International Journal of Engineering Innovation & Research
Volume 2, Issue 3, ISSN: 2277 – 5668

Software System Development through Software Engineering Methods

Arati Dandavate

***Abstract* – The development of modern day computer was the result of advances in technologies and man's need to quantify. One cannot imagine life without computers in today's world. You go to a railway station for reservation, you want to web site a ticket for a cinema, you go to a library, or you go to a bank, you will find computers at all places. Since computers are used in every possible field today, it becomes an important issue to understand and build these computerized systems in an effective way.**

Building such systems is not an easy process but requires certain skills and capabilities to understand and follow a systematic procedure towards making of any information system. For this, experts in the field have devised various methodologies. Waterfall model is one of the oldest methodologies. Later Prototype Model, Object Oriented Model, Dynamic System Development Model, and many other models became very popular for system development. For anyone who is a part of this vast and growing Information Technology industry, having basic understanding of the development process is essential. For the students aspiring to become professionals in the field a thorough knowledge of these basic system development methodologies is very important.

In this paper author tried to focus on basic methodologies and concepts involved in software system development and their significance and tried to make comparisons between different techniques.

***Keywords* – Close System, Open System, Prototype, System, Systema.**

I. INTRODUCTION

To understand System Analysis and Design, one has to first understand what exactly systems are. The term "System" is derived from the Greek word systema. It means an organized relationship among functioning units or components. We can define a System as a combination of resources or functional units working together to accomplish a given task.

The term "working together" in system definition is very important as all the components are interrelated and interdependent and cannot exist independently. As the definition says, these components interact with each other to accomplish a given task, which is actually the objective of the system[1].

The components that comprise a system may be the various inputs required by the system, the outcomes or the outputs of the system, the resources required to make the system functional etc.

1.1 Classifications of System

There are various types of system. To have a good understanding of these systems, these can be categorized in many ways. Some of the categories are open or closed, physical or abstract and natural or man made information systems. Classification of systems can be done in many ways [2,3].

A) Physical or Abstract System

Physical systems are tangible entities that we can feel and touch. These may be static or dynamic in nature. For example, take a computer center. Desks and chairs are the static parts, which assist in the working of the center. Static parts don't change. The dynamic systems are constantly changing. Computer systems are dynamic system. Programs, data, and applications can change according to the user's needs[4].

Abstract systems are conceptual. These are not physical entities. They may be formulas, representation or model of a real system.

B) Open Closed System

Systems interact with their environment to achieve their targets. Things that are not part of the system are environmental elements for the system. Depending upon the interaction with the environment, systems can be divided into two categories, open and closed.

Open systems: Systems that interact with their environment. Practically most of the systems are open systems [2]. An open system has many interfaces with its environment. It can also adapt to changing environmental conditions. It can receive inputs from, and delivers output to the outside of system. An information system is an example of this category.

Closed systems: Systems that don't interact with their environment. Closed systems exist in concept only.

C) Man made information system

The main purpose of information systems is to manage data for a particular organization. Maintaining files, producing information and reports are few functions. An information system produces customized information depending upon the needs of the organization [3]. These are usually formal, informal, and computer based.

Formal Information Systems: It deals with the flow of information from top management to lower management. Information flows in the form of memos, instructions, etc. But feedback can be given from lower authorities to top management.

Informal Information Systems: Informal systems are employee based. These are made to solve the day to day work related problems.

Computer-Based Information Systems: This class of systems depends on the use of computer for managing business applications.

II. Software Life Cycle Models

Nearly three decades ago the operations in an organization used to be limited and so it was possible to maintain them using manual procedures. But with the growing operations of organizations, the need to automate the various activities increased, since for manual procedures it was becoming very difficult and cumbersome. Like maintaining records for a thousand plus employees company on papers is definitely a cumbersome job. So, at that time more and more companies started going for automation [4,5].

Since there were a lot of organizations, which were opting for automation, it was felt that some standard and structural procedure or methodology be introduced in the industry so that the transition from manual to automated system became easy. The concept of system life cycle came into existence then. Life cycle model emphasized on the need to follow some structured approach towards building new or improved system. There were many models suggested. A waterfall model was among the very first models that came into existence [6].

