A Framework for Well-being Integrated Development Environments (WIDEs): Research Preview

Sara Hassan[1], Andrew-Sean Wilson[1] and John Galvin[2]

[1]School of Computing and Digital Technology, Birmingham City University, Birmingham, United Kingdom
[2]School of Social Sciences Birmingham City University Birmingham, United Kingdom

Abstract

[Context and Motivation] To date there are very few approaches for systematic requirements modelling of software from a mental-well-being-awareness perspective. There are even less technological innovations that address the need for mental-well-being-aware learning and teaching environment among Science, Technology, Engineering and Mathematics higher education (STEM HE) students. STEM HE student that study computing would typically work with integrated development environments (IDEs) for prolonged periods. This can make them more susceptible to technostress factors. Technostress is the inability to cope in a healthy way with technology. Technostress factors threaten the long-term mental health, productivity, and achievement outcomes of these students. **[Question/Problem]** In this paper, our target problem is the lack of systematic requirements modelling support and design guidelines for producing mental-well-being-aware IDEs that cater for technostress factors. **[Contribution]** We address this problem by proposing a novel idea for a framework that includes guidance for modelling the requirements for and designing the architectures of Well-being Integrated Development Environments (WIDEs). WIDEs aims to communicate programming errors such as runtime and syntax errors in a mental-well-being-aware manner.

Keywords

mental well-being, software design framework, software architectural modelling, software design guidelines

1. Introduction

Commonly used computing tools such as Integrated development environments (IDEs) are important for STEM HE students to learn for their future careers. Prolonged usage of technological tools can lead to a higher prevalence of technostress factors among this cohort [1, 2]. Technostress is defined as the inability to adapt or cope with information and communication technologies (ICTs) in a healthy manner [3]. If technostress is not addressed, it can lead to long-term deteriorated mental health, lower productivity, and deteriorated achievement outcomes of these students. Additionally, research indicates a significant participation of students with mental health issues in the STEM fields. [4, 5].

To our knowledge, there are currently very few technological innovations that address the need for mental-well-being-aware learning and teaching environment for STEM HE students.

REFSQ'22: 28th Intl. Working Conference on Requirements Engineering: Foundation for Software Quality, Birmingham, UK

Sara.Hassan@bcu.ac.uk (S. Hassan); Andrew.Wilson@bcu.ac.uk (A. Wilson); John.Galvin@bcu.ac.uk (J. Galvin)
0000-0001-7481-0434 (S. Hassan); 0000-0001-7064-6681 (A. Wilson); 0000-0003-4526-7529 (J. Galvin)

Nevertheless, considering technostress factors when designing learning/teaching software is integral to creating a mental-well-being-aware learning experience for STEM HE students. This has motivated us to address the following target problem: *the lack of systematic requirements modelling and design guidelines for producing an IDE that communicates guidance, runtime errors, and syntax errors in a mental-well-being-aware manner to computing HE students.* To pave the way to addressing this problem, we propose an idea to develop for a framework that aids in designing Well-being Integrated Development Environments (WIDEs) which aim to address technostress factors systematically when designing IDEs used by STEM HE students. Our work aims to expand the communities of requirement modelling and software engineering with a new track that treats mental-well-being as a quality of service dimension to be systematically optimised for during requirements and software engineering. For this paper, we consider a mental-well-being-aware IDE as one which does not impose a high level of technostress in its users.

The paper is organised as follows: Section 2 motivates our work by highlighting the significance of mental health and stress issues among STEM students. Section 3 will give a short motivating scenario to set the context which the WIDEs framework will target. Section 4 will present the proposed components of the WIDEs framework, their roles, and steps envisioned for producing these components. In Sections 5 and 6 we conclude the paper by a discussion of the challenges for producing the WIDEs framework and how they can be addressed.

