In this chapter, we will present what architecture evaluation is and what it consists of. We will break down the overall method of architecture evaluation into five clearly delineated checks: (1) checking the integrity of the drivers, (2) checking the solution adequacy of an architecture, (3) checking the architecture documentation quality, (4) checking the compliance of the implementation with the architecture, and (5) checking the code quality in general. We will introduce confidence levels as a response to the risk-driven idea of architecture evaluation: we only want to invest as much as needed to gain enough confidence. We will show how evaluation results can be interpreted, aggregated, and represented to senior management by mapping them to a color-coded rating scale.

### 3.1 What Is the Point?


The mission of architecture evaluation is to determine the quality of the (envisioned) software system and the quality of the (auxiliary) artifacts created during architecting or derived from the architecture.

Architecture evaluation aims at achieving confidence in the (architectural) design decisions made about the software system with respect to an evaluation question. It summarizes all activities aimed at answering critical concerns regarding the software system, its environment, its evolution (debts from the past and anticipated needs for the future), and its artifacts documenting and manifesting the design decisions.
The evaluation of the (envisioned) system quality analyzes the following questions:

- How well are the stakeholders’ requirements (driving the architecture) understood and agreed on?
- How well are the solution concepts and design decisions of the architecture suited to adequately addressing the requirements?
- How well are the solution concepts manifested in the implementation?

The evaluation of the artifact quality (documentation, models, source code) analyzes the following questions:

- How well is the documentation of the architecture structured and how consistent is it?
- How well is the source code of the software system structured and how readable is it?

Thus, architecture evaluation requires several checks and ranges from requirements via the architecture to the implementation/system level:

- Check for ambiguities, unclarities, or drift in stakeholder concerns and derived architecture drivers.
- Check for flaws and issues in solution concepts and identify inadequate architecture decisions.
- Check for problematic deficiencies and inconsistencies in the system’s architecture documentation.
- Check for drift between a system’s intended architecture and the architecture as realized in the implementation.
- Check for anomalies in the source code with respect to best practices, quality models, style guides, and formatting guidelines.

**Q.015. What Does Our Architecture Evaluation Method Consist of?**

Our approach to architecture evaluation applies a set of checks—driver integrity check, solution adequacy check, documentation quality check, architecture compliance check, and code quality check—to deliver answers with a certain level of confidence to the stakeholder concerns.

Figure 3.1 provides a conceptual overview of our approach for architecture evaluation, which is called RATE (Rapid ArchiTecture Evaluation). It is not intended to be yet another architecture evaluation method. Rather, it is a compilation and collection of best practices of existing evaluation approaches tailored towards pragmatic (or rapid) application in industry. We derived this approach from