Vision: Improved development of mobile eHealth applications

John Grundy
Faculty of Information Technology,
Monash University
Melbourne, Australia
john.grundy@monash.edu

Mohamed Abdelrazek
School of Information Technology,
Deakin University
Melbourne, Australia
mohamed.abdelrazek@deakin.edu.au

Maheswaree Kissoon Curumsing
School of Information Technology,
Deakin University
Melbourne, Australia
m.curumsing@deakin.edu.au

ABSTRACT

Mobile eHealth applications have become very popular, not just using mobile phones but also wearables, mobile AR/VR, and increasingly "smart houses" and "smart care" sensing and interaction facilities. However, a large majority of these solutions, despite early promise, suffer from a range of challenges including effort to develop, deploy and maintain; lack of end user acceptance; integration with other health systems; difficulty in tailoring to divergent users; lack of adequate feedback to developers; lack of sustainable adoption; and ultimately lack of success. In this MobileSoft vision paper we characterise these key issues from a Software Engineering perspective and present and discuss some approaches to mitigating them, building on our and others prior work.

CCS CONCEPTS

• Software and its engineering → Software development techniques; **• Human-centered computing → Ubiquitous and mobile devices**; **• Applied computing → Consumer health**;

KEYWORDS

mobile eHealth applications,living lab,behavioural requirements,emotion-oriented requirements,model-driven engineering,configuration and adaptation,user feedback

1 INTRODUCTION

Mobile eHealth applications have become increasingly popular in recent years [8, 27]. Many mobile phone eHealth applications have been developed including consumer advice; tracking steps, diet, activity, heart rate and other vital signs; programmes for those suffering from chronic disease e.g. heart disease or obesity; quit smoking programmes; and monitoring of elderly or those with serious mental or physical conditions [15, 27]. Various lightweight devices including wearables and pendants have provided further targeted mobile eHealth support [12]. Increasingly, augmented and virtual reality solutions leveraging mobile phones and/or tablets have been developed [24]. Smart home/care facility solutions include diverse devices, some mobile and some fixed, that, often along with mobile phones, tablets and wearables, provide "intelligent" living spaces to realise various eHealth applications [15]. Figure 1 illustrates some of these increasingly common approaches and examplar uses for eHealth.

However, the software engineering of these various mobile eHealth applications is in large part still in its infancy. In part this is due to the inherent diversity of users, domains, tasks, devices and health challenges the domain naturally presents. In part, this is due to the mis-match or limitations of current mobile application engineering approaches in addressing various challenges that present for different phases of mobile eHealth application development and deployment.

Figure 1: Some examples of mobile eHealth applications and uses

Development Process - Mobile app development has begun to adopt DevOps practices with greater incremental releases and reduced turn-around time. However, there is still a considerable lag from receiving user feedback and issues to releasing new applications. In some regards, eHealth apps are no different to others. However, given their criticality for health-related tasks – e.g. implementation of critical care plans for heart attack recovery, strict dietary and pharmacology routines, and remove monitoring for patient safety in their home – slowness to identify issues, security

MobileSoft2018, May 2018, Gothenburg, Sweden

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

and privacy vulnerabilities, correct, and re-deploy app may have very serious health consequences.

Requirements Engineering - Again, mobile eHealth apps in some regards are no different to many other mobile apps in terms of requirements elicitation, capture and validation approaches. Given the target end users of many mobile eHealth app solutions – e.g. elderly users being monitored and provided emergency alarms, obese patients being supported to improve dietary practices, and busy health workers in very challenging deployment environments – mobile eHealth apps often do have a range of very non-functional requirements for their supported features, end users and deployment that many other mobile app domains lack.

Design and Build of Mobile Apps - While some generation approaches exist for mobile apps, most are still developed via conventional means of translating requirements into implementations substantially by hand [7]. Given the challenge of mobile eHealth apps in terms of interface design, different QoS challenges, and critical care application domains, this may be argued to be required in order to implement the very challenging sorts of apps in this domain. However, this makes development and especially maintenance of mobile eHealth apps costly and slow [13].

Configuration - One approach to reducing design and development effort is to make mobile eHealth applications – where sensible and feasible – more "configurable". For example, tailoring the app interface and features and constraints to each individual user or group of users. Ideally, some mobile eHealth apps may support end user configuration themselves, either via end user configuration support or AI-based auto-adaptation e.g. monitoring "normal" behaviour or activity for each user and detecting anomalies to alert carers, users or handle differently. Current development practices to support such end user configuration and AI-based configuration are however still limited [23].

