itestra
be excellent

Operationalized Software Architecture Documentation

☒ Bachelorthesis
☒ Masterthesis

Context

Adequate architecture documentation can ease the maintenance and further development of a software system, help to keep the architecture consistent over time and serve as a common knowledge base for changing or distributed teams. However, documentation tends to become forgotten and outdated, in particular if it is kept separate from the source code, such as in word documents or wikis. Also, violations of the documented architecture are likely to go unnoticed unless explicit manual code reviews are performed.

Several aspects of a software architecture, such as the allowed and forbidden dependencies between components or the naming rules, can be checked by automated tools. This continually reminds developers of those rules and ensures adherence. However, the machine-readable rule definition usually forms *another* artifact (or one for each tool) besides the source code and the human-readable documentation, thus allowing for new inconsistencies.

Goals

The idea behind this thesis is to have machine-readable artifacts that serve *both* as a definition for automated analyses of the system *and* to document the architecture and its rationale. These artifacts could be kept close to the code e.g. in annotations or configuration files and include explanations of the rules. A tool could regularly aggregate all artifacts into a human-readable overall document*. This would increase the chances of the documentation to be noticed and updated and allow the rule violation messages for the developer to be augmented with the corresponding rationale.

The goals of this thesis are

- to gather the most important aspects of a software architecture (focus: business information systems) that need to be documented – through literature research and interview of itestra engineers.
- to determine which of those aspects can be checked in an automated way and for which tools already exist.
- to create a prototypical definition of architecture artifacts as described above and a prototypical tool that interprets it and checks existing sourcecode (or further items) for conformance.
- prototypically generate a human-readable overall document, e.g. HTML, from the architecture definition artifacts and further sources such as JavaDoc comments.
- to evaluate the approach on an industrial software project – provided by itestra.

Company profile

This project is offered in cooperation with itestra GmbH (www.itestra.de). itestra GmbH is an independent, innovative software solution provider and consultancy. Its services include business process analysis, development of core software systems as well as renovation, optimization and migration and strategic consulting.

Supervisor (itestra GmbH)

Jonathan Streit (streit@itestra.de)

* a similar approach is used for the JavaDoc tool which, however, operates on a detail level that is mostly inadequate for architecture documentation.