hamdi, M.M.F.: Ux evaluation of video streaming application with teenage users **9**(2-11), 129–131 (2017)
37. Huynh, K., Benarivo, J., Xuan, C., Sharma, G., Kang, J., Grundy, J.C., Madugalla, A.: Improving human-centric software defect evaluation, reporting, and fixing. In: 2021 IEEE Int. Conf. on Computers, Software, and Applications Conf. (COMPSAC2021), July 12-16 2021, online. IEEE (2021)
38. Izquierdo, D., Huesman, N., Serebrenik, A., Robles, G.: Openstack gender diversity report. IEEE Software **36**(1), 28–33 (2018)
39. Jefferson, L., Harvey, R.: Accommodating color blind computer users. In: Proc.8th Int. ACM SIGACCESS Conf. on Computers and accessibility. pp. 40–47 (2006)
40. Jim, A.Y., Shim, H., Wang, J., Wijaya, L.R., Xu, R., Khalajzadeh, H., Grundy, J., Kanij, T.: Improving the modelling of human-centric aspects of software systems: A case study of modelling end user age in wirefame designs. In: ENASE. pp. 68–79 (2021)
41. Kamalrudin, M., Grundy, J., Hosking, J.: Maramaai: tool support for capturing and managing consistency of multi-lingual requirements. In: 2012 Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering. pp. 326–329. IEEE (2012)
42. Khambati, A., Grundy, J., Warren, J., Hosking, J.: Model-driven development of mobile personal health care applications. In: 2008 23rd IEEE/ACM Int. Conf. on Automated Software Engineering. pp. 467–470. IEEE (2008)
43. Kim, H., Powell, M.P., Bhuyan, S.S.: Seeking medical information using mobile apps and the internet: are family caregivers different from the general public? Journal of medical systems **41**(3), 38 (2017)
44. Kumar, J.: Gamification at work: Designing engaging business software. In: International conference of design, user experience, and usability. pp. 528–537. Springer (2013)

45. LeRouge, C., Ma, J., Sneha, S., Tolle, K.: User profiles and personas in the design and development of consumer health technologies. International journal of medical informatics **82**(11), e251–e268 (2013)
46. Li, C., Yu, Y., Leckning, J., Xing, W., Fong, C., Grundy, J.C., Karolita, D., McIntosh, J., Obie, H.: A human-centric approach to building a smarter and better parking application. In: 2021 IEEE Int. Conf. on Computers, Software, and Applications Conf. (COMPSAC2021), July 12-16 2021, online. IEEE (2021)
47. Liang, H., Xue, Y.L., Ke, W., Wei, K.K., et al.: Understanding the influence of team climate on it use **11**(8), 2 (2010)
48. Longo, F., Nicoletti, L., Padovano, A.: Modeling workers' behavior: A human factors taxonomy and a fuzzy analysis in the case of industrial accidents **69**, 29–47 (2019)
49. Luy, C., Law, J., Ho, L., Matheson, R., Cai, T., Madugalla, A., Grundy, J.C.: A toolkit for building adaptive user interfaces for vision-impaired users. In: 2021 IEEE Symposium on Visual Languages and Human-centric Computing (VLHCC2021), 10-13 October, St Louis, USA. IEEE (2021)
50. Masood, M., Thigambaram, M.: The usability of mobile applications for preschoolers **197**, 1818–1826 (2015)
51. Mata-Greve, F., Johnson, M., Pullmann, M.D., Friedman, E.C., Fillipo, I.G., Comtois, K.A., Arean, P., et al.: Mental health and the perceived usability of digital mental health tools among essential workers and people unemployed due to covid-19: Cross-sectional survey study. JMIR Mental Health **8**(8), e28360 (2021)
52. McDonald, C.: IT 'professionals' are to blame for robodebt - what happened to ethics? [Online.] Available: https://ia.acs.org.au/article/2020/it–professionals–are-to-blame-for-robodebt.html
53. McIntosh, J., Du, X., Wu, Z., Truong, G., Ly, Q., How, R., Viswanathan, S., Kanij, T.: Evaluating age bias in e-commerce. In: 2021 IEEE/ACM 13th Int. Conf. on Cooperative and Human Aspects of Software Engineering (CHASE). pp. 31–40. IEEE (2021)
54. Menezes, Á., Prikladnicki, R.: Diversity in software engineering. In: Proc.11th Int. Workshop on Cooperative and Human Aspects of Software Engineering. pp. 45–48 (2018)
55. Michaels, G.M.: Colour preference according to age **35**, 79–87 (1924)
56. Miller, T., Pedell, S., Lopez-Lorca, A.A., Mendoza, A., Sterling, L., Keirnan, A.: Emotion-led modelling for people-oriented requirements engineering: the case study of emergency systems **105**, 54–71 (2015)
