Towards a Connected Health Delivery Framework

Craig Kuziemsky
Telfer School of Management,
University of Ottawa, Canada
Kuziemsky@telfer.uottawa.ca

Raja Manzar Abbas
Lero - the Irish Software Research
Centre, University of Limerick, Ireland
raja.abbas@lero.ie

Noel Carroll
Lero - the Irish Software Research
Centre, NUI, Galway, Ireland
noel.carroll@nuigalway.ie

ABSTRACT

The field of Connected Health (CH) has emerged in response to growing demands for improved connectivity to deliver patient-centred care through innovative healthcare management systems. As a result, there is a growing need to develop new healthcare delivery models and new ways of thinking about how information and communication technology can support healthcare delivery. This paper examines the role of software engineering in CH and presents the CH Delivery Framework as a first step to encapsulate key factors software engineers need to consider.

CCS CONCEPTS

• **Applied computing→ Health informatics**

KEYWORDS

Software Engineering, Health Information Technology, Connected Health, Healthcare Transformation

ACM Reference format:

TBC

1 INTRODUCTION

On January 13, 2018, an employee at the Hawaii Emergency Management Agency employee selected the wrong menu option and inadvertently set off an alert that incoming missiles were coming. The situation caused subsequent panic due to a 38-minute gap between the initial alert and a subsequent wireless alert stating the missile warning was a mistake. The event begs some key questions such as:

1. How were such critical yet very distinct items residing in the same menu? - **A "hot spot" for user error**
2. Why were there no reasonable safeguards?
3. Why was no system in place for rapid retraction?

While the incident received considerable attention for the distress it caused, it should be noted that similar human factor errors are common in healthcare. Selecting the wrong patient or the wrong dose is a common occurrence in healthcare and has led to fates worse than a false alarm. Patients have been killed because of such errors and unintended consequences (UICs) post Health information technology (HIT) implementation are very common and well described [1, 2]. Part of the reason for the emergence of UICs is that the degree of technology facilitated connectivity has increased and thus there are more potential interactions for outcomes (both good and bad) to occur. The emergency management example showed how small degrees of connectivity can have very significant consequences and that we need to understand the nature of these interactions to enable us to better engineer software to overcome them.

Communities such as '*Software Engineering in Healthcare Systems (SEHS)*', has emphasized the substantial role software engineering (SE) will play in the transformation of healthcare systems into collaborative patient centered systems. However, the actual design and evaluation of software systems remains problematic. HIT are frequently unsuitable for the tasks they have been designed for leading to numerous unintended consequences including workflow, communication and safety issues [3].

Weber and colleagues have proposed a knowledge translation (KT) framework for SE in healthcare, based on the premise that SE principles and knowledge cannot just be applied to healthcare, but rather must be translated to fit the unique context of healthcare delivery [4]. They also argue that despite frameworks and methods for SE in healthcare, we still struggle with the on-the-ground implementation of actual technologies.

As a research community, we have placed considerable effort in designing healthcare delivery frameworks at the macro system level but we need to shift our focus on understanding and designing for individual tasks at the micro level. To that end we heed the call of Weber and colleagues where they state that the SE in healthcare community must focus on understanding context and how to adapt SE methods to best enable the design of systems to support healthcare transformation [4].

This paper begins to address the above challenge by looking at Connected Health (CH) delivery and the design of software to support it. Section 2 reviews the relationship between SE and safe HIT. Section 3 describes CH delivery and the challenges in

SEHS'18, May 27, 2018, Gothenburg, Sweden
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5734-0/18/05…$15.00
https://doi.org/10.1145/3194696.3194703

providing it. Section 4 integrates sections 2 and 3 by introducing a framework for SE for CH.

2 SOFTWARE ENGINEERING FOR SAFER HIT

Care delivery models such as chronic disease management and community based care delivery are based on the need to connect various touchpoints across the continuum of care. Software plays a significant role in facilitating connected health at the micro level (e.g. EMRs and other clinical HIT), the meso level (tools to support collaboration) and the macro level (e.g. analytics and performance management applications). Historically these touchpoints have been a source of safety, workflow and usability issues [5]. HIT can be certified and still not work as intended or introduce safety. To address these issues we need to rethink how we design software for use in healthcare settings. Sittig et al. describe how addressing safety issues in electronic health records (EHR) is a shared responsibility between technical and non-technical agencies [6]. They make an analogy to the aviation industry where airline safety has been managed as a shared responsibility between personnel (e.g. pilots, aircraft manufacturers, government, and policy developers), the developers of safety standards, and the reporting, investigation and dissemination of aviation safety issues and adverse events.

