

## CONSTRUCTION TESTING

Construction testing is a combination of test practices performed during the construction of a component, by the engineer(s) constructing that component, with the intent of finding defects as close to the injection point as possible.

<b>Main Benefits</b>	Construction testing uncovers defects effectively, efficiently, and early.
<b>Keys to Success</b>	Determining what type(s) of testing should be done, using construction testing on both new and modified code, and performing the testing immediately after the code is completed.
<b>When to Use</b>	Construction testing should be used on virtually any type of project. Depending on the nature of the system, it may not be possible for engineers to test components in both the full system context and the component level.
<b>Main Risks</b>	Construction testing has no major inherent risks. Not doing construction testing is a major risk because you lose the opportunity to correct many common types of errors at a point where the marginal effort of doing so is trivial.

### Overview

Construction testing occurs as the component is built in a local environment, and is often used as a gate for releasing new code or configuration data into the project (e.g., before checking code into the source archives).

Construction testing verifies additions or modifications both at the component level and in the context of the system. It encourages small iteration and integration steps during construction by making sure additions or changes to the system are solid before moving on to new ones. It is an efficient mechanism for catching many common construction errors very close to their introduction. Rigor and formality of construction testing will vary based on the needs of the project, and although it can be a very effective form of testing is not a replacement for downstream testing and other quality activities such as reviews.

Depending on the needs of the project and the specific component, construction testing may be formal or informal, manual or automated, structural or functional, or a combination of all of these.

### CxOne Support

CxOne provides direct implementation support for construction testing through the *testing* materials. The test plan checklists and templates provide a means for determining which types of testing are most applicable for any particular project. Test checklists offer guidance to the individuals performing the various types of testing.

### Interactions with other Best Practices

Construction testing works well with Automated Daily Builds by supporting small frequent integration and stabilization cycles during construction.

## Construction Testing Activities

The following test activities make up the core of construction testing. Tests are normally informal with test cases based on undocumented expert judgment. In a more formal or test-first environment, documented test cases may be used. Automated test support, especially for unit testing, allows for easy capture and execution of new and regression tests during construction testing.

### Single Step Testing

Single step testing is performed by stepping through new or modified statements of code with a debugger. Single step testing is normally manual and informal.

### Unit Testing

Unit testing is testing of the smallest thing that can be conveniently tested, often a subroutine. Unit testing is often automated and may require creation of a harness, stubs, or drivers.

### Component Testing

Component testing is testing of an aggregate of units. Component testing generally includes called components and data types. Component testing is often automated and may require creation of harness, stubs, or drivers.

### Bench Testing

Exercising changes to a component while running the entire system on the local engineer's development environment. Similar to developer integration testing, but doesn't necessarily require full integration with latest release of the system.

## Further Reading

*CxStand\_Testing*

McConnell, Steve. *Code Complete*. Redmond, WA: Microsoft Press, 1996

Myers, Glenford J. *The Art of Software Testing*. New York: John Wiley, 1979

Hetzel, Bill. *The Complete Guide to Software Testing*. 2d ed. Wellesley, Mass.: QED Information Systems, 1988