

## Table of Contents



Introduction to Prototype model

Illustration of Prototype model

Phases of Prototype model

1. Requirements Gathering
2. Prototype Design
3. Prototype Development
4. Prototype Evaluation
5. Feedback Analysis
6. Prototype Refinement
7. Final Development
8. Testing and Deployment
9. Maintenance and Iteration

Advantages of Prototype Model

Disadvantages of Prototype Model

# Introduction To Prototype Model

- It requires that before carrying out actual software its prototype (model) must be created.
- Prototype model is a toy implementation of system.
- A prototype usually a demo version of actual system, possibly with limited functionality, low reliability, and inefficient performance compare to actual system.
- Detailed information is not available in it.
- Idea behind it is to create a prototype to gather the basic requirements.
- Prototype is built on the basis of current available requirements.
- It gives the “actual feel” of the system.
- Prototype is not complete system many of the details are not built into the prototype.
- The goal is to provide system with overall functionality.

# Illustration Of Prototype Model



## Phases Of Prototype Model

### 1. Requirements Gathering

In this phase, the project team works closely with stakeholders to gather initial requirements

and understand the objectives of the software. The focus is on capturing the essential features and functionalities that need to be included in the prototype.

## 2. Prototype Design

Based on the gathered requirements, a preliminary design of the prototype is created. This design may include user interfaces, navigation flows, and key functionalities that will be implemented in the prototype.

## 3. Prototype Development

The development phase involves building the initial version of the prototype based on the design specifications. This phase focuses on creating a functional representation of the software that demonstrates key features or functionalities.

## 4. Prototype Evaluation

The created prototype is shared with the stakeholders and users for evaluation and feedback. This phase involves conducting usability testing and gathering feedback on the prototype's functionality, design, and user experience.

## 5. Feedback Analysis

The feedback received from users and stakeholders is analyzed to identify necessary modifications and enhancements to the prototype. This analysis helps refine the requirements and improve the understanding of user needs.

## 6. Prototype Refinement

Based on the feedback and analysis, the prototype is refined and iteratively improved. This may involve making changes to the design, adding or modifying features, or addressing usability issues identified during the evaluation phase.

## 7. Final Development

Once the prototype is approved, the final development of the software begins. This phase involves building the complete and fully-functional software product based on the refined prototype and requirements.

## 8. Testing and Deployment

The developed software undergoes thorough testing to ensure that it meets the requirements and functions as expected. Once the software is deemed ready, it is deployed for use by the end-users.

## 9. Maintenance and Iteration

After deployment, the software may require maintenance and further iterations based on user feedback and evolving requirements. This phase involves addressing any issues, making updates, and providing ongoing support to users.

# Advantages Of Prototype Model

1. **Enhanced User Involvement:** The Prototype Model allows for early and continuous user involvement and feedback, ensuring that the final software product meets user expectations and requirements.
2. **Improved Requirement Understanding:** Prototypes help in better understanding and refining the requirements by providing a tangible representation of the software early in the development process.
3. **Early Detection of Issues:** The iterative nature of prototyping allows for early detection and resolution of design flaws, usability issues, and other potential problems, reducing the likelihood of costly changes later in the development cycle.
4. **Reduced Development Time:** Prototypes enable faster development cycles as they focus on specific features or functionalities, allowing for rapid iterations and incremental improvements.
5. **Increased Stakeholder Collaboration:** Prototypes facilitate effective collaboration and communication among stakeholders, including developers, users, and other project participants, leading to a shared understanding and alignment of project goals.
6. **Mitigated Risk:** By allowing stakeholders to visualize and interact with a working model of the software early on, the Prototype Model helps mitigate the risk of developing a final product that does not meet user needs or expectations.
7. **Opportunity for Innovation and Creativity:** The prototyping approach encourages creativity and exploration, providing an opportunity to experiment with different design ideas and innovative solutions.

8. Better User Experience: Through user feedback and iterative refinement, the Prototype Model helps in delivering a software product that provides a better user experience and usability.

## Disadvantages Of Prototype Model

1. Insufficient Documentation: The focus on prototyping and iterative development may result in inadequate documentation, making it challenging to maintain and understand the software in the long run.
2. Time and Cost Overruns: The iterative nature of prototyping can lead to project delays and increased costs due to the need for multiple iterations and incorporating changes.
3. Misinterpretation of Requirements: Without a thorough understanding of requirements, there is a risk of misinterpretation, resulting in prototypes that may not fully meet user needs.
4. Scope Creep: User feedback and continuous iterations may lead to scope creep, with requirements expanding beyond the initially defined scope, causing challenges in managing project scope, time, and budget.
5. Limited Scalability: Prototypes often focus on specific features, neglecting scalability and integration aspects, which can pose challenges when scaling the software or integrating it with other systems.
6. Reduced User Involvement: While user involvement is emphasized, there may be

limitations in involving all user groups or capturing the perspectives of all stakeholders, potentially leading to an incomplete understanding of user needs.