
# Addis Ababa Science and Technology University

## College of Engineering

## Department of Software Engineering

## Software Component Design

# Title:Prototyping Model

**Group Members:**

1. Amanuel Ayana ETS 0110/13
2. Amanuel Dagnachew ETS 0112/13
3. Amanuel Mandefro ETS 0121/13
4. Amanuel Solomon ETS 0126/13
5. Beamanuel Tesfaye ETS 0180/13

**Dec, 2024**

**Submitted to: Mr.**

# Table of Content

# Introduction

The **Prototyping Model** is a widely used software development methodology that focuses on building a working model of the system early in the development process. This model allows stakeholders to visualize the system, provide feedback, and refine requirements before full-scale development begins. By creating a prototype a preliminary version of the software developers and clients can identify misunderstandings, explore alternative solutions, and ensure that the final product meets user expectations.

The Prototyping Model is particularly beneficial for projects where requirements are not well-defined or are expected to evolve during development. It emphasizes **collaboration, flexibility, and user involvement** to achieve a more accurate and functional software solution.

# Phases of the Prototyping Model

The **Prototyping Model** follows an iterative and incremental approach to software development, where the system is built in phases to refine requirements and ensure stakeholder satisfaction. Below are the key phases involved in the Prototyping Model:

## 1. Requirement Gathering

- In this initial phase, basic requirements for the system are collected from stakeholders (e.g., clients, end-users).
- Unlike traditional methods, the requirements are not expected to be fully detailed at this stage. The focus is on understanding the **core objectives** and **functionality** of the system.
- This phase involves discussions, brainstorming sessions, and meetings to identify the initial scope.

**Example**: A client may request an e-commerce app to display products, allow purchases, and track orders. The exact features are left flexible for refinement during prototyping.

## 2. Quick Design

- A simple and **preliminary design** of the system is created based on the initial requirements gathered.
- This design focuses on the **user interface (UI)** and key functionalities rather than technical or backend details.
- The purpose of this phase is to **visualize the system** so stakeholders can understand and interact with it early on.

**Key Output**: Wireframes, mockups, or sketches of the application interface.

**Example**: For an e-commerce app, designers may create a quick layout showing a homepage with product categories, a cart button, and a profile section.

## 3. Prototype Development

- In this phase, the actual **working prototype** of the system is developed based on the quick design.
- The prototype is a simplified version of the software with limited functionality but enough to demonstrate key features.
- Developers use rapid tools or frameworks to quickly build the prototype, ensuring faster turnaround.

**Purpose**:

- To give users a tangible idea of the system.
- To allow stakeholders to interact with the prototype and evaluate its features.

**Example**: A basic clickable version of the e-commerce app with dummy data to simulate product browsing and adding items to the cart.

## 4. User Evaluation

- The prototype is presented to stakeholders (users or clients) for evaluation.
- Stakeholders interact with the prototype and provide feedback on:
    - The user interface design.
    - Functionality and usability.
    - Missing features or changes needed.
- This phase helps identify misunderstandings, gaps, or evolving requirements.

**Key Activity**: Collecting **feedback** through user testing, interviews, or surveys.

**Example**: Users may request features like "search products" or "filter by price" after testing the e-commerce prototype.

## 5. Refinement

- Based on user feedback, the prototype is refined, updated, or rebuilt to include changes and enhancements.
- This process can repeat multiple times (iterative development) until the prototype meets user expectations and requirements.
- Refinements ensure that the final product aligns closely with what stakeholders need.

**Key Outcome**: A more polished and accurate prototype that serves as a blueprint for the final product.

**Example**: After adding requested features like product search and filtering, the updated prototype is presented for re-evaluation.

### 6. Final Product Development

- Once the prototype is accepted and refined, the final system development begins.
- The approved prototype acts as a **reference** for developers to implement the complete product, including:
    - Technical architecture.
    - Backend functionalities.
    - Integration of all features.
- Rigorous testing is performed to ensure the system works as intended.

**Key Focus**: Delivering a **full-fledged, production-ready** version of the software.

**Example**: Developing the e-commerce app with all finalized features, live product data, payment gateway, and user authentication.

| Phase | Purpose | Outcome |
|---|---|---|
| Requirement Gathering | Understand core requirements. | Initial scope of the project. |
| Quick Design | Visualize features and interface. | Simple wireframes/mockups. |
| Prototype Development | Create a working version of the system. | Functional but limited prototype. |
| User Evaluation | Collect user feedback. | Insights into changes needed. |
| Refinement | Update and enhance the prototype. | Improved and refined prototype. |
| Final Product Development | Develop the final, complete system. | Full production-ready software. |

## Advantages of Prototyping Model

1. Improved Requirement Clarity
2. Early Feedback
3. Better Communication
4. Reduced Risks
5. Flexibility in Development

6. Improved User Satisfaction
7. Facilitates Complex Systems

## Disadvantages of Prototyping Model

1. Time-Consuming
2. Increased Costs
3. Misunderstanding of the Prototype
4. Scope Creep
5. Dependency on User Feedback
6. Poor Design Choices
7. Not Suitable for Large Systems

## Application Areas

The Prototyping Model is well-suited for specific types of projects and environments where requirements are dynamic or unclear. Below are the key application areas:

1. **Systems with Unclear or Evolving Requirements**
    - When clients are unsure about their exact needs, the Prototyping Model helps clarify requirements through iterative feedback.
    - Example: Developing a custom CRM system where features are refined based on user input.
2. **User-Interface Intensive Systems**
    - Applications where the user interface (UI) and user experience (UX) play a critical role benefit greatly from prototyping.
    - Example: Web applications, mobile apps, or graphical design tools.
3. **Complex Systems with High User Interaction**
    - For systems requiring significant user interaction, prototypes help ensure usability and functionality before full development.
    - Example: Online booking systems, e-commerce platforms, and gaming applications.
4. **Innovative or New Technology Projects**
    - Prototyping is useful for projects involving emerging technologies where stakeholders may need to validate concepts.
    - Example: AI-based tools, augmented reality (AR) applications, and IoT systems.
5. **Web and Mobile Application Development**
    - Rapid development tools make it easy to build prototypes for web and mobile apps, gather user feedback, and refine designs iteratively.
    - Example: Social media apps, fintech applications, and food delivery platforms.