

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.

Q.002. Why Invest in Software Architecture, Which Is Only an Auxiliary Construct in Software Engineering?

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.

Q.003. What Is the Role of Architecture Evaluation in Software Engineering?

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.

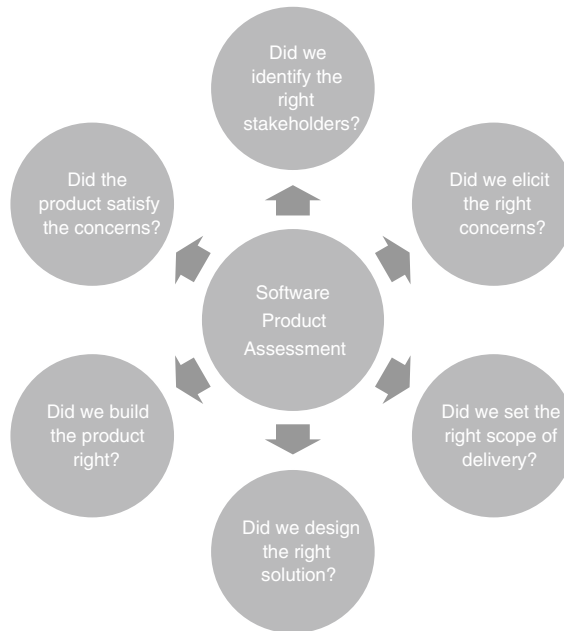


Fig. 1.1 Software product assessment questions. © Fraunhofer IESE (2015)

Consequently, the mission of architecture evaluation is twofold: (1) to determine the quality of the (envisioned) software system and (2) to determine the quality of the (auxiliary) artifacts created during architecting or derived from the architecture.

Architecture evaluation supports informed decision-making: It creates awareness of risks and allows understanding potential trade-offs and side effects in decision-making. It aims at delivering quantitative and/or qualitative facts as input to decision-making. The architecture evaluation reveals design decisions (already made or under discussion) to address concrete stakeholders concerns and make them explicit. This enables challenging these design decisions and reasoning about their adequacy for addressing the stakeholder concerns. Risks associated with the design decisions can be made explicit, either to discuss mitigation measures or to find an alternative solution that fits the given context.

Architecture evaluation supports tracking of the realization of decisions made: Architecture evaluation creates awareness of drifts between the intended architecture (as designed) and the implemented architecture (as codified). The implemented architecture is typically not visible directly; rather, it is usually buried deeply in the source code and has to be extracted by means of reverse engineering or architecture reconstruction techniques. Since the architecture describes not only static artifacts of a system at development time, it might be necessary to mine for information from the system behavior at runtime. Risks associated with drift are detected and support the tracking of decisions made.

Q.004. What Are the Benefits of Architecture Evaluation?

The key benefit of architecture evaluation is to deliver guidance—for making business decisions, for making technology decisions, for controlling product quality, and last but not least for managing the evolution and migration of software systems. Any decision comes with the risk of being inadequate: thus, architecture evaluation aims at sustaining the investments made.

The obvious advantage of an architecture evaluation is, of course, the subsequent improvement of the architecture as such. Given the fact that the architecture (or the design decisions made) often determines the success or failure of arbitrary projects (e.g., software construction, software acquisition, software customization, retirement), it is clear that effort should be spent in every project on getting the architecture right. Architecture evaluation helps to avoid running into problems with respect to quantity, quality, budget, and/or time. This stresses the importance of architecture in general and architecture evaluation as a means for assuring its quality in particular. Consequently, the need for a systematic approach for architecture evaluation is obvious—and it is no surprise that architecture evaluation is considered one of the software engineering best practices in both research and industry. Numerous success stories in industry are proof of the crucial role of architecture evaluation in reaping benefits for risk management.

