

Improved Software Quality and Design Standards Based on Customer Preferences by Applying Evolutionary Prototyping Software Development Model

S. Adarsh, D. Harish, K. Balaganapathy, R. Venkatachalapaty, E. Abishiek, M. Nagarajan

Abstract— The concept of Evolutionary prototyping software development model came into existence that emphasized on the need to follow some structured approach towards building new or improved system. In this paper, we focus on the analysis of the Evolutionary Prototyping model and its application in the design and development of Graphical User Interface (GUI). The IT industries invest huge capital sums for developing consumer products. These products must satisfy all requirements of the customers. Sometimes the requirements may not be clear, or there may be different solutions for a problem. In such cases, the developers often make a prototype of the product before the actual product is produced. This prototype helps the producers to better understand the project requirement. This prototype can be used to improve the design of the product, so that it is favoured by the customer. Also the prototype helps in identifying errors in design at a very early stage of production, or rather before the production begins, thus improving the product quality.

Index Terms— Design standards, Evolutionary prototyping, Graphical User Interface, Mobile Launchers, Software development model, Software quality.

I. INTRODUCTION

There are several models for such processes, each describing approaches to a variety of activities that take place during the process. Any software development process is divided into several logical stages that allow a software development company to organize its work efficiently in order to build a software product of the required functionality within a specific time frame and budget. In this paper we focus on the Evolutionary prototyping model and its application in development of GUIs for various devices. We particularly focus on the application of this method in development of Launcher applications employed in smart phones and other handheld electronic devices.

Change is inevitable in software projects. A technology

S. Adarsh, Information Technology, P.S.G College of Technology, Coimbatore, Tamil Nadu, India.

D. Harish, Information Technology, P.S.G College of Technology, Coimbatore, Tamil Nadu, India.

K. Balaganapathy, Information Technology, P.S.G College of Technology, Coimbatore, Tamil Nadu, India.

E. Abishiek, Information Technology, P.S.G College of Technology, Coimbatore, Tamil Nadu, India.

M. Nagarajan, Information Technology, P.S.G College of Technology, Coimbatore, Tamil Nadu, India.

R. Venkatachalapaty, Information Technology, P.S.G College of Technology, Coimbatore, Tamil Nadu, India.

keeps on replacing the older ones continuously. As the technology changes, the system requirement may change due to factors like changing customer priorities. New technologies also introduce new design and implementation possibilities. Therefore the product developed must be able to accommodate changes. Business and product requirements often change as the product goes through various production stages, thus making linear approach unrealistic.

The change in design and requirements result in increased costs, time and increases the possibility of introduction of new bugs. This includes re-analyzing relationship between requirement, re-analyzing requirements, re-testing the product for errors and going through the various stages of development. This is called rework.

Consider few more cases, where the market deadline is tight for complete development of a product; a set of core product requirements is well understood, but the details are yet to be defined. In other cases, the developer may be unsure of the efficiency of an algorithm, or the human-machine interaction needed, or the adaptability of the product to different working environment. In such cases an Evolutionary Prototyping development model can be used to reduce the cost of the rework involved.

An Evolutionary prototyping model helps in change avoidance, by anticipating the changes before production phase. This model is iterative and is characterized in a way to help Software engineers develop increasingly more complete versions of software. For example: A prototype may be developed to show the key features of a software product to the customer. The customer can then help them to refine the requirements before production begins.

II. EVOLUTIONARY PROTOTYPING

A Prototype is an initial version of a software system that is used to demonstrate concepts, try out design options, and find out more about the problem and its possible solutions. Rapid iterative development of the prototype is essential to control the cost and allow stakeholders to experiment with the prototype early in the production process. A prototype serves as a mechanism for identifying the requirements.

The prototyping model has an iterative approach and each iteration has five major steps as listed below.

- Communication
- Quick Plan
- Modelling

Improved Software Quality and Design Standards Based on Customer Preferences by Applying Evolutionary Prototyping Software Development Model

- Construction
- Deployment, delivery and feedback

A. Communication:

The prototype paradigm begins with Requirement elicitation and validation of system requirements. The software engineer and customer meet and define the overall objectives for the software and outline areas where further definition and detailing is required.