Technology alone will not solve all healthcare issues and in fact poorly engineered systems will make safety and other issues more prominent. Case in point is the increased implementation of HIT that occurred in response to the 1999 "To Err is Human" report that outlined a significant occurrence of patient deaths occurring from medical errors [7] . While increased adoption of HIT such as EHR and Computer Provider Order Entry systems reduced the occurrence of known errors they also led to the emergence of a new category of technology induced errors [8]. This emphasizes that SE for safer HIT cannot just look at the design and implementation of software for an isolated need, user task, or moment in time, but rather must be an interdisciplinary endeavor that identifies and considers multiple interactions over time.

3 CONNECTED HEALTH (CH)

Connected Health (CH) is described by Carroll et al., as: "a patient or consumer-centered socio-technical healthcare management model which exploits the use of ICT during clinical or wellness decision-making tasks." [12]. CH facilitates the connectivity of information sources and processes across the full extent of the health care system, for example from the patient's home, to the clinician's surgery, and into out-patient support centers. Information and communications technology (ICT) development is a key part of CH delivery [9].

CH delivery is a complex social-technical model that brings several implementation challenges. One such challenge is that CH software is a hybrid of many different types of systems. Medical software, a subset of CH technologies, is subject to Software-as-a-Medical Device (SOMD) regulation [10]. However, other software may not be classified as SOMD and thus not subjected to the same regulation, a second challenge is that ICT spending within healthcare has been significantly lower than many other sectors, resulting in low quality software systems, requiring healthcare professionals to implement inefficient processes and practices. Healthcare professionals unquestioningly place their trust in (often inaccurate and unreliable) software [11, 12]. Finally, healthcare connectivity is not linear or static, but rather is a complex system with many interrelated concepts [13].

Designing for complex CH systems introduces challenges such as dynamic processes (e.g. collaborative care delivery) that evolve over time, presenting with dynamic requirements and evaluation metrics [13]. CH software design needs to incorporate this complexity into systems design as much as possible.

4 SOFTWARE ENGINEERING FOR CH

Weber et al. suggest that we need to understand how to translate general SE research into healthcare [4]. While they describe numerous healthcare issues (e.g. ill-defined data and workflows, verification and validation, evolving processes) that present SE challenges, they do not describe the integration of the issues as part of SE for a particular context. CH presents a unique challenge in that the interaction of people, processes and technology in specific CH contexts will define how software needs to be engineered and subsequently the analytics we need to measure and manage CH delivery models. As a first step for operationalizing SE to support CH delivery we developed the 'Connected Health Delivery Framework' (Figure 1).

The main contribution from our framework is that it identifies pain points, business model development, analytics, and evaluation as four main linkages between users (e.g. patients and providers) and technology. As described earlier, CH is a patient centered social-technical healthcare model that uses technology to support healthcare processes as part of continuous care delivery. Traditional SE approaches based on predefined hierarchies and/or structured workflows are not suitable for modeling interactive domains like CH because the degree of connectivity must be defined prior to engineering software to support it. Our framework and the four linkage points provides an approach to define the context of a CH domain in order to support SE tasks. Below we describe each of the four linkage points and how design thinking provides an interdisciplinary way of integrating the four points.

'Pain points' refer to the need to understand and define the context of a CH system prior to designing software to support the system. Examples of pain points are the cardinality of a CH system. Software to connect a primary care provider with his/her patients in a single setting is very different than software to connect multiple patients and providers in a team based care delivery model such as community based care delivery. The former is a 1:1 type of connectivity while the latter is a Many: Many type of connectivity. Other examples of contextual pain points that need to be identified include variations in workflow (e.g. synchronous or asynchronous), data entry and retrieval (e.g. structured or free text), and how communication and collaboration occurs.

