Digital Addiction: a Requirements Engineering Perspective

Amen Alrobai, Keith Phalp and Raian Ali

Bournemouth University, UK
{aalrobai, kphalp, rali}@bournemouth.ac.uk

Abstract. [**Context and motivation**] Digital Addiction, (hereafter referred to as DA), has become a serious issue that has a diversity of socio-economic side effects. [**Question/problem**] In spite of its high importance, DA got little recognition or guidance as to how software engineering should take it into account. This is in stark contrast to other domains known for traditional addiction (e.g., drugs, gambling, and alcohol) in which there are clear rules and policies on how to manufacture, market and sell the products. [**Principal ideas/results**] In this position paper, we suggest that software engineering in general and requirements engineering in particular need to consider DA as a first class concept in developing software systems. [**Contribution**] As an early step in this area, we conduct an empirical investigation of DA by reviewing the literature and analysing web discussion forums on the topic and use that to design a mind-map of its main causes. We also provide a basic model to articulate the DA problem from requirements perspective and elaborate research challenges for a future work.

Keywords: Digital Addiction, Requirements Engineering

1 Introduction

Digital Addiction (DA) can be described as a significant degree of dependent behaviour that is triggered and facilitated by software products. It can lead to both pleasure and relief of discomfort, but unfortunately, in a way that can harm a person socially, physically and psychologically. However, despite its impact on society, DA is still considered outside the boundary of the software engineering community. That is, unlike the situation with drugs or alcohol, software engineering has, so far, not been charged with the responsibility for dealing with or mitigating the effects of DA.

DA is still seen as a problem on the user's side, rather than the responsibility of the software or the software developers. Hence, the problem of DA is typically articulated in a way that makes the solution entirely within the domain of other disciplines, such as psychology, sociology and health care. For example, Beard [1] highlighted different factors related to the content, style of use and activity. Widyanto and Griffiths [2] emphasized the addiction 'on' rather than 'to' the Internet. As such, the Internet is treated as a single entity, without considering the features of the applications used, the way they are designed or the goals and values they help to achieve. Similarly, software is still seen, implicitly, as just a medium in which its requirements, features, values and design are not studied as primary causes of DA. In contrast, this paper suggests that the study of these factors inherently belongs to the early stages of developing software; namely requirements engineering. DA strongly relates to the requirements of users in the first place. People use software as a means to reach certain requirements, however, while doing so, they may get addicted.

There are a variety of different perspectives and debates on DA. While some works view it as a mental disorder, others believe that it is no more than a personal choice [2]. Similar debate could be found for tradition addiction [3]. However, individuals' exposure to technology advancements has led to patterns of use that seem to match the criterion of Diagnostic and Statistical Manual of Mental Disorders (DSM). Therefore, the American Psychiatric Association has added this type of addiction as an Appendix in DSM-5, which is the latest version of DSM. Such a debate in the application domain is not new to the world of software engineering, e.g., we still lack consensus in a variety of domains, such as Green Computing, Digital Citizenship and Agent Computing. For this reason, we would encourage approaches that do not interfere with the decision-making about DA, but rather provide tools and platforms to facilitate taking those decisions effectively.

DA has roots in the software design, the requirements or goals for which this software is being used and the context of use. Other latent causes relate to the personal, physiological and mental characteristics of the user, over which we, as software engineers, have little control. However, we may still aim to accommodate these factors in the design of software, similar to disciplines like design for accessibility and universal design [4]. We contend that software engineering is required to attempt to provide ways to develop software that do not lead to addiction and to accommodate users who are genuinely vulnerable to addictive behaviour.

DA raises new challenges to software engineering in general and requirements engineering in particular. This paper argues for novel approaches, which are able to cater for the diversity, subjectivity and also the private nature of information related to DA. We review the literature and analyse a discussion thread in a set of online forum discussions about DA and present a mind-map articulating the causes of DA. Some of these causes are within a software engineering remit. Finally, we use our findings to suggest a foundation, or baseline ontology for DA, and propose areas of research on DA for the software and requirements engineering community.

2 Empirical Investigation of DA

There are already existing studies on sub-areas of DA (internet addiction [2] and game addiction [5,6]) which focus on the perception of users and those user characteristics which lead to DA. However, crucially, these studies do not focus on the peculiarities of the object on which DA is centred, i.e., the software.