End User Feedback - In order to improve mobile eHealth apps developers need proactive, high quality feedback. Reporting some defects, such as usability defects, are still problematic in software engineering in general, let alone mobile apps in general or eHealth mobile apps [34]. These defects might negatively impact app adoption/reviews.

Sustainability of Change - While not all eHealth mobile applications are designed to "change" behaviour, a large number are, or at least aim at supporting healthy lifestyles, proactively capturing key health indicators, or providing target health information and support [11]. A recurring challenge with mobile eHealth apps is them actually working e.g. a wearable app counting activities actually helping someone carry out a desired exercise regime or a calorie counting app actually helping someone accurately track food intake and weight loss (or gain) [31]. A great many mobile eHealth apps are great in principle but fail to affect desire change, fail to be accepted by end users, or after a short time of effectiveness, their use trails off. In short, they do not work. Worse still, some are actually dangerous to users' health[32].

2 THE VISION

We envisage a more integrated approach to the development of mobile eHealth apps to address the above challenges. Figure 2 outlines such an approach and some of its key features.

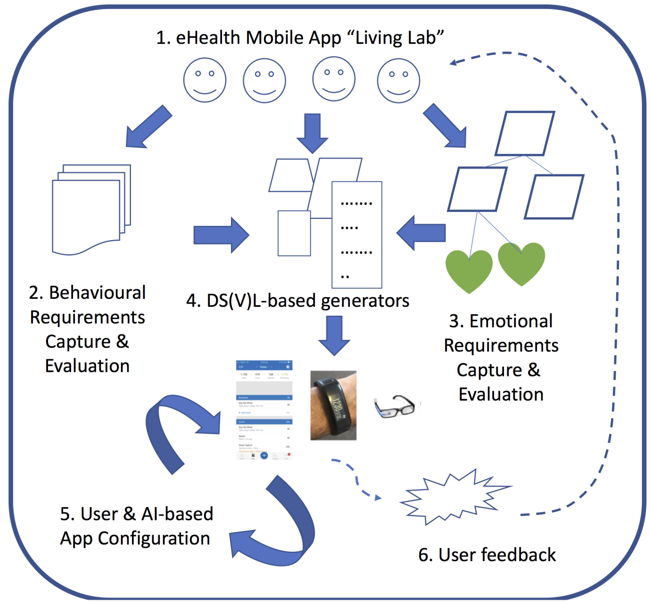

Figure 2: Our vision of an approach to improve mobile eHealth app development

2.1 Mobile eHealth App Living Lab

Continuous end user input into and refinement of requirements is needed for more effective mobile eHealth apps [5]. We envisage the use of a "living lab" concept as a joint collaboration of clinicians – since these are health-supporting apps – developers, end users, and carers for end users, whether professional or family [5]. The living lab would provide an overarching context in which apps are imagined, designed, prototyped, tested, and rolled out. This would compliment the other following innovations by providing a more effective grounding of eHealth apps in evidentiary-based clinical practices, include all stakeholders in all phases of development and deployment, and identify key negative impacting factors, whether technical or socio-technical, in a proactive way.

2.2 Behaviour Goal Modelling and Evaluation

The majority of mobile eHealth apps are intended to effect some form of behaviour change, monitor end user behaviours, reinforce behaviours, encourage more positive behaviour or discourage negative behaviour, and inform as to benefits and costs of health-related behaviours [8, 31]. Better modelling of such behaviour change goals and ensuring these goals are being met during development and during deployment would be very beneficial. We envisage the use of goal-oriented approaches to these behaviour change targets, not just during requirements engineering but also during development of the mobile eHealth apps and during their deployment and evaluation. Such deployment-time assessment of whether goals are being met by the app require enhanced user feedback, as per below. New approaches are needed to assess both behaviour change desired, the way the app supports this (or not), and the degree of change [11]. This suggests improved techniques are needed both to model these goals but also to assess them and then act on the assessment and feedback.

2.3 Emotional Goal Modelling and Evaluation

Recent work has shown the importance of considering user emotions during requirements gathering and evaluation of mobile eHealth apps, particularly those targeted to elderly users [12]. Related to affecting behaviour change above, negative emotional goals have been shown to adversely impact the acceptance and usage of mobile eHealth applications. For example, users – or their carers – perceiving loss of control, lack of safety, unpredictable behaviour or other negative emotions towards the app are serious challenges for many mobile ehealth app developers to address [15]. Similarly, positive emotions users feel towards their apps, and positive support from others when using their eHealth apps [20, 25], tend to lead to better adoption and adherence to health care plans. To better address this when engineering mobile eHealth apps requires further development of emotional impacts of the apps, both during requirements engineering but addressing these during development, during deployment and when assessing effectiveness of the apps.

