International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

Selecting Application Oriented Approach For A Project

Adarsh Dhuliya[1]
Arzoo Verma[2]
Gokul Pillai[3]
Er. Ramandeep Sandhu[4]

Abstract

Computers are commercially being used for over the past sixty years. If we examine the way computers have evolved over this period, we can see that in the early day's computers were very slow and lacked sophistication. The computational power and sophistication of computers have increased ever since, while their prices have dropped dramatically. The improvements to their speed and reductions to their cost were brought about by several technological breakthroughs that occurred at regular intervals. The more powerful a computer is, the more sophisticated programs it can run. Therefore, with the every increase in capabilities of computers, software engineers have been called upon to solve large and complex problems, and that too in cost effective ways. All these innovations and experiences have given rise to the discipline of software engineering.

Keywords:SDLC model, iterative waterfall model, prototyping model, evolutionary model, spiral model.

1 B. Tech* Deptt. Of C.S.E, C.G.C Technical Campus, Jhanjeri, Mohali,

2 B. Tech* Deptt. Of C.S.E, C.G.C Technical Campus, Jhanjeri, Mohali,

3 B. Tech* Deptt. Of C.S.E, C.G.C Technical Campus, Jhanjeri, Mohali,

4 Associate. Prof. Deptt. Of C.S.E, C.G.C Technical Campus, Jhanjeri, Mohali

Introduction

The essence of all programming experiences and innovations for writing good quality programs in cost effectives and efficient ways have been systematically organized into a body of knowledge. Software engineering discusses systematic and cost effective techniques to software development. Firstly there was an exploratory approach but later on modern approach came into existence which emphasizes adherence to a well-defined life cycle model. SDLC, an acronym for Software Development Life Cycle, is a well-defined and systematic approach, put into practice for the development of a reliable high quality information system. It's a methodology that is employed by

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

business analysts which describes the activities carried out at each juncture of the development of software. [2] A life cycle model prescribes the different activities that need to be carried out to develop a software product and the sequencing of these activities. We are going to discuss about all such models. All such models which are used to develop a software are also known as software development lifecycle models (SDLC MODELS).

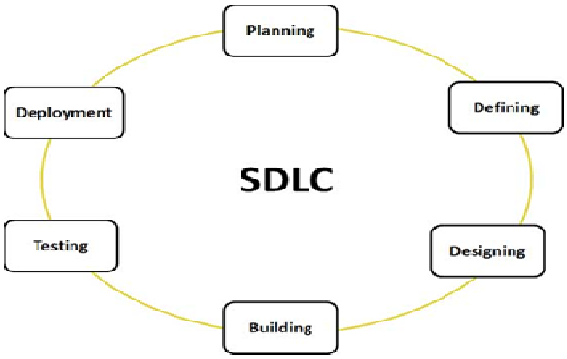

Fig. 1: Stages of SDLC models

Stage 1: Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational, and technical areas.

Stage 2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through .SRS. . Software Requirement Specification document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the product architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

Stage 4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Stage 5: Testing the Product

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However this stage refers to the testing only stage of the product where products defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

I. NEED OF LIFE CYCLE MODELS

The primarily advantage of adhering to a life cycle model is that it encourages development of software in a systematic and disciplined manner. As we have already pointed out, when a program is developed by a single programmer, he has the freedom to decide the exact steps through which he will develop the program. However when a software product is developed by a team, it is necessary to have precise understanding among the team members as to- what to do and when. Also by using a software life cycle model, target can be achieved in a precise manner. Thus use of suitable life cycle model is crucial to the successful completion of a project. The life cycle models are as follows:

I. Iterative Waterfall Model:

Iterative waterfall model was a result of the disadvantage of CLASSICAL WATERFALL MODEL as rework at previous phases is not possible in classical waterfall model and no concept of error correction is given by it. But Iterative waterfall model provides a feedback path from each next phase to each previous phase instead of feasibility.

