Daily Build and Smoke Test

The daily build and smoke test is a process in which a software product is completely built every day and then put through a series of tests to verify its basic operations.

Main Benefits	Daily builds minimize integration risk, reduce the risk of low quality software, support easier defect diagnosis, and improve progress monitoring capabilities.
Keys to Success	▪ Automating as much of the build and smoke test as is feasible, and having it run every night. ▪ Ensuring the smoke test exercises the entire system from end to end. It does not have to be an exhaustive test, but it should be capable of detecting major problems. ▪ Creating a penalty for breaking the build. ▪ Establishing a dedicated build engineer or group ▪ Adding code into the build when it makes sense. Developers should converge code into the build frequently, but not necessarily daily. ▪ Requiring developers to bench test code before its added into the build
When to Use	Can be used during construction of almost any project.
Main Risks	Daily builds is they can increase the pressure to provide external releases.

Overview

This is a construction-stage process, and it can be initiated even on projects that are already underway. The complete daily build and smoke test produce its savings by reducing the likelihood of several common, time-consuming risks—unsuccessful integration, big-bang integration, low quality, poor progress, and poor visibility.

The process provides critical control for projects in recovery mode, but is just as valuable to new projects starting from scratch. Automating the build process and smoke test should be one of the first construction activities, and should evolve with the system over the life of the project.

The success of this practice depends on developers taking the process seriously and on well-designed smoke tests. The daily build and smoke test can be used effectively on projects of virtually any size and complexity.

Interactions with other Best Practices

Daily builds are especially effective when used in conjunction with miniature milestones by supporting small frequent integration. Daily builds provide support for incremental-development lifecycles such as staged delivery.

If automated unit testing is being used, adding the running of such unit tests to the build can provide a useful addition to the smoke test, or in some cases serve as the smoke test.

Further Reading

McConnell, Steve. *Rapid Development*. Microsoft Press. 1996