

System Development Method with The Prototype Method

Azhar Susanto; Meiryani

Abstract: some users have difficulty expressing their desire to get an application that suits their needs. & this difficulty needs to be resolved by analysts by understanding user needs and translating them into prototypes. This model is then continually improved until it is according to user needs. The application of the system is important for the information system developer in assessing the success of the information system whether it runs according to the plan and whether it has fulfilled the wishes of its users. Unclear planning and analysis often creates problems that will affect development activities. This study will provide an overview of the use of prototyping models in information system development activities in the hope of producing a prototype as one of the first steps in developing an information system. Prototype is created with the aim of providing equal perception and initial understanding of the basic processes of the system to be developed, so that there will be good communication between developers and system users.

Index Terms: System Development, Prototyping, Method, Information Systems.

1 INTRODUCTION

Prototyping is a system development process often using a prototype approach. This method is very well used to solve the problem of misunderstanding between the user and the analyst that arises due to the user being unable to clearly define his needs (Mulyanto, 2009). Prototyping is the rapid development and testing of working models (prototypes) of new applications through an interaction and repetitive process that is commonly used by information system experts and business experts. Prototyping is also called rapid application design (RAD) because it simplifies and speeds up system design (O'Brien, 2005). Some users have difficulty expressing their desire to get an application that suits their needs. Prototyping is the rapid development and testing of working models (prototypes) of new applications through an interaction and repetitive process that is commonly used by information system experts and business experts. Prototyping is also called rapid application design (RAD) because it simplifies and speeds up system design (O'Brien, 2005). This method is very well used to solve problems of misunderstanding between users and analysts arising from the user being unable to clearly define his needs (Mulyanto, 2009). Raymond McLeod, prototype is defined as a tool that gives ideas to potential makers and users about how the system functions in its full form, and the process of producing a prototype called prototyping. Prototyping is a process of making simple models that allow users to have a basic description of the program and do initial testing. Prototyping provides facilities for developers and users to interact with each other during the manufacturing process, so that developers can easily model the software that will be created. Prototyping is one of the most widely used software development methods. A prototype is a part of a product that expresses the logic and physical external interface that is displayed. Potential consumers use prototypes and provide input to the development team before large-scale development begins. Seeing and trusting is what is expected to be achieved in a prototype.

Using this approach, consumers and the development team can clarify their needs and interpretations. Software prototyping (software prototyping) or life cycle using prototyping (life cycle using prototyping) is one of the system life cycle methods based on the working model concept. The aim is to develop the model into a final system. This means that the system will be developed faster than traditional methods and the costs will be lower. There are many ways to photograph, as well as their use. A distinctive feature of this methodology is the system developer (system developer), the client, and the user can view and experiment with parts of the computer system from the beginning of the development process. The model shows the client what the system will do, but is not supported by detailed structural design. When designers and clients experiment with various ideas on a model and agree with the final design, the actual design is made exactly like a model with better quality. Prototyping helps in finding needs in the early stages of development, especially if the client is not sure where the problem originates. Besides that prototyping is also useful as a tool for designing and improving user interfaces - how the system will be seen by the people who use it.

2 LITERATURE REVIEW

2.1 Prototype Form

Based on the characteristics of a system prototype can be in the form of low fidelity and high fidelity. Fidelity refers to the level of detail of a system (Walker et al, 2003). Low fidelity prototypes do not describe the system too detailed. The characteristics of the low fidelity prototype are having limited functions or interactions, more describing the concept of design and layout compared to the interaction model, not showing in detail the operational system, Generally demonstrate the feel and look of the user interface and only describe the concept of the approach in general (Walker et al, 2003). High fidelity prototype more detailed describes the system. This prototype has full interaction with users where users can enter data and interact with the system, representing core functions so that they can simulate most functions of the final system and have a very similar appearance to the actual product (Walker et al, 2003). Features that will be implemented on a system prototype can be limited to vertical or horizontal techniques. Vertical prototypes contain detailed functions but only for a few selected features, not for all system features. Horizontal

- Azhar Susanto; Accounting Department, Faculty of Economics and Business, Padjadjaran University, Bandung, Indonesia
- Meiryani; Accounting Department, Faculty of Economics and Communication, Bina Nusantara University, Jakarta, Indonesia 11480
meiryani@binus.edu

prototypes include all user interface features, but without the main functions, only in the form of simulations and cannot be used to do actual work (Walker et al, 2003).

