What is a framework in programming?

The system is a house, a car, a theory, or a mobile app, the concept of the framework is the same: it provides the support and basic 'guide' of the structure being built.

A framework in programming is a tool that provides ready-made components or solutions that are customized in order to speed up development. A framework may include a library, but is defined by the principle of Inversion of Control (**IoC**). With traditional programming, the custom code calls into the library to access reusable code. With **IoC**, the framework calls on custom pieces of code when necessary.

A framework can include support programs, compilers, code libraries, toolsets, and APIs to develop software and create systems. Open-source frameworks are always been updated and improved.

Why Frameworks been used in Software Development?

The purpose of a framework is to assist in development, providing standard, low-level functionality so that developers can focus efforts on the elements that make the project unique.

The use of high quality, pre-vetted functionality increases software reliability, speeds up programming time, and simplifies testing. With an active base of users and ongoing code improvements, frameworks help improve security and offer a base of support.

Ultimately, frameworks are been used to save time and money.

What are the Features of a Good Framework?

There are many kinds of frameworks, with some more popular than others. Developers often choose frameworks they are most familiar with, but that framework may not be the right for the job. Instead, consider the following features of good frameworks when deciding on the right framework for the project at hand:

- **Functionality** – choose a framework that provides the functionality needed for the project at hand, respecting that each framework has its limits and not investing in a framework that does far more than your project will ever need.
- **Consistency** – a framework can assist in consistency for large or distributed teams
- **Documentation** – choose a framework that has well-documented code and that provides implementation training
- **Active Community** – frameworks are only as strong as the user base of support. Choose a framework that is been well established with an active user base.

Challenges of Using a Software Framework:

Software frameworks can become a costly crutch if developers are not strong in the language the framework is based upon or if the developer over-relies on the framework instead of custom code, a problem which can lead to software bloat and performance issues.

Mobile App Development Frameworks:

A mobile development framework (mobile app development framework) supports the development of mobile applications for a specific environment, classified as native (platform specific), hybrid (native back end shell with web app front-end), or cross-platform (shared codebase for all). There are still lot of mobile app frameworks available which is not been covered here including, Swift, Native Scripts, and Ionic, React Native.

There are risks in choosing a framework that is too new or not well supported, which could require costly re-tooling if the framework becomes obsolete. Similarly, if the framework has limitations that are not well understood up front, this could affect the project.

Types of Programming Frameworks:

There are a variety of different programming frameworks, each built upon a programming language and specializing in its function, whether it is working on a web app, database, or mobile app. In this section, we are discuss about mobile application in flutter.

What is Flutter?

Flutter is Google’s open-source mobile UI framework for creating cross-platform applications and, most recently with Flutter 2, web and desktop (native) apps. Flutter based upon Dart and considered a modern framework, being simple to use and leveraging hot reload to see results in real-time.

Flutter Framework Examples: Amazon, Microsoft, Adobe, eBay, Google ads

****