



## The Straight and Methodical Way of Representing the Prototype Model

Shivani Sharma<sup>1</sup>, Aman Kaushik<sup>2</sup><sup>1</sup>M.Tech, <sup>2</sup>Assistant Professor<sup>1,2</sup>CSE Department, Baddi University of Emerging Science & Technology,  
Baddi, Solan, Himachal Pradesh, India

**Abstract-** This paper describes the general review on Prototype model. In the product design and development process, the prototyping or model making is one of the important step to finalize a product which help in conceptualization of a design. The prototype pattern support the software engineer and the customer to satisfactory understand what is to be built when requirements are fuzzy. Another important situation where the prototyping model can be used is when technical solutions are not clear to the development team. The Prototyping pattern starts with communication. The goal of this review paper isto help how they can help designers generate and share new ideas, get feedback from user, choose among design, alternatives and articulate reasons for their final choices.

**Keywords:** Process of prototype model, Problem, Types, Dimensions, Need, Merits and Demerits, Scope.

### I. INTRODUCTION

The prototype model is start by developing a small prototype than and followed by a mini waterfall process, primarily to gather requirements. Then the first prototype is reviewed is subsequent loops. The prototype paradigm begins with communication. The software engineer and customer meet and define to overall objectives for the software, identify whatever requirements are known. The fundamental purpose here is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements. This prototype is developed based on the currently known requirements. The prototyping model is applied when detailed information related to input and output requirements of the system is not available. In this model, it is assumed that all the requirements may not be known at the start of the development of the system. After the finalization of software requirement document, the prototype is discard and actual system is then develop using waterfall approach. Thus it is used as an input to waterfall model and produces main table and good quality software. A Prototype is toy implementation of the system. [1]

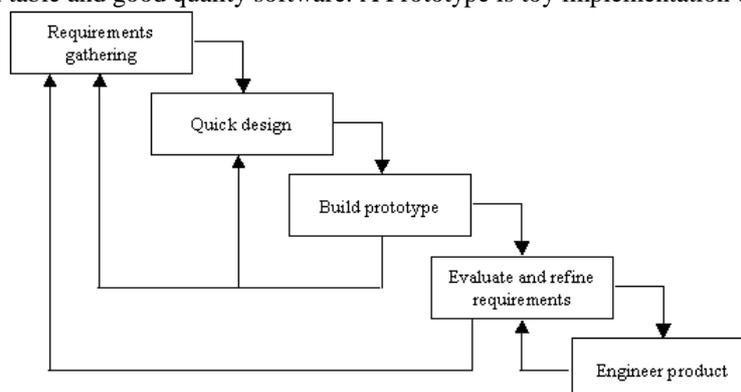


Fig. 1 Prototype model

### II. OUTLINE OF THE PROTOTYPE

- **REQUIREMENTS GATHERING:** Determine basic requirements including the input and output information desired. Details, such as security, can typically be ignored. The more intricate details of the internal design and external aspects like performance and security can be ignored at this stage.
- **QUICK DESIGN:** The initial prototype is developed that includes only user interfaces. The initial Prototype is developed in this stage, where the very basic requirements are showcased and user interfaces are provided.
- **BUILD PROTOTYPE:** The customers, including end-users, examine the prototype and provide feedback on additions or changes. The feedback is collected in an organized manner and used for further enhancements in the product under development.
- **EVALUATE AND REFINE PROTOTYPE:** Using the feedback both the specifications and the prototype can be improved. Negotiation about what is within the scope of the contract/product may be necessary. The changes

accepted are again incorporated in the new Prototype developed and the cycle repeats until customer expectations are met.

- **ENGINEER PRODUCT:** After the clearly specified the requirements a complete product is ready for the deliver to customer. [2]

### III. DIMENSIONS OF PROTOTYPE

#### HORIZONTAL PROTOTYPE

#### VERTICAL PROTOTYPE

- **HORIZONTAL PROTOTYPE:** A common term for a user interface prototype is the horizontal prototype. It provides a broad view of an entire system or subsystem, focusing on user interaction more than low-level system functionality, such as database access.
- **HORIZONTAL PROTOTYPES ARE USEFUL FOR:** Confirmation of user interface requirements and system scope. Demonstration version of the system to obtain buy-in from the business. Develop preliminary estimates of development time, cost and effort.
- **VERTICAL PROTOTYPE:** A vertical prototype is a more complete elaboration of a single subsystem or function. It is useful for obtaining detailed requirements for a given function, with the following benefits:
- Refinement database design Obtain information on data volumes and system interface needs, for network sizing and performance engineering Clarifies complex requirements by drilling down to actual system functionality. [3]

### IV. NEED OF PROTOTYPE

This type of System Development Method is employed when it is very difficult to obtain exact requirements from the customer. While making the model, user keeps giving feedbacks from time to time and based on it, a prototype is made. Completely built sample model is shown to user and based on his feedback, the SRS (System Requirements Specifications) document is prepared. After completion of this, a more accurate SRS is prepared, and now development work can start using Waterfall Model.