2.4 Application Generation

Mobile Apps in general take a lot of effort to design and build. Many mobile eHealth Apps in particular often exhibit a number of challenging features including complex user interfaces, interfacing with different sensors, wearables and other data sources, detailed data exchange with web servers and/or hospital information systems, a variety of challenging non-functional requirements including security, privacy, performance, and usability, and increasingly various forms of machine learning or data analytics built into the app (vs processed on a server). In previous work to assist development of mobile apps we have developed model-driven approaches to generating mobile eHealth apps from domain-specific visual languages [19], and general purpose, data-driven domain-specific language-based mobile app generation [7]. Many others have also developed various platforms and frameworks for modelling and generating mobile apps to address these issues [2, 4, 21]. While promising, such approaches face major challenges in finding suitable abstractions for models, trading off generality and tailoring to domain needs, effective integration of generated apps with other (health) IT systems components, supporting multiple devices effectively, and ensuring highly usable solutions. We want to extend our earlier general mobile bootstrapping approach [7] to include domain-specific support and include incorporation of key non-functional requirements as discussed above.

2.5 Configuration and Adaptation

We have developed a number of approaches to support end user configuration of applications [14, 16, 17]. Building on this work, we envisage providing much improved support for end user configuration of mobile eHealth apps using a range of techniques. This is necessary to tailor eHealth applications to specific user needs [19], and to tailor them to usage contexts without the need for heavy weight redesign and reengineering. Supporting such approaches requires architectural, user interface, integration API, and end user domain-specific languages or by-example configuration support [10, 14, 17].

Increasingly, AI-based adaptation of mobile eHealth applications is desired. This includes learning end user behaviours in order to detect anomalies, example-based training of apps, and using big data analytics approaches on devices to provide improved interaction, detection, feedback or other features [3, 15, 22]. Using machine learning approaches to help adapt mobile eHealth applications to user profile, task, context and preferences is a promising approach to supporting (semi-)automated eHealth app configuration and adaptation. We plan on using a range of techniques to compliment the explicit configuration approaches outlined above.

2.6 Continuous Feedback and Update

The large increase in number and interest in mobile eHealth apps has resulted in considerable interest in how to evaluate these apps [9, 26, 29]. Challenges include determining if the apps actually work [31], what are suitable metrics to use and getting suitable feedback to developers [26], ensuring apps meet various privacy and regulatory standards [9], and matching end user impressions, emotions and activities with target requirements [15]. Our recent work has shown that current mobile testing techniques are still limited in both capability and usage [36], current usability defect reporting techniques have several major problems [34], and reviews of mobile apps are often not sufficient for developers to use to improve [30]. Considerable further work is needed in improving the evaluation of apps, including not only functional capabilities but emotion and other non-functional characteristics [12], improved usability defect reporting [35], and better support for quality review and defect capture [33]. This user feedback will be applied to existing apps as well as new apps developed as above.

Relating to the over-arching use of the living lab concept above, development processes that incorporate high degrees of change and continuous integration based on these reviews are also needed [18]. We previously proposed similar approaches for improved software security engineering [1] and the engineering of big data oriented software systems [28]. Along with improved model-driven engineering based app generation approaches, we believe applying such approaches to mobile eHealth apps will produce much more responsive app development and effective mobile health apps.

3 WHY IS IT NEW?

There is great interest and activity in developing many new mobile eHealth apps but as discussed, current software engineering practices are still lacking. Of particular concern are those including continuous development and deployment, capture of key requirements including behaviour change and emotion goals, efficient and effective generation of high quality apps, supporting end user and AI-assisted configuration of apps, and capturing high quality feedback to improve apps quickly and proactively.

Some techniques are currently in use for mobile eHealth app development, such as the use of living labs and various attempts at model-driven engineering of apps. However, these are still not integrated with other critical supporting approaches. For example, development methodologies for living labs, while using participatory design and other related approaches, typically move to traditional requirements refinement, app development and testing [5]. Most mobile app generators used to date show the concept has promise but have several severe limitations to date [7].