Besides, architecture evaluation provides other positive side effects: Its results can be the basis for improving the documentation of the software system under evaluation. Implemented violations of architecture decisions can be revealed as well and can be refactored afterwards. Furthermore, the organization's awareness for the architecture is raised (at least for the stakeholders directly or indirectly involved in the evaluation).

In summary, the advantages of architecture evaluation are significant. We believe it should be part of the quality assurance portfolio of any software development organization. However, the return on investment for evaluating architecture is only positive if the results can be achieved with adequate effort and if fast feedback is provided regarding the design decisions made with respect to the achievement of stakeholder concerns.

Q.005. Who Should Ask for an Architecture Evaluation?

There is no single stakeholder (see Fig. 1.2) who is typically interested in an architecture evaluation. Depending on the overall situation of the development and the evaluation goals, very different stakeholders may ask for an architecture evaluation.

Stakeholders who own an architecture evaluation may be from the company that develops the software system or from another company. In the same company, the owner of an architecture evaluation may be the architect or the development team itself, or higher development management. In particular, architects of a system should be interested in evaluating their own architectures. However, other

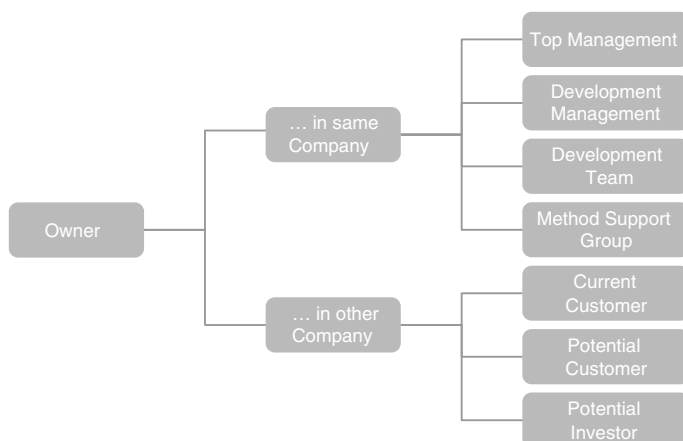


Fig. 1.2 Typical initiators of architecture evaluations. © Fraunhofer IESE (2014)

stakeholders may also be interested: for example, method support groups that intend to introduce and manifest architecture practices for improving overall software development quality. In addition, top management stakeholders may be interested in an architecture evaluation when they have to make decisions about software that will have a far-reaching impact.

On the other hand, stakeholders from external companies might also be interested in architecture evaluations. Potential customers or potential investors might be interested in the quality of a system and how adequate it is for the customer's purposes and requirements in the context of product selection and risk mitigation. Current customers have to make decisions about the future of acquired software systems: Is there enough confidence that the system will evolve in the right direction? What will happen if the customer's own context changes—for example, if its customer base strongly increases? Will the acquired system still be adequate or would it need to be replaced?

Q.006. Who Executes Architecture Evaluations?

Determining who is best suited for an architecture evaluation strongly depends on the goals of the evaluation.

An architecture evaluation can be done by very different people. Software architects who design a system should continuously challenge their own design decisions. Whenever they make a decision, they directly reflect on how well the tackled drivers are fulfilled by the decision they just made as well as those they made earlier. They should also reflect on how the current decision affects other drivers. These continuous evaluations can be augmented by the opinion of other people with direct involvement in decision-making, such as developers or product managers.

As architects are human beings, they tend to overlook deficiencies in their own work. Thus, it is often useful to get neutral people involved in the performance of an architecture evaluation at critical points. These neutral people may be from the same company: Internal auditors perform architecture evaluations as a (regular) quality engineering instrument. Internal auditors may be other architects, technology experts, or experienced developers. They typically have in-depth domain knowledge, are at least partially aware of the architecture/system history, and are potentially familiar with the overall vision and roadmap of the software system, its business goals, and the requirements to be satisfied in the next releases. Internal auditors may face the challenge of being low priority when requesting information or they may be ignored when summarizing the results and pinpointing identified risks.

