

Agile and Lean Software Engineering and the SWEBOK

Position Paper for Panel: Educational and Professional Implications of SWEBOK

David Parsons
The Mind Lab
Auckland, New Zealand
david.parsons@themindlab.ac.nz

Abstract—This position paper addresses the usefulness (or otherwise) of the Software Engineering Body of knowledge (SWEBOK) version 3 for software practitioners in industry, and the consequent need for the SWEBOK to evolve to better address current industry practice. The position taken in this paper is that agile and lean methods are now the predominant approach to software engineering, and that the limited and anachronistic coverage of agile methods in the SWEBOK, coupled with the absence of any acknowledgement of lean approaches, is undermining software engineering education, the career prospects of graduates, and the software industry as a whole. It is therefore proposed that the agile methods section of the SWEBOK is revised and expanded such that it provides a valid body of knowledge for contemporary software engineering

Keywords—*Agile; Lean; Kanban; Scrum; DevOps; SWEBOK; SEEK*

I. INTRODUCTION – SWEBOK, AGILE, AND LEAN

While the Software Engineering Body of Knowledge (SWEBOK) [1] has been guiding software engineering education for decades, it has continuously struggled to provide software engineering students with appropriate skills to excel in their jobs [2], and many graduates face difficulties when beginning their professional careers due to a skills mismatch between what is taught and what is needed [3]. One reason for this may be the failure of the SWEBOK to meaningfully provide a useful body of knowledge in contemporary practices in agile and lean software development. Of course, the SWEBOK is not the curriculum, but directly underpins it through the IEEE software engineering curricula guidelines [4]. Links between the Software Engineering Education Knowledge (SEEK) and the SWEBOK also strongly emphasize software engineering tools and methods [5], so both should reflect the dominant nature of agile methods in contemporary software engineering. This position paper first outlines the growth of agile methods over the last 20 years or so, along with the associated increase in the use of lean methods and practices such as Kanban, often integrated with agile processes. It then provides some commentary on the limitations of SWEBOK 3 in addressing these major trends and concludes by suggesting

some ways in which the next version of the SEWBOK can improve its coverage in these areas.

A. *The growth of agile methods*

Agile methods, which emerged from lightweight methods in the 2000s, have continued to increase their influence over how software is engineered. Exactly how this increase has unfolded is to some extent unclear. One set of data suggests that uptake has been increasing markedly since about 2010 and is “now the norm” [6]. Snapshots using different data across different years provide varying perspectives. An internal Microsoft study in 2007 suggested that about a third of the teams were using agile methods [7]. Ten years later, a broader study of 153 practitioners gave a similar number, suggesting that about a third of organizations were using agile methods [8].

More recently, the 15th State of Agile Report [9] noted a significant growth in agile adoption, from 37% in 2020 to 86% in 2021. This seems in no small part to have been driven by DevOps initiatives, a complementary set of agile practices for iterative delivery in short cycles [10], which require the core agile practices of collaboration, automation, and tooling [11]. Of course, definitions and measures of agile adoption can be variable. A 2015 study by HP noted that only 15% of respondents claimed to be using “pure agile”, while 51% were “leaning towards agile” [6]. A 2018 article in the Harvard Business Review noted that although about 40% of organizations had applied agile methods in parts of their operations, adoption was neither broad nor deep [12]. As Hoda et al. note, after more than two decades of agile practice, many organizations still consider themselves still maturing in this space [13].

These figures may suggest that, at the time of publication of SWEBOK 3 in 2014, agile methods had not yet reached the level of dominance in software engineering that they now appear to hold. In addition, the inconsistent application of agile methods suggests that better coverage in the SWEBOK might, through the improved knowledge of graduates and early career software engineers, lead to more mature usage in industry. In addition, software engineering education has already broadly embraced agile methods. A 2021 study found that 79.4% of software engineering education studies were associated with

Agile Software Development [2]. A revised SWEBOK would help to address this de facto move towards agile methods as a predominant software engineering approach.

B. Lean development

The application of lean thinking to software engineering is by no means as widespread or embedded as agile methods in industry. However, its links with agile methods, particularly in the sharing of practices such as Kanban boards in agile teams, and more explicitly in the Scrumban method, mean that we cannot fully address agile software engineering without at least acknowledging the influence of lean thinking. Its use is also growing significantly, for example the most recent “State of Agile” report shows that 22% of respondents were using some kind of lean approach. [9]. However, like agile methods, we must question to what extent this usage is broad and deep, with only a small number of organizations implementing Kanban beyond the “still maturing” stage [14]. Perhaps the SWEBOK can provide more support for this evolving area of software engineering?

C. Agile, Lean and SWEBOK 3

So, what of SWEBOK 3? There are a few references to agile scattered through the document, some of which raise questions about how it is categorized, for example “Agile development” (actually an example of traditional incremental delivery) (p. B-17) suggests a somewhat dismissive tone. The main agile methods section (one page out of 335) certainly shows its age, as we might expect from a document that is around ten years old. It refers to the most popular methods as being Rapid Application Development (RAD), eXtreme Programming (XP), Scrum, and Feature-Driven Development (FDD). It is doubtful that this was true even in 2014, and certainly is not the case now. The most recent state of agile report shows that 66% of respondents were using Scrum. Only 1% used XP, with no sign of RAD or FDD [9]. The agile development section of the SWEBOK continues with a discussion around combinations of agile and more plan-based methods, but this is neither referenced nor illustrated with any examples. Essentially the problem is that the SWEBOK does not provide any kind of body of knowledge for agile software development. Neither does it provide any support at all for an understanding of any aspects of lean development or DevOps.