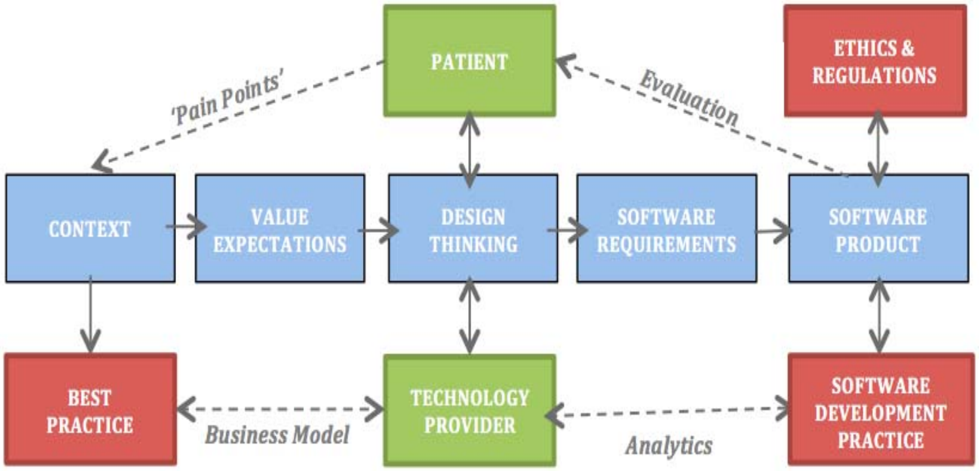

Figure 1: Connected Health Delivery Framework

There are two main concepts that pain points need to identify. First is the number and extent of connectivity in the CH system. The more connections that exist the more complex the degree of connectivity that needs to be engineered. From a SE perspective, pain points represent hazards that must be addressed by system requirements [14]. Second, pain points must identify any system discrepancies between users, processes or policies. For example, individual user workflows may have to change as part of moving to collaborative care workflows and such workflow changes are not always well received [1]. These discrepancies must be identified and reconciled prior to designing software to automate the transformed workflow.

The context that is defined via the pain points is then used as the basis for developing a business model to link existing CH best practices, the unique CH context that was identified in the previous step, and existing SE practices. Business model development can draw upon Business Process Management (BPM) approaches, particularly Social BPM approaches that enable the understanding of connected communities and how emerging and/or dynamic connectivity is established over time [15]. A fundamental role of the business model is to reconcile the pain points that were identified in the above step. For example, reconciling the issue of transitioning from individual to collaborative workflows requires developing common ground across the users, processes and technology that will be used to support collaboration [5]. Individual processes may have to be adapted and/or trade-offs may have to be made in the movement to collaborative teamwork [1]. While the business model defines the linkage between the context, best practices, technology and users, it must be remembered that CH systems are dynamic and thus the business model will need to be revised as processes or user needs evolve.

Once a contextually grounded business model has been developed, the next step is to identify relevant metrics to define the analytics for monitoring how the software supports the connected health practices. Defining metrics for CH delivery is challenging because many CH processes, for example, collaborative care delivery, are dynamic and will develop over time. Dynamic metrics will need to be developed to measure and monitor the development of collaborative processes. For example, metrics can be developed to measure the development of the aforementioned common ground among healthcare team members. This metric will measure the current state of common ground that a team has in order to identify where SE applications need to be developed to support common ground development.

Evaluation links the software product to the CH system defined by the business model and should draw upon formal quality management system standards (i.e. ISO 13485) and standards for medical device software life cycle processes (i.e. IEC 62304). Evaluation also must close the loop on the 'pain-point' hazards that were identified in the first stage of the framework to make sure the software product addresses the pain points. Evaluation needs to integrate the multiple perspectives (e.g. clinical, business, political, and technological) that exist within a CH system while also demonstrating the social and economic value of a software product designed to support CH delivery. Finally, CH solutions are iterative in nature and the design and evaluation of them are ongoing [16].

The central point to our framework is the use of the Design Thinking approach to understand the relationship between and explorative interplay between people, processes, technology and business needs [18]. Design Thinking complements traditional systems and software requirements engineering techniques for requirements analysis by encouraging the generation of innovative approaches to solving problems at the point where systems and SE techniques jointly move towards the software development stage. Design thinking provides an interdisciplinary approach for defining the four linkage points in our framework to understand the complexity of interactions within a CH system by providing an in-depth understanding of healthcare stakeholders "realities" in context of their day-to-day experiences as part of a pre-software requirements phase. Thus, rather than focusing on technology solutions in the early stages of the process, we become more immersed in the interactive problem space of a CH system.

Trust in technology influences use or adoption of a technology [12] and enforces the need for trustworthy software models in CH. Others have advocated using Design Thinking to build trust between healthcare professionals and healthcare technology providers in order to better enable the adoption and use of technology [17, 18].

5 DISCUSSION AND CONCLUSION

Healthcare delivery is being transformed into a connected system that personalizes, tracks, and manages patient information across people, places and technologies. Software engineering (SE) is playing a fundamental role in this transformation by developing software applications that enable safer and more effective, efficient, equitable, and user-centered services across multiple providers and settings. This paper presented the CH Delivery Framework as a first step to identify CH requirements and how SE approaches can operationalize the requirements. We identified pain points, business model development, analytics, and evaluation as four main linkages between users (e.g. patients and providers) and technology We also described the growing need to establish trustworthy models and analytics to evaluate and ensure CH solutions deliver safe and quality care across multiple providers and settings. Overall, CH provides a novel and multidisciplinary opportunity to support health systems transformation from a healthcare service, technology (software and hardware), data analytics, social science, and management context.