2. Motivation: Stress and Mental Health Issues in STEM

The literature on stress related to ICTs tends to focus on the phenomenon of technostress. The negative consequences of technostress include increased anxious and depressive symptomatology, decreased motivation, and intention to quit [1], [2]. For instance, Chirikov et al. [1] recently reported that over 1 in 3 STEM students are suffering from clinical levels of anxiety and depression which impacted their work.

Hands-on programming practice is an integral part of STEM degrees. IDEs are the main tools that students use in this practice. Repeated exposure to IDE related technostress in computing students can therefore threaten longer-term mental health, productivity, and achievement outcomes, justifying the need for systematic design guidelines for producing well-being-aware IDEs. Several technology-related factors (summarised in Table. 1) have been identified as determinants of technostress [6] and are important considerations for human interaction with IDEs. Although previous studies have established the importance of technostress [1, 2, 6, 7] it is not clear which characteristics of the technology create stress. The boundaries and relationship between technology characteristics and stress is ambiguous. What is stressful for one person is not necessarily stressful for another, with individual differences variables such as personality characteristics, coping styles, self-esteem, self-efficacy, accounting for a high proportion of variance in well-being [8]. Situational factors such as computer system performance, restricted access to technology, or blurred boundaries between work and home life can also exacerbate technostress [1, 6]. It is necessary therefore that the WIDEs framework is flexible and considers how IDEs can be tailored to individual circumstances.

Table 1
Definitions of technostress factors

Technostress factor	Definition
Techno-overload	When the technology usage results in high workload, resulting in the user feeling forced to work faster and longer.
Techno-complexity	Whether the user feels competent enough to use a technological tool and achieve the desired results.
Techno-reliability	The perception of the consistency or dependability of the tool.
Pace of change	An individual's perception of frequent tool-related changes and upgrades.

3. Motivating Scenario

Consider a scenario where a first-year undergraduate STEM student is studying online due to COVID, personal, or work-related circumstances. These hours are all spent in a seated position in front of a laptop screen where the laptop is used both for personal and educational purposes.

On any given weekday, about an hour is initially spent reading a programming task on the screen which was asynchronously provided by the teacher. The task instructions include step-by-step instructions and screenshots of the outputs and inputs of each task. The screenshots are in a different version of software from the one the student is using. The student therefore spends about 2 hours researching how to install and set up an IDE that is compatible with their laptop and how to translate the instructions from the given task to the IDE version at hand. This initial task already exposes the student to techno-overload.

The following 2-3 hours are then spent to complete the given programming task after setting up the required IDE. The written code now includes errors. The error is explained using technical logs unreadable to the student. This common situation now leads to techno-complexity. Finally, the student resorts to trial and error making insignificant changes every time in hope for the program to produce the required output; eventually the output is as required. The student however does not understand why. This now leads to techno-reliability issues since the IDE was not consistent as far as the student is concerned. The above common narrative can happen fully or partially on any given day for a STEM student or in fact for any student.

It is scenarios such as the above that motivate us to propose WIDEs in order to aid in reducing some of the technostress factors that STEM HE students can experience when using "traditional" IDEs.

4. WIDEs Framework Outline

In this section we outline the proposed components of the WIDEs framework. The framework would comprise a requirements modelling language and design guidelines for tailoring the design of an existing or new IDE to be mental-well-being-aware. Both components of the framework are intended to be used iteratively together. Fig. 1 summarises the intended usage of the WIDEs framework. Software engineers can create iterations of the model before they embark on improving an IDE design. They can then choose to adopt some of the designs from a proposed set of design guidelines. Upon adopting them, the modelling language can