57. Morris, L.J., Stauffer, L.A.: A design taxonomy for eliciting customer requirements **27**(1), 557–560 (1994), 16th Annual Conf. on Computers and Industrial Engineering
58. Mosqueira-Rey, E., Alonso-Ríos, D., Moret-Bonillo, V.: Usability taxonomy and context-of-use taxonomy for usability analysis. In: 2009 IEEE Int. Conf. on Systems, Man and Cybernetics. pp. 812–817 (2009)
59. Obie, H.O., Hussain, W., Xia, X., Grundy, J., Li, L., Turhan, B., Whittle, J., Shahin, M.: A first look at human values-violation in app reviews. In: 2021 IEEE/ACM 43rd Int. Conf. on Software Engineering: Software Engineering in Society (ICSE-SEIS). pp. 29–38. IEEE (2021)
60. Pan, X.: Research of iphone application ui design based on children cognition feature. In: 2010 IEEE 11th Int. Conf. on Computer-Aided Industrial Design & Conceptual Design 1. vol. 1, pp. 293–296. IEEE (2010)
61. Parker Software: Key considerations for making age-friendly software (2019)

62. Perry, K., Shearer, E., Sylvers, P., Carlile, J., Felker, B.: mhealth 101: an introductory guide for mobile apps in clinical practice. Journal of Technology in Behavioral Science **4**(2), 162–169 (2019)
63. Pew Research Center: Internet, Science & Tec: Internet/broadband fact sheet (2019)
64. Ralph, P.: Toward methodological guidelines for process theories and taxonomies in software engineering **45**(7), 712–735 (2018)
65. Rodríguez-Pérez, G., Nadri, R., Nagappan, M.: Perceived diversity in software engineering: a systematic literature review. Empirical Software Engineering **26**(5), 1–38 (2021)
66. Rokeach, M.: The Nature of Human Values (1973)
67. Seneler, C.O., Basoglu, N., Daim, T.U.: A taxonomy for technology adoption: A human computer interaction perspective. In: PICMET '08 - 2008 Portland Int. Conf. on Management of Engineering Technology. pp. 2208–2219 (2008)
68. Shamsujjoha, M., Grundy, J., Li, L., Khalajzadeh, H., Lu, Q.: Human-centric issues in ehealth app development and usage: A preliminary assessment. In: 2021 IEEE Int. Conf. on Software Analysis, Evolution and Reengineering (SANER). pp. 506–510. IEEE (2021)
69. Silva, S., Kenney, M.: Algorithms, platforms, and ethnic bias **62**(11), 37–39 (2019)
70. Silveira, K.K., Prikladnicki, R.: A systematic mapping study of diversity in software engineering: a perspective from the agile methodologies. In: 2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE). pp. 7–10. IEEE (2019)
71. Sim, G., Shrivastava, A., Horton, M., Agarwal, S., Haasini, P.S., Kondeti, C.S., McKnight, L.: Child-generated personas to aid design across cultures. In: IFIP Conference on Human-Computer Interaction. pp. 112–131. Springer (2019)
72. Singh, J., Lutteroth, C., Wünsche, B.C.: Taxonomy of usability requirements for home telehealth systems. In: Proc. 11th Int. Conf. NZ ACM Special Interest Group on Human-Computer Interaction. p. 29–32. CHINZ '10, ACM (2010)
73. Soper, S.: Fired by bot at amazon: 'it's you against the machine'. [Online.] Available: https://www.bloomberg.com/news/features/2021-06-28/fired-by-bot-amazon-turns-to-machine-managers-and-workers-are-losing-out?sref=EJ3iffSv
74. Strengers, Y., Kennedy, J.: The smart wife: Why Siri, Alexa, and other smart home devices need a feminist reboot. MIT Press (2020)
75. Vangeepuram, N., Mayer, V., Fei, K., Hanlen-Rosado, E., Andrade, C., Wright, S., Horowitz, C.: Smartphone ownership and perspectives on health apps among a vulnerable population in east harlem, new york. Mhealth **4** (2018)
76. Vorvoreanu, M., Zhang, L., Huang, Y.H., Hilderbrand, C., Steine-Hanson, Z., Burnett, M.: From gender biases to gender-inclusive design: An empirical investigation. In: Proc.2019 CHI Conf. on Human Factors in Computing Systems. pp. 1–14 (2019)
77. (WAI), W.W.A.I.: W3c accessibility standards overview. [online]. Available: https://www.w3.org/WAI/standards-guidelines/
78. Xu, T., White, J., Kalkan, S., Gunes, H.: Investigating bias and fairness in facial expression recognition. In: European Conference on Computer Vision. pp. 506–523. Springer (2020)
79. Yan, P., Schroeder, R.: Variations in the adoption and use of mobile social apps in everyday lives in urban and rural china. Mobile Media & Communication **8**(3), 318–341 (2020)
80. Yusop, N.S.M., Grundy, J., Schneider, J.G., Vasa, R.: A revised open source usability defect classification taxonomy **128**, 106396 (2020)