



Assignment SAD – 3.0 ECTS

PA1454 – Software Architecture and Quality



Published on August 31st, 2023

Due on September 28th, 2023

General Information

You are supposed to work on the assignment in your groups of three or four students. You will receive feedback on your work after submission, which we expect you to address in case of a re-submission.

The goal of this assignment is to learn to apply a process for **designing** and **documenting** a software architecture. To this end, you will be guided through one iteration of the architecture design process “*Attribute-Driven Design*” (short: ADD; detailed in [T2, Ch. 3]). The first part of the assignment corresponds to step 1 of ADD, where you identify desired quality attributes and other architectural drivers for software system. The second part of the assignment corresponds to steps 2 to 6, during which you will apply architectural *tactics* and *patterns* to satisfy the selected architectural drivers and document your decisions. Both evaluation (step 7) and iterating over the architecture design are crucial parts in an actual ADD process, but they are not part of this assignment.

Note: Many of the topics touched upon in this assignment will also be addressed in the lectures. But don’t worry: Those lectures are *not necessary* for working on and completing the assignment, but instead provide more background about the challenges of large-scale software projects going beyond the scope of the assignment. Everything you need for completing the assignment are Lectures 1 and 2, and the literature references provided throughout the assignment. In fact, you are invited to bring your lessons learned from working on the assignment as *input* for discussion in the lectures.

1 Part 1: Architectural Drivers (50 points in total)

Part 1 of this assignment is about *architectural drivers*, which include a design purpose, quality attributes, primary functionality, architectural concerns, and constraints – all of which are detailed in the following subsections (see also: [T2, Sec. 2.4]). You are expected to extract/elicit, analyze, and prioritize these architectural drivers from the *system description* (linked with the assignment in Canvas) and also define additional requirements. So, read the document “*System Description*” which describes the intended use of a system first.

1.1 Introduction and Design Purpose (5 points)

First, you need to identify and describe the purpose of the design that you want to achieve. It should address the following questions:

- When and why are you doing this architecture design? For example, is the architecture for a new system or for a portion of an existing system that is being refactored? Is it being done as part of a process of creating an evolutionary prototype?
- Which business goals is the organization most concerned about at this time?

Note: You should provide all the assumptions you have for the system in this section.

1.2 Quality Attributes (25 points)

Quality attributes (e.g., performance, modifiability, security etc.) are the architectural drivers that most significantly shape the architecture. One of the most important tasks of a software architect is eliciting, specifying, prioritizing, and validating quality attributes.

For this assignment, you are to elicit, specify, and prioritize quality attributes. All group members should actively participate in this work to brainstorm the quality requirements and to work towards achieving consensus. The expected outputs from this activity are:

- Select a set of **at least 3** quality attributes that your group thinks are the most relevant to the system. Clearly motivate your selection. (Reference: a selection of common attributes is in [T1, Ch. 4–14])
- For the selected quality attributes, develop *quality attribute scenarios* using the six-part scenario template (Reference: [T1, Sec. 3.3]). For each of your selected quality attributes, you must describe a general scenario to explain the attribute and at least one concrete scenario to specify the system.
- Develop a *utility tree* to *relate* the scenarios to business drivers and to *prioritize* the quality attribute requirements according to their business risk and, where possible, technical difficulty. (Reference: [T1, Sec. 19.4])

1.3 Primary Functionality (10 points)

Functionality represents a system's ability to do its intended work. It normally does not have a large influence on the system structure – quality attributes have a more profound and direct influence on the structure of the system. When designing systems, however, it is still important to consider at least the *primary* functionality. Functionality is considered *primary* when it is critical to achieving the business goals that motivate the development of the system. Furthermore, technically complex features requiring interaction amongst many architectural elements can also be classified as part of primary functionality. Consideration of primary functionality helps architects to think about how to allocate it to different architectural elements (e.g., modules) to promote maintainability (a quality attribute), for example.

All group members are expected to brainstorm the primary functionality. The goal of this task is to elicit, prioritize, and document the important use cases of the system. In particular:

- Prepare a use case model (with 4-5 use cases). Besides a use case diagram, you are required to describe each use case in detail. (Examples: [T2, Secs. 4.2.1, 5.2.1, 6.2.1])
- Explain for each use case your reasons for including it as primary functionality, i.e., why is it important to discuss this use case now as part of the architectural process? How is it related to the business goals?

