



Proceedings of the Informing Science + Information Technology Education Conference

*An Official Publication
of the Informing Science Institute
InformingScience.org*

InformingScience.org/Publications

Online July 24 – 25, 2024

AGILE SOFTWARE DEVELOPMENT: THE EXPERIENCE OF WORKING IN SPRINTS [RESEARCH IN PROGRESS]

Maayan Nakash

Department of Management;
Department of Information Science;
Bar-Ilan University, Ramat Gan, Israel

Maayan.Nakash@biu.ac.il

ABSTRACT

Aim/Purpose	This exploratory study seeks to surface a richer understanding regarding the direct experience of software developers (SDs) working in sprints as part of the adoption of agile methodology principles.
Background	The software development industry is increasingly adopting agile development methods. In the agile methodology, the delivery cycle is broken down into sprints, also known as iterations. There is relatively little empirical knowledge produced to date on agile software development methods. Specifically, the SDs' authentic work experience in agile sprints is a subject that is missing from the literature.
Methodology	This paper presents results based on data collected through preliminary in-depth interviews with SDs working in research and development teams of Israeli high-tech companies. As a heuristic device for describing their experience, an interpretive phenomenological perspective was adopted. Content analysis was used as a central methodology to analyze the data.
Findings	The preliminary results present the ways of managing a development project in sprints, put a spotlight on the value of the agile method, and provide a unique glimpse of its challenges.
Contribution	Our work adds to the growing body of research investigating the SDs' perceptions. The preliminary results develop the theoretical and practical knowledge related to managing agile sprints, with reference to optimal communication, monitoring, control, and knowledge sharing among SDs in agile projects.

Accepted by Editor Eli Cohen | Received: December 17, 2023 | Revised: January 18, 2024 |

Accepted: January 24, 2024.

Cite as: Nakash, M. (2024). Agile Software Development: The Experience of Working in Sprints [Research in Progress]. In M. Jones (Ed.), *Proceedings of InSITE 2024: Informing Science and Information Technology Education Conference*, Article 2. Informing Science Institute. <https://doi.org/10.28945/5252>

(CC BY-NC 4.0) This article is licensed to you under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/). When you copy and redistribute this paper in full or in part, you need to provide proper attribution to it to ensure that others can later locate this work (and to ensure that others do not accuse you of plagiarism). You may (and we encourage you to) adapt, remix, transform, and build upon the material for any non-commercial purposes. This license does not permit you to use this material for commercial purposes.

Keywords software development, agile, sprint, software project management, adaptive software development

INTRODUCTION

Agile has been described as “a set of recommendations for a more adaptive and efficient approach” to project management (Annosi et al., 2020, p. 62), and it originates in software development (Beck et al., 2001). Recent reports find that agile practices are pervasively used across various industries around the world (Benlian, 2022). Agile software development (ASD) methods, such as Extreme Programming (XP), Scrum, or Kanban (Alsaqqa et al., 2020; Dyba & Dingsoyr, 2009; Sahin & Celikkan, 2020; Traini, 2022), are essentially based on values and principles defined in the Agile Manifesto (Beck et al., 2001). Since it was introduced, practitioners have tried to create and apply agile methodologies in many disciplines, and the research community has devoted a great deal of attention to the subject (Dingsøy et al., 2012; Larson, 2012).

ASD methods are designed to help software developers (SDs) improve their ability to estimate the amount of time required to complete a task (Edeki, 2015). Little attention has been paid to the potential negative consequences of the implementation of agile in large-scale settings (Annosi et al., 2020). The complexity inherent in ASD adoption is sometimes considered high due to organizational culture, resistance to changes, and the need for upper management sponsorship and involvement (Campanelli & Parreiras, 2015). The need to learn the skill of working collaboratively is of primary importance for SDs (Goel & Kathuria, 2010). Specifically, the failure to adopt ASD methodology was seen as a learning failure by SDs within the team (McAvoy & Butler, 2009).