B. Planning and Modelling:

The development process then involves planning a quick iteration and modelling a quick design. This design leads to the construction of a prototype.

C. Construction:

The model is then used to construct a static or working prototype. The prototype undergoes some basic testing, to check if it meets the requirements. The prototype is tested to have only a minimum quality. Software quality is not given priority in the initial iterations. Once the customer is satisfied with the prototype, the production phase starts where strict quality testing is done and the software quality is enhanced.

D. Deployment and Feedback:

The prototypes helps the customers see how well the product serves its purpose. Customer feedback is used to refine the requirements.

Furthermore as the prototype is developed, new bugs may be identified and a few requirements may be dropped. Iterations help in tuning the customer requirements and also help the developer to better understand what need to be done. According to Brooks [BRO75], "In most projects, the first system built is barely usable. It may be too slow, too big, awkward to use or all three. There is no alternative but to start again, smarting but smarter, and build a redesigned version in which the problems are solved". The prototype can serve as "the first system". The prototyping is one of the popular development models because the user can get a feel of the actual system, and the developer get to develop something immediately.

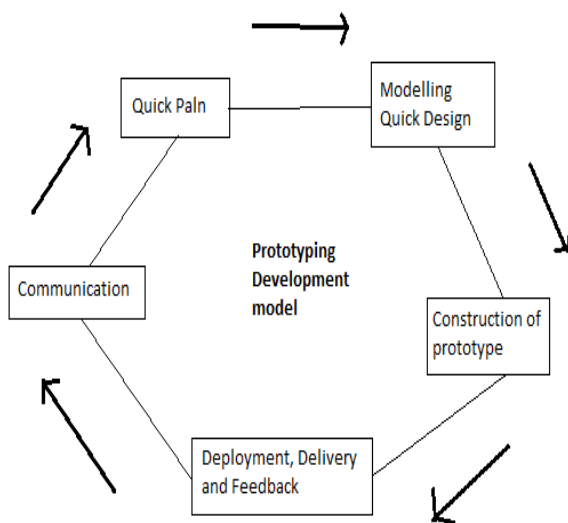


Fig 1: Prototype Development Cycle Model

III. CASE STUDY: GRAPHICAL USER INTERFACE

The prototyping model is suitable for products where its functionality can be tuned only through user feedback, after they get a feel of it. One such application is the Graphical User Interface (GUI). The graphical user interface is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation. GUIs were introduced in reaction to the perceived steep learning curve of command-line interfaces (CLIs), which require commands to be typed on a computer keyboard.

The Graphical User Interface was first introduced in the Xerox Alto, which became popular with the arrival of Windows 95 in the 1990s. The GUI has revolutionized the way humans interact with the machine. Mouse came into use which helped us to point at various tools, icons, and options in the screen. Graphical interface is used to communicate with all electronic devices today. The GUI has completely replaced the CLIs.

Beyond computers, GUIs are used in many handheld mobile devices such as MP3 players, portable media players, gaming devices, smart phones and smaller household, office and industrial controls.

A. GUI in handheld devices

Today phones and other handheld electronic devices have become smarter. Everyone today has a smart phone. Android phones especially have become popular globally. The user communicates with these devices through touch screens. These smart phones have a colourful and user friendly Graphical interface. We concentrate on the mobile launchers that help us to customize the home screen.

