Swinburne University of Technology

Gothaer

# ASSIGNMENT 2 CASE STUDY AGILE

## The (Go)SMART way to agility: managing a Scrum subproject in a waterfall environment

Student Name: Phuong Linh Tang

Student ID: 103844450

Course Name: INF30033-IT Program and Project Management

# 1.1- PROS OF SCRUM IN THE CURRENT SITUATION

| Issues (Before adopting Scrum) | Pros of Scum (after adopting the methodology) |
|---|---|
| **The Inflexibility of Waterfall Methodology-** The process is rigid as once plans are finalized and frozen, making subsequent changes becomes undesirable and difficult. | • Facilitates accommodating changes through short sprints and constant feedback<br>=> Allows teams to focus on specific features within manageable periods for quicker adjustments and ongoing reassessment of priorities. |
| • Joe's subproject team struggled to adapt to the rapid **two-week development cycles** required for new increments.<br>• The team was overwhelmed by **the volume and frequency of requirements** in the collab with Alpha Systems. | • Helps teams to focus on completing a predefined set of tasks within a limited duration, typically ranging from one to four weeks.<br>=> Encourages quicker learning, faster adaptation, and continual refinement. |
| **The lack of stakeholders' engagement** - team members often worked in isolation with "little" communication during the execution of the defined plan. | • Enhances collaboration among stakeholders by including sprint planning, daily scrum, sprint review, and sprint retrospective.<br>=> Facilitates ongoing communication, quick problem-solving, and ensure continuous alignment with project goals and stakeholder expectations. |

# 1.2 - CONS OF SCRUM IN THE CURRENT SITUATION

| Issues (After adopting Scrum) | Cons of Scrum |
|---|---|
| • **Sharing the Product Owner role** between two individuals( 20-30% of their time) and do not attend the daily scum and refinment meetings<br>• **Overload of Scrum Master:** Joe- the Scrum Master but also have to handles project management tasks outside his team. | Scrum's effectiveness depends on the experience and collaborative efforts of the team e.g: a lack of commitment, cross-team interaction, and physical team separation can significantly increase the risk of project failure and effect decision-making. |
| • Some team members find **sprint planning meetings overly lengthy and crowded**, indicating potential inefficiencies in meeting management. | Regular meetings can occasionally become a source of frustration for team members. |
| • Since Joe's team is the only team adopts Scrums, they **struggle to coordinate their Agile workflow with other teams following different methodologies**, leading to synchronization issues. | When Scrum teams try to align their work with groups using Waterfall, they face issues coordinating schedules and dependencies, resulting in conflicts and misaligned timelines. (Gren et al., 2014) |

# 2.1- ASSESS THE USE OF SCRUM

I would consider several criteria to assess the use of Scrum in this case study:

01

### Flexibility and Responsiveness to change

Scrum facilitated increased flexibility for the team in responding to requirement changes, which was vital within the dynamic insurance industry. This adaptive capability represents a significant advancement over the traditional method.

02

### Team Collaboration and Communication:

Scrum emphasizes teamwork and communication through daily stand-ups and sprint reviews, enhancing workplace cooperation and stakeholder engagement, likely leading to improved problem-solving and better alignment with project goals.

03

### Integration with non-agile teams

Integrating Scrum within an environment with Waterfall methodology team , it leads to synchronization issues and delays when dependencies arose across teams with differing methodologies.

04

### Role Confusion

The splitting of the Product Owner role between two part-time employees led unclear task allocation and availability issues, which could disrupt the workflow and decision-making processes.

# 2.2- SOME ASPECTS THAT COULD BE IMPROVED:

## The Role of Product Owner

Assigning a dedicated, full-time Product Owner could resolve many of the ambiguities and availability issues. This is a significant role in Scrum and  by having a dedicated individual , the project could considerably improve the process's efficiency and effectiveness.

## Enhanced Integration Techniques:

Developing better integration strategies for teams using different methodologies could alleviate synchronization issues. This might include more comprehensive planning sessions that involve all stakeholders and align sprint timelines as closely as possible.

## Training and Cultural Adaptation:

Additional training and cultural change initiatives could help further embed agile principles within the organization. This would aid in smoothing out many operational hitches related to Scrum adoption, particularly around communication and collaboration practices.

