

# FreeRTOS Architecture Part 1

Name

Universidad Panamericana

Presentation August 5, 2024



# Contents

- 1 Why good practices?
- 2 Use other programming languages
- 3 Requirements
- 4 SDD
- 5 SDG
- 6 UML
- 7 Testing
- 8 TPD

## Why good practices?

# Why good practices?

- **1. Maintainability**

- This is crucial for long-term projects where multiple developers might be working on the same codebase over time.

- **2. Readability**

- Clear and consistent coding standards make it easier for developers to read and understand each other's code.

- **3. Reusability**

- This means that code can be reused in different parts of a project or even in different projects, saving time and effort.

- **4. Bug Reduction**

- Identifying and fixing bugs early in the development process.

- **5. Performance**

- This is particularly important in applications where performance is critical, such as real-time systems or high-traffic web services.

# Why Good Practices?

- **6. Scalability**

- Be easily extended with new features wiYout significant rework.

- **7. Security**

- Secure data handling are crucial in preventing security breaches.

- **8. Documentation**

- For future maintenance, debugging, and onboarding new developers.

- **9. Consistency**

- It allows developers to switch between different parts of the codebase wiYout needing to adjust to different coding styles.

- **10. Professionalism**

- It can enhance the reputation of a development team or company and build trust with clients and stakeholders.

# 10 commandments

- ① **You shall prioritize Maintainability**
- ② **You shall value Readability**
- ③ **You shall strive for Reusability**
- ④ **You shall reduce Bugs early**
- ⑤ **You shall optimize Performance**
- ⑥ **You shall ensure Scalability**
- ⑦ **You shall secure thy code**
- ⑧ **You shall document thoroughly**
- ⑨ **You shall maintain Consistency**
- ⑩ **You shall uphold Professionalism**

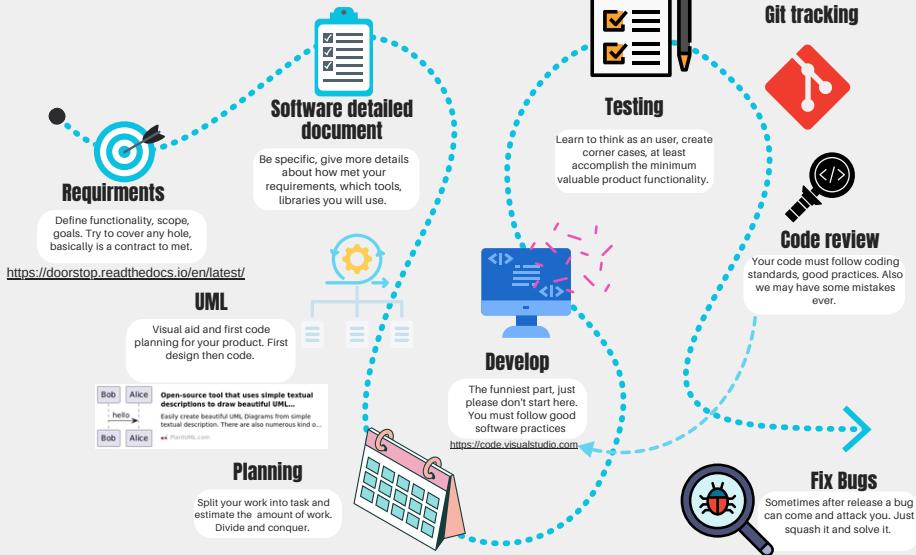
Use other programming languages

# You shouldn't be opposed to learning programming languages.

- Latex : Documentation
- Java: PlantUML
- JSON: For automatization stuff
- Python: Unit Testing
- C: Pretty basic programming
- Cpp: Robust programming
- C#: Windows App



# GOOD SOFTWARE PRODUCT CYCLE



# Requirements

# Requirements

SDD

# Software Detailed Document (SDD)

# SDG

# Software Development Guide (SDG)

# UML



# Unified Modeling Language with PlantUML (UML)

UML diagrams serve as excellent documentation tools that are useful throughout the system's lifecycle. They help new team members understand the system quickly and can also be valuable for maintenance and future upgrades.

# Testing

# Black box vs Whitebox testing

TPD

# Test plan document (TPD)