2.2 Prototype Making Process

The process of making a prototype is an interactive and repetitive process that combines traditional development cycle steps. The prototype was evaluated several times before the end user stated that the prototype was received.

Prototype steps:

a. System Requirements Analysis

Development of information systems requires investigation and analysis of the reasons for ideas or ideas to develop and develop information systems. Analysis is done to see the various components used by the system that are currently running, including hardware, software, networks and human resources. The analysis also documents information system activities including input, processing, output, storage and control (O'Brien, 2005). Then conduct a feasibility study to formulate information needed by end users, resource requirements, costs, benefits and feasibility of the proposed project (Mulyanto, 2009). System requirements analysis as part of the initial study aims to identify system specific problems and needs. System specific needs are specifications regarding the things the system will do when implemented (Mulyanto, 2009). System requirements analysis must define specific system requirements including:

- 1) Input needed by the system (input)
- 2) Output produced (output)
- 3) Operations carried out (process)
- 4) Source of data handled
- 5) Control (control)

System Requirements Specifications

The stage of system requirements analysis requires evaluation to determine the ability of the system by defining what the system should be able to do then determining the criteria that must be met by the system. Some criteria that must be met are achievement of goals, speed, cost, quality of information produced, efficiency and productivity, accuracy and validity and reliability or reliability (Mulyanto, 2009).

b. System Design

System analysis (system analysis) describes what the system must do to meet the user's information needs. System design (system design) determines how the system will meet these objectives. System design consists of design activities that produce functional specifications. System design can be seen as the design of interfaces, data and processes with the aim of producing specifications that are appropriate to the product and user interface methods, database structure and processing and control procedures (Ioanna et al., 2007). System design will produce a prototype software package, a good product should include seven parts:

- 1) Quick and easy menu features.
- 2) Display input and output.
- 3) Printable reports.
- 4) Data dictionary that stores information on each field including field length, editing in each report and the format of the field used.
- 5) Database with optimal format and key record.

- 6) Display online queries precisely to data stored on the database.
- 7) A simple structure with a programming language that allows users to do special processing, time of occurrence, automatic procedures and others.

C. System Testing

The prototype software package is tested, implemented, evaluated and modified over and over until it is acceptable to the user (O'Brien, 2005). System testing aims to find errors that occur in the system and make system revisions. This stage is important to ensure that the system is error free (Mulyanto, 2009). According to Sommerville (2001) system testing consists of:

- 1) Test units to independently test individual components without other system components to guarantee the correct operating system.
- 2) Testing modules consisting of interconnected components.
- 3) Testing the sub-system consisting of several modules that have been integrated.
- 4) Testing the system to find errors caused by interactions between subsystems and their interfaces and validating functional and non-functional requirements.
- 5) Testing of acceptance with data entered by the user and not the test of simulation data.
- 6) Documentation in the form of recording each step of the work from the beginning to the end of making the program.

Web-based information system testing can use traditional software testing techniques and methods. Testing web applications includes link testing, browser testing, usability testing, load testing, voltage and external testing (Simarmata, 2009). Acceptance of users (users) of the system can be evaluated by measuring user satisfaction with the system being tested. Satisfaction measurements include the appearance of the system, conformity with user requirements, speed and accuracy of the system to produce information that the user wants. There are several models for measuring user satisfaction with the system, including the Technology Acceptance Model (TAM), End User Computing (EUC) Satisfaction, Task Technology Fit (TTF) Analysis and Human Organizational Technology (HOT) Fit Models. One measurement model that has been translated into several different languages and does not show significant differences in measurement results is End User Computing (EUC) Satisfaction. This model emphasizes user satisfaction with technological aspects including aspects of content, accuracy, format, time and ease of use of the system (Chin & Mathew, 2000).

