SOFTWARE COMPONENTS

A **software component** is basically a **software** unit with a well-defined interface and explicitly specified dependencies. A **software component** can be as small as a block of reusable code, or it can be as big as an entire application.

They are parts of a system or applications.Components are means of breaking the complexity of software into manageable parts. Each components hides the complexity of its implementation behind an interface.Components can be swapped in and out like the interchangeable parts of a machine. This reduces the complexity of software development, maintenance , operations and support and allows the same code to be reused in many places.

There are types of components

1. Off the shelf components : Existing software that can be acquired from the third party
2. Full experience Components: Existing past projects that are similar to the software to be built for the current project and team members have full experience.
3. Partial Experience components: Existing past project that are related to the software to be built for current project but needs substantial modifications.
4. New Components: Software components that must be built by the software team specially for the needs of the current project.

   Examples:
   1. Views(interfaces): User interface components for different requests, views and scenarios. For example difficult components can be used to display the same information in a webpage and mobile app.
   2. Models : Components that handle requests or events including business rules and data processing. For example, a model might handle a bill payment request for an internet banking website.
   3. Controllers: A controller is a component that decides what components to call for a particular request or event. For example, a controller might dynamically load different views for a bill payment based on factors such as language, transaction status or channel.