1.4 Architectural Concerns (5 points)

Architectural concerns are other issues that should be considered during the architecting phase, but are not documented as requirements. These include the following:

- To start with, as part of the general concern, you need to attempt to establish an overall system structure. You shall do that by creating a context diagram. (Examples: [T2, Figs. 3.3, 4.2, 6.1])
- For more specific concerns, describe detailed system-internal issues such as authentication, authorization, logging, exception management, etc., that are common across large number of applications.
- Other issues may arise during the design and evaluation of the architecture.

1.5 Constraints (5 points)

As part of the architecting design process, you are also required to document constraints on the development. These constraints may be in the form of mandated technologies (e.g., use of a particular DBMS, use of open source systems), other systems that your system needs to interface with, standards, deadlines, compatibility issues etc.

- You are required to document the constraints. You also need to provide a clear justification for the selected constraints. You may get some hints from the system description. Furthermore, it could also be based on your assumptions about the system requirements. (Examples: [T2, Secs. 4.2.3, 5.2.3, 6.2.3])

2 Part 2: Software Architecture Description (50 points in total)

You will now continue with the *Attribute-Driven Design* process (ADD). For this assignment, you will go through one iteration of that process. In part 1 of the assignment, you elicited and described the architectural drivers.

Your goal now is to design an overall system architecture, i.e. to produce a *module structure* by selecting and applying appropriate architectural tactics and/or patterns to satisfy the identified architectural drivers. For your report, you need to document your decisions and the resulting structures as a *module view* using the view template as described in [T3, Ch. 10]. Please refer to that book for more details on what to write in each of the template's parts. Number the subsections in your report accordingly:

- **Section 2.1:** Primary presentation (10 points)
 - That's a *module view* representing the *module structure* that you decided on, e.g. as an UML package diagram.
- **Section 2.2:** Element catalogue (20 points)
 - **Section 2.2.1:** Elements and their properties
 - **Section 2.2.2:** Relations and their properties
 - **Section 2.2.3:** Element interfaces (if needed)
 - **Section 2.2.4:** Element behavior (if needed)
- **Section 2.3:** Context diagram (5 points)
- **Section 2.4:** Variability guide (5 points)
- **Section 2.5:** Rationale (10 points)

This section must contain:

 - Clear justification for selecting a particular pattern(s) for this view
 - Clear justification for other architectural decisions you made for this view (e.g., use of specific tactics).
 - Architectural drivers (i.e., mainly use cases and quality attribute scenarios) are mapped/linked to the decisions made to address them.

References

- T1** L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 4th ed. Addison-Wesley Professional, 2021. Available: <https://learning.oreilly.com/library/view/software-architecture-in-9780136885979/>
- T2** H. Cervantes and R. Kazman, *Designing Software Architectures: A Practical Approach*. Addison-Wesley Professional, 2016. Available: <https://learning.oreilly.com/library/view/designing-software-architectures/9780134390857/>
- T3** P. Clements *et al.*, *Documenting Software Architectures: Views and Beyond*, 2nd ed. Addison-Wesley Professional, 2010. Available: <https://learning.oreilly.com/library/view/documenting-software-architectures/9780132488617/>

Submission

To complete this assignment, you need to submit a written report.

Report Structure

Use the same section structure as the above assignment description (i.e., 1., 1.1, ..., 2., 2.1, ...).

Formatting

- Formatting specifications: single column, 12-point serif font (e.g. Times), and regular margins.
- Page limit: There is no page limit (but avoid repetition)
- References and citations: Select any style (APA, Harvard etc.) for references and citations. You shall then use the selected style consistently throughout the report.
- Tables and figures: Number and label the tables and figures consistently, and also refer/cite them in the text.
- Sections and sub-sections: Use the section and sub-section numbering style consistently throughout the document.

NOTE: You can find a [LaTeX DIPT Template here](#), and we recommend using it.

Notes on Submission & Grading

The report, as a PDF file, has to be submitted on Canvas before the deadline date. You are advised to **proof-read and critically review the final report before submission** to identify and remove language related issues (e.g. grammatical mistakes) and formatting issues.

The minimum passing marks are 60/100, i.e., 60% for the minimum passing grade.