Prototyping gives you the opportunity to demonstrate your idea to potential buyers, get their feedback and perhaps improve the product. A good prototype can also encourage investment because they can see it in action and it will be easier for anybody wishing to provide funding, to better understand the value of the product.

Perhaps you need a prototype engineered to a high standard and built by professionals for presentation to potential buyers or investors, or you need a one-off model for photography or market research and feedback. You may simply need to test that a part works and require a batch of small component prototypes.

It is possible several prototypes will be required before your idea - and product model - is complete, since the need for refinement or amending design faults for example, is not uncommon. [4]

### V. TYPES OF PROTOTYPE

- **THROW AWAY PROTOTYPING:** Throw away prototyping is one type of approach where an initial prototype is built mainly focusing on the poorly understood requirements. Once the requirements are understood requirements document is updated and a conventional development process is followed to build system. Throwaway prototyping is also called as rapid or close ended prototyping. This type of prototyping uses very little efforts with minimum requirement analysis to build a prototype.
- **EVOLUTIONARY PROTOTYPE:** Evolutionary Prototypes have an advantage over Throwaway Prototypes in that they are functional systems. Although they may not have all the features the users have planned, they may be used on an interim basis until the final system is delivered. Evolutionary prototyping also called as breadboard prototyping is based on building actual functional prototypes with minimal functionality in the beginning.
- **INCREMENTAL PROTOTYPE:** Incremental prototyping refers to building multiple functional prototypes of the various sub systems and then integrating all the available prototypes to form a complete system. The final product is built as separate prototypes. At the end the separate prototypes are merged in an overall design. By the help of incremental prototyping we can reduce the time gap between user and software developer.
- **EXTREME PROTOTYPE:** Extreme Prototyping is an architectural process for developing applications, especially web applications, in terms of increasingly functional prototypes. At a high level it breaks down the web development into three distinct phases. The first phase is the static prototype, consisting of HTML pages and possibly a logical data model supporting those pages. The second phase is a coding process in your chosen web framework whereby the screens are fully functional using a simulated services layer. The third phase is where the services are implemented. The process is called Extreme Prototyping. [5]

### VI. STRENGTH OF PROTOTYPE

**IMPROVED AND INCREASED USER INVOLVEMENT:** Prototyping requires user involvement and allows them to see and interact with a prototype allowing them to provide better and more complete feedback and specifications.

- Time required to complete the project after getting final the SRS reduces, since the developer has a better idea about how he should approach the project.
- Errors can be detected much earlier.

- It quickly flushes out system requirements that the user may have not known or forgot to tell you about before you create the whole thing.
- Provides visual feedback to users and gives them something this type of approach of developing the software is used for non-IT-literate people. They usually are not good at specifying their requirements, nor can tell properly about what they expect from the software.
- Confusing or difficult functions can be identified Requirements validation, Quick implementation of, incomplete, but Functional, application.
- Reduced time and costs: Prototyping can improve the quality of requirements and specifications provided to developers. [6]

#### **VII. WEAKNESS OF PROTOTYPE**

- Once we get proper requirements from client after showing prototype model, it may be of no use. That It is a slow process.
- Too much involvement of client, is not always preferred by the developer.
- Too many changes can disturb the rhythm of the development team.
- Leads to implementing and then repairing way of building systems.
- Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.
- It can be time consuming if users continuously tweak demo after demo.
- Prototyping is usually done at the cost of the developer. So it should be done using minimal resources.[7]

#### **VIII. CONCLUSION**

Prototyping is an essential component of interactive system design. Prototypes may take many forms, from rough sketches to detailed working prototypes. They provide concrete representations of design ideas and give designers, users and Developers and managers an early glimpse into how the new system will look and feel. Prototypes increase creativity, allow early evaluation of design ideas, help designers think through and solve design problems, and support communication within multi-disciplinary design teams. Prototypes, because they are concrete and not abstract, provide a rich medium for exploring a design space. They suggest alternate design paths and reveal important details about particular design decisions. Perhaps most important, prototypes provide one of the most effective means for designers to communicate with each other, as well as with users, developers and managers, throughout the design process.

#### **ACKNOWLEDGEMENT**

I would like to first thank my Academic advisor, Asst. Professor Aman Kaushik, Who has been guiding my study and teaching me how to do great review on paper.

#### **REFERENCES**

- [1] Software Engineering, 6<sup>th</sup> edition by Roger S. Pressman
- [2] Fundamentals of software Engineering, 2nd edition, by Rajiv Mall
- [3] Software Prototyping by Vishnu Chai Tanya reddy Nara.
- [4] <http://www.ianswer4u.com>
- [5] Software Engineering, 3rd edition by K.K Aggarwal.
- [6] <http://www.iotap.com/blog/entryid/124/advantaged>
- [7] <http://istqbexamcertification.com/>