System development begins with the recognition of user needs. Then there is a preliminary investigation stage. It includes evaluation of present system, information gathering, feasibility study, and request approval. Feasibility study includes technical, economic, legal and operational feasibility. In economic feasibility cost-benefit analysis is done. After that, there are detailed design, implementation, testing and maintenance stages [4].

III. Activities Involved in any Life Cycle Model

Following list of activities is the part of any life cycle model. The sequence may be not being exactly same. However, each of these is necessarily covered in the life cycle.

1. Preliminary Investigation
2. Determination of System's requirements
3. Design of System
4. Development of Software
5. System Testing
6. Implementation and Maintenance

A. Preliminary Investigation.

The main aim of preliminary analysis is to identify the problem. First, need for the new or the enhanced system is established. Only after the recognition of need the proposed system is done then further analysis is possible[7].

B. Determination of System's requirements: Analysis phase

After preliminary investigation, analysis phase begins. Analysis is a detailed study of the various operations performed by the system, relationships among the various sub-systems or functional units and finally the relationships outside the system. Major questions under consideration during analysis are:

1. What is being done?
2. How is it being done?
3. Does a problem exist?
4. If a problem exists, how severe is it?
5. How frequently does it occur?
6. What is the underlining cause for the problem?

C. Design of System.

After the system has been analyzed by the analyst, the design stage of system life cycle begins. In design phase, the structure or design for the proposed system is finalized. Structure of files, databases, input, output, processes, and screens (interfaces) are decided. After design is finalized, it is clearly documented in what is called Design Document.

This design document contains various graphical representations of reports, user interaction screens, etc. This design document is referred by developers during development of system.

Correct designing of the system is very crucial. If we have a wrong design we won't be able to a get correct desired system. From various studies it is observed that nearly 50% errors are made during design phase in any software development while 33% errors are pertaining to program logic and only 17% are syntactical errors. Also cost of fixing errors is maximum for design phase. Cost of fixing errors for design phase nearly amounts to 80% while it is only 20% for logic and programming.

D. Development of Software

In this phase, the actual development of the system takes place. That is, design representations are translated into actual programs. Software developers may install (or modify and then install) purchased software or they may write new, custom-designed programs[6].

Programmers are also responsible for documenting the program, providing an explanation of how procedures are coded. Documentation is essential to test the program and carry on maintenance once the application has been installed. It is also helpful to user in knowing the system well.

E. System Testing

After a system has been developed, it is very important to check if it fulfills the customer requirements. For this purpose, testing of the system is done. For testing (Which is one of the part of system development phases) the systems, various test cases are prepared. A test case is a certain made up situation on which system is exposed so as to find the behavior of system in that type of real situation [5,7]. These test cases require data. The data can be also made up artificial data or the real data provided by the user.

There are various types of tests which are used to test the system. These include unit, integration, and acceptance testing.

The smallest unit of software design is module. In unit testing these modules are tested. Since the modules are very small even individual programmer can test them. Once the individual modules are tested, these are integrated to build the complete system. But testing

individual module doesn't guarantee if the system will work properly when these units are integrated.

Acceptance testing ensures that the system meets all the requirements. If it fulfils the needs then the system is accepted by the customer and put into use.

F. Implementation and Maintenance

Implementation of system means putting up system on user's site. Like any system, there is an aging process. Therefore, the system requires periodic maintenance.

Maintenance can be for software or hardware. User priorities, changes in organizational requirements, or environmental factors call for system enhancements. This is very crucial for the system's life.

IV. DIFFERENT LIFE CYCLE MODELS

We have studied the various stages that are involved in the development of systems. There are system development models, which follow these stages. There is sequential traditional model also called waterfall model. Along with this there are many other approaches to system development. They are

- Iterative Prototype Model
- Dynamic Systems Development Model (DSDM).
- Spiral Model, Incremental Model
- Object-Oriented Model.