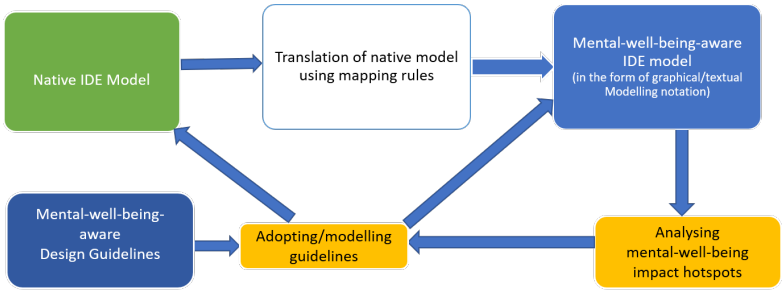

Figure 1: Overview of the WIDEs framework usage: the green box indicates input to the WIDEs framework; the blue boxes indicate components of the WIDEs framework; the yellow boxes indicate activities to be done by software engineers using the WIDEs framework; the white box indicates an activity to be done automatically within the WIDEs framework

then be used again to create updated versions of the IDE requirements model. Background research, focus groups and surveys are required to refine the proposed features for each of the components proposed below.

4.1. A Novel Mental-well-being-aware Software Requirement Modelling Support

The role of this modelling support would be to capture mental- well-being as a priority of the non-functional requirements engineering exercise. A model of an existing IDE's requirements can be created using this modelling language. This model will be the basis of enhancing the design using guidelines from the next component of the framework described below. The modelling support would include constructs to capture:

- Complexity of data displayed to the IDE user.
- Mental-well-being impact of each IDE component.
- Mental-well-being impact of each user-IDE interaction.

Existing interventions and guidance for incorporating mental-well-being into software design either focus on changing the user's behaviour and/or on changing the layout of a software without eliciting the requirements systematically [9, 10, 11]. Producing such modelling support requires the following activities:

1. Examining state-of-art requirements modelling languages and leveraging them with the above constructs. The state-of-the-art modelling languages include but are not limited to: goal-oriented, feature-oriented, aspect-oriented, object-oriented.
2. Developing a graphical and/or textual notation to enable efficient use of the modelling language by software engineers.
3. Developing a set of mapping rules to facilitate translating requirement models from other modelling languages to our modelling language.
4. Evaluating our language for expressiveness, our notation for flexibility, and our mapping rules for comprehensiveness among other criteria.

4.2. Novel Mental-well-being-aware Architectural Design Guidelines

The role of this set of guidelines would be to aid software engineers in improving the overall mental-well-being-awareness of existing or prospective IDEs, thereby transforming their IDE to a WIDE. The guidelines would be categorised according their focus: data complexity, component design and/or user-IDE interaction.

The choice of adopting guidelines introduced in this component would be up to the software engineer designing the IDE. At a high level, software engineers would use the modelling support to identify "hot-spots" in the IDE design which have a particularly negative technostress impact. Guidelines can then be used to refine the requirements in this hot-spots.

Producing this set of guidelines requires the following:

1. Examining current classical software design practices and analysing whether they can be tailored into more mental-well-being-aware guidelines. The tailored version of the practice should be described using the aforementioned modelling notation. In this way, the link between both contributions will become apparent allowing both the modelling language and the design guidelines to fit under a packaged WIDEs framework.
2. Qualitatively evaluating the tailored guidelines for their flexibility and coverage among other criteria. Software engineers can be included at this stage of the evaluation. Surveys and/or focus groups with software engineers can be conducted to assess the practicality and understandability of both the modelling language and the design guidelines.

5. Discussion and Road map

In this section we summarise the WIDEs framework implementation challenges and potential solutions for them. In particular:

- Identifying evaluation metrics to quantify mental-well-being impact
- Setting a representative evaluation case study
- Identifying and recruiting a representative and inclusive sample of STEM students to act as sample end users.

To address the challenges above we propose the following steps for a controlled experiment evaluation of the WIDEs framework:

1. Identifying relevant IDEs for comparative assessment. A combination of mixed-methods (qualitative and quantitative), subjective data collection, and objective biometric measurements will be used to profile the user experience, satisfaction and cognitive interaction with the IDEs using an initial testing cohort of users.
2. Using the above as a basis for creating a heuristic of key issues that would have to be addressed in a mental-well-being-aware IDE and thereafter producing the initial WIDEs framework.
3. Creating controlled version(s) of the IDE where the WIDEs framework is not used and a mental-well-being-aware IDE version of the IDE where the WIDEs framework is applied.
4. Using an iterative software development lifecycle, the WIDE heuristics will be systematically addressed until a final mental-well-being-aware IDE is produced.

