

Component Essentials

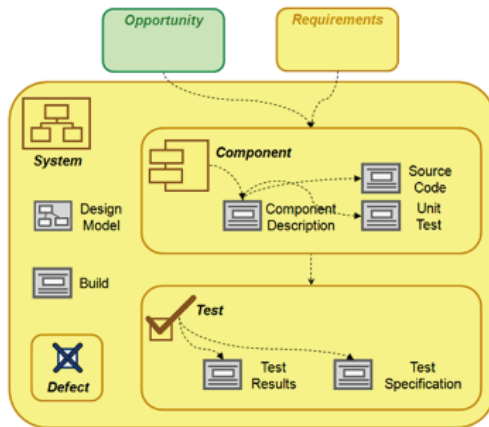
Simple, scalable, component-based development

Use this practice to develop complex systems as assemblies of smaller and simpler components.

This practice allows teams to:

- Manage the complexity associated with the development of software systems.
- Develop complex systems in an extensible and maintainable fashion.
- Develop and verify the separate parts of a system independently and in parallel.
- Identify opportunities for reuse and exploit reusable components.
- Understand the impact of proposed solutions upon the business.
- Utilize third-party frameworks and component libraries.

Things to Produce

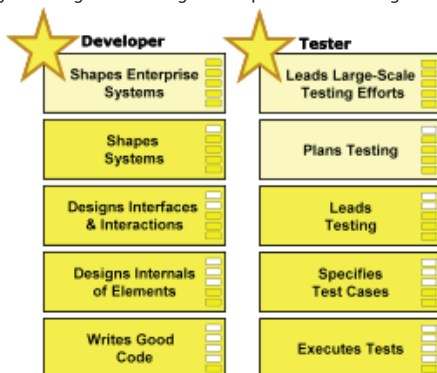


This practice involves the production of a number of implementation and test artifacts:

- A design model of the implemented system, identifying the set of required components.
- A description of each component, including its required behavior and interfaces.
- Source code and unit tests for each component.
- Integrated builds of the component system and the tests and test results used to verify the builds.

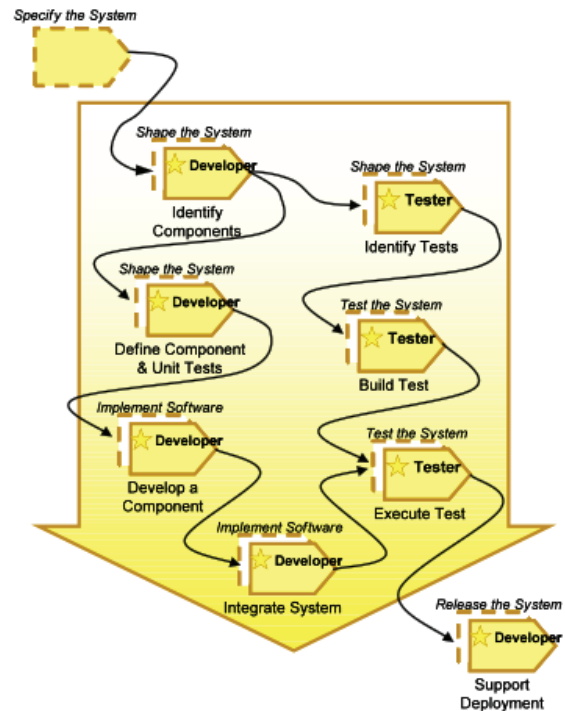
Key Competencies

This practice requires the team to be skilled in software design, implementation, unit testing and integration testing. This requires the following competencies:



Things to Do

The practice starts by identifying the set of components that are needed to meet the system requirements as captured outside this practice in Specify the System. This includes identifying an appropriate set of tests to verify the system.



It continues by defining the components, including their interfaces and unit tests and developing the components to implement the interfaces and pass these tests. It concludes by integrating the system, executing the tests to verify the system produced and then supporting deployment as the components are released.