We have applied a number of concepts outlined above in other domains to date. For example, we have successfully shown that complex end user configuration of security controls and user interface elements can be supported at run-time with suitable software architectures and implementation platforms [1, 14]. We have shown in our early work on mobile eHealth app generation that abstract, high level domain-specific visual languages can be used to generate (functional) aspects of useful mobile eHealth apps [19]. We have shown that improved usability defect reports and mobile app review quality can be improved, if not for eHealth apps then for other domains [33]. However, we have not advanced or integrated these in a systematic way for mobile eHealth app development.

In this domain, we want to improve our mobile app generation techniques [7, 19] to better support user interface specification and end user configuration of user interfaces [14]. We want to better capture and incorporate key health behaviour change requirements along with key positive and negative emotional goal requirements when designing the apps with end users, carers and clinicians [15]. In addition we want to make sure these key requirements are met during development, by ensuring design models used to generate the apps incorporate required support for these requirements, and improve feedback and evaluation of how these key requirements are met during app deployment [12]. While we have made progress on some of these areas, much further work is needed, especially in terms of incorporating particular mobile eHealth app non-functional requirements during development and improving feedback and evaluation on deployed apps from end users.

As others have recognised, more continuous, agile and inclusive of end user need development approaches are needed when building mobile eHealth apps [6, 18]. We want to adapt some of our ideas for improving the software engineering of big data-oriented systems to the mobile eHealth development domain to provide better continuous development, delivery and feedback [28].

4 THE RISKS

Capturing, realising and assessing behavioural and emotional goals is hard – behavioural goal and emotional goal requirements are not commonly explicitly captured during app development. Similarly, there do not yet exist rigorous methods to ensure they are incorporated in development and evaluated during deployment.

Finding the right balance between abstraction and fine-tuned customisation is hard – our own prior work, and much other app generation work, has demonstrated the significant challenge of being able to model – at high enough levels of abstraction – all the key components of a mobile eHealth app and then generate a highly usable, efficient, integratable, and effective app solution.

Too hard to get good user feedback – our recent work on usability defect reporting has shown this is an area still needing significant further development in software engineering in general, let alone mobile eHealth app evaluation, review and feedback.

Too hard to support good end user configuration and AI-based app adaptation – while a number of attempts have been made to support device adaptation, task adaptation, understand user preferences, and support end user programming and configuration, there is considerable further research to be done to actually produce a broadly affective solution. While using AI in mobile eHealth applications has greatly increased in recent years, this also faces significant challenges in terms of engineering effort, suitable training data, appropriate usage, explaining recommendations and analyses, maintaining privacy, and managing bias.

Feedback loop too slow or lacks quality – refinement of existing eHealth apps show that there is still going to be a challenge in adapting mobile eHealth app features to target users given that users for many apps are likely to be very diverse, have a widely varying needs, and provide widely varying forms of feedback. Lack of timely and quality feedback for developers will greatly hinder take up and usefulness.

Still not sustainable or effective – these is still a risk resultant mobile eHealth apps do not achieve their intended health outcome aims. Given the target communities for many eHealth apps, we still don't fully understand many of the challenges in using, adopting and evolving these apps e.g. to support patients with early onset dementia, health works in dangerous environments, people who move between highly heterogeneous usage domains, and changing needs and requirements of the end users.

5 NEXT STEPS

Development of living lab idea – we are collaborating with our health industry partners to develop a living lab environment in which to imagine, design, prototype, evaluate and evolve mobile eHealth apps for a range of target domains, including aged care support, early onset dementia support, various forms of chronic disease, better supporting carers of patients and health professionals.

Emotion-based Development – we have been using Emotion-oriented requirements engineering and evauation approaches on several health domain applications, including several mobile eHealth applications [15]. We want to extend this to include how to incorporate such requirements during app development and testing, as well as further improve feedback and issue reporting.

App Generation – our recent work on mobile app generation has not been confined to the health domain. We want to explore greater use of domain-specific models for the mobile eHealth app domain in our generation tools to try and improve usability and performance (of task) of generated apps.

Defect Reporting – we want to apply our defect reporting approaches to eHealth apps, including improving user review creation but also detailed usability, behaviour goal, emotional goal, security and privacy goal, and other non-functional characteristics.

Continuous Development – finally, we want to improve the feedback from usage of both existing and new mobile eHealth apps with the living lab-based team to more quickly validate and improve ideas and prototypes.

6 ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of ARC Discovery grant DP170101932, Deakin University, Monash University and the Deakin Software Technology Innovation Lab.

