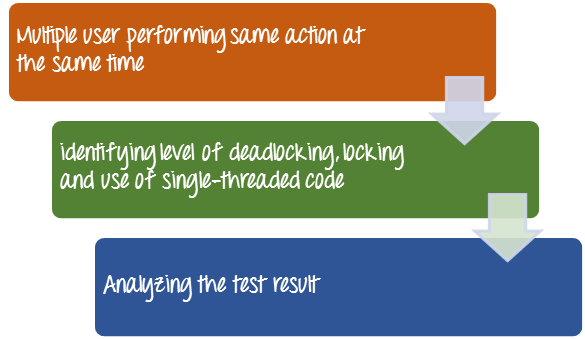
Concurrency Mechanism for Verification & Validation

Verification + Validation = Testing

Concurrency analysis examines the interaction of tasks being executed simultaneously within the product to insure that the overall specifications are being met. Concurrent tasks may be executed in parallel or have their execution in-between. Concurrency analysis is sometimes referred to as “background testing”.

For products with tasks that may execute in parallel, concurrency analysis must be performed during each of the product’s V & V activities.

Concurrency Testing is defined as a testing technique to detect the defects in an application when multiple users are logged in. In fact we can say that monitoring the effect while multiple users perform the same action at the same time. The image below show the concurrent testing:



Concurrent testing is also referred as **multi-user** testing. Testing concurrent program is more challenging then testing sequential program, due to non-determinism and synchronization issues.

Why concurrency testing?

* Identifies the effects of accessing the same database records, modules or application code at the same time.
* Identifies and measures the level of deadlocking, locking and use of single-threaded code and constraining access to shared resources.

Benefits of Concurrent Testing

* It relatively reduces the amount of effort needed to test an application by restricting the scope of concurrent interactions to a few widely used well-tested components.
* Encapsulation value allows to analyze the behavior of a portion of a program without reviewing the entire code or program
* It helps in improving the reliability and robustness of concurrent programs.

Key Challenges in Concurrent Testing

Challenges that might be encounter by test while performing concurrent testing

* Need to test on multiple platforms
* Requires more intensive tests
* Functions do not return their result to the caller immediately, but instead, it can be delivered later via notifications, blocks, callback functions, or similar mechanisms, which makes testing more difficult.
* The information or program flow is not reflected in the call stack.
* The number of execution paths in the system can be extremely large as such the processes in a concurrent system can interact with each other while they are executing
* Concurrent programs have more ratio of failure than sequential ones
* Debugging of concurrent programs

Concurrent or concurrency testing technique is done to detect the defects in an application when multiple users are logged in.