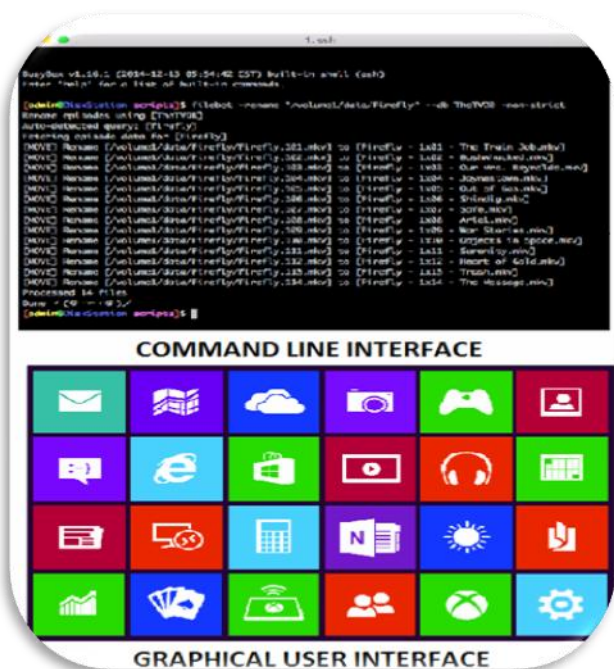


Fig. 2 Difference between GUI and CLI

Launcher is the name given to the part of the Android user interface that lets users customize the home screen (e.g. the phone's desktop), launch mobile apps, make phone calls, and perform other tasks on Android devices (smart phones and other devices that use the Android mobile operating system). We must agree that the success of smart phones is due to the ease with which the user can operate them. Thanks to these Launchers.

IV. PROTOTYPING THE GUI

The above discussed prototyping model can be employed to design such mobile launchers. As said earlier the success of the phone depends on the ease of usage and the appearance of the screen. In such cases the designers must ensure that their launchers are to the liking of the user.

It is not easy to gather the requirements for designing a launcher. The designer may know the basic system requirements like the Operating system, the environment, and basic functionalities like launching applications, and so on. But other requirements like the desktop design, the animations used to change between screens are not properly designed. The user is attracted to the appearance of the launcher than to any other factor. So the designer has to ensure that the product is popular among different types of user (ranging from kids, students, office goers, the aged etc.)

The designer will have a tight timeframe to develop such software products, because of business competitions. Also, it is not possible to explain the design of the launcher to the user through diagrams and sketches. The user can have some idea about the launcher only by feeling them. So Evolutionary prototyping is the only realistic solution to design software products where the user needs to have a feel of the product to understand about it.

Consider developing the mobile launchers using some other development model. The designer will take time to gather all the requirements, then make a design, construct the launcher, perform quality analysis, and finally deliver the product. Only after the final stage the user gets to feel the product. If the user is not satisfied with the design, then the entire time and cost spent on the project goes waste. This problem can be overcome using Prototyping model.

Steps:

A. Communication:

The prototyping process starts with the requirement elicitation step. The designer is aware of few requirements initially, that doesn't tend to change like the operating environment of the Launcher, the basic functionalities that the Launcher must support and so on. But still the requirements are imprecise.

B. Quick design and construction:

Based on the known requirements, the designers can perform a quick modelling, and construct a basic prototype of the Launcher. The prototype is constructed in a very short timeframe. The designer doesn't concentrate much in the quality at the earlier iterations.

C. Feedback:

The user now has a feel of the product. The user is able to actually use the interface to communicate or work with the device. As the user now has a feel of the interface, he/she is in a position to give a better feedback which will lead to more precise requirements. Based on the user feedback the requirements are refined and the features of the product are tuned.



MOBILE LAUNCHERS

Fig. 3 Mobile Launchers

The process repeats until the user is satisfied with the prototype of the Launcher. Once the designer and the user are satisfied with the Launcher, the software product can be developed from this prototype.

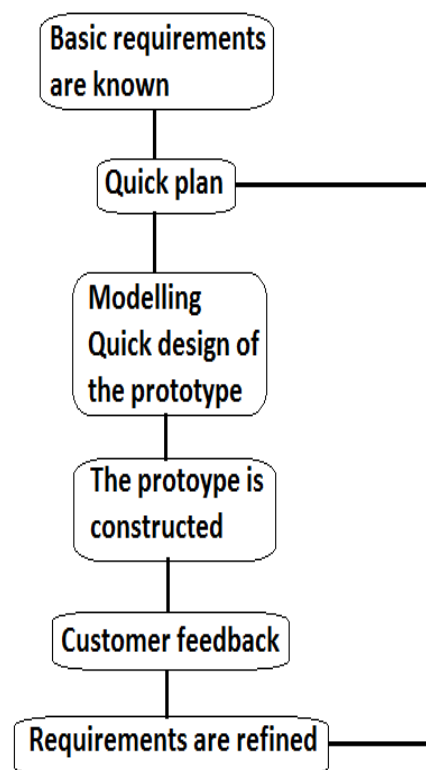


Fig. 4 Flow chart of stages in Development of GUI

This method thus anticipates the future requirement, and saves time and money that had to be spent changing needs. It also gives satisfaction to both the designer and the user, as the user gets the feel of the launcher in a shorter time, and the