Instead of the long release cycles of the previously popular waterfall methodology, the agile technique offers regular short sprint release cycles (Edeki, 2015). A sprint is an iteration of a development round in the agile technique, and it requires careful planning to ensure the project’s success. This is the basic development unit, which is a time-boxed effort. That is, the sprint is restricted to a specific duration (Project Management Institute, 2017). At each iteration, a subset of user stories is developed. The estimated complexity, business value, and affinity of the user stories may influence the optimality of sprint plans (Boschetti et al., 2014). The literature recognizes that agile projects have a failure rate of 10%, while waterfall projects fail 30% of the time (Mishra & Alzoubi, 2023).

The ASD methods present clear advantages, such as accelerated time to market, collaboration with customers, learning among developers, and enhanced flexibility (Abrahamsson et al., 2017; Annosi et al., 2020; Campanelli & Parreiras, 2015). Moreover, they allow customers and stakeholders to be more involved in the software development process (Edeki, 2015). Some studies have suggested that agile software projects may incorporate changes more easily and demonstrate business value more effectively than traditional models (Dyba & Dingsoyr, 2009). In addition, the predominant view in the software engineering literature is that the ADS methods have positive ramifications for SDs (Benlian, 2022).

Works carried out in this domain focused, for example, on identifying similarities and differences between certain ASD methodologies (e.g., Zayat & Senvar, 2020). However, in general, academic research on ASD methodologies is relatively scarce (Abrahamsson et al., 2017). Even after about two decades of ASD, there is still a lack of empirical knowledge gained from practitioners regarding their worldview related to ways of working in sprints. This exploratory research sought to examine how SDs experience, interpret, and give meaning to work in sprints as part of the adoption of agile methodology principles.

METHOD

This paper presents a study in progress, highlighting the initial findings and ongoing research efforts. Initial in-depth interviews were conducted with four professional SDs who use agile development methodologies in their work. An interpretative phenomenological perspective was used as a heuristic device to describe their experience with working in sprints. The choice of this approach was because the descriptive power of the phenomenological perspective enables a presentation of diverse views and explanations (Smith et al., 2021).

The interviewees work in research and development (R&D) teams of organizations with distinct technological characteristics in the Israeli high-tech industry, known as the start-up nation. Interviewees were asked to describe a typical working day in a software development project based on agile principles. The meanings they attribute to working in sprints were also enquired about, including how they experience the opportunities and challenges it raises. The interviews were recorded with the consent of the participants and transcribed in full. Content analysis was applied to the data by breaking down the text into meaning units and conducting open axial coding (Krippendorff, 2019).

RESULTS

AGILE SPRINT MANAGEMENT

It is understood that, given the changes it enables on the go, the agile sprint methodology is suitable for development projects that require a flexible approach to workflow. In fact, development in agile sprints provides SDs with a methodical and time-bound work template. Agile management, according to an iterative concept, divides clear and well-defined mission objectives, as described below:

Working in sprints requires setting deadlines, prioritizing, and deciding which software feature we will work on. The product department comes with a request list to the R&D department and after prioritization, they present us – the developers – with the development requirements in a kick-off meeting ... Development of a standard feature will usually be conducted in one sprint, and sometimes, when it comes to several features that are related to each other, the development can be spread over several sprints ... Each sprint lasts two weeks, and when it ends, the feature is released to the customer ... The aspiration is to reach a product that is strong, stable, and ripe enough. (SD1)

In other words, the development is done “in a very modular and flexible way ... At the end of the sprint, we examine what can be further improved in the product and what should be preserved” (SD2). The importance of the synchronization process between R&D team members was supported by another testimony:

The head of the team assigns as limited and specific tickets as possible so that we [the SDs] can do big things in small steps ... We have daily and weekly in all the teams for synchronization and an opportunity to find out if and why the product is delayed. When there is justification for this, we move the task to the next sprint and explain the cost involved in the delay. (SD4)

A dashboard for evaluating sprint management in an agile project is shown in Figure 1. The figure illustrates data on the total work hours in the project and includes components like tasks by type, tasks by status, unfinished tasks by priority, story points over time, and so forth.

