Architecture evaluation is a valuable, useful, and worthwhile instrument for managing risks in software engineering. It provides confidence for decision-making at any time in the lifecycle of a software system. This chapter motivates architecture evaluation by explaining its role and its initiators, and by pointing out its benefits. Architecture evaluation requires investments, but saves time and money (if done properly) by preventing wrong or inadequate decisions.

1.1 What Is the Point?

Q.001. What Is Architecting?

Architecting, in its core essence, is the creative activity of software engineers making principal design decisions about a software system to be built or to be evolved. It translates concerns and drivers in the problem space into design decisions and solution concepts in the solution space.

Architecting is the process of coming up with some kind of solution for some kind of problem. Key to success for the architect is awareness of the problem to be solved and the design of an adequate solution considering given context factors. Consequently, any principal design decision made in advance is an investment in software engineering: the design decisions capture the output of reasoning based on facts, assumptions, and experiences and prescribe an implementation plan to satisfy the desired properties of the software system.

As a matter of fact, every software system has an architecture. The question is whether it is known or not, whether it has been designed proactively and intentionally or has emerged accidentally, and whether the design decisions have been
explicitly documented or are only known and communicated implicitly. Therefore, architecting denotes the process of deliberately designing, using, communicating, and governing the architecture of a system.


Because it saves time and money (if done properly) by preventing repeated trial and error.

The architecture offers an escape from the complexity trap. It systematically provides abstractions to enable coping with the complexity of modern software systems. Common software engineering practice is to use software architectures as the central vehicle for prediction, analysis, and governance over the whole lifecycle of a software system. The architecture enables the architects to reason about the pros and cons of design decisions and alternatives. Any principal design decision, and thus architecting itself as an activity, is always accompanied by the potential risk of being wrong or turning out to be extremely expensive.

The architecture enables making the right trade-offs between functionality and quality attributes. It assures that the technical aspects of realizing business goals are met by the underlying software system. Without investing effort into software architecture, it is unlikely that a software system with adequate quality (e.g., maintainability, performance, security, etc.) can be developed in time and on budget.


Architecture evaluation is a crucial assessment technique for mitigating the risks involved in decision-making. It contributes to the typical assessment questions depicted in Fig. 1.1. Note that architecture evaluation does not aim at answering the question “Is this design (or the decisions leading to it) good or bad?”. It rather evaluates whether the architecture is adequate to address the stakeholder concerns or not. However, having a well-designed architecture is just half the rent. The best-designed architecture does not help at all if it is not properly realized in the source code (the so-called drift between intended and implemented architecture). Hence, architecture evaluation further aims at making sure that the right design decisions have been manifested correctly in the implementation. The architecture is only of value if the implemented system is built as prescribed by the architecture; otherwise, all investments made into architecting become delusive and useless.