Improved Software Quality and Design Standards Based on Customer Preferences by Applying Evolutionary Prototyping Software Development Model

designer has the satisfaction of developing a basic launcher quickly. As the probability of some requirement to change in the future is less, the chance of introduction of new bugs is reduced. Once the Launcher passes through the quality analysis, there is very less chance for Software wearing. One such example is the Windows XP Operating system. Ever since its introduction in the early 2000's its popularity has not yet faded, because of the user friendly interface. It has been a successful product in the market for the past 15-20 years.

V. IMPROVED DESIGN AND QUALITY

The application of prototyping model in the development of Launchers for mobile phones was discussed above. The same procedure can be employed for developing GUI for other electronic devices also. This method is generally employed when the requirements are not well understood. The prototype model helps in refining the requirements based on the customer feedback. So it supports the problems with requirements that tend to change frequently. The prototype model provides guarantee of success in a short timeframe. The process is a simple one and the risk involvement is low. This development model is highly flexible, provides high priority to user feedback and, assures successful production of the product.

The repeated iterations refine the requirements. This leads to better design and modelling of the prototype and the iterations continue. Finally when the customer is satisfied with the prototype, the production phase is imitated. The final prototype will have the best design, as the requirements will be well defined. So this method provides a improved design of the product when compared with all other methods.

As the prototype model is follows an iterative approach, the possible error, and bugs are identified in various iterations which are corrected before the next iteration. So the final prototype would have dealt with all possible quality problems. So there is least possibility for bugs in the final product. Moreover the product itself undergoes a quality testing before deployment. So because of several testing stages the quality of the product is assured. So this method provides an improved product design and quality.

VI. ADVANTAGES OF PROTOTYPING

This method has several advantages over other software development methods, some of which were discussed above. In general the prototyping method has the following advantages:

- Supports frequently changing requirements
- Guarantees success of the project
- Has a simple development procedure
- Risk Involved is very low
- Low or Medium expertise is required
- Facilitates easy incorporates of changes
- User involvement is high
- Highly flexible process
- Project can be completed in a short timeframe

VII. DIFFICULTIES

Every development model has its own pros and cons. A few of the difficulties in prototype model has been discussed

here. The developers are sometimes pressured to deliver throwaway prototypes, particularly when there is a tight schedule.

- The customer misunderstands the prototype to be a working version of the software.
 - The designer makes implementation compromises to make the prototype working quickly. Example: while developing the launchers, an inappropriate Operating system may be used, or a less efficient and less secure algorithm may be used. After getting the customer's nod, the developer may be tempted to use the same choices which were less than ideal. This risks the software quality of the product.
 - Overall cost involved in this method is High
 - Does not provide either Cost control or Resource control
 - Routine maintenance is required
 - Integrity and security is weak
 - Doesn't encourage reusability
- Improper documentation

VIII. CONCLUSION

Evolutionary prototyping model is the only possible and realistic solution to those problems where the user cannot be made to understand the working of a product through diagrams and sketches. This development process actually provides the user with a working model to the user, which the later can feel and understand its working better.

This model refines the requirements and gives a clear picture of the final product to both the customer and the developer. The prototypes are very cheap to develop and can be constructed in a few days. This model reduces the cost of rework by helping the product to have two features: change avoidance and change tolerance. Thus evolutionary prototyping model can be used for such cases where the user needs a feel of the product to understand its working, and when the requirements are fuzzy.

IX. ACKNOWLEDGMENT

We would also like to show our gratitude to Dr. Raja Mohana S.P., PSG College of Technology, Coimbatore for sharing her pearls of wisdom with us which helped us in enhancing our work. We are also immensely grateful to Dr. Raja Mohana S.P. for her comments on an earlier version of the paper, although any errors are our own.