5. Profiling the interaction of the initial testing cohort and/or a subset of them with the mental-well-being-aware IDE produced.
6. Evaluating the produced IDE in two ways. Firstly, mixed methods and biometric approaches outlined in point 1 will be used with the initial testing cohort of IDE users. Secondly, evaluation will also be conducted with a fresh cohort of "tissue testers" who have not previously experienced the IDE or the mental-well-being-aware IDE produced. The aim of this evaluation is to study the mental-well-being impact of WIDE compared to IDE usage.

6. Conclusion

In this paper we present our vision for a framework that aids in designing Well-being Integrated Development Environments (WIDEs). We present an outline for WIDEs' components, shed the light on implementation challenges and possible solutions for them. Once these challenges are addressed, we intend to investigate the suitability of our framework for software products beyond IDEs and eventually beyond teaching and learning environments.

References

[1] J. Galvin, M. Evans, K. Nelson, G. Richards, E. Mavritsaki, T. Giovazolias, K. Koutra, B. Mellor, M. C. Zurlo, A. Smith, F. Vallone, Technostress, coping, and anxious and depressive symptomatology in university students during the COVID-19 pandemic, Europe's Journal of Psychology (2021). doi:10.5964/ejop.4725.

[2] P. Upadhyaya, et al., Impact of technostress on academic productivity of university students, Education and Information Technologies 26 (2021) 1647–1664.

[3] C. Brod, Technostress: The Human Cost Of The Computer Revolution, Basic Books, 1984. Google-Books-ID: CtMmAAAAMAAJ.

[4] S. Baron-Cohen, S. Wheelwright, C. Stott, P. Bolton, I. Goodyer, Is there a link between engineering and autism?, Autism 1 (1997) 101–109. URL: https://doi.org/10.1177/1362361397011010. doi:10.1177/1362361397011010. arXiv:https://doi.org/10.1177/1362361397011010.

[5] G. C. Windham, K. Fessel, J. K. Grether, Autism spectrum disorders in relation to parental occupation in technical fields, Autism Research: Official Journal of the International Society for Autism Research 2 (2009) 183–191. doi:10.1002/aur.84.

[6] R. Ayyagari, V. Grover, R. Purvis, Technostress: Technological Antecedents and Implications, MIS Quarterly 35 (2011) 831–858. URL: https://www.jstor.org/stable/41409963. doi:10.2307/41409963, publisher: Management Information Systems Research Center, University of Minnesota.

[7] X. Wang, S. C. Tan, L. Li, Technostress in university students' technology-enhanced learning: An investigation from multidimensional person-environment misfit, Computers in Human Behavior 105 (2020) 106208.

[8] G. M. Mark, A. P. Smith, Stress models: a review and suggested new direction, Nottingham

University Press, Nottingham, 2008, pp. 111–144. URL: https://orca.cardiff.ac.uk/31085/, issue: 3 Num Pages: 312 Number: 3.

[9] E. Jagroep, J. M. van der Werf, S. Brinkkemper, L. Blom, R. van Vliet, Extending software architecture views with an energy consumption perspective, Computing 99 (2017) 553–573. URL: https://doi.org/10.1007/s00607-016-0502-0. doi:10.1007/s00607-016-0502-0.

[10] Q. E. Booker, C. M. Rebman Jr, F. L. Kitchens, A model for testing technostress in the online education environment: An exploratory study., Issues in Information Systems 15 (2014).

[11] T. Bickmore, A. Gruber, R. Picard, Establishing the computer-patient working alliance in automated health behavior change interventions, Patient Education and Counseling 59 (2005) 21–30. doi:10.1016/j.pec.2004.09.008.