The study examines software components and their features. Although components have already gained a significant position in the software industry, they are still much disputed elements. At the moment, there is even no consensus of the definition of components.

Because of the above, the purpose of this research was to clarify the concept of a software component and examine features that software components are purported to have. The study describes the structure, behavior, principle, and infrastructure of a software component. The theory was gathered by combining and evaluating views of several authors.

This study verifies that a software component is an entity that has an internal part and one or more interfaces. Components must protect their internal information by allowing only such interaction that happens through interfaces. A software component can have both horizontal and vertical interfaces. Horizontal interfaces can define required and provided services. However, the same interface should not define both of them.

A component can receive and send messages through interfaces and change the runtime configuration of the system. The principles of components cover modularity and adaptability. Modularity covers modular design and encapsulation; adaptability covers substitutability, late binding, and binary reuse. Finally, the infrastructure was examined as a part of the description of a component, and it was noticed that its powerful nature creates a conceptual difference between objects and components.

The theory presented in this study was compared to a real component model that is Enterprise JavaBeans (EJB). It was found that EJB is an elegant component model that supports well the theory excluding the language dependence and the lack of basic interfaces. In the other hand, it should be noted that the present study does not cover details but only higher-level features.

Finally, different requirements for components and component-based systems were listed in the end of the study. A set of problems that hinder meeting the requirements was presented, and the possible solvability of these problems was considered.

Keywords: Software Component, Component Model, Infrastructure, Framework, Component-Based Architecture, Enterprise JavaBeans, EJB