A. Traditional/Waterfall Software Development Life Cycle Model

Waterfall model is a systematic and sequential approach towards software development. Each stage begins or originates only after the previous stage has finished. There are different levels, corresponding to which each stage starts. Probably due to different leveling of stages, this model is called Waterfall, organized in a linear order. The first stage deals with the system study. The rest of the stages are analysis, coding, testing and maintenance. Next stage can begin only when the previous stage is over. Fig 4.1.1 shows Waterfall Model [8,9]

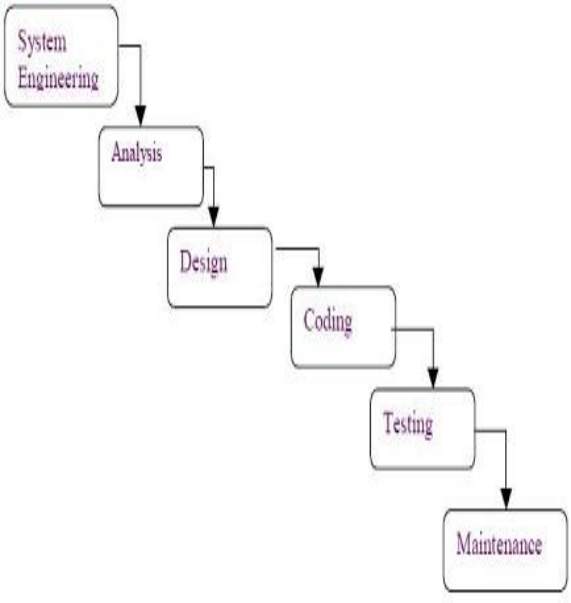

Fig.4.1.1. Waterfall Model

B. Prototyping Software Life Cycle Model

A prototype is a working system that is developed to test ideas about the new system. And prototyping is a process of building a model of the system to be developed.

This approach is used when it is difficult to know all the requirements in the beginning of the project. Such situations arise in the following cases:

* The user has not provided all the requirements.
* No other system like the proposed system was built earlier.

This model can be also used in a situation when the customer wants a quick delivery. May be not the entire system but a part of it. The complete system is based on an iterative model where the customer keeps specifying the requirements according to the model made available and the changes are incorporated to build a better model.

After design is finalized a working prototype or model is built. It is built to show the functionality of the system to the user before the development of the actual product. Once you have a prototype, it is shown to the customer. The customer verifies the prototype. In case there are some suggestions from customers then again that functionality is added to the prototype and again it is evaluated by customer. This cycle gets repeated till the time the customer is fully satisfied with the prototype. Then the actual product is built. The same prototype can be used for the final product or a new product can be built based on the prototype.

In the first customer evaluation stage suppose customer identifies that they require one more function then designer will again go to the design stage and incorporate this function in the prototype. This cycle will be repeated till the customer is fully satisfied with the prototype.

The prototyping model of System Development is thus very different from the traditional Waterfall approach. In this model, the phases of development are not sequential, but they are iterative, where any changes or additions suggested by the end user are incorporated by modifying the prototype model and again giving it to the user[9].

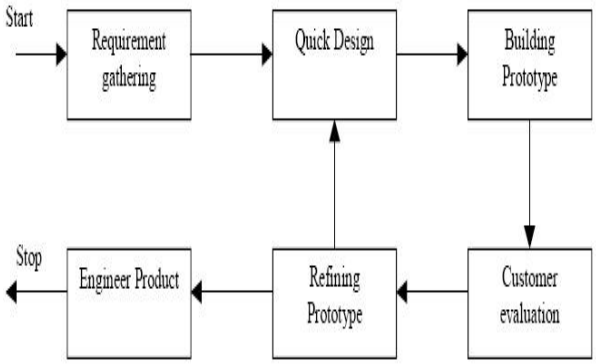

Fig.4.2.1. Prototype Model

C. Object Oriented Methodology.

We live in a world of objects. These objects exist in nature, in man-made entities, in business, and in the products that we use. They can be categorized, described, organized, combined, manipulated and created. Therefore, an object-oriented view has come into picture for creation

of computer software. An object-oriented approach to the development of software was proposed in late 1960s.

