Sustainable software engineering – have we neglected the software engineer's perspective?

Binish Tanveer
Department of Software Engineering
Blekinge Institute of Technology
Karlskrona, Sweden
0000-0003-2300-068X

***Abstract*—Background: Software development is a knowledge-intensive and creative activity requiring human ingenuity, thus, sustainability from an engineer's perspective (i.e., meeting an engineer's personal and professional needs while maintaining their mental and physical well-being) must be achieved and maintained. Aims: With this paper, we aim to bring attention to the currently overlooked aspect of sustainability from an individual engineer's perspective. Method: Through an analysis of systematic literature reviews and mapping studies, this research demonstrates a lack of research investigating the individual (human) dimension of sustainability in the current software engineering literature. Results: The analysis of the literature reviews reveals that the current research has mainly focused on reducing the energy footprint of software systems. Thus, revealing the renewed need for paying attention to the individual engineer's perspective. Conclusion: Future research should leverage contributions from related research areas like human aspects in software engineering (e.g., topics like cognition and motivation). There is a need for identifying factors that impact sustainability at an individual level and their interplay with the team and organization level practices, policies, and decisions. The overall ambition is to develop empirically validated guidelines and best practices to measure, improve and maintain sustainability from an engineer's perspective. Such measures are expected to ensure engineers' sustainability and enable high-quality software development.**

***Index Terms*—sustainable software engineering, software development, individual sustainability**

I. Introduction

The rapid rate of technology changes [1], the increasingly shorter development cycles and the stress [2] and cognitive load [3] that characterizes the modern software development context make software engineers work untenable. The situation has become further demanding when these modern methods (like DevOps [4], [5]) require additional skills, which were previously reserved for specialized roles [6].

The software has become the core of almost every product or service on the face of the globe today. Whether it is the banking sector, automotive industry, other modern transportation, manufacturing, or supply-chain management all are powered by software. Airbus A380 includes 120 million mission-critical lines of code [7]. Taking the automotive industry, for example, in 2020 a Volvo vehicle has about 100 million lines of code (a software equivalent to 6000 books) [8]. This trend will probably continue because of autonomous drive and connected cars, and by 2030 Volvo will have a software "equivalent to the number of books of MIT Science Fiction Society Library, which is nearly 90% of all science fiction books that were ever printed in English (as of 2010)" [8].

As software has become an essential part of our lives, its ever-increasing usage has also increased demands for energy and resource consumption. Information technology (IT) promotes sustainability by finding more efficient software solutions to address environmental problems [9]. Green IT addresses environmental aspects of sustainability by considering computer hardware only. Hardware is useless without its application layer in an IT system [10]. The software runs on hardware and provides functionality like enabling smart logistics, smart lighting, and smart heating or paperless offices, etc.

Software development is a human activity and human sustainability should be considered in development. Researchers have explored mainly the environmental and energy dimensions of sustainability. However, there is a lack of research investigating the individual (human) dimension of sustainability, though this dimension has been considered as a basis for other dimensions [11]. This research intends to explore this missing dimension of sustainability in the context of software development.

Software systems have become central to operations of the industrial society today, though in an unsustainable manner. The local interests and convenience take priority over the global responsibility resulting in adverse effects on the environment [12]. The Karlskrona Manifesto [13] recognizes software sustainability as "the capacity of the software-intensive system itself to endure" and maintains the view that sustainability is a construct constituting five dimensions: environmental, economic, individual, social, and technical. Therefore, it is essential to consider all five dimensions as non-functional requirements in the software engineering process [12].

II. Sustainability in software engineering - state-of-the-art analysis

A rising concern for sustainability in and for software engineering (SE) has attracted researchers' attention. A thorough search revealed several secondary studies on the topic.

