

1. **Remember to maximize the test coverage every time you test any application.**
 - Though testing an application, 100 percent test might not be possible, always try to reach near it.
2. **To ensure maximum test coverage, break your application under test into smaller functional modules.**
 - Write test cases for each individual module using the QA template. Also, if possible break these modules into smaller parts.
3. **Make your test cases available to developers prior to coding.**
 - Let developers analyze your test cases thoroughly before testing to help develop quality applications. This will also save the re-work time.
4. **Think positive and go beyond requirement testing.**
 - Start testing the application with the intention of finding bugs/errors. If you test the application with the intention of finding bugs, you will succeed in finding those subtle bugs; test application for what it is not supposed to do.
5. **Think like an end user.**
 - Do not think only like a technical professional, think like customers or end users. Think how an end user will be using your application. Technical combined with end user thinking will assure that your application is user friendly and will pass acceptance tests easily.
6. **Conduct performance testing.**
 - Applications requiring critical response time should be thoroughly tested for performance. If it is not possible to create test data manually then write scripts to create test data for the performance test.
7. **Jot down the new terms and concepts you learn while testing.**
 - Keep a text file open while testing an application. Jot down observations and the testing progress. This is a good habit that will help you to provide test reports and release details.
8. **Increase your conversation with developers to know more about the application.**
 - Whenever possible, make face-to-face communication for resolving questions and also to avoid any misunderstandings. Make sure to communicate any changes to requirements over written communication channels like emails.
9. **Write clear, descriptive, unambiguous issue reports.**
 - Use MS Word, MS Excel or Mantis to log the bug symptoms and the effect of the bug with all possible solutions.
10. **Don't run out of time to do high priority testing tasks.**
 - Prioritize your testing work from high to low priority and plan your work accordingly. Analyze all associated risks to prioritize your work.