This lack of consideration of the software motivated us to carry out our own investigation by reviewing the literature to identify those factors that appeared to lead to DA and then to analyse discussion forums on DA which we found in widely accessed and well-reputed websites to validate and enhance our initial findings. In doing so, we identified a range of factors and then classified them under five main categories, namely; software-mediated activity, attractiveness, personal, cultural and situational. The last three categories are directly related to qualities of the software while the personal and cultural dimensions are factors that would fit studies in psychology and sociology. Our findings are summarized in Figure1.

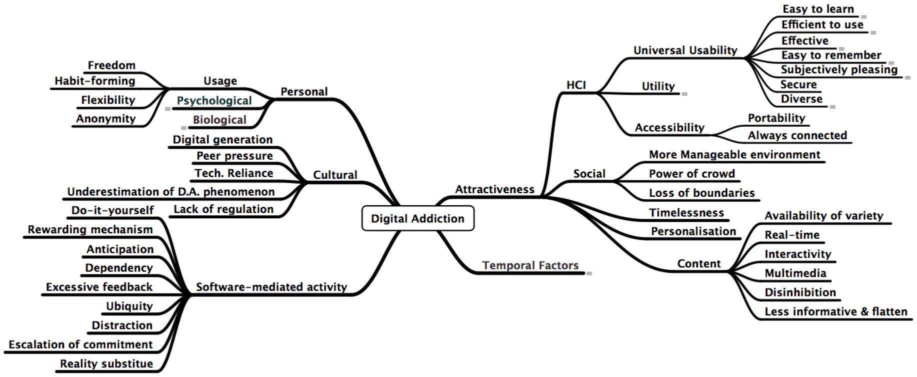

Fig. 1. A mind-map for Digital Addiction

This mind-map merely provides answers for the“what” question, that is, what has an impact on DA? Ultimately we might hope that studying user experience (UX) could, ideally, provide insights on the “why” question. Several studies, e.g., [7,8,9,10], showed that user experience is not negatively affected even when social software such as YouTube, Facebook, Wikipedia have poor compliance to usability principles [11]. Therefore, to understand the true nature of DA, the broader scope of UX may need to incorporate not only the “felt experience” such as “pleasure, curiosity, and self- expression”, but also what users gain, rightly or wrongly, from particular behaviours. To some extent, we could see these users as using the ‘addictive’ behaviour to satisfy some ‘internal’ requirement (an aspect which we explore more fully below). Hence, it may not be enough simply to describe the associations among aspects of the software and specific, possibly addictive, behaviours, but rather to understand the nature of the satisfaction and how it relates to user’s internal and private requirements, and their individual values.

However, whilst ultimately wanting to reach such rich understanding, a more pragmatic, or medium term, view would be that, even should we not be able to understand fully, say the psychological reasons for why certain features appear to trigger or exacerbate particular addictive behaviours, we could still learn which features have those impacts. That is, from a behavioural perspective, we should be able to learn to produce software products that are less likely to stimulate addictive behaviours. Hence, in having such an engineering goal, we turn again to consideration of DA from a requirements perspective.

3 Digital Addiction: a Requirements Perspective

There is a wide debate on the meaning of DA [2]. In brief there is a general agreement on the existence of the phenomenon but different viewpoints on its nature. In order to make practical progress, for a discipline like requirements engineering, we suggest a working definition. Hence, we take the initiative here and define DA from a requirements perspective as:

Digital Addiction is the excessive use of certain software-mediated operations to reach certain requirements. This includes the case when the use itself is compulsive or impulsive and also the case when the user cannot switch to other available alternatives to reach the same requirements without a good reason.

Requirements Engineering is the natural place in which users' goals and values are captured and analysed. Users' goals and values are different in that the values are 'cognitive representations' of the goals and are able to sustain users' positive emotion towards a software design [12]. On the other hand, goals are the explicit requirements that users can express. In terms of DA, values are very hard to identify due to their private nature. For example, increasing the number of followers in a social network, e.g., Twitter, is the explicit *goal* for some users, while raising the reputation in the virtual community is the latent *value.* If we can validate such values, we might help users to switch to another goal and/or alternative software design, perhaps less addictive, as long as it still can satisfy that value. Understanding what and how to do to achieve that is a requirements challenge in the first place.