A systematic literature review was conducted on SE practices for sustainability, investigating models and methods for sustainable software [11]. A Systematic Mapping Study

(SMS) [14] investigated the software process improvement approaches in sustainability. Another SMS [15] was done which extended the previous work [11] by focusing on research topics and facets, application domains in finding how sustainability is supported in SE. Both the reviews concluded that the research is limited to few knowledge areas in SE and further research is required. Another SMS exists that captures the state-of-the-art of sustainable SE [16]. The study's primary focus was existing models, guidelines, and practices. The results indicated an increased interest in the field, with few prominent authors and venues publishing research within the scope of sustainable SE. The prominent venues publishing research within the scope of sustainable SE are relatively new, e.g., the international workshop on Green and Sustainable Software (GREENS) with the first workshop held in 2012, the international conference on ICT for sustainability[1] with the first instance held in 2013 and the international Green and sustainable computing conference[2] with the first conference held in 2010. Journal of Sustainable Computing: Informatics and Systems (SUSCOM), and the book Harnessing Green IT: Principles and Practices were other prominent venues [16]. An International Workshop on Sustainable Software Engineering (SUSTAINSE) has recently taken off, with its first workshop held in 2020[3]. Another SMS was done with a focus on the role of software analytics in the field of Green SE [17]. The results indicate a lack of analytics techniques usage in green SE with a strong need to develop new analytics tools and metrics that correlate the tools and their energy usage [17]. A more recent SMS was conducted in 2018, with the focus to investigate existing software methods, processes, tools, and metrics in Green SE [18]. They also concluded that Green and Sustainable software research field is recently getting attention and also a need to conduct more studies on sustainable techniques, tools, and metrics covering implementation, testing, and maintenance phases in SDLC [18].

These secondary studies' have a consensus that Green and sustainable software research is a rising field with less research coverage in different knowledge areas with respect to dimensions of sustainability and hence demands further research. However, none of the secondary studies have discussed the dimension of human sustainability. Software development activity is a human endeavor that requires critical thinking and creativity. Therefore, to enable high-quality software development, it is essential to realize the engineer's personal, professional needs and maintain their well-being.

III. Engineer's sustainability and research vision

Individual (human) sustainability refers to "maintaining individual human capital (e.g., education, skills, knowledge, leadership, self-respect, mobility, and access to services)". It also includes "the mental and physical well-being of humans as individuals" [13]. In our context we refer to the individual as engineer.

[1]https://conf.researchr.org/home/ict4s-2022

[2]https://www.igscc.org/archive

[3]https://sites.google.com/view/sustainablese-workshop/home

The capacity of the human working memory and the amount of cognitive load it can process (i.e., cognitive bandwidth) are closely related. According to cognitive load theory [19], overloading a human's "limited" working memory inhibits his learning ability and problem-solving skills [19] [20]. The innate complexity of a task, its environment, the use of tools, information flow and management, work processes, and even workplace layouts are seen as the cognitive load drivers in large-scale software development [3] that affect the mental and physical well-being of engineers. Hence, cognitive load is considered a waste in SE [20]. Removing this waste is associated with more efficient learning outcomes [21]. Unfortunately, the most researched cognitive aspect in SE has been limited to "program comprehension" whereas only a small percent of the studies focused on the "soft or human-related topic" [3] like cognitive biases [22]. The topic of cognitive biases also got attention in the recent past, however, the rise in the number of published articles is still low. The focus lies mainly in the knowledge area of software management where critical areas including requirements, design, testing, quality and processes/methods are under represented [22].

Another aspect that is related to and can affect human sustainability is "burnout [23] - the state of long-term exhaustion accompanied by diminished interest and performance" [24]. A lack of independence or control at work and prolonged pressure increases the risks of burnout in software engineers [23]. Increasing and deep embedding of information technology in daily working processes have also been observed as a cause of burnouts [24]. A consequence of burnout was seen when ten percent of sick days in German organizations resulted from employee burnout, which in turn cost billions of euros to the economy [24].

While considering the above-mentioned areas, we need to relate the findings to the context of software development. Among different software development approaches, agile methodology emerged as a human-centric approach with autonomous teams that value individuals and interactions over processes and tools. Although this approach has been beneficial in the last decades due to its ability to reduce time to market and high productivity, yet, the engineer's sustainability has been neglected. Researchers have also indicated that the initial productivity gains in agile are challenging to maintain at a sustainable pace over time. Agile development also has certain side effects, e.g., cognitive load, constant pressure on delivering working code, information overload, team constant interactions in daily meetings [25], and team intensive open space layouts resulting in employee burnouts (for example in XP, where one has to constantly work with another engineer with little time for mental rest). Developing software under such circumstances not only affects the mental health of the engineers but also compromises the quality of the produced software [26]. These findings also indicate a need to enable and maintain engineer's sustainability while coping with the pace of changing technology.

