1 Definition of Architecture

First four definitions of software architecture are stated and their differences and simularities are discussed. Second, our own view on software architecture is given.

1.1 Definitions on architectures

1. Software Architecture is the set of structures needed to reason about the software system, which comprise the software elements, the relations between them, and the properties of both elements and relations.[1]

2. Software Architecture is an abstract system specification consisting primarily of functional components described in terms of their behaviors and interfaces and component-component interconnections[2]

3. Software Architecture is the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.[3]

4. Software architecture is the study of the large-scale structure and performance of software systems. Important aspects of a system's architecture include the division of functions among system modules, the means of communication between modules, and the representation of shared information.[4]

The definitions show a great deal of simularities. In all software architecture is defined as a set of structures/components/modules, their properties and the relationships between them. However each of the definitions highlights certain aspects of architecture more than the others.

- In [1] and [2] the focus has been put on the end result. The architecture is defined as the components themselves (their properties/interfaces) and the relationships between them (interconnections).
- The definition stated in [2] lays a focus on the functional components of a software architecture.
- The definition of [3] is more focussed around the process of designing and maintaining an architecture. It explicitity states that the design process and the evolution are part of the architecture itself.
- The last definition ([4]) focusses more on comparing architectures and how the systems themselves perform. This also becomes clear from the title of the article *Studying Software Architecture Through Design Spaces and Rules*.

References

[1] Bass et al. *Software Architecture in Practice*. Addison Wesley, Boston, 3nd Edition, 2012.

[2] Hayes-Roth, F *Architecture-Based Acquisition and Development of Software: Guidelines and Recommendations from the ARPA Domain-Specic Software Architecture (DSSA) Program*, Teknowledge Federal Systems, 1994

[3] IEEE Std 1471-2000, *Recommended Practice for Architecture Description of Software-Intensive Systems*, United States, 2000

[4] Lane, Thomas, *Studying Software Architecture Through Design Spaces and Rules*. Software Engineering Institute, Carnegie Mellon University, 1990.