II. A PROPOSAL FOR SWEBOK 4 – AGILE AND LEAN BODIES OF KNOWLEDGE

The practice of software engineering is evolving all the time. A systematic literature review by Garousi et al. (2019) revealed that professional practice and project management are becoming increasingly important and emphasize the soft skills that are essential to modern agile software development [3]. It is clear that both agile and lean software engineering are becoming increasingly popular, but also that usage is immature. The SWEBOK can contribute to addressing this problem by providing an improved body of knowledge that can help to ensure that the core principles of contemporary agile and lean development are properly understood by those entering the profession.

Of course there will always be calls to expand the SWEBOK, such as providing better coverage of testing, maintenance and configuration [15]. To some extent, new coverage can be introduced simply by recontextualizing what is already there, for example by creating a DevOps curriculum from various existing components of the SWEBOK [16]. However, the current single page entry for agile development is simply not sufficient for today’s software engineering environment and must be revised and expanded.

REFERENCES

- [1] P. Bourque, R. E. Fairley, and IEEE Computer Society, *SWEBOK v.3.0: Guide to the software engineering body of knowledge*. IEEE, 2014.
- [2] O. Cico, L. Jaccheri, A. Nguyen-Duc, and H. Zhang, “Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends,” *Journal of Systems and Software*, vol. 172, p. 110736, Feb. 2021, doi: 10.1016/j.jss.2020.110736.
- [3] V. Garousi, G. Giray, E. Tuzun, C. Catal, and M. Felderer, “Closing the Gap Between Software Engineering Education and Industrial Needs,” *IEEE Software*, vol. 37, no. 2, pp. 68–77, Mar. 2020, doi: 10.1109/MS.2018.2880823.
- [4] P. Bourque, R. E. Fairley, and IEEE Computer Society, *Guide to the software engineering body of knowledge*. 2014.
- [5] S. Frezza, M. Tang, and B. J. Brinkman, “Creating an Accreditable Software Engineering Bachelor’s Program,” *IEEE Software*, vol. 23, no. 6, pp. 27–35, Nov. 2006, doi: 10.1109/MS.2006.156.
- [6] J. Jeremiah, “Agile vs. waterfall: Survey shows agile is now the norm,” *TechBeacon*. <https://techbeacon.com/app-dev-testing/survey-agile-new-norm> (accessed Feb. 20, 2022).
- [7] A. Begel and N. Nagappan, “Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study,” in *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, Sep. 2007, pp. 255–264. doi: 10.1109/ESEM.2007.12.
- [8] L. R. Vijayarathy and C. W. Butler, “Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter?,” *IEEE Software*, vol. 33, no. 5, pp. 86–94, Sep. 2016, doi: 10.1109/MS.2015.26.
- [9] Digital ai, “15th Annual State Of Agile Report | Digital.ai,” 2021. <https://digital.ai/resource-center/analyst-reports/state-of-agile-report> (accessed Feb. 20, 2022).
- [10] L. Leite, C. Rocha, F. Kon, D. Milojevic, and P. Meirelles, “A Survey of DevOps Concepts and Challenges,” *ACM Comput. Surv.*, vol. 52, no. 6, pp. 1–35, Nov. 2020, doi: 10.1145/3359981.
- [11] W. Luz, G. Pinto, and R. Bonifacio, “Adopting DevOps in the Real World: A Theory, a Model, and a Case Study,” *Journal of Systems and Software*, vol. 157, 2019, doi: 10.1016/j.jss.2019.07.083.
- [12] S. Panditi, “Survey Data Shows That Many Companies Are Still Not Truly Agile,” *Harvard Business Review*, Mar. 22, 2018. Accessed: Feb. 20, 2022. [Online]. Available: <https://hbr.org/sponsored/2018/03/survey-data-shows-that-many-companies-are-still-not-truly-agile>
- [13] R. Hoda, N. Salleh, and J. Grundy, “The Rise and Evolution of Agile Software Development,” *IEEE Software*, vol. 35, no. 5, pp. 58–63, 2018, doi: 10.1109/MS.2018.290111318.
- [14] N. Tsonev, “The First-Ever State of Kanban Report Is Live!,” *Kanbanize Blog*, May 20, 2021. <https://kanbanize.com/blog/state-of-kanban-report/> (accessed Feb. 20, 2022).
- [15] V. Garousi, G. Giray, and E. Tuzun, “Understanding the Knowledge Gaps of Software Engineers: An Empirical Analysis Based on SWEBOK,” *ACM Trans. Comput. Educ.*, vol. 20, no. 1, p. 3:1-3:33, Nov. 2019, doi: 10.1145/3360497.
- [16] A. Capozucca, N. Guelfi, and B. Ries, “Design of a (yet another?) devops course,” in *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, Mar. 2018, pp. 1–18.