

# Traditional Challenges of Teaching Software Architectures

Mirja Assyne\*

Department of Computer Science, Aalto University, Finland

**Corresponding author:** Mirja Assyne, Department of Computer Science, Aalto University, Finland, Email: mirjaassynehamid23@gmail.com

**Received date:** December 28, 2022, Manuscript No. IPACSIT-23-15953; **Editor assigned date:** December 30, 2022, PreQC No. IPACSIT-23-15953 (PQ); **Reviewed date:** January 13, 2023, QC No. IPACSIT-23-15953; **Revised date:** January 20, 2023, Manuscript No. IPACSIT-23-15953 (R); **Published date:** January 30, 2023, DOI: 10.36648/2349-3917.11.1.2

**Citation:** Assyne M (2023) Traditional Challenges of Teaching Software Architectures. Am J Compt Sci Inform Technol Vol: 11 No: 1: 002.

## Description

Software architecture has received a lot of attention from academia and industry, and numerous educational establishments include software architecture topics in their curriculum. Nevertheless, teaching software architectures remains challenging. It necessitates teamwork, sufficient complexity, a realistic setting, and extensive coaching. The software architecting community has encountered a new obstacle as a result of the growing popularity of agile software development practices. Common architectural design activities receive very little attention in the majority of agile methodologies. Kent Beck, for instance, believes that architectures emerge and change in everyday design. Predict that software architecture will be recognized as an essential component of agile software development's foundation. The second most pressing issue that agile practitioners face is the amount of architectural work required for agile projects.

## Software Architecture Design and the Documentation

Approaches, visions, and experiences have been published by researchers and practitioners who have accepted this challenge; surveys on how agile development methods can or do approach software architecting. It would appear that there is a widespread agreement regarding the significance of paying particular attention to software architectures in agile projects. In agile projects, however, there are various approaches to architecting. According to Jan Bosch, the community is moving away from quality-based architecture and toward user-driven architecture. In accordance with the agile principles, teams should initially concentrate on the requirements that will provide the client with value rather than the architecturally significant requirements, recognizing that the initial architecture might not be the best one. Given the advantages gained in terms of agility and customer satisfaction, he believes that the cost of constant refactoring is manageable. Waterman and others the same conclusion in their research. Friedrichsen looks at the concept of emergent architecture and comes to the conclusion that while it is a good approach for creating detailed architectures, it does not adequately support critical architectural activities like choosing architectural choices,

creating high-level designs, and validating architecture. The literature has not yet sufficiently addressed the connection between software architectures and agile methods in education. Cleland-Huang and others demonstrate a method for treating software architectures in educational agile projects. The approach ignores the actual dynamics of an agile project and instead focuses on the architecture design phase, specifically the role of stakeholders. In our course, students work in teams to create a software system using an agile scrum methodology. System design is one of the skills they need to demonstrate in this course. In the past, we required a software architecture design and the documentation for it, but we did not provide any instructions on how to do this in an agile project. Most of the time, the students approached this activity as a Big Design Up Front or end-of-project documentation activity.

## Software Architecting in Waterfall Projects

The agile principle of working software preceding comprehensive documentation drove the first behavior, while prior knowledge of software architecting in waterfall projects influenced the second. The first kind of group didn't want to learn more about software architecture because they were too focused on the development process. In order to pass the course, the architecture was discussed hurriedly and superficially in the teams at the beginning of the projects and documented at the conclusion of the projects without realizing the purpose or the advantages of architecting in an agile project. Students' software architecting abilities were graded incorrectly and erroneously due to the absence of an explicit approach to software architecture in agile projects. We design and introduce a software architecting approach to enhance our course. The method is presented in this paper. The traditional challenges of teaching software architectures, such as realistic context, teamwork, sufficient complexity, and extensive coaching, must be addressed by an agile software architecting approach. In addition, architecting and agile activities must be seamlessly integrated without going against agile principles or practices. As a result, taking into account our educational context, we design our strategy using publications on software architecting in agile development and teaching software architectures.