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

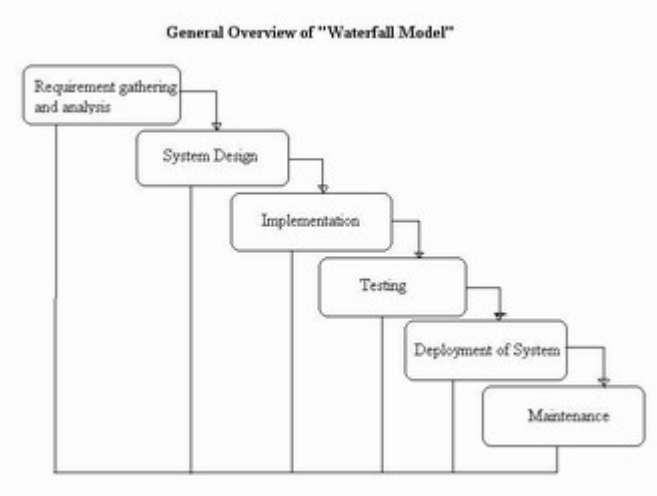

Fig. 2 [4] General Overview of "Waterfall Model"

The phases of iterative waterfall model are as follows [4]-

1. Feasibility study:

It checks two main types of feasibilities- technical feasible and financial feasible. It define problem in abstraction and also mentions various strategies to solve it.

2. Requirement analysis and specification:

The aim of this phase is to understand the actual requirements of the customer and to document them properly. This phase consist of two distinct activities, namely requirement gathering and analysis and requirement specification as follows-

a. Requirement gathering and analysis- the activity consist of first gathering all the information required and then analyzing them. The goal of this phase is to gather all the relevant information regarding the product to be developed from the customer with a view to clearly understand the customer requirements. Once the requirements have been gathered, the analysis activity is taken up. On the other hand an incomplete requirement is one where some parts of the requirements may have been omitted inadvertently.

b. Requirements specification- the customer requirements identified during the requirements gathering and analysis activity are organized into software requirement

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

specification (SRS) document.The three basic contents of this document are:

(a) Functional requirement- which includes all the functions that are linked with the specific project.

(b) Non-functional requirement- they are not actions of projects but kind of performances, constraints relevant to project.

(c) Goals of implementation- it includes general suggestion regarding the development. It is not confirmed by customer at the time of accepting it.

3. Design**:**
The goal of the design phase is to transform the requirements specified in the SRS document into a structure that is suitable for implementation in some programming language. Two distinctly different design approach are being used at present:
(a) TRADITIONAL APPROACH- it is being used by many software development houses. It is based on the data flow oriented design approach.
(b) OBJECT ORIENTED DESIGN APPROACH- it is relatively new technique. In this technique various objects that occur in the problem domain and the solution domain are first identified and the different relationships that exist among these objects are identified.

4. Coding and Unit Testing:

The output of design phase is given to next phase for creating its code. It is also called implementation phase. The whole project is coded into modules, once coding is complete must perform unit testing on each module.

5. Integration and System Testing:

After all modules are tested individually, must integrate them in well manner and according to a plan. It is not possible in one spot for a perfect project .after coupling modules and performing integration testing finally carry system testing on all modules which are coupled.

There are three types of system testing:-

(a) ALPHA TESTING- the testing performed by development team.

(b) BEETA TESTING- the system testing performed by friendly type of customers.

(c) ACCEPTANCE TESTING- the system testing performed by customer itself at the time of product delivery to check whether to accept it or reject it.

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

6. Maintenance:

This phase requires much high efforts then all other phases. To increase the life cycle of a product high maintenance is must. It is of three types:

(a) CORRECTIVE MAINTAINENCE- it involves; correct those errors which were not discovered during previous phases.

(b) PERFECTIVE MAINTAINENCE- it involves; improve the implementation of the system and enhance the function of system.

(c) ADAPTIVE MAINTAINENCE- it includes; maintain your product means it must be able to use in new environment.

II. Prototyping Model:The prototyping model is a system development method (SDM) in which a prototype (an early approximation of a final system or product) is built, tested and then reworked as necessary until an acceptable prototype is finally achieved. [4, 5]. Working as shown in figure 3.

Advantages of Prototyping Model:

1) When prototype is shown to the user, he gets a proper clarity and 'feel' of the functionality of the software and he can suggest changes and modifications.

2) 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.

3) When client is not confident about the developer's capabilities, he asks for a small prototype to be built. Based on this model, he judges capabilities of developer.

4) Sometimes it helps to demonstrate the concept to prospective investors to get funding for project.

5) It reduces risk of failure, as potential risks can be identified early and mitigation steps can be taken.