Object-Oriented development requires that object-oriented techniques be used during the analysis, and implementation of the system. This methodology asks the analyst to determine what the objects of the system are, how they behave over time or in response to events, and what responsibilities and relationships an object has to other objects. Object-oriented analysis has the analyst look at all the objects in a system, their commonalties, difference, and how the system needs to manipulate the objects.

Object Oriented Process

The Object Oriented Methodology of Building Systems takes the objects as the basis. For this, first the system to be developed is observed and analyzed and the requirements are defined as in any other method of system development. Once this is done, the objects in the required system are identified. For example in case of a Banking System, a customer is an object, a cheque book is an object, and even an account is an object.

In simple terms, Object Modeling is based on identifying the objects in a system and their interrelationships. Once this is done, the coding of the system is done. Object Modeling is somewhat similar to the traditional approach of system designing, in that it also follows a sequential process of system designing but with a different approach[6]. The basic steps of system designing using Object Modeling may be listed as:

1. System Analysis
2. System Design
3. Object Design
4. Implementation

a) System Analysis

As in any other system development model, system analysis is the first phase of development in case of Object Modeling too. In this phase, the developer interacts with the user of the system to find out the user requirements and analyses the system to understand the functioning.

Based on this system study, the analyst prepares a model of the desired system. This model is purely based on what the system is required to do. At this stage the implementation details are not taken care of. Only the model of the system is prepared based on the idea that the system is made up of a set of interacting objects. The important elements of the system are emphasized.

b) System Design

System Design is the next development stage where the overall architecture of the desired system is decided. The system is organized as a set of subsystems interacting with each other. While designing the system as a set of interacting subsystems, the analyst takes care of specifications as observed in system analysis as well as what is required out of the new system by the end user.

As the basic philosophy of Object-Oriented method of system design is to perceive the system as a set of interacting objects, a bigger system may also be seen as a set of interacting smaller subsystems that in turn are composed of a set of interacting objects. While designing the system (Ian Sommerville), the stress lies on the objects comprising the system and not on the processes being carried out in the system as in the case of traditional Waterfall Model where the processes form the important part of the system.

C) Object Design.

In this phase, the details of the system analysis and system design are implemented. The Objects identified in the system design phase are designed. Here the implementation of these objects is decided as the data structures get defined and also the interrelationships between the objects are defined.

d) Implementation.

During this phase, the class objects and the interrelationships of these classes are translated and actually coded using the programming language decided upon. The databases are made and the complete system is given a functional shape [8].

The complete OO methodology revolves around the objects identified in the system. When observed closely, every object exhibits some characteristics and behavior. The objects recognize and respond to certain events. For example, considering a Window on the screen as an object, the size of the window gets changed when resize button of the window is clicked [3].

Here the clicking of the button is an event to which the window responds by changing its state from the old size to the new size. While developing systems based on this approach, the analyst makes use of certain models to analyze and depict these objects.

D. Dynamic System Development Method.

Dynamic System Development Method is another approach to system development, which, as the name suggests, develops the system dynamically. This methodology is independent of tools, in that it can be used with both structured analysis and design approach or object-oriented approach.

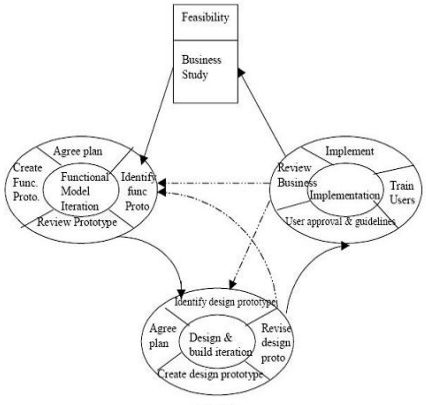

Fig.4.4.1. DSDM Model

The DSDM development process is dynamic as it is a Rapid Application Development method that uses incremental prototyping. This method is particularly useful for the systems to be developed in short time span and where the requirements cannot be frozen at the start of the application building. Whatever requirements are known at

a time, design for them is prepared and design is developed and incorporated into system. In DSDM, analysis, design and development phase can overlap. Like at one time some people will be working on some new requirements while some will be developing something for the system. In DSDM, requirements evolve with time. Fig 4.4.1 shows DSDM Model [1].