A further challenge is whether we can help stakeholders to articulate these hidden requirements. Conventional elicitation methods, such as workshops, interviews and focus groups, suffer from a threat to validity when used for DA. This is due to the private nature and the tacit nature of users' values. To handle this, we suggest exploiting techniques that enable stakeholders, say addicts, to communicate through a lifelong collaborative and social activity, e.g. designated forums, and facilitate capturing this knowledge at runtime. Techniques like Crowdsourcing [13] used in the context of obtaining knowledge about software, as in [14], could be promising here.

One might argue that design time surveys can provide similar results. However, these requirements are dynamic as users' interactions with the system evolve with time, e.g., due to changes in different factors including users' familiarity with the software, the competitive technology or peer pressure. Hence, addiction not only arises from software features, but also depends on the interaction with the software in a particular context (technical, environmental and social). The fact that DA is both dynamic and context dependent makes it necessary to have more novel elicitation technique to sustain the validity of elicited DA knowledge. Software could utilize that knowledge from addicts at runtime and use it to switch to a behaviour shown by the users to be less-addictive or addiction-free. Such adaptation is called Social Adaptation [15] and it aims to harness the "wisdom of crowd" [16] in the context of software adaptation.

Social networking websites provide a wide range of features that have distinct functional traits such as tagging, likes, notifications, walls, and new features will continue to emerge. Kietzmann et al. [17] presented the Honeycomb framework as an attempt to define social media based on the peculiarity of their activities. The framework consists of seven functional blocks, identity, conversations, sharing, presence, relationships, reputation and groups. This contribution aimed to help firms to understand the engagement needs of their audiences. Such approach could provide a starting point and help to analyse software features, mainly social features, based on the addiction aspects (see Figure 1) the users' engagement requirements and values.

In our preliminary suggested approach, we first create links between users' values and requirements and analyse the software features against the mind-map factors. This analysis can be done individually, by user, or collectively through designated social platform or community of interests. The challenge is on how to adapt and provide users with alternative feature configuration that are less addictive whilst at the same time maintain users' values and requirements. We can view this as a particular kind of Dynamic Software Product Lines [18] where the addictive aspects of features, as shown by users feedback or patterns of use, could be the driver for adaptation. Figure 2 shows the meta-model which contains the main concepts of our suggested direction.

We emphasise that there is a fine line between a commitment to a task or a high level of satisfaction with software and DA. As requirements engineering we cannot, and perhaps should not, impose our definition of an addictive behaviour. Hence, we advocate that users, individually or in groups, provide and update that knowledge. Developing that user-led knowledge elicitation is an obviously challenging problem.

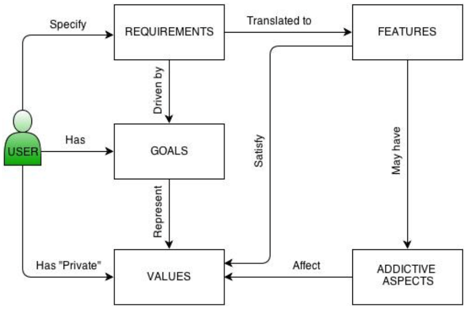

Fig. 2. DA from Requirements Perspective

4 Challenges and Future Work

Having articulated a vision for Requirements Engineering accommodating DA, we note that there are still a number of significant challenges, notably, those outlined below.

– Diversity of both Software and Users: both product features and user diversity in terms of their needs and patterns of use justify the need for enabling users to act as modellers to express personal perception toward software. A key challenge is to develop the social platform to act as a communication channel so we understand better addiction sources and stimuli within the different users groups and software features.
– Elicitation: DA relates heavily to users perceptions, expectations and personal requirements which are not easy to express in words for most users, i.e. tacit [19], fuzzy in nature and also very sensitive and private. This maximizes the challenge of capturing DA knowledge even via Crowdsourcing.
– We have identified some addictive aspects of different social software products. Questions remain as to how, or even whether, users would like to be aware of DA when they have it, what decisions would be taken by software and what other decisions are to be taken by users when the software is running? This introduces also ethical and legal issue on the accountability and responsibility of software, developers, and users.
– From a developer's (business) perspective, user's satisfaction should not be compromised. Thus, how can we, as software engineers, regulate addiction or even prevent it without affecting negatively users' experience?
– Our work focuses on the software as a core entity within DA. However, complementary work should be conducted to look at a particular set of personality traits that make individuals predisposed to DA.