6) Iteration between development team and client provides a very good and conductive environment during project.

7) 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.

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

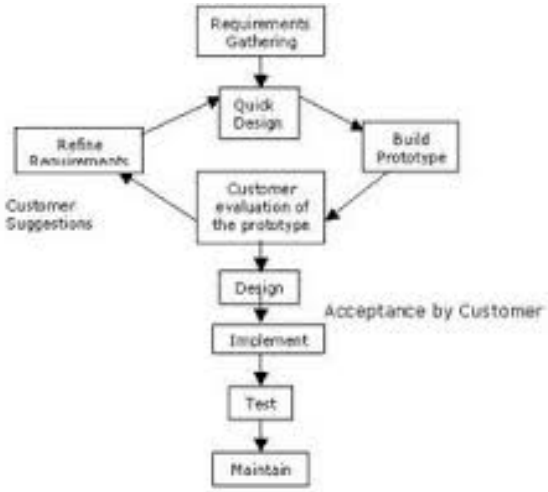

Fig. 3 [5]: General Overview of "Prototype Model"

Disadvantages of Prototyping Model:

1) Prototyping is usually done at the cost of the developer. So it should be done using minimal resources. It can be done using Rapid Application Development (RAD) tools. Please note sometimes the start-up cost of building the development team, focused on making prototype, is high.

2) Once we get proper requirements from client after showing prototype model, it may be of no use. That is why, sometimes we refer to the prototype as "Throw-away" prototype.

3) It is a slow process.

4) Too much involvement of client is not always preferred by the developer.

5) Too many changes can disturb the rhythm of the development team.

III. Evolutionary Model:

This lifecycle model is also referred to as the successive versions model and sometimes as the incremental model. In this life cycle model first a simple working system is built, which subsequently undergoes many functionality and additions until the desired system is realized. The evolutionary software development process is

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

therefore sometimes referred to as design a little, build a little, test a little, and deploy a little model. That is, once the requirements have been specified, the design, build, test and deployment activities [1] are interleaved.

1. Life Cycle Activities:

In the evolutionary model, the software requirements is first broken down into several modules that can be incrementally constructed and delivered. The development team first develops the core modules of the system. The core modules are those who do not need services from the other modules. On the otherhand non- core modules need services from the core modules. This initial product skeleton is refined into increasing levels of capability by adding new functionalities in successive versions. Each evolutionary version may be developed using an iterative waterfall model of development. Each successive version of the product is a fully functioning software capable of performing more work than the previous versions.

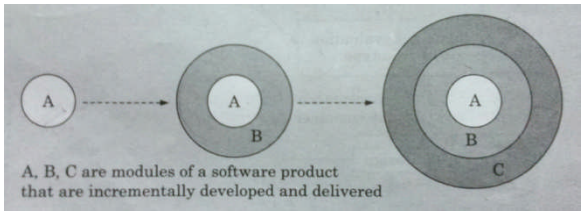

Fig. 4 [1]: General Overview of “Evolutionary Model”

2. Advantages:

This model of development has several advantages. In this model, the user gets a chance to experiment with a partially developed software much before the complete version of the system is released. Thus this model helps to accurately elect user requirements during the delivery of different versions of software.

3. Disadvantages:

The main disadvantage of the successive versions model is that for most practical problems it is difficult to divide the problem into several versions that would be acceptable to the customer and which can be incrementally implemented and delivered.

Thus the evolutionary model is a very natural model to use in object- oriented software development projects.

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

IV. Spiral Model:

The diagrammatic representation of this model appears like a spiral with many loops. The exact no of loops of the spiral is not fixed and can vary from project to project. Each loop of the spiral is called a phase of the software process. It can be seen that this model is much more flexible compared to the other models, since the exact no of phases through which the product is developed is not fixed [6].

Fig. 5 [6]: General Overview of "Spiral Model"

1. Risk Handling In Spiral Model:

A risk is essentially any adverse circumstance that might hamper the successful completion of a software project. As an example the risk involved in accessing data from a remote database can be that the data access rate might be too slow. This risk can be resolved by building a prototype of the data access subsystem and experimenting with the exact access rate. The spiral model provides direct support for coping up with the risks by providing the scope to build a prototype at every phase of software development.

