Nowadays, the software industry forms the very foundation on which our modern civilization is built. They have become the engine driving mankind toward the future. Likewise, software development methods are also progressing, with new principles and techniques being invented and improved constantly. This essay will discuss and compare Agile (Scrum in particular) and Lean software development, as well as how they can support each other.

Software development models illustrate how to traverse the complicated process of software development. The quality, timeliness, cost, and ability to satisfy clients’ expectations of a software product are all influenced by the model chosen (Shiklo, 2019). The most common models employed in modern professional software development are the Agile and the Lean method.

The term Agile emerged in the 2001 ‘Manifesto for Agile Software Development’ when 17 prominent software developers gathered in Snowbird, Utah to discuss efficient development techniques. Agile could be interpreted as an umbrella term to describe different sets of frameworks and practices based on the manifesto (Agile Alliance, 2013). Because of this common root, all Agile models agree that incremental development and implementation of software should be the primary technique. Sommerville (2015, pp.76) noted that they have shared values, which are 1) individuals and interactions over processes and tools, 2) working software over comprehensive documentation, 3) customer collaboration over contract negotiation, and 4) responding to change over following a plan. Agile also includes 12 principles, which are summarized in this table by Gill (2013):



Agile practices place a heavy emphasis on optimizing the time required to deliver a product through incremental development and continuous internal communication, which enables high adaptability to alternating business requirements. Sommerville said that Agile models are usually better suited to small, medium-sized companies making commercial software and tailored systems within a business. However, agile methods do not perform well when it comes to the development of large, lifetime systems installed for an external customer. He explained that there could be numerous complications to the legal approach, obstacles to tight communication, and difficulty in maintaining software (2015, pp.89).

On the other hand, Lean development originated from the manufacturing field, adapted from Toyota Production System, and focuses on eliminating ‘waste’ during development. Poppendieck and Cusumano found out that this technique developed by Toyota effectively cuts down in-process and final inventories by optimizing the flow through the ‘pull’ instead of the ‘push’ philosophy. Lean methods do not concentrate solely on the development process, but instead highlighting 7 principles: 1) optimize the whole, 2) eliminate waste, 3) build quality in, 4) learn constantly, 5) deliver fast, 6) engage everyone, and 7) keep getting better (2012). In short, Lean relentlessly eliminates anything that doesn't add value and only includes the utmost necessities at the time of execution. It reduces time planning what is ‘expected’ to be required in the future as customers’ requirements change over time and removes inefficient ways of developing – like multitasking – to deliver products faster.

As such, the Lean model, in contrast to Agile, prioritizes the system being developed itself. Lean adopts a comprehensive approach to the product and ensures that each developer is respected and entrusted with working on their fields of expertise. This method involves learning constantly, so delaying decisions to discover more information is essential. Furthermore, the Lean technique focuses on ‘built-in quality’, which means having to revert to the previous stage of development is less likely as stability of the product is already taken into consideration during development.

Whether Lean is Agile and Agile is Lean remains a matter of contention. On the surface, they seem to be different from theories to applications. However, both models have a wide range of similarities, since they all aim to improve product quality via close communication, iterative deployment, and developers’ freedom. A combination of Agile and Lean is a common approach. Fronczak noted that characteristics of Lean development like value stream management, optimizing the whole, and customer validation go hand in hand with Agile’s continuous delivery, test-driven development, ceremonies (Scrum in particular: sprint planning, daily stand-ups, sprint demos, etc.), co-location. Blending the best practices and learnings of each guarantee a fast, adaptable, and smooth delivery of software products and consistent clients’ satisfaction (2021).

Nevertheless, a hybrid model between Agile and Lean still suffers from limitations usually seen in Agile development. Rodríguez et al. (2014) pointed out that whilst teams’ autonomy is relatively easy to accomplish, scaling this combined Agile and Lean method for large corporations is met with obstacles such as flexibility, reducing waste, and hierarchical management.

In conclusion, both models have a lot of advantages over older methods. In the software industry, a combination of these 2 models is used to achieve maximum efficiency. Choosing the right model is critical since the product must be supplied on time and meet the necessary quality standards.

# References

SHIKLO, B., 2019. *8 Software Development Models: Sliced, Diced and Organized in Charts* [online]. ScienceSoft. [viewed 1 October 2021]. Available from: https://www.scnsoft.com/blog/software-development-models

AGILE ALLIANCE, 2013, *What is Agile?* [online]. Agile Alliance. [viewed 1 October 2021]. Available from: https://www.agilealliance.org/agile101/

BECK, K. et al, 2013. *Manifesto for Agile Software Development.*

SOMMERVILLE, I., 2015. *Software Engineering.* 10th ed. Harlow: Pearson Education Limited.

GILL, S., CHANA, I., 2013. Introducing Agility in Cloud Based Software Development through ASD. *International Journal of u- and e- Service, Science and Technology* [online].**6**(5), pp. 191-202. [viewed 1 October 2021]. DOI:10.14257/ijunesst.2013.6.5.17

POPPENDIECK, M., CUSUMANO, M. A., 2012, Lean Software Development: A Tutorial. *IEEE Software* [online]. **29**(5), pp. 26-32. [viewed 1 October 2021]. DOI:10.1109/MS.2012.107

FRONCZAK, S., 2021, *How Lean and Agile Relate and How You Can Win by Using Both* [online]. Plutora. [viewed 1 October 2021]. Available from: https://www.plutora.com/blog/how-lean-agile-relate

RODRÍGUEZ, P., KUJAVA, P., PARTANEN, J., OIVO, M., 2014, Combining Lean Thinking and Agile Methods for Software Development: A Case Study of a Finnish Provider of Wireless Embedded Systems. In: University of Hawaiʻi at Mānoa ed. 2014 47th Hawaii International Conference on System Sciences. Waikoloa, USA, 2014. Washington, DC: IEEE Computer Society, pp. 4770-4779. DOI: 10.1109/HICSS.2014.586