Sprint Cycles in Agile Development

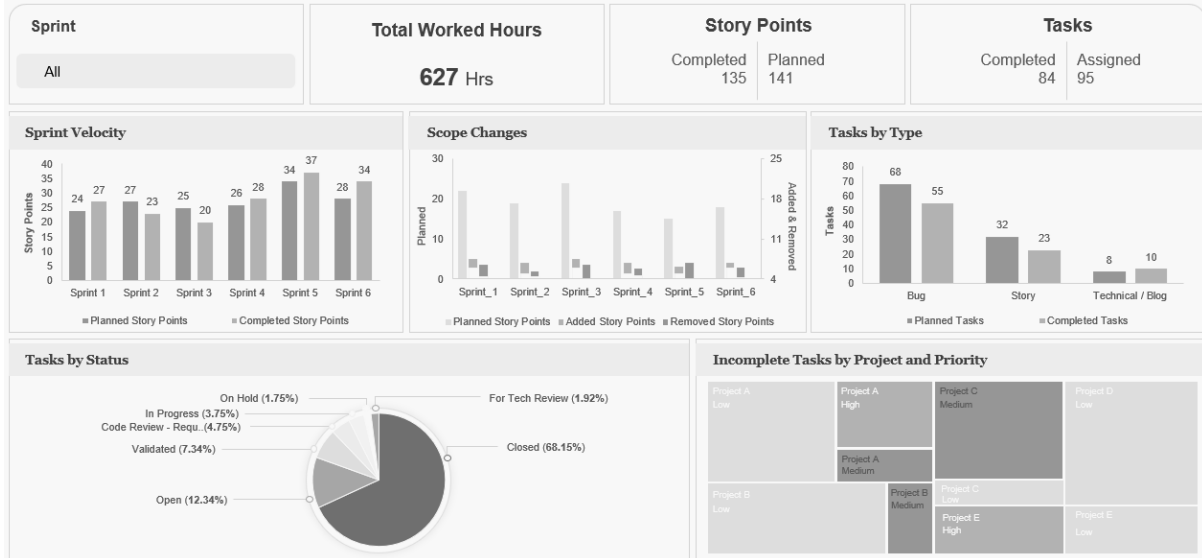


Figure 1. Illustration – agile sprint management summary dashboard

An example of a report that illustrates agile sprint defect management can be seen in Figure 2. This figure includes defects by project, defects by type, defects by status, bug fix time, and so forth.



Figure 2. Illustration – agile sprint defect management summary report

THE VALUE OF DEVELOPING IN AGILE SPRINTS

The interviewees often mentioned the benefits inherent in the adoption of the agile methodology, as shown in the following quote:

The transition we made to work in the agile methodology, and in particular the accelerated adoption of the sprints approach, significantly improved the communication between the SDs and the product people and actually led to an increased knowledge sharing. There is [today] a deeper understanding of the development requirements, the schedules have become defined, and a clear order has been made in the tasks. (SD1)

A similar point of view was described as follows:

The method allows for a very open dialogue with the parties involved in the project and each of them can ‘raise a flag’ when they identify something unusual. This dialogue produces a better product and allows adjustments to be made on the fly. (SD4)

The added value of the ASD was also described in comparison to other development approaches:

Most of the time we work in agile with the belief that it’s a very effective way. Unlike, for example, the waterfall model, where you prepare requirements and work on the complete product from start to finish, in agile the planning is according to specific capacities and in the short term. There is a lot of modularity here, maximum flexibility. Meeting the goals is more realistic ... When working in agile it’s easier to estimate how long it will take to work on each feature and it’s also simpler to make changes. (SD3)