# 2.3- REQUIRED QUALIFICATIONS TO EMBODY A SCRUM ROLE IN THE SUBPROJECT:

1. **The Scrum team**

- Is responsible for their goal setting and accomplishment. Members are individually empowered to set goals and make commitments, yet the responsibility for achieving the final outcome is collectively held by the entire team to enhance cross-functionality and communication both within and outside the team (Schwaber & Sutherland, 2020; Moe et al., 2009).
- Must understand a comprehensive grasp of Scrum principles and practices which is essential to guide effectively and ensure adherence to Agile methodologies. (Baxter & Turner, 2021)

2. **Product Owners** must have exceptional communication and coordination skills to successfully engage teams in strategy development, manage stakeholders, and articulate product needs. Their deep understanding of the product and ability to prioritize user stories are essential for driving project success.(Verwijs & Russo, 2022; Berntzen et al., 2019)
3. **The Scrum Master** distrubutes task and leads the planning and Sprint Review meetings, ensuring that the Scrum team follows the methodology's principles, practices, and rules.(Jiménez et al., 2020)

# 3. Alternative Agile Methodologies

## Kaban

Kanban in Agile software development, it was introduced by David J. Anderson (2011) to assist teams in managing tasks efficiently. Kanban aims to prevent overloading teams with tasks, emphasizing the timely delivery of features based on demand. It uses visual boards to display workflow stages and focuses on limiting work in progress, managing flow, clarifying policies, and fostering continuous improvement and collaboration. (Anderson, 2010)

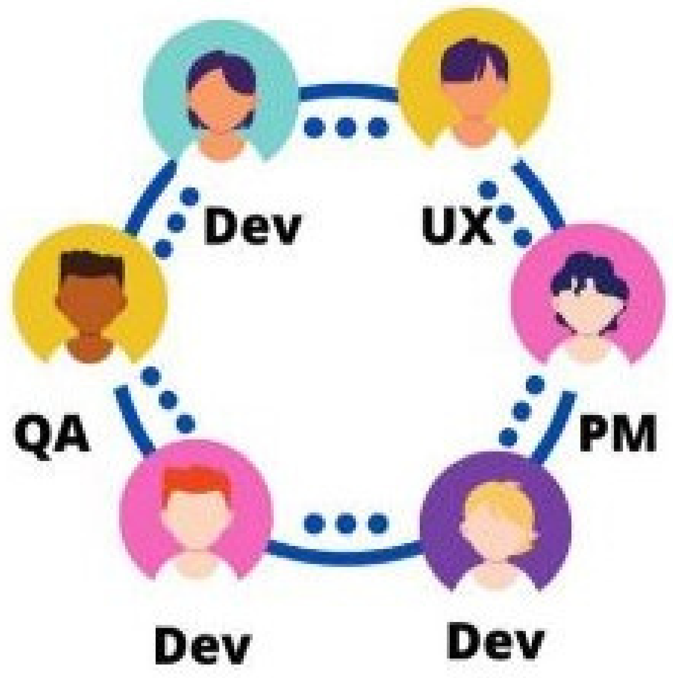
## Extreme Programming (XP)