D. Implementation

After the prototype is received, then at this stage it is the implementation of a system that is ready to operate and then there is a learning process towards the new system and comparing it with the old system, technical and operational evaluation and user, system and information technology interactions.

2.3 Prototype Model

All Prototyping model is a process of making software that is repetitive and with rapid planning where there is feedback that allows the occurrence of repetition and improvement of software until the software meets the needs of the user. Prototyping models are one simple model of making software which allows users to have an initial / basic description of the program and carry out initial testing based on the working model concept.

The purpose of the prototype?

Prototyping models have the goal of developing the initial software model into a final system.

Prototype processes

The prototyping model processes are:

- 1) First communication is carried out between the customer and the software development team regarding the specifications of the desired requirements.
- 2) Planning and modeling will be done quickly in the form of quick design (quick design) and then will begin construction of the prototype.
- 3) The prototype will then be submitted to the stakeholders for further evaluation before being submitted to the software makers.
- 4) Making software in accordance with the prototype that has been evaluated which will then be submitted to the customer.
- 5) If it has not met the needs of the customer, it will return to the initial process until the needs of the customer have been fulfilled.

The processes in the prototyping model in general are as follows:

1) Collection of needs

Developers and clients will meet first and then determine general goals, known needs and an overview of the parts that will be needed next.

2) Designing

The design was done quickly and the design represented all known aspects of the software, and this design became the basis for making prototypes.

3) Evaluation of the prototype

The client will evaluate the prototype made and used to clarify the software requirements.

2.4 Data Mining Stages

To model a software it takes several stages in the development process. This stage will determine the success of a software. Software developers must pay attention to the stages in the prototyping method so that the final software can be accepted by its users. The stages in the prototyping are as follows:

- 1) Collection of needs. Customers and developers together define the format and overall software requirements, identify all needs, and outline the system that will be created.
- 2) Building prototyping. Build prototyping by creating a temporary design centered on presenting to

customers (for example by making input and sample output).

- 3) Evaluation of prototyping. This evaluation is carried out by the customer whether the prototyping that has been built is in accordance with the wishes of the customer. If it is appropriate, the fourth step will be taken. If not, then prototyping is corrected by repeating steps 1,2 and 3.
- 4) Encoding the system. In this stage the agreed prototyping is translated into the appropriate programming language.
- 5) Test the system. After the system has become a software that is ready to use, it must be tested before use. This test is carried out with white boxes, black boxes, base paths, architectural testing and others.
- 6) Evaluate the system. Customers evaluate whether the finished system is as expected. If it is, then the seventh step is done, if not then repeat steps 4 and 5.
- 7) Using the system. Software that has been tested and accepted by customers is ready to use.

Stages of prototype methodology:

- 1) Collection of needs and improvements. Establish all requirements for software development.
- 2) Fast design. The translation phase of the needs or data that has been analyzed into a form that is easily understood by the user.
- 3) Form of prototype. Translating data that has been designed into a programming language.
- 4) Customer evaluation of the prototype. Programs that have already been tested by customers, and if there are deficiencies in the program can be added.
- 5) Repair of prototypes. Improved program that has been made, according to consumer needs. Then the program is re-created and evaluated by consumers until all user needs are met.
- 6) Engineering products. The finished program and all user needs have been fulfilled.
- 7) Features that will be implemented on a system prototype can be limited to vertical or horizontal techniques. Vertical prototypes contain detailed functions but only for a few selected features, not for all system features. Horizontal prototypes include all user interface features, but without the main functions, only in the form of simulations and cannot be used to do actual work (Walker et al, 2003).
- 8) What is different from this prototype methodology, when compared to waterfall, is the creation of a prototype of an application, before the application enters the design phase. In this phase, the prototype that has been designed by the developer will be given to the user to get evaluated. This stage will continue to be repeated until both parties really understand the requirements of the application to be developed. When the prototype is complete, the application stage will continue to the design stage and return to following the steps in the waterfall model. The disadvantage of this type is that the application developer development team must have good capabilities because in developing this prototype there is only a short time. A prototyping is a system in very minimal functions.

