Nowadays, the software industry forms the very foundation on which our modern civilization is built. They have become the engine driving humanity toward the future. In software engineering, the architecture of a system is essential as it forms the foundation to build larger software. This essay will discuss the role and importance of software architecture in Agile development.

In Agile development, defining and designing the architecture of the system is usually considered the early stage of the development process. According to Sommerville, architectural design is the blueprint for a system, primarily concerning the organization method and the overall structure of a software system. These structural models could be either static – showing design structure – or dynamic – showing system organization during execution (2015). Architecture usually follows patterns, such as layered, repository, client-server, etc., which could be further adapted to document a new architectural design or reuse for other subsequent projects with similar characteristics. Therefore, software architecture forms the foundation of software systems and enables numerous beneficial traits in the long term. Sommerville (2015) noted that these properties include performance, security, availability, maintainability, and reusability. Therefore, we could say that software architect plays a crucial role in the quality of a software product. The steps to design system architecture are not fixed as they depend on different scenarios. Nevertheless, Sommerville (2015) found out that some common activities are 1) define the context and the external interactions with the system, 2) design the system architecture, 3) identify the principal system objects, 4) develop design models, and 5) specify object interfaces. Therefore, it remains correct to assume that several phases must be followed, and the terminating documentation should be precise and unambiguous.

The process of design architecture is usually done by a software architect, who is also an experienced programmer (Marić et al., 2016). Software architects offer value to clients and provide support to developers, allowing direct interaction between them. Marić et al. suggested that architects’ responsibilities should be based primarily on their knowledge and expertise at the beginning of the project so that they could be indispensable members of the team (2016). To achieve this, architects should participate in meetings with stakeholders and acknowledge problems first-hand. Hopkins and Harcombe (2014) noted that to ensure quality and efficiency, software architecture requires that architects analyze problems from their very roots and diverse perspectives. Architects must identify functional and non-functional requirements clearly to produce concise and compact architecture documentation, preventing future fatal errors from popping up and making post-installation maintenance easier. Furthermore, software architects act as a linking bridge between two important concepts of modern software development: architecture and Agile.

In the software industry, many believe that there exists a conflict between architectural design and Agile development, arising from the minimalist principles of Agile and the advocate for planned, well-defined architectural documentation. In traditional software development (e.g., Waterfall), the step of architecture design occurs after requirements specifications and before initial development and is accompanied by detailed documentation. On the other hand, Agile seems to be the complete opposite, where rapid and flexible implementation, transforming process-centric development to being human-centric (Hadar and Sherman, 2012). Abrahamsson et al. claimed that the conflict between them is ‘a false dichotomy’ since Agile criticizes software architecture for redundant ahead of time planning, but a path of reconciliation had been found. He said that Agile methods do not distance themselves from architecture concepts, but neglect how to identify architecturally significant requirements, perform incremental architectural design, validate architectural features, and so on (2010). Blair et al. (2010) noted that in the context of Agile development, architects should be able to figure out these hidden features and serve the development team in a supportive and informative manner, guaranteeing an architecturally applicable system. Architects in Agile might delay their decisions until the later phases of the development process to better adjust system architecture to changing requirements. Booch (2010) said that architecture itself does not present an obstacle to Agile but only acts just enough to assist development - “not so much that we prematurely and unnecessarily bind decisions, yet not so little that we avoid tackling the big problems”. Agile architects differ from their traditional colleagues as they spread architectural design tasks throughout the development process, thus reducing effort put in planning up-front and contributing more to activities that improve architectural integrity. Therefore, software architect has become an emergent subject thanks to its incorporation into Agile development, rather than the planned property of the system arising from the traditional model.

In conclusion, software architecture design is an integral and indispensable part of software development as it lays the foundation for later components of the system to be built on. The primary duty of software architects remains designing the architecture; therefore, they play a crucial role in connecting developers and customers. Thanks to the work of these architects, Agile architecture design has become an emergent alternative to its plan-driven predecessor.

# REFERENCE LIST

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

MARIĆ, M., TUMBAS, P., 2016. The Role of the Software Architect in Agile Development Processes. *Strategic Management* [online]. **21**(1), pp. 16-22. [viewed 17 October 2021]. Available from: https://www.researchgate.net

HOPKINS, R., & HARCOMBE, S. (2014). Agile Architecting: Enabling the Delivery of Complex Agile Systems Development Projects. In: BABAR, A. l., BROWN, A. W., MISTRIK, I. ed. *Agile Software Architecture: Aligning Agile Processes and Software Architectures*. New York, NY: Elsevier, pp. 291-314.

HADAR, I., SHERMAN, S., 2012. Agile vs. plan-driven perceptions of software architecture. 2*012 5th International Workshop on Co-operative and Human Aspects of Software Engineering (CHASE)*.Zurich, Switzerland. New York, NY: IEEE, pp. 50-55.

ABRAHAMSSON, P., BABAR, M. A., & KRUCHTEN, P., 2010. Agility and Architecture: Can They Coexist? *IEEE Software* [online]. **27**(2), pp. 16-22. [viewed 17 October 2021]. Available from: https://ieeexplore.ieee.org

BLAIR, S., WATT, R., & CULL, T., 2010. Responsibility-Driven Architecture. *IEEE Software* [online]. **27**(2), pp. 26-32. [viewed 17 October 2021]. Available from: https://ieeexplore.ieee.org

BOOCH, G., 2010. An Architectural Oxymoron. *IEEE Software* [online]. **27**(5), pp. 95-96. [viewed 17 October 2021]. Available from: https://ieeexplore.ieee.org

   
 