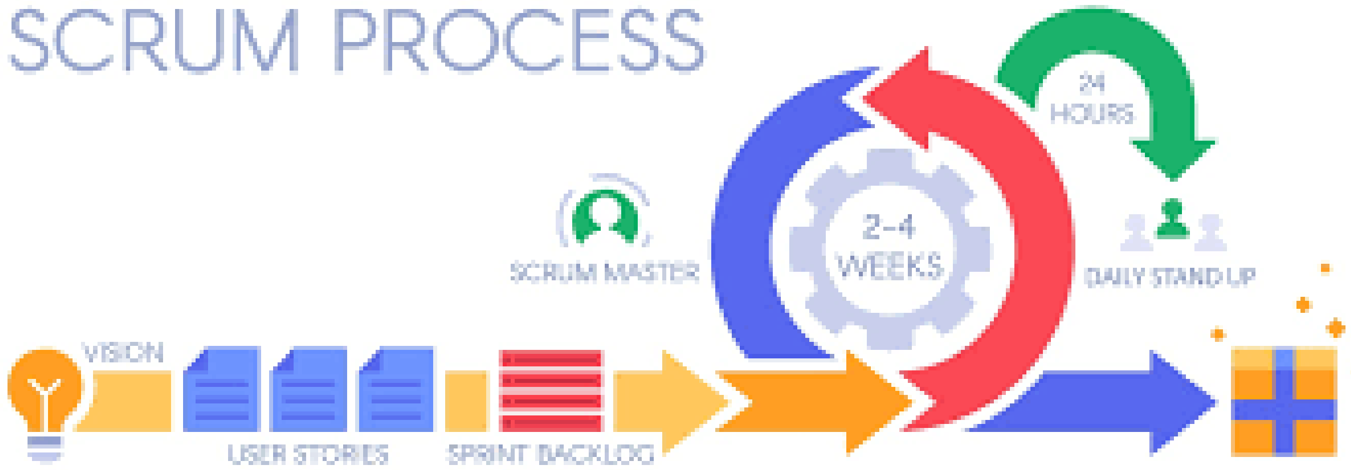
Extreme Programming (XP), which indeed is a framework within the Agile software development model. XP emphasizes producing higher quality software through various practices aligned with Agile principles such as adaptive planning, evolutionary development, continuous improvement, and rapid response to changes. XP also prioritizes customer satisfaction, simplicity, and continuous collaboration between developers and customers. (English, 2002; Goto et al., 2014; van Deursen, 2001)

| ISSUES | KABAN | XP |
|---|---|---|
| The inflexibility | • Based on Just-in-Time (JIT) principles and emphasizing a focus on flow and shorter feedback loops, enables teams to develop features when necessarybut still allow continuous process adaptation and frequent reassessments. (Ahmad et al., 2013; Alqudah & Razali, 2017) | • XP's continuous design and frequent releases encourage adaptability (Shrivastava et al., 2021)<br>• Supports on-the-fly changes and updates from customers as project understanding deepens over time |
| • The rapid two-week development cycles required for new increments.<br>• The voluminous and frequency of requirements | • Allows for categorizing features into smaller pieces, enabling delivery of value in less than one week, even on a daily/hourly basis. (Cuellar, 2011) | • Short, frequent release cycles deliver fully functioning features after each iteration, enhancing responsiveness and customer satisfaction.(Shrivastava et al., 2021) |
| The lack of stakeholders' engagement | • Emphasizes on minimizing lead time results in faster delivery of requested features or changes, enhancing stakeholder involvement and satisfaction.(Shalloway, 2011) | • Constant feedback streamlines design and pinpoints improvement areas.<br>• Involves high engagement and communication and respectful collaboration between customers enhances project success (Beck 2004) |

# 4.1- WOULD SCRUM BE RECOMMENDED USING IN OTHER SUB-PROJECTS?

My answer is **YES** since Scrum might be benfitcial for subprojects because:



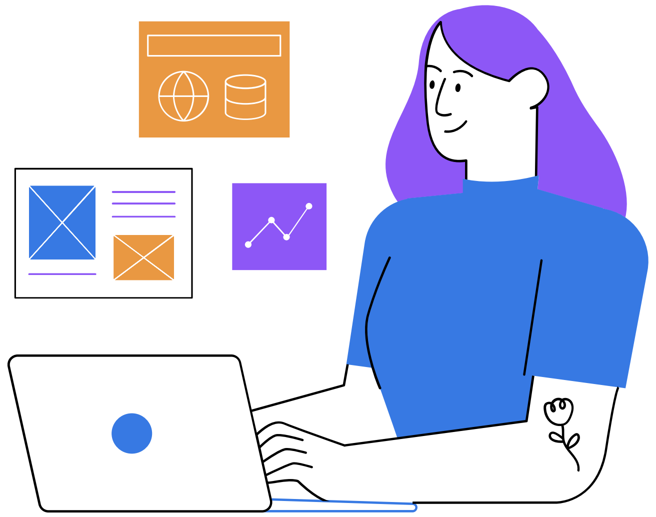


**Flexibility and Adaptability**

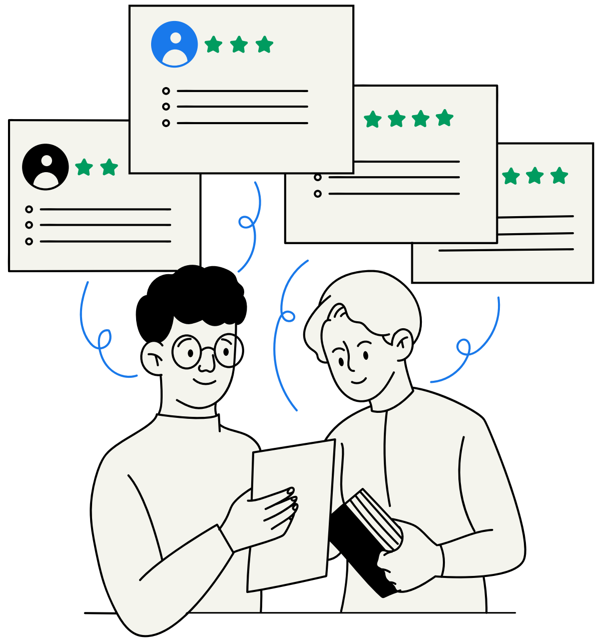
**Cross-functional Team**

- **Feedback loops**
- **Quickly response to change**

# 4.2- WHICH PREREQUISITES MUST BE FULFILLED

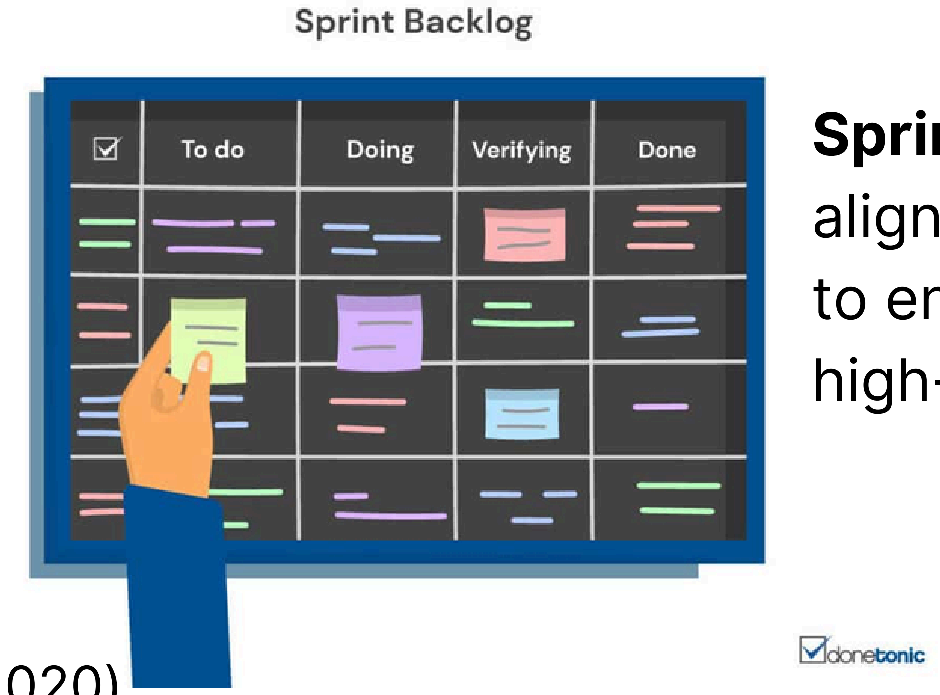


**Well-defined roles** including a Scrum Master, Product Owner, and Development Team, each with a clear understanding of their respective responsibilities.



**Maintaining transparency and regular communication** across all project phases, along with continuous inspection of artifacts and progress

**Continuous and effective interaction with stakeholders** to ensure adaptability to feedback and changes; secure their commitment to the Scrum process and its objectives.



**Sprint Backlog Management** must align with the "Definition of Done" to ensure each contributes to a high-quality product increment.

(Schwaber & Sutherland, 2020)

# 4.3- POTENTIAL RISKS IN OTHER SUB-PROJECTS WHEN USING SCRUM

Coordinating team's Agile workflow with other teams that used Waterfall methodology led to significant delays and conflicts in meeting project timelines, highlighting the risks of isolated Scrum adoption in a mixed-methodology environment.

