Addis Ababa Science and Technology University

Software Component Design

Title: Prototyping Model

Section D

Group Members **ID**

1. **Yordanos Negusu----------ETS1370/13**
2. **Yoseph Zewdu----------ETS1395/13**
3. **Rediet Teklay----------ETS1093/13**
4. **Yoseph Shemeles----------ETS1392/13**

Submission Date: 12/18/24

Submitted To: Mr. Gizatie

Table Of Contents

Prototyping Model in Software Development

The Prototyping Model is a user-centric software development methodology that emphasizes the creation of a working prototype during the early stages of a project. This approach is particularly beneficial when requirements are ambiguous or likely to evolve over time. The model enables developers and stakeholders to collaborate closely, ensuring the final product meets user needs.

1. What is a Prototype?

A prototype is a simplified, working version of the intended software that demonstrates key functionalities. It provides a tangible representation of the application, allowing users and developers to visualize the system and refine its features before full-scale development begins.

1.1. Key Characteristics of the Prototyping Model:

1. **Iterative Process**: Involves repeated cycles of building, testing, and refining the prototype.
2. **User Involvement**: Users actively participate in providing feedback and shaping the product.
3. **Focus on Critical Areas**: Emphasizes understanding and validating core functionalities to minimize misunderstandings.

1.2. Phases of the Prototyping Model

1. Requirements Gathering

- Developers and users collaborate to identify core requirements.
- The focus is on understanding essential functionalities rather than finalizing every detail.

Example: For an e-commerce platform, initial requirements may include browsing products, adding items to a cart, and checkout functionality.

2. Quick Design

- A basic system design is created to outline the user interface and critical workflows.
- This phase prioritizes simplicity and usability.

Example: A rough layout of the homepage and product categories for the e-commerce platform.

3. Prototype Development

- A working model is built based on the quick design.
- This version includes essential features needed to gather user feedback.

Example: Users can browse products and add them to the cart, but payment processing is not implemented yet.

4. User Evaluation

- Users test the prototype and provide feedback on usability, missing features, and potential improvements.
- Feedback helps identify gaps and refine requirements.

Example: Users may suggest adding a search bar or filtering options for product categories.

5. Refinement

- The prototype is updated based on user feedback.
- This process is iterative and continues until stakeholders are satisfied.

Example: Adding the requested search bar and improving the navigation menu based on feedback.

6. Final Product Development

- Once the prototype is approved, the final system is developed.
- Additional features are implemented, and the product undergoes rigorous testing before deployment.

Example: The e-commerce platform is finalized with payment processing, user accounts, and mobile responsiveness.

2. Types of Prototypes

✓ **Throwaway (Rapid) Prototype**:

- Created quickly to clarify requirements.
- Discarded after feedback is incorporated.
- Example: A basic wireframe demonstrating website navigation.

✓ **Evolutionary Prototype**:

- Continuously refined and evolves into the final product.
- Example: A chatbot that starts with simple responses and grows smarter over time.

✓ **Incremental Prototype**:

- Developed in parts (modules) and integrated later.
- Example: First building the login page, then the dashboard.

✓ **Extreme Prototype**:

- Used for web applications; begins with the UI and adds backend functionality later.
- Example: A visually complete website that connects to APIs in later stages.

3. Advantages of the Prototyping Model

- **Improved User Involvement**:
    - Users actively participate, ensuring their needs are met.
- **Reduced Risk of Failure**:
    - Early feedback helps identify and fix potential issues.
- **Clarified Requirements**:
    - Prototypes help refine unclear or incomplete requirements.
- **Better Design**:
    - Iterative refinement leads to a more user-friendly and functional design.
- **Faster Delivery**:
    - Early versions of the product can be delivered for feedback, even if not fully complete.

4. Disadvantages of the Prototyping Model

- **Scope Creep**:
    - Users may continuously request changes, delaying the project timeline.
- **Increased Cost**:
    - Building and refining prototypes can require additional resources.
- **Inadequate Documentation**:
    - Iterative processes may lead to incomplete documentation of requirements and designs.
- **Misuse of Prototype**:
    - Stakeholders may mistake the prototype for the final product, leading to unrealistic expectations.

5. Applications of the Prototyping Model

Web Applications: Refining UI/UX and testing navigation workflows.

- Example: Prototyping a social media platform’s user interface.

Mobile Apps: Testing navigation and feature usability.

- Example: A fitness app prototype to validate functionality and user experience.

E-commerce Platforms: Visualizing customer journeys like browsing and checkout.

- Example: Prototyping an online store to test filtering and search functionalities.

Innovative Projects: Exploring new ideas and concepts.

- Example: Prototyping a virtual reality app to gather user insights and refine functionality.

6. Real-Life Example

Prototyping a Food Delivery App

Requirements Gathering:

- Users want to browse restaurants, view menus, and place orders.

Quick Design:

- A simple interface with a list of restaurants and an "Order Now" button.

Prototype Development:

- A working model that allows browsing restaurants but does not process orders.

User Evaluation:

- Feedback highlights the need for cuisine filters and delivery time estimates.

Refinement:

- Add filters and time estimates, then retest the prototype.

Final Product Development:

- Fully develop the app with payment gateways, GPS tracking, and user accounts.

7. Comparison with Other Models

Aspect	Prototyping Model	Waterfall Model	Agile Model
Flexibility	High (iterative improvements)	Low (linear process)	High (continuous iterations)
User Involvement	Extensive	Minimal	Extensive
Risk	Low (early feedback)	High (late problem detection)	Medium (requires team discipline)
Documentation	Minimal	Comprehensive	Moderate

Conclusion

The Prototyping Model is a user-focused and iterative approach ideal for projects with unclear or evolving requirements. By prioritizing user feedback and refining the system through prototypes, it ensures that the final product aligns closely with user needs. However, careful management is essential to prevent scope creep and increased costs.

References

- Pressman, Roger S., and Maxim, Bruce R. 2020. *Software Engineering: A Practitioner's Approach*. 9th Edition. New York: McGraw-Hill.
- Sommerville, Ian. 2020. *Software Engineering*. 10th Edition. Boston: Pearson Education.
- IBM. 2023. "What Is the Prototyping Model?" IBM Cloud Education. https://www.ibm.com/cloud/learn/prototyping-model.