

How to Launch a Bug Free App?

The year is 2041, and humans are embarking on its first spaceflight to Saturn's largest moon; Titan. The commander orders the ship to land on the moon's surface. The ship's AI computer obliges while performing millions of mathematical calculations to ensure a smooth and easy landing with no casualties.

In this case, the ship's landing software is a bug-free application. It doesn't malfunction, crash or need to be rebooted. As a bug in vital software like this could be catastrophic to the human lives on the spaceship.

How do the ship's apps remain bug-free?

The ship's apps are bug-free as the advanced AI conducts periodic checks on its code for errors. Even if errors do pop up, these would be single line errors in data which would be immediately corrected by the ship's large team of engineers.

This app is almost perfect. It doesn't need app testers or updates to keep it functioning without a hitch. Yet, this is a vital part of a spaceship's architecture. If this software has bugs, a lot of people die. So it's maintained by hundreds of men and women to ensure that the software delivers as it's supposed to with no danger to human lives.

Launching bug-free applications

Technically humans can write and launch bug-free applications like the one on the spaceship. In a practical sense, however, it's not as straightforward as it sounds. Today, releasing a bug-free app is not economically viable. Unless the app has a mission or a life-critical application.

Causes of bugs

Evolution of Technology

Technology advances with time, and with time even technology starts to age. Time doesn't spare software, and so bugs begin to arise. For example; a version of Android starts slowing down after a few months due to errors. Google releases bug patches and updates to make sure that it runs without a hitch.

Updated APIs

The software contains APIs for sharing on social media or payment integrations. To ensure they run smoothly, these APIs are regularly updated. These updates may cause bugs to appear, and so these bugs need to be removed to keep users satisfied with the app.

Complex software

Over time, software becomes more complex due to advancements in technology. Technologies like, AI, VR and AR are possible due to ultra-complex programs with hundreds

of lines of code. These programs are rarely error-free due to its high level of sophistication. They are even more prone to errors when new features are added.

Execution environment

The execution environment is the place where users interact with the app. End users will always find bugs and problems. Therefore, offering customer support and bug reporting options is time-saving and cost-effective.

How to maintain bug-free apps?

Code Reviews

Periodic review of the app's code prevents common bugs.

Beta Tests

Releasing the beta version is the best way to find out how the users receive it. Letting users find bugs in reduce the time and costs required for debugging.

Handling Error Reports

Using automated error reporting procedures makes preparing error reports easier and faster.

Using Virtual Machines

Testing an app in different virtual machines helps investigate its compatibility with different operating systems.

App testing

Subjecting the app to regular tests throughout its development guarantees fewer bugs:

Unit testing:

This test involves examining Individual modules and functions of the project. It prioritises bug reports and patches.

Integration testing:

The objective of this test is to examine Module synergies and dependencies.

System testing:

System testing involves testing the app's single and group module functions. It further ensures that the app performs as its requirements.

Acceptance testing:

In this test, the app's core functions and features are thoroughly checked before it is prepared for an official release.

Conclusion

Having a 100% bug-free app is an insurmountable task as the apps are as imperfect as the humans who coded them. Yet, taking the necessary action to prevent bugs is essential.