Skepticism or unfamiliarity with Agile practices among team members can hinder the adoption and effectiveness of Scrum methodologies.

The absence of adequately trained Scrum Masters or Product Owners presented a significant barrier to effectively implementing Scrum

# REFERENCES

1. Ahmad, M. O., Markkula, J., & Oivo, M. (2013). Kanban in software development: A systematic literature review. In Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on (pp. 9-16). IEEE
2. Alqudah, M., & Razali, R. (2017). A comparison of scrum and Kanban for identifying their selection factors. 2017 6th International Conference on Electrical Engineering and Informatics (ICEEI). https://doi.org/10.1109/iceei.2017.8312434
3. Anderson, D. J. (2010). Kanban : successful evolutionary change for your technology business. Blue Hole Press.
4. Anderson, D. J., & Roock, A. (2011). An agile evolution: why Kanban is catching on in Germany and around the world. Cutter IT Journal, 24(3), 6.
5. Bass Julian M., Beecham Sarah, Razzak Mohammad Abdur, Canna Clodagh Nie, and Noll John. 2018. An empirical study of the product owner role in Scrum. In Proceedings of the International Conference on Software Engineering. 123–124.
6. Baxter, D., & Turner, N. (2021). Why Scrum works in new product development: the role of social capital in managing complexity. Production Planning & Control, 1–13. https://doi.org/10.1080/09537287.2021.1997291
7. Beck, K. (2004). Extreme programming explained : embrace change. Addison-Wesley.
8. Berntzen, M., Moe, N. B., & Stray, V. (2019). The Product Owner in Large-Scale Agile: An Empirical Study Through the Lens of Relational Coordination Theory. Lecture Notes in Business Information Processing, 121–136. https://doi.org/10.1007/978-3-030-19034-7_8
9. Coad, P., Lefebvre, E., & De Luca, J. (1999). Java Modelling in Color with UML. Chapter 6.
10. Cuellar, R. (2011). Kanban for Help Desks: Managing the Unplannable [Review of Kanban for Help Desks: Managing the Unplannable]. Cutter IT Journal, 24, 16–23.
11. English, A. (2002). Extreme Programming: It's Worth a Look [Review of Extreme Programming: It's Worth a Look]. IT Pro, 48–50.
12. Goto, T., Tsuchida, K., & Nishino, T. (2014). EPISODE: An Extreme Programming Method for Innovative Software Based on Systems Design. https://doi.org/10.1109/iiai-aai.2014.157
13. Gren, L., Torkar, R., & Feldt, R. (2014). Work motivational challenges regarding the interface between agile teams and a non-agile surrounding organization: A case study. In 2014 Agile Conference, 11-15. IEEE.
14. International Conference on Engineering, Computing & Information Technology ICECIT 2017, Akdeniz University, Antalya, Turkey, 21–23 August 2017; pp. 1–6
15. Jiménez, V., Afonso, P., & Fernandes, G. (2020). Using Agile Project Management in the Design and Implementation of Activity-Based Costing Systems. Sustainability, 12(24), 10352. mdpi. https://doi.org/10.3390/su122410352
16. Moe, N. B., Dingsøyr, T., & Dybå, T. (2009). Overcoming Barriers to Self-Management in Software Teams. IEEE Software, 26(6), 20–26. https://doi.org/10.1109/ms.2009.182
17. Schwaber Ken and Sutherland Jeff. (2021). The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game.
18. Schwaber, K., & Sutherland, J. (2020). The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game.
19. Shalloway, A. (2011). Demystifying Kanban [Review of Demystifying Kanban]. Cutter IT Journal, 24(3), 12-17.
20. Shrivastava, A., Jaggi, I., Katoch, N., Gupta, D., & Gupta, S. (2021). A Systematic Review on Extreme Programming. Journal of Physics: Conference Series, 1969(1), 012046. https://doi.org/10.1088/1742-6596/1969/1/012046
21. van Deursen, A. (2001). Program comprehension risks and opportunities in extreme programming. Software Engineering [SEN]. CWI.
22. Verwijs, C., & Russo, D. (2022). A Theory of Scrum Team Effectiveness. ACM Transactions on Software Engineering and Methodology, 32(3). https://doi.org/10.1145/3571849