

Some thoughts on Software Testing and debugging by Srinivas Pavan Addanki

1. Bugs can easily be created. They cannot however, be easily destroyed. But they can easily get transferred from one platform to another.
2. There is no such thing as a bug free application. But yes, bugs do come free with every application.
3. Bugs do not get reproduced when you demo them to the developer / project manager.
4. But will magically reappear from nowhere when you demo your product to the client / customer.
5. Your most bug-free product version will reside in your developer's local machine / session. Bugs never happen there.
6. The more the number of developers working on the product, the more bugs there will be to find. Too many cooks spoil the broth.
7. No matter how extensive your test coverage is, you will always miss / overlook one aspect of the product.
8. No matter how many critical bugs you find, you will be pilloried for the one bug you failed to detect.
9. The PM and delivery manager will always consider the testing team as dead weight.
10. The bug which causes the maximum amount of damage always requires the smallest of fixes.
11. Customers / End users usually find more bugs than your testers' do.
12. Bugs will never be apparent if you keep looking for them, but when you are not, they will pop up from every nook and corner of the application and will hit you in the face.
13. Automation might get things done quicker, but the amount of time it takes to script the tests and run them will make you think of running the tests manually.
14. The most critical and time consuming bug for the developers to fix will always occur at the far end of the testing cycle or when the product is ready to ship.
15. The Developers will fix the build and get it ready for testing at the far end of the day, just when you are about to leave office. And that build happens to be a critical release which just needs to be tested ASAP.
16. Most critical issues will always occur when most of your team is on vacation.
17. Only thirty percent of your effort is spent in detecting bugs. The remaining seventy goes in convincing developers and project managers that it is actually a bug.
18. The product / feature for which you have detected the maximum number of bugs will most certainly be De-scoped.

Developers' ultimate justification for bugs:

"But it does not happen on my local machine!!" :)

"This issue is not in scope."

The Fundamental Law of Software Engineering

"If it works, then don't touch the code."

Murphy's Ultimate Law for all products:

"It just won't work!"