This research envisions the future of engineer's sustainability for enabling high-quality software development. Though

the published literature on this dimension is sparse, yet this research will use and build upon the existing knowledge of sustainability. However, this research will take into consideration the work done on other areas such as motivation in SE, human factors in SE, cognition in SE. These research areas are very much related and hence considering them in the context of software development can be very useful in understanding and improving engineer's sustainability.

IV. Research goals and required research work

The aim of this research is to enable high-quality software development while ensuring engineer's sustainability. To achieve this aim, the goal (using GQM approach [27]) is to analyze the software development process for the purpose of understanding with respect to the engineer's sustainability from the point of view of the software development team in the context of agile software development. Another goal is to improve the engineer's sustainability through identifying the metrics and required process to operationalize the proposed improvements. Hence, this research would require investigating for example the factors (policies, practices, processes, and environment) impacting engineers' sustainability in relation to their responsibilities. Furthermore, research is required to identify the metrics and required processes to implement and measure the improvement. The following research questions (RQ) serve as a starting point to conduct this investigation:

- RQ1: How is an engineer's sustainability perceived in SE research and practice?
- RQ2: What are the factors that impact an engineer's sustainability? (e.g. contextual factors like existing policies, practices or technologies, organizational and team structures, etc.)
  - RQ2.1: What are the sources of internal and external stress for software engineers?
  - RQ2.2: What is the impact of a short development cycle on an engineer's sustainability?
  - RQ2.3: What is the impact of meetings (e.g., daily stand-up, planning, retrospectives etc.) on an engineer's well-being?
- RQ3: What factors need to be adapted and how, in agile development life cycle to ensure engineer's sustainability?
- RQ4: How can we measure the effectiveness of the improvement with respect to engineer's sustainability?

Since software development is a knowledge-intensive activity that largely relies on human cognitive skills, creativity, knowledge, and abilities, therefore, we need to explore the factors that facilitate this human endeavor sustainably. Although we are interested in exploring the human's (engineer) perspective, we need to explore through the teams and organization. This means we need to understand the social structures, organization policies (reward policies), development processes because all these aspects have implications on the engineer's sustainability. Hence, the following work is required to understand and operationalize engineer's sustainability:

1) A deeper state-of-the-art analysis of related research areas as mentioned above i.e., motivation in SE, human factors in SE, cognition in SE, process improvement needs to be done while relating it to engineers' sustainability in the context of software development. This will help discover the diverse factors impacting engineers' sustainability.
2) Another complementary work required is to capture the practitioners' perceptions about sustainability, the factors impacting engineer's sustainability like organizational structures, development processes, (reward) policies, and prevailing ways of working.
3) Work is required on identification of metrics and processes on how to operationalize and measure the improvement.

Currently, we are working on the design of a literature review to capture the state-of-the-art (as indicated in 1) as well as a survey to capture the practitioner's perspective (as mentioned in 2).

V. Expected outcomes

This research is expected to make the following outcomes and contributions to software engineering research and practice:

- The first outcome is the establishment of understanding of the connection between engineers' sustainability and their tasks and responsibilities.
- The second outcome is the development of taxonomy of factors that impact sustainability at an individual level and their interplay with team and organization level practices, policies, and decisions. The taxonomy could also be used by practitioners to operationalize sustainability concepts during decision-making activities in software development [28].
- The third outcome is the development of empirically validated practices and guidelines for processes, policies, practices to ensure engineer's sustainability. The practitioners could also use these guidelines to do essential reforms in their development processes or other policies and practices in their organizations.

An overview of related research areas and results is depicted in Figure 1.