3 DISCUSSION

The advantages of prototyping are:

- 1) There is good communication between the developer and the customer.
- 2) Developers can work better in determining customer needs.
- 3) Customers play an active role in system development.
- 4) Save more time in system development.
- 5) Application becomes easier because the user knows what is expected.

While the disadvantages of prototyping are:

- 1) The customer does not see that the software does not reflect the overall quality of the software and has not considered maintenance for a long time.
- 2) Developers usually want to quickly complete projects so that they use simple algorithms and programming languages.
- 3) Customer relationships with computers may not describe good design techniques.

This development model (Prototyping Model) has several advantages, including:

- o There is good communication between developers and customers
- o Developers can work better in determining customer needs
- o Customers play an active role in developing the system
- o Save more time in developing the system
- o Application becomes easier because the user knows what is expected
- o make the client get the initial picture of the prototype
- o Helps get better detailed needs.

Implementation of Model Prototyping

The prototyping method as a new paradigm in the development of management information systems, is not just an effusion of existing information system development methods, but at the same time is a revolution in the development of management information systems. This method is said to be reflux because it changes the old information system development process (SDLC). According to the literature, what is meant by a prototype (prototype) is "the first model", which is often used by industrial companies that produce goods in a period. But in relation to the second definition information system from Webster which states that "the prototype of individual exhibits of the essential features of later type", which when applied in the development of management information systems can mean that the prototype is an information system that describes important things from the upcoming information system. The information system prototype is not something complete, but something that must be modified again, developed, added or combined with other information systems if necessary. In some cases software development is different from manufacturing products, each stage or phase of information system development is an integral part of the entire process that must be done. This process is generally only for one product and the characteristics of the product cannot be determined exactly like manufactured products, so the use of the "first model" for software development is not appropriate. The term prototyping in relation to the development of management information system software is more of a process rather than a prototype

as a product. For example, the car maker can develop a prototype that can be used in a specific aiming track and then displayed in a showroom. Information obtained from such treatment can be used to improve the design before implementation / production is done in bulk.

4 CONCLUSION

Based on the description and discussion in the study of the prototype model conclusions can be drawn as follows:

1. Required careful planning in system planning and development
2. Planning starts from a good system design, personnel selection, software and hardware determination, and network architecture Department of Information Technology Faculty - Merdeka University Pasuruan 61
3. Clear commitment between developers, users, and leaders of the organization.

ACKNOWLEDGMENT

The authors wish to thank to Padjadjaran University, Bandung Indonesia and Binus University, Jakarta, Indonesia.

REFERENCES

- [1] Connolly, T.M.; Carolyn E. Begg. 2010. "Database Systems: A Practical Approach to Design, Implementation, and Management". 5th Edition. America: Pearson. Education.
- [2] Ladjamudin, 2013, Analisis dan Desain Sistem Informasi. Graha Ilmu. Yogyakarta.
- [3] Ogedebe, P.M., & Jacob, B.P. , 2012, Software Prototyping: A Strategy to Use When User Lacks Data Processing Experience. ARPN Journal of Systems and Software. VOL. 2, NO.6 , 2012, http://scientific-journals.org/journalofsystemsandsoftware/archive/vol2no6/vol2no6_4.pdf.
- [4] Satzinger, Jackson, Burd, 2012, Systems Analysis And Design In Changing World, Ed. 9.
- [5] Sommerville, Ian. 2011. Software Engineering (Rekayasa Perangkat Lunak). Jakarta: Erlangga.
- [6] <https://abhique.blogspot.com/2012/11/metode-prototyping-dalam-pengembangan.html>.
- [7] <http://rizalloa.ilearning.me/?p=132>
- [8] <http://awaysidik.blogspot.com/2013/07/metode-prototyping-menurut-raymond.html>