UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

TEMA:

Framework Selection

.

BY:

Derian Omar Tarango Mendez

GROUP:

10 B

COURSE:

Mobile software development

PROFESSOR:

Ray Brunett Parra Galaviz

Tijuana, Baja California, 10 of january 2025

Framework Selection

1. Introduction

Selecting the right framework is a crucial step in software development that directly impacts the efficiency, scalability, and maintainability of a project. This document outlines the factors to consider when choosing a framework and provides examples to guide the selection process.

2. Importance of Framework Selection

1. **Efficiency: Frameworks provide pre-built tools and components, reducing development time.**
2. **Scalability: The right framework ensures the application can grow with user demands.**
3. **Maintainability: A well-supported framework simplifies future updates and feature additions.**
4. **Team Productivity: Familiar frameworks enable teams to work faster and more effectively.**

3. Factors to Consider in Framework Selection

3.1 Project Requirements

- **Type of Application: Web, mobile, desktop, or hybrid.**
- **Functional Needs: Features required, such as authentication, data processing, or API support.**
- **Non-Functional Needs: Performance, security, and scalability.**

3.2 Language and Ecosystem

- **Choose a framework compatible with the programming language your team is skilled in.**
- **Example:**
  - **For JavaScript: React, Angular, Vue.js.**
  - **For Python: Django, Flask.**
  - **For Java: Spring, Hibernate.**

3.3 Community and Support

- **Evaluate the size and activity of the framework's community.**
- **Look for documentation, tutorials, and third-party integrations.**

3.4 Scalability and Performance

- **Ensure the framework can handle high traffic or large datasets if needed.**
- **Example:**
  - **For high-performance needs: Node.js (event-driven) or Spring (multi-threaded).**

3.5 Development Speed

- **Assess how quickly the framework allows developers to prototype and build features.**
- **Example:**
  - **Rapid development: Ruby on Rails, Laravel.**
  - **Custom flexibility: Flask, Express.js.**

3.6 Cost and Licensing

- **Check for open-source or proprietary licensing fees.**
- **Ensure compatibility with your project's budget.**

4. Examples of Framework Selection

4.1 Scenario: Building an E-Commerce Web Application

1. **Requirements: Scalability, secure user authentication, and payment gateway integration.**
2. **Recommended Framework:**
   - **Frontend: React for dynamic UI.**
   - **Backend: Django (Python) or Laravel (PHP) for robust server-side logic.**

4.2 Scenario: Developing a Mobile Application

1. **Requirements: Cross-platform compatibility and fast deployment.**
2. **Recommended Framework: Flutter (Dart) or React Native (JavaScript).**

4.3 Scenario: Data-Driven Applications

1. **Requirements: Real-time data processing and visualization.**
2. **Recommended Framework: Node.js with Express for backend, and Angular or Vue.js for frontend.**

5. Best Practices

1. **Involve the Team: Ensure the development team agrees on the framework choice.**
2. **Prototype First: Build a small proof of concept to evaluate framework suitability.**

3. **Consider Longevity: Choose frameworks with active support and regular updates.**
4. **Optimize for Scalability: Plan for future growth when evaluating frameworks.**