

## Capstone Project Fundamentals

- **Background Section:** Its primary purpose is to explain the project's origin and context.
- **Scope:** This part of Chapter I defines the project's boundaries and limitations.
- **Problem Statement:** It is crucial because it clarifies the main issue the project will solve.
- **Objectives vs. Goals:**
  - An example of an objective is "Improve user experience within 3 months".
  - An example of a goal is to "Improve organizational decision-making".
- **Stakeholders:** These are typically the beneficiaries and evaluators of the project.
- **Significance and Relevance:** This section focuses on who will benefit from the project and why it's important.
- **Structure of the Document:** This provides chapter summaries and the flow of the document.
- **Context and Scope:** This section includes project limitations, the project environment, and the problem background, but it excludes covered modules.
- **Related Literature:** This refers to past studies and frameworks that support your concept.
- **Methodology Chapter:** This chapter primarily explains how the project was executed.
- **Capstone Methodology:** A capstone methodology should be clearly defined and replicable.
- **Past Research:** Reviewing past research is important to build on existing knowledge and justify the solution.

## Agile Scrum

- **Agile Scrum:** It is primarily used for managing projects iteratively.
- **Sprint:** A sprint is a short, time-boxed development cycle.
- **Product Owner:** This role is responsible for prioritizing the product backlog.
- **Scrum Ceremonies:** The ceremonies include Sprint Review, Daily Standup, and Sprint Planning, but "Code Compilation" is not part of them.
- **Scrum Artifacts:** These are key outputs like the Product and Sprint Backlog.
- **Scrum Master:** This role acts as a facilitator and removes obstacles.
- **Development Team:** The team is responsible for implementing product backlog items.
- **Sprint Cycle:** A sprint cycle includes planning, standups, and review, but not "Post-mortem" as a standard part of the cycle.
- **First step in Scrum:** The first step in applying Scrum is Sprint planning.
- **Benefits of Agile Scrum:** A key benefit is frequent delivery and flexibility.

## Enterprise Architecture & TOGAF

- **Enterprise Architecture:** This is a structured framework for IT design and integration.
- **TOGAF:** It stands for The Open Group Architecture Framework.
- **TOGAF Domains:**
  - **Business Architecture:** Addresses business processes and governance.
  - **Application Architecture:** Defines software systems and their interactions. This domain also concerns app designs and interactions.
  - **Data Architecture:** Concerns data storage and management.
  - **Technology Architecture:** Focuses on platforms and infrastructure.
- **Usefulness of TOGAF:** TOGAF is most useful for strategic enterprise system planning.
- **Key Reason for using TOGAF:** It ensures architectural consistency across the enterprise.
- **Documentation:** TOGAF principles help ensure proper documentation of integrated systems.

## Microservices & System Integration

- **Microservices Architecture:** A key benefit is having modular and scalable services.
- **Inter-service Communication:** API (Application Programming Interface) is a technology that supports inter-service communication in microservices.
- **Deployment of Microservices:** Services are usually deployed as independently deployable components.
- **Large-scale Applications:** Microservices are preferred in these applications because of independent scaling and deployment.
- **Monolithic vs. Microservices:** A limitation of monolithic systems is centralized failure compared to microservices.
- **System Integration:** An example is linking HR and payroll systems. A successful integration approach must focus on compatibility and communication.

## DevOps

- **DevOps:** Its main focus is the integration between development and operations.
- **CI/CD Pipeline:** It is used for automating code build, test, and deployment.
- **Tools:** Jenkins is a tool typically used in a CI/CD pipeline. Docker is also a tool for DevOps automation.
- **DevOps Culture:** It is described by collaboration, automation, and continuous improvement.
- **Deployment Pipeline:** It ensures automation and consistency.
- **Continuous Integration (CI):** This is a DevOps practice that runs tests on every commit.

## Additional Concepts

- **Product Backlog vs. Sprint Backlog:** The Product Backlog is a list of all features and bugs to be implemented, while a Sprint Backlog includes the items selected for the current sprint.