THE CHALLENGES ASSOCIATED WITH WORKING IN AGILE SPRINTS

According to the interviewees, the agile technique adoption forces the development team to hold many work meetings in a way that sometimes has a negative impact on labor productivity:

Due to the method’s characteristics, the working day is dedicated to many meetings, so that we can exchange knowledge between us. But we must remember that if this time had been spared, I could have invested it in de facto development. The need for effective communication in a certain sense detracts from productivity. (SD3)

But the challenges inherent in agile sprint environments do not end only with this, as shown in the following testimony:

Sometimes the speed required by the method and the sense of urgency impair a deep understanding of the [customer’s] needs. The desire to bring the product to the customer’s use as quickly as possible means that only after the delivery, we actually examine the code’s effectiveness and optimize a duplicated code. Only after the release [that is, at the end of the current sprint], we approach to optimize the software code. (SD4).

CONCLUSION

Many companies are challenged by dynamic market environments caused by increased digitization, the expansion of information technologies, and the emergence of the fourth industrial revolution. Therefore, the ability to act flexibly, quickly, and efficiently in product development becomes inevitable for business success. Agile principles play a pivotal role in modern software development (Al-saqqa et al., 2020; Traini, 2022). The ASD strategies are being widely accepted among the software development community (Edeki, 2015).

The paper is crafted for researchers, practitioners, and professionals in the software development industry who are keen on embracing agile methodology principles and effectively managing development projects through sprints. This exploratory study shed light on the meanings that SDs give to working in sprints, relying on the adoption of the principles of the agile methodology. This paper presents the ways of managing a development project in sprints, puts a spotlight on the value of the agile method, and provides a unique glimpse of its challenges.

This work adds to the growing body of research investigating the SDs’ perceptions. The preliminary results develop the theoretical and practical knowledge related to managing agile sprints, with reference to optimal communication, monitoring, control, and knowledge sharing among SDs in agile projects. This empirical knowledge is particularly valuable for administrators who are seeking insight into the internal dynamics of decision-making and timelines within the realm of software development.

This research work has made only a modest beginning in understanding the phenomenon under study. Considering the work in progress, further iterations will contribute additional comprehension, shaping the overall narrative of the study. It is appropriate to expand the research sample to derive additional meaningful insights, which may turn out to be of great value to the ASD theory and practice.

The study's participants were selected from Israeli high-tech companies, which may limit the generalizability of the findings to other cultural and organizational contexts. Future research could consider a more diverse participant selection to provide a broader perspective on the topic. Furthermore, future researchers are encouraged to join in exploring requirements-gathering processes, performance assurance, and challenges of ASD efforts in companies with different characteristics (such as size and domain). Research in these areas may be useful and valuable for strategic planning of effective agile sprint management.