5 Conclusions

Digital Addiction (DA) is as a growing and important societal issue. In this paper we suggest the need to recognise DA, and our responsibilities for it within Requirements Engineering (RE). We also suggest some potential approaches to incorporating consideration of DA within RE and articulate key challenges for DA. We expect a multidisciplinary research to address the many diverse aspects of DA aiming to develop addiction-aware software. Our future work will investigate approaches which allow not only experts but also users to contribute knowledge on the addictive aspects of a software. We anticipate that this will lead to a more holistic view of the reasons and potential treatment of DA.

Acknowledgement

The research was supported by an FP7 Marie Curie CIG grant (the SOCIAD Project) and by Bournemouth University through the Fusion Investment Fund (the BBB and the VolaComp projects) and the Graduate School PGR Development Fund.

References

1. Beard, K.W., Yarnall, C.: Internet addiction in children and adolescents. Computer science research trends (2008) 59–70
2. Widyanto, L., Griffiths, M.: internet addiction: a critical review. International Journal of Mental Health and Addiction **4**(1) (2006) 31–51
3. Davies, J.B.: Myth of Addiction. Psychology Press (1997)
4. Story, M.F.: Maximizing usability: the principles of universal design. Assistive technology **10**(1) (1998) 4–12
5. Schüll, N.D.: Addiction by Design: Machine Gambling in Las Vegas. Princeton University Press (2012)
6. Park, S., Hwang, H.S.: Understanding online game addiction: Connection between presence and flow. In: Human-Computer Interaction. Interacting in Various Application Domains. Springer (2009) 378–386
7. Hart, J., Ridley, C., Taher, F., Sas, C., Dix, A.: Exploring the facebook experience: a new approach to usability. In: Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges, ACM (2008) 471–474
8. McCarthy, J., Wright, P.: Technology as experience. interactions **11**(5) (2004) 42–43
9. Silva, P.A., Dix, A.: Usability: not as we know it! In: Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI... but not as we know it-Volume 2, British Computer Society (2007) 103–106
10. Thompson, A.J., Kemp, E.A.: Web 2.0: extending the framework for heuristic evaluation. In: Proceedings of the 10th international conference NZ chapter of the ACM's special interest group on Human-Computer Interaction, ACM (2009) 29–36
11. Rosson, M.B., Carroll, J.M.: Usability engineering: scenario-based development of human-computer interaction. Morgan Kaufmann (2002)
12. Kujala, S., Väänänen-Vainio-Mattila, K.: Value of information systems and products: Understanding the users' perspective and values. Journal of Information Technology Theory and Application **9**(4) (2009) 23–39
13. Howe, J.: The rise of crowdsourcing. Wired magazine **14**(6) (2006) 1–4
14. Ali, R., Solis, C., Salehie, M., Omoronyia, I., Nuseibeh, B., Maalej, W.: Social sensing: When users become monitors. In: Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering. ESEC/FSE '11, New York, NY, USA, ACM (2011) 476–479
15. Ali, R., Solis, C., Omoronyia, I., Salehie, M., Nuseibeh, B.: Social adaptation: when software gives users a voice. In: ENASE'12: 7th International Conference Evaluation of Novel Approaches to Software Engineering. (2012)
16. Surowiecki, J.: The wisdom of crowds. Random House Digital, Inc. (2005)
17. Kietzmann, J.H., Hermkens, K., McCarthy, I.P., Silvestre, B.S.: Social media? get serious! understanding the functional building blocks of social media. Business Horizons **54**(3) (2011) 241–251
18. Hallsteinsen, S., Hinchey, M., Park, S., Schmid, K.: Dynamic software product lines. Computer **41**(4) (2008) 93–95
19. Gacitua, R., Ma, L., Nuseibeh, B., Piwek, P., Roeck, A.D., Rouncefield, M., Sawyer, P., Willis, A., Yang, H.: Making tacit requirements explicit. In: Second International Workshop on Managing Requirements Knowledge (MaRK'09). (September 2009)