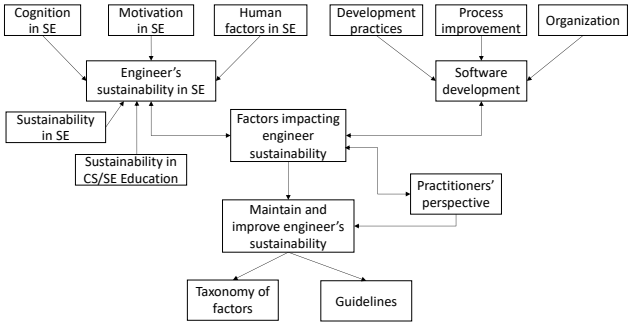

Fig. 1. Overview of research areas and results

VI. Conclusions

Software plays a critical role in our lives. Software development activity is a human endeavor that requires critical thinking and creativity. Therefore, to enable high-quality software development, it is essential to realize the engineer's personal, professional needs and maintain their well-being. The current software engineering literature lacks the individual (human) dimension of sustainability. Hence, this research envisions the future of enabling, maintaining, and improving engineer's sustainability while coping with the pace of changing technology. Overall, this research relies on using empirical research with a theoretical basis in sustainability in SE, software development practices, motivation, human factors in SE, cognition in SE, and software process improvement. The resulting taxonomy of factors and guidelines will influence the decision-making models and methods to take into consideration the concepts of sustainability [28] for example, decisions regarding team structure, organization structure, policies and prevailing software development practices. The factors would also serve as a foundation for future research to find interrelations between them, categorize them and devise ways to control them.

Acknowledgment

We would like to acknowledge that this work was supported by the Knowledge Foundation through the projects SERT – Software Engineering ReThought.

References

[1] M. Kersten, "A cambrian explosion of devops tools," *IEEE Softw.*, vol. 35, no. 2, pp. 14–17, 2018.

[2] F. Asgari, J. Mostafa *et al.*, "Investigating the relationship between organizational factors of stress and delay of software projects in a large knowledge based company." *Independent Journal of Management & Production*, vol. 8, no. 1, pp. 15–33, 2017.

[3] D. Helgesson, E. Engström, P. Runeson, and E. Bjarnason, "Cognitive load drivers in large scale software development," in *Proceedings of the 12th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE@ICSE*. Montreal, Quebec, Canada: IEEE / ACM, 2019, pp. 91–94.

[4] R. Jabbari, N. B. Ali, K. Petersen, and B. Tanveer, "What is devops?: A systematic mapping study on definitions and practices," in *Proceedings of the Scientific Workshop Proceedings of XP2016*. Edinburgh, Scotland, UK: ACM, 2016, p. 12.

[5] ——, "Towards a benefits dependency network for devops based on a systematic literature review," *J. Softw. Evol. Process.*, vol. 30, no. 11, 2018.

[6] A. Wiedemann, M. Wiesche, and H. Krcmar, "Integrating development and operations in cross-functional teams - toward a devops competency model," in *Proceedings of the Conference on Computers and People Research, SIGMIS-CPR*. Nashville, TN, USA: ACM, 2019, pp. 14–19.

[7] C. C. Venters, R. Capilla, S. Betz, B. Penzenstadler, T. Crick, S. Crouch, E. Y. Nakagawa, C. Becker, and C. Carrillo, "Software sustainability: Research and practice from a software architecture viewpoint," *Journal of Systems and Software*, vol. 138, pp. 174–188, 2018.

[8] V. Antinyan, "Revealing the complexity of automotive software," in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2020. New York, NY, USA: Association for Computing Machinery, 2020, p. 1525–1528.

[9] S. Naumann, M. Dick, E. Kern, and T. Johann, "The GREENSOFT model: A reference model for green and sustainable software and its engineering," *Sustain. Comput. Informatics Syst.*, vol. 1, no. 4, pp. 294–304, 2011.

[10] K. Erdelyi, "Special factors of development of green software supporting eco sustainability," in *Proceedings of IEEE 11th International Symposium on Intelligent Systems and Informatics, SISY*. Subotica, Serbia: IEEE, 2013, pp. 337–340.

[11] B. Penzenstadler, V. Bauer, C. Calero, and X. Franch, "Sustainability in software engineering: A systematic literature review," in *Proceedings of the 16th International Conference on Evaluation & Assessment in Software Engineering, EASE*. Ciudad Real, Spain: IET - The Institute of Engineering and Technology / IEEE Xplore, 2012, pp. 32–41.