REFERENCES

[1] Mohamed Abdelrazek, John Grundy, and Amani Ibrahim. 2017. Adaptive Security for Software Systems. In *Managing Trade-Offs in Adaptable Software Architectures*. Elsevier, 99–127.

[2] Roberto Acerbis, Aldo Bongio, Stefano Butti, and Marco Brambilla. 2015. Model-driven development of cross-platform mobile applications with webratio and ifml. In *Proceedings of the Second ACM International Conference on Mobile Software Engineering and Systems*. IEEE Press, 170–171.

[3] Mark V Albert, Konrad Kording, Megan Herrmann, and Arun Jayaraman. 2012. Fall classification by machine learning using mobile phones. *PloS one* 7, 5 (2012), e36556.

[4] Nour Ali and Carlos Solis. 2014. Mobile architectures at runtime: research challenges. In *Proceedings of the 1st International Conference on Mobile Software Engineering and Systems*. ACM, 41–44.

[5] Tariq O ANDERSEN, Jørgen P BANSLER, Finn KENSING, and Jonas MOLL. 2017. From Prototype to Product: Making Participatory Design of mHealth Commercially Viable. *Participatory Design & Health Information Technology* 233 (2017), 95.

[6] Oresti Banos, Rafael Garcia, Juan A Holgado-Terriza, Miguel Damas, Hector Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga. 2014. mHealthDroid: a novel framework for agile development of mobile health applications. In *International Workshop on Ambient Assisted Living*. Springer, 91–98.

[7] Scott Barnett, Rajesh Vasa, and John Grundy. 2015. Bootstrapping mobile app development. In *Proceedings of the 37th International Conference on Software Engineering-Volume 2*. IEEE Press, 657–660.

[8] Edwin D Boudreaux, Molly E Waring, Rashelle B Hayes, Rajani S Sadasivam, Sean Mullen, and Sherry Pagoto. 2014. Evaluating and selecting mobile health apps: strategies for healthcare providers and healthcare organizations. *Translational behavioral medicine* 4, 4 (2014), 363–371.

[9] Maged N Kamel Boulos, Ann C Brewer, Chante Karimkhani, David B Buller, and Robert P Dellavalle. 2014. Mobile medical and health apps: state of the art, concerns, regulatory control and certification. *Online journal of public health informatics* 5, 3 (2014), 229.

[10] Cinzia Cappiello, Maristella Matera, Matteo Picozzi, Alessandro Caio, and Mariano Tomas Guevara. 2012. MobiMash: end user development for mobile mashups. In *Proceedings of the 21st International Conference on World Wide Web*. ACM, 473–474.

[11] David E Conroy, Chih-Hsiang Yang, and Jaclyn P Maher. 2014. Behavior change techniques in top-ranked mobile apps for physical activity. *American journal of preventive medicine* 46, 6 (2014), 649–652.

[12] Maheswaree Kissoon Curumsing, Sonja Pedell, and Rajesh Vasa. 2014. Designing an Evaluation Tool to Measure Emotional Goals. *International Journal of People-Oriented Programming (IJPOP)* 3, 1 (2014), 22–43.

[13] Isabel de la Torre-Díez, Miguel López-Coronado, Cesar Vaca, Jesús Saez Aguado, and Carlos de Castro. 2015. Cost-utility and cost-effectiveness studies of telemedicine, electronic, and mobile health systems in the literature: a systematic review. *Telemedicine and e-Health* 21, 2 (2015), 81–85.

[14] John Grundy and John Hosking. 2002. Developing adaptable user interfaces for component-based systems. *Interacting with computers* 14, 3 (2002), 175–194.

[15] John Grundy, Kon Mouzakis, Rajesh Vasa, Andrew Cain, Maheswaree Curumsing, Mohamed Abdelrazek, and Niroshine Fernando. 2018. Supporting Diverse Challenges of Ageing with Digital Enhanced Living Solutions. In *Studies in Health Technology and Information series*. IOS Press.

[16] John Grundy and Wenjing Zou. 2003. AUIT: Adaptable user interface technology, with extended Java Server pages. *Multiple user interfaces: Cross-platform applications and context-aware interfaces* (2003), 149–167.

[17] John C Grundy, John Hosking, Karen Na Li, Norhayati Mohd Ali, Jun Huh, and Richard Lei Li. 2013. Generating domain-specific visual language tools from abstract visual specifications. *IEEE Transactions on Software Engineering* 39, 4 (2013), 487–515.

