



UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

TOPIC:

Framework Selection

STUDENT:

Maciel Leyva Ariana Lizeth

GROUP:

10A

SUBJECT:

Comprehensive Mobile Development

TEACHER:

Ray Brunett Parra Galaviz

Tijuana, Baja California, January 8th of 2025

“Framework Selection”

What is a framework?

A framework, or software framework, is a platform that provides a foundation for developing software applications. Think of it as a template of a working program that can be selectively modified by adding code. It uses shared resources such as libraries, image files, and reference documents and puts them together in one package. That package can be modified to suit the specific needs of the project. With a framework, the developer can add or replace features to give new functionality to the application. A framework can include support programs, compilers, code libraries, toolsets, and APIs to develop software and create systems. Open-source frameworks are constantly being updated and improved.

Why are Frameworks Used in Software Development?

The purpose of a framework is to assist in the development, providing standard, low-level functionality so that developers can focus efforts on the elements that make the project unique. 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 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 proper 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 that can assist in consistency for large or distributed teams.
- **Documentation:** Choose a framework that has well-documented code and provides implementation training.
- **Active Community:** Frameworks are only as robust as the user base of support. Choose a framework that is well-established with an active user base.

What are the different types of frameworks?

With the huge variety of programming languages out there, naturally, you'll find that there are a large number of frameworks available. Each of these frameworks is used to solve different problems that come up in the process of software development. They're used in many industries, including web development, data science, and more. Here, we'll go into some of the broader categories of frameworks with examples of specific frameworks within them.

In practice, frameworks can be used to develop a broad range of digital products, namely desktop applications, mobile apps, web-based software applications, etc. Therefore, they are often categorized by the type of development environment they are equipped for:

- Mobile development frameworks
- Web development frameworks
- Desktop application frameworks
- Cross-platform development frameworks
- Data science frameworks
- Front-end frameworks
- Backend frameworks

Mobile Application Frameworks

Mobile application frameworks are specifically designed for the development of feature-rich mobile apps. The reusable code structures, UI components and libraries in mobile application frameworks support faster and more efficient mobile app development.

Mobile application frameworks are typically categorized as:

Native Frameworks: Native mobile application frameworks empower developers to write applications for specific mobile operating systems, like iOS, Android, or Windows. When you build an app using native frameworks, it can access all the features of the specific operating system directly. Kotlin, Java, Objective-C, and Swift are some of the most well-known native mobile application frameworks.

Cross-platform Frameworks: When you write an app while using a cross-platform mobile application framework, code written once can be run on multiple operating systems. Some commonly used cross-platform mobile application frameworks include React Native, Flutter, Xamarin, and more.

Web application frameworks

These frameworks are designed to simplify the development of web applications by providing a structure and set of tools. They often include features like routing, database integration, session management, and user authentication. For example, Ruby on Rails follows the MVC architectural pattern and includes a wide range of built-in functionalities, making it efficient for building web applications.

Data Science Frameworks

Data science frameworks are abstractions that can be altered selectively to get application-specific capabilities. They are like toolkits that give data scientists certain pre-built functions, classes, and modules that can be used as and when required.

Popular examples of data science frameworks include Pandas, NumPy, Scikit-learn, TensorFlow, and more.

Frontend frameworks

These open-source JavaScript frameworks are used for the development of the front end (user facing) of a web application, including the design and functionality of the user interface:

- **Angular:** A framework for building web applications with an emphasis on TypeScript.
- **Vue:** A lightweight framework with a component-based architecture.
- **Next.js:** A React-based framework for building web applications.
- **Svelte/SvelteKit:** A newer, fast framework that offers increased performance by shifting work from runtime to the build process.

Backend frameworks

These frameworks are used for the development of the back end of web apps, or the components that handle server-side functionality.

- **Ruby on Rails:** A Ruby-based, server-side framework for developing web applications.
- **Django:** A high-level Python framework for web development.
- **Express:** A Node.js framework for developing web and mobile applications.
- **Spring:** A Java framework for microservices, serverless, and the cloud.

Framework Selection

Choosing the right framework is a critical decision in software development, as it can significantly impact the project's success, scalability, and maintainability. Here are the key considerations and steps involved in selecting a framework:

1. Understand the Project Requirements

- **Type of Application:** Define whether the project is a web app, mobile app, desktop app, or another type of software.
- **Features and Complexity:** Identify the specific features needed, such as real-time updates, integrations, or high scalability.
- **Timeline and Budget:** Assess the project's time constraints and financial resources.

2. Evaluate Framework Features

- **Core Functionality:** Ensure the framework supports the primary features required for the project.
- **Extensibility:** Check if the framework allows easy addition of plugins or extensions.
- **Cross-Platform Support:** If needed, ensure compatibility with multiple platforms (e.g., iOS and Android for mobile apps).

3. Consider Community and Ecosystem

- **Community Support:** A larger community means more tutorials, libraries, and quicker resolutions for issues.
- **Documentation:** Well-documented frameworks are easier to adopt and use.
- **Third-Party Tools:** Look for compatibility with popular libraries or tools.

4. Assess Performance and Scalability

- **Performance:** Review benchmarks and ensure the framework can handle the app's expected load.
- **Scalability:** Choose a framework that supports horizontal and vertical scaling.

5. Evaluate Learning Curve

- **Team Expertise:** Consider the team's familiarity with the framework.
- **Ease of Learning:** Some frameworks are beginner-friendly, while others require advanced knowledge.

6. Long-Term Viability

- **Active Development:** Check if the framework is actively maintained and updated.
- **Future Trends:** Ensure the framework aligns with industry trends and won't become obsolete soon.

7. Compare Popular Frameworks

- **Web Development:** React, Angular, Vue.js, Django, Flask
- **Mobile Development:** Flutter, React Native, Swift, Kotlin Multiplatform
- **Desktop Development:** Electron, WPF, PyQt
- **Game Development:** Unity, Unreal Engine, Godot

8. Run a Proof of Concept

Before fully committing, implement a small part of the application to test the framework's capabilities in real-world scenarios.

By carefully considering these factors, you can select a framework that meets your project's needs, aligns with your team's skills, and supports long-term goals.

Sources

- Bartsch, K. (2023, April 24th). Frontend framework selection considerations for existing teams. *Medium*. Retrieved January 8th, 2025, of <https://medium.com/@kbartsch/frontend-framework-selection-considerations-for-existing-teams-32d3e17c1706>
- Net Solutions. (2023, June 30th). *What is a Framework in Programming? And Why You Should Use One - Net Solutions*. Retrieved January 8, 2025, of <https://www.netsolutions.com/insights/what-is-a-framework-in-programming>
- O'Grady, B. (2024, April 4th). *What is a Framework? Why We Use Software Frameworks - Code Institute IE*. Code Institute IE. Retrieved January 8th, 2025, of <https://codeinstitute.net/global/blog/what-is-a-framework/>
- Sood, S. (2024, January 12th). Programming Frameworks - A complete development frameworks guide. *Latest News, Updates and Events*. Retrieved January 8th, 2025, of <https://www.seasiainfotech.com/blog/your-complete-guide-to-programming-frameworks/>
- Stahl, L. (2024, August 29th). What is a framework in programming? All about software frameworks. *Contentful*. Retrieved January 8th, 2025, of <https://www.contentful.com/blog/what-is-a-framework/>