For future research, SE will play a significant role in the design and evaluation of a CH system to support innovative care delivery models such as collaborative and patient centered care delivery. Our CH Delivery Framework provides initial guidance on how to translate SE practices into the healthcare domain.

ACKNOWLEDGEMENTS

CK acknowledges funding from a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada. We also acknowledge financial support of the Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

REFERENCES

1. Kuziemsky, C.E., R. Randell, and E.M. Borycki, *Understanding Unintended Consequences and Health Information Technology:. Contribution from the IMIA Organizational and Social Issues Working Group.* Yearb Med Inform, 2016(1): p. 53-60.
2. Ash, J.S., M. Berg, and E. Coiera, *Some unintended consequences of information technology in health care: the nature of patient care information system-related errors.* Journal of the American Medical Informatics Association, 2004. **11**(2): p. 104-112.
3. Committee on Patient, S., T. Health Information, and M. Institute of, in *Health IT and Patient Safety: Building Safer Systems for Better Care*. 2011, National Academies Press (US). Copyright 2012 by the National Academy of Sciences. All rights reserved.: Washington (DC).
4. Weber-Jahnke, J.H., M. Price, and J. Williams. *Software engineering in health care: Is it really different? And how to gain impact*. in *2013 5th International Workshop on Software Engineering in Health Care (SEHC)*. 2013.
5. Kuziemsky, C.E., J. Weber-Jahnke, and J. Williams. *Engineering the healthcare collaboration space*. in *2012 4th International Workshop on Software Engineering in Health Care (SEHC)*. 2012.
6. Sittig, D.F., E. Belmont, and H. Singh, *Improving the safety of health information technology requires shared responsibility: It is time we all step up.* Healthcare, 2017.
7. Institute of Medicine Committee on Quality of Health Care in, A., in *To Err is Human: Building a Safer Health System*, L.T. Kohn, J.M. Corrigan, and M.S. Donaldson, Editors. 2000, National Academies Press (US). Copyright 2000 by the National Academy of Sciences. All rights reserved.: Washington (DC).
8. *Borycki, E., Dexheimer, J., Cossio, C. H. L., Gong, Y., Jensen, S., Kaipio, J., Kennebeck, S., Kirkendall, E., Kushniruk, A. & Kuziemsky, C. 2016. Methods for Addressing Technology-induced Errors: The Current State. Yearbook of medical informatics, 30.*
9. Carroll, N., M. Travers, and I. Richardson. *Connecting Multistakeholder Analysis Across Connected Health Solutions*. in *International Joint Conference on Biomedical Engineering Systems and Technologies*. 2016. Springer.
10. Carroll, N. and I. Richardson, *Software-as-a-medical device: demystifying connected health regulations.* Journal of Systems and Information Technology, 2016. **18**(2): p. 186-215.
11. Richardson, I., L. Reid, and P. O'Leary, *Medical Software Quality: Ensuring Correctness in the Clinical Environment.*
12. Abbas, R.M., Carroll, N., Richardson, I. and Beecham, S., 2017, February. The Need for Trustworthiness Models in Healthcare Software Solutions. In HEALTHINF (pp. 451-456).
13. Kuziemsky, C., *Decision-making in healthcare as a complex adaptive system.* Healthcare Management Forum, 2016. **29**(1): p. 4-7.
14. Weber, J.H. and C. Kuziemsky. *Towards Continuous Certification of Clinical Information Systems*. 2017. Cham: Springer International Publishing.
15. Russell, N. and A. Barros. *Business Processes in Connected Communities*. in *Business Process Management Workshops*. 2015. Cham: Springer International Publishing.
16. O'leary, P., Carroll, N., Clarke, P. and Richardson, I., 2015, October. Untangling the complexity of connected health evaluations. In Healthcare Informatics (ICHI), 2015 International Conference on (pp. 272-281). IEEE.
17. RM Abbas, N.C., I Richardson, S Beecham, *Trust Factors in Healthcare Technology: A Healthcare Professional Perspective*, in *HEALTHINF*. 2018: Madeira, Portugal.
18. Carroll, N., *Key Success Factors for Smart and Connected Health Software Solutions.* Computer, 2016. **49**(11): p. 22-28.