V. Comparing Different Life Cycle Models

A. Waterfall model.

Major limitations:

" Real projects rarely follow sequential flow. Iterations are required.

" Requirements may not be clearly defined in the beginning itself.

" Major errors may require going back to Design phase or even Feasibility study phase.

" Customer would have no idea of the working program until the end. Very less interaction with the customer.

" Major blunders may be noticed only after development. " Product delivery takes much time.

Advantages:

" It is the oldest method. It is easily understood.

B. Prototyping model.

Limitations:

" Because of quick design, there can be compromise on quality.

Advantages:

" Fits user work patterns.

" Reduces risk of uncertainty.

" Allows early stoppage of unsuccessful projects.

" User interaction and involvement is more

" Less chances of error during system design.

C. DSDM Model.

Limitations:

" It is a relatively new model. It is not very common. So it is difficult to understand.

Advantages:

" Active user participation throughout the life of the project and iterative nature of development improves quality of the product.

" DSDM ensures rapid deliveries.

" Both of the above factors result in reduced project costs.

D. Object Oriented Method.

Advantages:

" OO Model closely represents the problem domain. Because of this, it is easier to produce and understand designs.

" The objects in the system are immune to requirement changes. Therefore, allows changes more easily.

" OO designs encourage more re-use. New applications can use the existing modules, thereby reduces the development cost and cycle time.

" OO approach is more natural. It provides nice structures for thinking and abstracting and leads to modular design.

VI. Conclusion

Software development is not a precise science. Software development must be accompanied by quality assurance activities. It is typical for developers to spend around 40% of the total project time on testing. For life critical software (e.g. flight control, reactor monitoring), testing can cost 3 to 5 times as much as all other activities combined. The destructive nature of testing requires that the developer discard preconceived notions of the correctness of his/her developed software system. In software system development project, errors can come at any stage during development. The main causes of errors are

1. not obtaining the right requirements,
2. not getting the requirements right, and
3. not translating the requirements in a clear and understandable manner that programmers implement them properly.

There are techniques available for detecting and eliminating these errors that originate in various stages of system development, implementing basic methods would reduce these errors to considerable extent. One should have through knowledge of these methods before development of system so that one can cut down time and money considerably on testing.

With these basic methodologies of software development one can achieve considerable success rate of development.

VII. Limitations

Agile development has not been focused in this paper, where system development using software engineering with agile development would be much better approach than the legacy methodologies.

References

[1] David Alex Lamb. *Software Engineering: Planning for Change*. Prentice-Hall, 1988.

[2] Hethe Henrickson and Scott Hofma,IIS 6.0: The Complete reference , Mc Graw- Hill Osborne media, first edition, 2003.

[3] Ian Sommerville, Software Engineering6th Edition, Person Education Ltd.

[4] Lara Ashmore ,A Web-based Application Design Overview, IT Journal On-Line: Spring ,1997.

[5] Pfleege, Software Engineering Theory and Practice, 2nd Edition, Pearson Education, 2001.

[6] Roger.S.Pressman,Software Engineering- A Practitioners Approach, 7th Edition,McGraw-Hill, 2007.

[7] William R. Stanek, Microsoft IIS 6.0 Administrator's Pocket Consultant , Microsoft Press, 2003

[8] Waman S Jawadekar, Software Engineering Principles and Practice, Tata McGrawHill, 2004.

[9] M.R.V. Chaudron, J.F. Groote, K.M. van Hee, C. Hemerik, L.J.A.M. Somers and T. Verhoeff. "Software Engineering References Frame work" Technical Report CS-Report 04-039, *Computer Science Reports*, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands, 2004.

AUTHOR'S PROFILE

Arati Dandavate
Head of Department,
Department of Computer Engineering
Dhole Patil College of Engineering, Wagholi,
Pune, Maharashtra State, India.
Teaching Experience: 14 Yrs
Email: arati_01@rediffmail.com,
www.dpcoe.com