**ITWorks Best Practice for Unit Testing:**

* Test should be simple. Test code is not difference with the production code, make the test code with simple execution path.
* Tests should be fast. When there is a change of the code being tested, the suite of the unit tests can be re-run fast.
* Tests should be readable. Test cases is a form of documentation. Keep the executable to be in sync with what are documenting. The executable specification must be readable.
* Use one logical assertion per method.
* Use BDD-style (behaviour driven development) test cases (i.e. Given/When/Then patten)
* Test shouldn’t duplicate implementation logic. i.e. the unit test should not have any of the test code logic in it.
* The test case shouldn’t depend on:
* Other test cases
* Environmental value such as current time, language setting of the computer running
* External dependencies such as the file system, network or APIs.
* The unit test should be part of the build process. i.e. Continuous testing. If possible, make the build process executes the unit tests and marks the build as broken when the test fail.
* Adopt naming convention for the tests. Your tests should have names that reflect the scenario they’re testing. May adopt the BDD style of naming convention.
* The unit test should not be too coupled to the internals of the code. The unit test should allow the developers to change internal implementation and refactor when needed without the modification of the tests.

**Steps in planning a test:**

* Product analysis
* Who will use the product?
* What is the main purpose of this product?
* How does the product work?
* What are the software and hardware specifications?
* Design a test strategy
* Scope of testing (i.e. what to test and what not to test)
* Type of testing
* Risk and issues (tight deadline, management support, cost estimate, effect of these risk on product)
* Defining objectives
* List of functionality, GUI or Performance must be tested.
* Expected result or benchmark i.e. to which the actual results will be compared.
* Establish test criteria
* Suspension Criteria or Exit Criteria for the product suitable for use or not.
* Plan resource allocation
* Resources include human, equipment, and the infrastructure required
* Planning setup of test environment
* Need real devices with browser and operating systems. i.e. Not only on simulators.
* Determine test schedule and estimate
* Employee availability, duration, deadlines, amount of efforts
* Risk associated with the project.
* Establish test deliverables
* Documentation on Test plan and test design
* Documentation on Test scripts, Test data, Error and Execution logs
* Documentation on Test results, defect Report and Release Notes