[12] B. Penzenstadler, A. Raturi, D. J. Richardson, and B. Tomlinson, "Safety, security, now sustainability: The nonfunctional requirement for the 21st century," *IEEE Softw.*, vol. 31, no. 3, pp. 40–47, 2014.

[13] C. Becker, R. Chitchyan, L. Duboc, S. Easterbrook, B. Penzenstadler, N. Seyff, and C. C. Venters, "Sustainability design and software: The karlskrona manifesto," in *Proceedings of the 37th IEEE/ACM International Conference on Software Engineering, ICSE*. Florence, Italy, May 16-24, 2015: IEEE Computer Society, 2015, pp. 467–476.

[14] G. A. García-Mireles, "Environmental sustainability in software process improvement: a systematic mapping study," in *International Conference on Software Process Improvement*. Ciudad Real, Spain: Springer International Publishing, 2017, pp. 69–78.

[15] B. Penzenstadler, A. Raturi, D. J. Richardson, C. Calero, H. Femmer, and X. Franch, "Systematic mapping study on software engineering for sustainability (SE4S)," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, EASE*. London, England, United Kingdom, 2014: ACM, 2014, pp. 14:1–14:14.

[16] K. R. Berntsen, M. R. Olsen, N. Limbu, A. T. Tran, and R. Colomo-Palacios, "Sustainability in software engineering - a systematic mapping," in *International Conference on Software Process Improvement*. Aguascalientes, Mexico: Springer International Publishing, 2017, pp. 23–32.

[17] H. Anwar and D. Pfahl, "Towards greener software engineering using software analytics: A systematic mapping," in *Proceedings of the 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. Vienna, Austria: IEEE, 2017, pp. 157–166.

[18] B. C. Mourão, L. Karita, and I. do Carmo Machado, "Green and sustainable software engineering - a systematic mapping study," in *Proceedings of the 17th Brazilian Symposium on Software Quality, SBQS*. Curitiba, Brazil, October 17-19, 2018: ACM, 2018, pp. 121–130.

[19] J. Sweller, "Cognitive load during problem solving: Effects on learning," *Cognitive science*, vol. 12, no. 2, pp. 257–285, 1988.

[20] T. Sedano, P. Ralph, and C. Péraire, "Software development waste," in *Proceedings of the 39th International Conference on Software Engineering, ICSE*. Buenos Aires, Argentina: IEEE / ACM, 2017, pp. 130–140.

[21] J. J. Van Merrienboer and J. Sweller, "Cognitive load theory and complex learning: Recent developments and future directions," *Educational psychology review*, vol. 17, no. 2, pp. 147–177, 2005.

[22] R. Mohanani, I. Salman, B. Turhan, P. Rodríguez, and P. Ralph, "Cognitive biases in software engineering: A systematic mapping study," *IEEE Transactions on Software Engineering*, vol. 46, no. 12, pp. 1318–1339, 2020.

[23] M. Mäntylä, B. Adams, G. Destefanis, D. Graziotin, and M. Ortu, "Mining valence, arousal, and dominance: possibilities for detecting burnout and productivity?" in *Proceedings of the 13th International Conference on Mining Software Repositories, MSR*. Austin, TX, USA, May 14-22, 2016: ACM, 2016, pp. 247–258.

[24] C. Maier, S. Laumer, and A. Eckhardt, "Information technology as daily stressor: pinning down the causes of burnout," *Journal of Business Economics*, vol. 85, no. 4, pp. 349–387, 2015.

[25] J. E. Hannay and H. C. Benestad, "Perceived productivity threats in large agile development projects," in *Proceedings of the International Symposium on Empirical Software Engineering and Measurement, ESEM*. New York, NY, USA: ACM, 2010.

[26] D. Smite, N. B. Moe, and P. J. Ågerfalk, Eds., *Agility Across Time and Space, Implementing Agile Methods in Global Software Projects*. Berlin: Springer, 2010.

[27] V. R. B. G. Caldiera and H. D. Rombach, "The goal question metric approach," *Encyclopedia of software engineering*, pp. 528–532, 1994.

[28] N. B. Ali, "Is effectiveness sufficient to choose an intervention?: Considering resource use in empirical software engineering," in *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM*. Ciudad Real, Spain: ACM, 2016, pp. 54:1–54:6.