However, sometimes it is even better to get a completely external opinion from external auditors. External auditors provide an objective evaluation delivered by qualified and experienced people. External auditors often apply a more formal approach for systematically eliciting and collecting information. They have predefined templates for audit presentations and reports that create higher visibility and awareness of the results. They avoid the “prophet in one’s own country” symptom. In case of conflicts, external auditors can be neutral towards all involved parties.

Q.007. What Is the Return on Investment for Architecture Evaluations?

In our experience, architecture evaluation is a worthwhile investment that pays off. Any wrong or inadequate decision prevented or any risk avoided may save sufficient time and/or effort to pay for the evaluation. The cost for reversing fundamental or business-critical design decisions typically outweighs the investment required for architecture evaluation.

In more than 75 architecture evaluation projects in the past ten years in which we performed architecture evaluations or were part of a team of external auditors, we received—without any exception—positive feedback about the results and the output. In most cases, the cost for the auditing contract consumed less than one percent of the overall project budget. In most cases, the architecture evaluation results were delivered within a few weeks and were perceived as valuable input to decision-making by the sponsor of the evaluation. Architecture evaluation results are always expected to be delivered immediately as they deal with critical stakeholder concerns. This means that the “time to evaluate” is in many cases a critical factor for the overall success of the architecture evaluation.

In a nutshell, architecture evaluations serve to improve the architecture and mitigate risks. They identify problems, allow early and quick corrective reactions, and provide valuable input to decision-making for future-proof systems. Architecture evaluation results can be achieved rapidly with comparably low effort and provide fast feedback regarding stakeholder concerns. Avoiding just a single risk can already save more time and effort than what is required for conducting the architecture evaluation.

1.2 What Mistakes Are Frequently Made in Practice?

Having no idea about architecture evaluation at all.

If stakeholders do not know what an architecture evaluation is (i.e., if they are not aware of its benefits, procedures, and limitations), they are unable to use architecture evaluation as an instrument for quality assurance. This means risks may remain uncovered and crucial design decisions may be realized without critical reasoning and challenging of the decision.

This book and in particular Chap. 3 will allow the reader to learn about the key issues and characteristics of architecture evaluation. Stakeholders can use the detailed descriptions and sample cases used throughout this book to match them to their particular situations. We consider architecture evaluation to be a crucial instrument for assessing software products and believe that it should be part of the quality assurance portfolio of every software development organization.

→ Questions [Q.003](#), [Q.004](#), [Q.014](#), Chaps. 3 and 4.

Not being able to engage in an architecture evaluation for a particular case.

Stakeholders might be aware of the power of architecture evaluation and currently have a concrete case for architecture evaluation at hand, but they might not be able to engage in the architecture evaluation. The advocates for an architecture evaluation might not be able to convince management, or technical stakeholders might not be willing to participate.

→ Questions [Q.028](#), [Q.029](#), [Q.030](#), [Q.100](#), [Q.101](#).

Not having the skills or the right people to execute an architecture evaluation.

Stakeholders willing to go for an architecture evaluation might face the challenge of identifying a skilled auditor for the task. Instead of starting immediately with a good enough candidate, they might spend too much time and effort on selecting an auditor. We experienced more than ten cases with long preparation phases prior to the actual architecture evaluation where the patient eventually died on the way to the hospital (before we could start evaluating, the project was canceled due to inadequate architecture). Being successful in evaluating

architectures requires skills and experiences and is driven by the stakeholders' level of confidence. To introduce architecture evaluation in an organization, one should think big, but start small. Incremental and iterative rollout of architecture evaluation promises to be the path to success if accompanied by coaching of architecture evaluation skills.

→ Questions [Q.006](#), [Q.098](#), [Q.113](#).



<http://www.springer.com/978-3-319-34176-7>

Pragmatic Evaluation of Software Architectures

Knodel, J.; Naab, M.

2016, XXIII, 154 p. 40 illus., Softcover

ISBN: 978-3-319-34176-7