2. Phases of the Spiral Model:

Each phase in this model is split into four sectors or quadrants. In the first quadrant some features of the product are identified based on the severity of the risk and how crucial it is to the overall product development. During the second quadrant, the alternative solutions are evaluated to select the best possible solution. Activities during the third quadrant consist of developing and verifying the next level of the product. At the end of the third iteration, the identified features have been implemented and the

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

next version of the product is available. Activities during the fourth quadrant concerned reviewing the results of the stages traversed so far, progressively more complete versions of the software get built.

3. Pros and Cons of the Spiral Model:

There are a few disadvantages of the spiral model that restrict its use to certain projects only. To the developers of the project, the spiral model usually appears as a complex model to follow, since it is risk driven and is more complicated than the other models. It therefore requires knowledgeable staff also, it is not suitable for development of product as outsourced projects, since the projects risks need to be continually assessed as it is developed.

COMPARISON OF DIFFERENT LIFE CYCLE MODEL:

The classical waterfall model can be considered as the basic model and all other life cycle models as embellishments of this model. However, classical waterfall model cannot be used in practical development projects, since the mechanism supports no mechanism to correct errors that are committed during any of the phases but detected at a later phase.

The iterative waterfall model is probably the most widely used software development model so far. This model is simple to understand and use. However, this model is suitable only for well-understood problems, and is not suitable for very large projects and for projects that suffer from many types of risks.

The prototyping model is suitable for projects for which either the user requirements or the underlying technical aspects are not well understood, however all the risks can be identified before the project starts.

The evolutionary approach is suitable for large problems which can be decomposed into a set of modules for incremental development and delivery. This model is used widely for object-oriented development projects. This model can only be used if incremental delivery of the system is acceptable by the customer.

The spiral model is considered a Meta model and encompasses all other life cycles models. Flexibility and risk handling are inherently built into this model. The spiral model is suitable for development of technically challenging and large software

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

products that are prone to several kinds of risks that are difficult to anticipate at the start of the project.

Viewpoint of the Customer:

Initially, customer confidence is usually high on the development team irrespective of the development model followed. During the lengthy development process, customer confidence normally drops off, as no working product is yet visible. Developers answer customer queries using technical slang, and delays are announced. This give rise to customer resentment. On the other hand evolutionary approach lets the customer experiment with a working product much earlier than the monolithic approaches. Also, from the customer's financial view point, incremental development does not require a large upfront capital outlay. The customers can order the incremental versions as and when he can afford them.

Selecting an Appropriate Life Cycle Model for a Project:

A suitable life cycle model can possibly be selected based on an analysis of issues such as following:

1. Characteristics of the software to be developed: if a software is a simple data processing application, an iterative waterfall model should be sufficient. An evolutionary model is a suitable model for object-oriented development projects.

2. Characteristics of the development team: if the development team is experienced in developing similar products, then even an embedded software can be developed using an iterative waterfall model. If the development team is entirely novice, then even a simple data processing application may require a prototyping model to be adopted.

3. Characteristics of the customer: if the customer is not quite familiar with computers, the requirements are likely change frequently as it would be difficult to form complete, consistent, and unambiguous requirements. Thus, a prototyping model may be necessary to reduce the later change requests from the customers.

Conclusion:

This paper is providing all the necessary details about the selection of a software development life cycle models to develop software. This approach is comparing different models and relating the work of each with thinking of customer. During the development of any type of software product, adherence to a suitable process model has become universal. Adoption of a suitable life cycle model is now accepted as a primary necessity for successful completion of projects.

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

References:

[1] Fundamentals of Software Engineering By Rajib Mall third edition.
[2] A Quick and Simple Introduction to Software Development Life Cycle (SDLC) [Online] Available: http://www.zyxware.com/articles/2037/a-quick-and-simple-introduction-to-software-development-life-cycle-sdlc-why-what-how
[3][Online] available: http://www.tutorialspoint.com/sdlc/images/sdlc_stages.jpg
[4] [Online] Available: http://www.onestoptesting.com/images/waterfall.jpg

[5] [Online] Available: http://rajeevprabhakaran.wordpress.com/2008/11/20/software-life-cycle-models/

[6] [Online] Available: http://upload.wikimedia.org/wikipedia/commons/thumb/3/37/Software_Development_Spiral.svg/2000px-Software_Development_Spiral.svg.png