REFERENCES

- Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). *Agile software development methods: Review and analysis*. arXiv:1709.08439. <https://doi.org/10.48550/arXiv.1709.08439>
- Alsaqqa, S., Sawalha, S., & Abdel-Nabi, H. (2020). Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies*, 14(11), 246-270. <https://doi.org/10.3991/ijim.v14i11.13269>
- Annosi, M. C., Foss, N., & Martini, A. (2020). When Agile harms learning and innovation: (and what can be done about it). *California Management Review*, 63(1), 61-80. <https://doi.org/10.1177/0008125620948265>
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for agile software development*. https://ai-learn.it/wp-content/uploads/2019/03/03_ManifestoofAgileSoftwareDevelopment-1.pdf
- Benlian, A. (2022). Sprint zeal or sprint fatigue? The benefits and burdens of agile ISD practices use for developer well-being. *Information Systems Research*, 33(2), 557-578. <https://doi.org/10.1287/isre.2021.1069>
- Boschetti, M. A., Golfarelli, M., Rizzi, S., & Turrlicchia, E. (2014). A Lagrangian heuristic for sprint planning in agile software development. *Computers & Operations Research*, 43, 116-128. <https://doi.org/10.1016/j.cor.2013.09.007>
- Campanelli, A. S., & Parreiras, F. S. (2015). Agile methods tailoring – A systematic literature review. *Journal of Systems and Software*, 110, 85-100. <https://doi.org/10.1016/j.jss.2015.08.035>
- Dingsoyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213-1221. <https://doi.org/10.1016/j.jss.2012.02.033>
- Dyba, T., & Dingsoyr, T. (2009). What do we know about agile software development? *IEEE Software*, 26(5), 6-9. <https://doi.org/10.1109/MS.2009.145>
- Edeki, C. (2015). Agile software development methodology. *European Journal of Mathematics and Computer Science*, 2(1), 22-27. <https://www.idpublications.org/wp-content/uploads/2015/05/Agile-Software-Development-Methodology.pdf>
- Goel, S., & Kathuria, V. (2010). A novel approach for collaborative pair programming. *Journal of Information Technology Education: Research*, 9(1), 183-196. <https://doi.org/10.28945/1290>
- Krippendorff, K. (2019). *Content analysis: An introduction to its methodology*. Sage Publications. <https://doi.org/10.4135/9781071878781>
- Larson, D. (2012). Agile methodologies for business intelligence. In A. A. Rahman El Sheikh & M. Alnoukari (Eds.), *Business intelligence and agile methodologies for knowledge-based organizations: Cross-disciplinary applications* (pp. 101-119). IGI Global. <https://doi.org/10.4018/978-1-61350-050-7.ch005>

- McAvoy, J., & Butler, T. (2009). A failure to learn by software developers: Inhibiting the adoption of an agile software development methodology. *Journal of Information Technology Case and Application Research*, 11(1), 23-46. <https://doi.org/10.1080/15228053.2009.10856152>
- Mishra, A., & Alzoubi, Y. I. (2023). Structured software development versus agile software development: A comparative analysis. *International Journal of System Assurance Engineering and Management*, 14, 1504–1522. <https://doi.org/10.1007/s13198-023-01958-5>
- Project Management Institute. (2017). *Agile practice guide*. <https://www.pmi.org/pmbok-guide-standards/practice-guides/agile>
- Sahin, Y. G., & Celikkan, U. (2020). Information technology asymmetry and gaps between higher education institutions and industry. *Journal of Information Technology Education: Research*, 19, 339-365. <https://doi.org/10.28945/4553>
- Smith, J. A., Larkin, M., & Flowers, P. (2021). *Interpretative phenomenological analysis: Theory, method and research* (2nd ed.). Sage Publications. <https://www.torrossa.com/it/resources/an/5282221>
- Traini, L. (2022). Exploring performance assurance practices and challenges in agile software development: An ethnographic study. *Empirical Software Engineering*, 27, Article 74. <https://doi.org/10.1007/s10664-021-10069-3>
- Zayat, W., & Senvar, O. (2020). Framework study for agile software development via scrum and Kanban. *International Journal of Innovation and Technology Management*, 17(4). <https://doi.org/10.1142/S0219877020300025>

AUTHOR



Dr. Maayan Nakash is a researcher and lecturer in the Technology Management Program in the Department of Management at Bar-Ilan University (BIU), Israel. Maayan is a graduate (summa cum laude) of the Department of Information Science and specializes in WEB technologies and IT. During her Ph.D., she was one of the Presidential Fellows at BIU. She won the Rector's Excellence Awards for Doctoral Students and the President's Awards for publishing articles in international scientific journals. She has extensive experience in mapping needs to improve business performance; modeling, characterization, and analysis of systems; planning strategic solutions and designing systematic methodologies for managing data, information, and knowledge. Her research deals with the implementation of systems, human capital management, digital transformation processes, organizational performance improvement, digital workplace environments, and big data analysis.