

Package structures of software systems

The package structure you choose has a considerable influence on the architecture and maintainability of your software system. It lays the foundation for whether your application stays manageable in the long term or becomes a big ball of mud. In this talk, we will show what matters.

The package structure is the basic structure of object-oriented software systems. It is not only the way of grouping classes but also relevant for every developer in their daily work. Package structures help to grasp and understand structures within the application quickly. Is it possible to derive the functionality based upon the package name and talk about the system on the functional level? A meaningful structuring of the application helps in day-to-day activities, in the implementation of new requirements and maintenance. All thanks to the fact, that a higher implementation speed can be achieved. In many projects, the package structure is based on the class stereotypes such as controllers, services or factories. This technical structuring is an intuitive procedure in smaller software systems, which leads to considerable disadvantages in larger software systems like an increase of technical debts. Reasons for that are the resulting lack of system understanding which leads to unclear responsibilities, undesired dependencies, cycles and high complexity. Finally, this causes applications to erode unnoticed, resulting in a reduction of productivity.

An alternative way of system decomposition helps to avoid the listed negative effects. Focusing on the mental model of the user and the developer leads to a functional system decomposition. This will be discussed looking at use cases illustrating genuine business transactions, which everyone can understand. The use of a functional structuring is motivated based on cognitive psychology as well as further design principles of software architecture. These things considered in the architecture leads to significant improvement in comparison to a technical structuring approach.