[18] Mona Erfani Joorabchi, Ali Mesbah, and Philippe Kruchten. 2013. Real challenges in mobile app development. In *Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on*. IEEE, 15–24.

[19] Abizer Khambati, Jim Warren, John Grundy, and John Hosking. 2009. A model driven approach to care planning systems for consumer engagement in chronic disease management. *electronic Journal of Health Informatics* 4, 1 (2009), 3.

[20] Maheswaree Kissoon Curumsing. 2017. *Emotion-Oriented Requirements Engineering*. PhD dissertation. Swinburne University of Technology.

[21] Chang Liu, Qing Zhu, Kenneth A Holroyd, and Elizabeth K Seng. 2011. Status and trends of mobile-health applications for iOS devices: A developer's perspective. *Journal of Systems and Software* 84, 11 (2011), 2022–2033.

[22] Brent Longstaff, Sasank Reddy, and Deborah Estrin. 2010. Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In *2010 4th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth),*. IEEE, 1–7.

[23] David D Luxton, Jennifer D June, and Samantha A Chalker. 2015. Mobile health technologies for suicide prevention: feature review and recommendations for use in clinical care. *Current Treatment Options in Psychiatry* 2, 4 (2015), 349–362.

[24] Stefan Mitrasinovic, Elvis Camacho, Nirali Trivedi, Julia Logan, Colson Campbell, Robert Zilinyi, Bryan Lieber, Eliza Bruce, Blake Taylor, David Martineau, et al. 2015. Clinical and surgical applications of smart glasses. *Technology and Health Care* 23, 4 (2015), 381–401.

[25] David C Mohr, Pim Cuijpers, and Kenneth Lehman. 2011. Supportive accountability: a model for providing human support to enhance adherence to eHealth interventions. *Journal of medical Internet research* 13, 1 (2011).

[26] Suhaib Mujahid, Giancarlo Sierra, Rabe Abdalkareem, Emad Shihab, and Weiyi Shang. 2017. Examining user complaints of wearable apps: a case study on Android wear. In *Proceedings of the 4th International Conference on Mobile Software Engineering and Systems*. IEEE Press, 96–99.

[27] Jonathan C Rawstorn, Nicholas Gant, Andrew Meads, Ian Warren, and Ralph Maddison. 2016. Remotely delivered exercise-based cardiac rehabilitation: design and content development of a novel mHealth platform. *JMIR mHealth and uHealth* 4, 2 (2016).

[28] Mark Staples, Liming Zhu, and John Grundy. 2016. Continuous validation for data analytics systems. In *Software Engineering Companion (ICSE-C), IEEE/ACM International Conference on*. IEEE, 769–772.

[29] Stoyan R Stoyanov, Leanne Hides, David J Kavanagh, Oksana Zelenko, Dian Tjondronegoro, and Madhavan Mani. 2015. Mobile app rating scale: a new tool for assessing the quality of health mobile apps. *JMIR mHealth and uHealth* 3, 1 (2015).

[30] Rajesh Vasa, Leonard Hoon, Kon Mouzakis, and Akihiro Noguchi. 2012. A preliminary analysis of mobile app user reviews. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*. ACM, 241–244.

[31] Ronald S Weinstein, Ana Maria Lopez, Bellal A Joseph, Kristine A Erps, Michael Holcomb, Gail P Barker, and Elizabeth A Krupinski. 2014. Telemedicine, telehealth, and mobile health applications that work: opportunities and barriers. *The American journal of medicine* 127, 3 (2014), 183–187.

[32] Y Tony Yang and Ross D Silverman. 2014. Mobile health applications: the patchwork of legal and liability issues suggests strategies to improve oversight. *Health Affairs* 33, 2 (2014), 222–227.

[33] Nor Shahida Mohamad Yusop, John Grundy, and Rajesh Vasa. 2015. Reporting Usability Defects: Limitations of Open Source Defect Repositories and Suggestions for Improvement. In *Proceedings of the ASWEC 2015 24th Australasian Software Engineering Conference*. ACM, 38–43.

[34] 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*. ACM, 38.

[35] Nor Shahida Mohamad Yusop, John Grundy, and Rajesh Vasa. 2017. Reporting Usability Defects: A Systematic Literature Review. *IEEE Transactions on Software Engineering* 43, 9 (2017), 848–867.

[36] Samer Zein, Norsaremah Salleh, and John Grundy. 2016. A systematic mapping study of mobile application testing techniques. *Journal of Systems and Software* 117 (2016), 334–356.