REFERENCES

- [1] Schmidt, D. C. (2006). 'Model-Driven Engineering'. IEEE Computer, 39 (2), 25-31.
- [2] Rettig, M. (1994). 'Practical Programmer: Prototyping for Tiny Fingers'. Comm. ACM, 37 (4), 21-7.
- [3] Rumbaugh, J., Jacobson, I. and Booch, G. (1999). The Unified Software Development Process.
- [4] Budgen, D. (2003). Software Design (2nd Edition). Harlow, UK.: Addison-Wesley.
- [5] Laura C. Rodriguez Martinez, Manuel Mora ,Francisco,J.Alvarez, "A Descriptive/Comparative Study of the Evolution of Process Models of Software Development Life Cycles",Proceedings of the 2009 Mexican International Conference on Computer Science IEEE Computer Society Washington, DC, USA, 2009.

- [6] Jovanovich, D., Dogsa, T., "Comparison of software development models," Proceedings of the 7th international Conference on, 11-13 June 2003, ConTEL 2003, pp. 587-592.
- [7] Comparative study of prototype model for software engineering with system development life cycle, Rajendra Ganpatrao Sabale, Dr. A.R. Dani, IOSR Journal of Engineering, ISSN: 2250-3021 Volume 2, Issue 7(July 2012).
- [8] Software Engineering, a Practitioner's Approach, Roger S Pressman, Tata McGraw-Hill.
- [9] A Case Study of Rapid Prototype as Design in Educational Engineering Projects, O. DIEGEL, W. L. XU and J. POTGIETER.
- [10] Software Engineering, Ian Sommerville, Addison-Wesley.
- [11] <http://www.mhhe.com/engineering/pressman>

S. Adarsh: I'm currently perusing B.Tech Information Technology (II year), in P.S.G College of Technology, Coimbatore. I completed Secondary schooling in Kendriya Vidyalaya, Dharmapuri, with 10 CGPA in the 10th board exams. Then I completed class 12 in Sri Vijay Vidyalaya Mat. Hr. Sec. School (Boys), Dharmapuri, scoring 99% in Board exams. Currently I have 9.84 CGPA in college.

E. Abishek: I'm currently perusing B.Tech Information Technology (II year), in P.S.G College of Technology, Coimbatore. I completed Secondary schooling in Adharsh Vidyalaya, Nagapatinam, and scored 91% in 10th board exams. Then I completed class 12 in SRV Mat. Hr. Sec. School (Boys), Salem, scoring 95% in Board exams. Currently I have 7.55 CGPA in college.

K. Balaganopathy: I'm currently perusing B.Tech Information Technology (II year), in P.S.G College of Technology, Coimbatore. I completed Secondary schooling in KVS Mat. School, Virududnagar and scored 95.6% in the 10th board exams. Then I completed class 12 in KVS Mat. school, Virududnagar scoring 96.8% in Board exams. Currently I have 7.74 CGPA in college.

D. Harish: I'm currently perusing B.Tech Information Technology (II year), in P.S.G College of Technology, Coimbatore. I completed Secondary schooling in St. Judes public school and junior college (ICSE), and scored 87% in 10th board exams. Then I completed class 12 in SRV Mat. Hr. Sec. School (Boys), Salem, scoring 96% in Board exams. Currently I have 8.53 CGPA in college.

R. Venkatachalabathy: I'm currently perusing B.Tech Information Technology (II year), in P.S.G College Of Technology, Coimbatore. I completed Secondary schooling in Arasunager Mat. School, Nagapatinam, and scored 94.2% in 10th board exams. Then I completed class 12 in SRV Mat. Hr. Sec. School (Boys), Trichy, scoring 96.25% in 12th Board exams. Currently I have 8.29 CGPA in college.

M. Nagarajan: I'm currently perusing B.Tech Information Technology (II year), in P.S.G College Of Technology, Coimbatore. I completed Secondary schooling in Carmel Hr. Sec. School, Kanyakumari, Nagarcovil, and scored 89.75% in 10th board exams. Then I joined P.S.G Polytechnic College, Coimbatore. I did a diploma course in Computer networking, and secured 87%.