

Chakra - A new era in Software Lifecycle modeling technique

R.P.Muthu

Assistant Professor,
Department of Computer Science
Indian Institute of Technology, Bombay.

Abstract:

Every old thing has to be modified into a newer version to stay in par with this world. This theory of changing applies to every field and Software development field is no exception. In all fields, especially Software field, time has become an important constraint. We have many models in Software Development lifecycle model. But it's been so long since we have a revised software development lifecycle model, due to a lack of development we many times land in a situation where we are not able to finish our project in time which paves way for the loss of project to the companies. Thus to avoid all these chaos in time and make the process of developing a software a much faster and simpler one, This paper has proposed a new kind of Software development model called chakra. We took two life cycle models- Prototyping and Spiral Model and combined them with some small modifications to land into our model Chakra. Chakra will have all the benefits of both Prototype and Spiral model and will not have their disadvantages. As we move into this paper we can find its usage and understand why this method was proposed.

Introduction:

Software Engineering can be defined as the study and application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of the software. We can easily define it as the application of the engineering to software. Thus Software Engineering can be taken as the systematic approach to develop and maintain a software product from scratch to final outcome in a cost effective and efficient way.

Software Engineering Paradigm is an abstract representation of a process. It is chosen based on nature of software and its application

Software Lifecycle Model:

Software Lifecycle model is a framework that tells the about different phases that are involved in a software development process. The Primary goal of this life cycle model is to develop software in a systematic and disciplined manner. Normally while developing a large size project, the task of creating a project gets divided and is given to different teams. Thus the project is split into different modules and is created by different teams concurrently. Thus it reduces the confusion and the completion time of a project. This is the main reason why go for Software Development Life cycle.

The steps or phases generally involved in a software lifecycle model are

- Customer communication
- Requirement Analysis
- Design
- Implementation
- Testing
- Deployment and Maintenance

There are Different Models that are already available some of them are

- 1) Waterfall Model

- 2) Incremental Model
- 3) Prototyping Model
- 4) Spiral Model
- 5) Win-Win Spiral Model

As *chakra* is a combination of Prototype and Spiral model let us have a glance about these two models.

Prototyping model:

This is one of the important Software Lifecycle Model. If we know the meaning of prototype we can easily define what the basic concept of the Prototype model is. Prototype means a sample which depicts the original. Thus the Prototype model gives the end-user a model to depict how his requirement looks when it is implemented as software. A Prototype model may be the best approach in many situations where the customer is unsure of the efficiency of an algorithm.

Although prototyping can be used as a standalone process model, it is commonly used as a technique that can be implemented within the context of anyone of the process models. In prototyping model the feedback from the user plays a vital role in creating the software as it determines whether the software has to be refined or can be developed from the prototype. If it has to be refined we have to again start from the planning phase. The diagram of the Prototyping model is given below

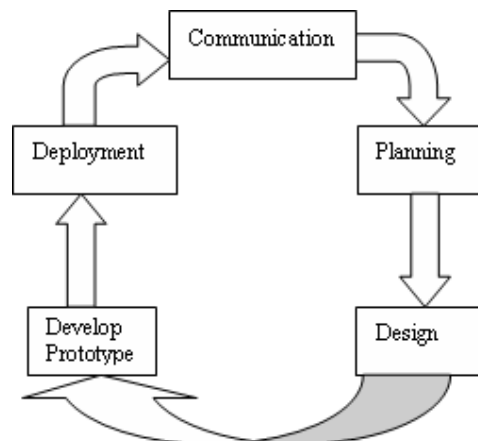


Figure 1: Prototype Model

The main advantage of prototype model is that the customer will get the actual feel of the product and the developer will get something to build immediately. However there are some disadvantages of Prototype model they are i) The Project has to be managed properly or else it'll lead to some infinite iteration. ii) Maintenance can be difficult if not provided with proper documentation.

Spiral Model:

Spiral model is unarguably the best among the software lifecycle model that has been found for developing a product. This model was developed by Boehm in 1988. It is particularly used for large sized products in the industrial side. The basic assumption behind the spiral model is that form of the software development cannot be determined in advance. So the developer has to keep on communicating with the end-user and know about his requirements. A spiral model is divided into different set of frameworks activities defined by the software engineering team. Each framework activities represent one segment of the spiral path as illustrated in Figure.

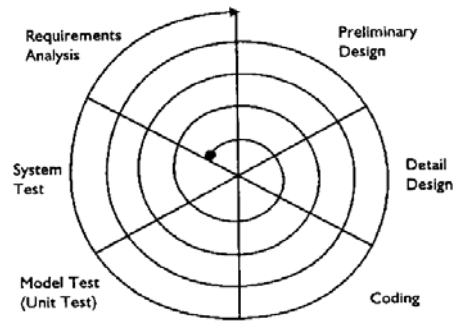


Figure 2: Spiral Model

The advantage of Spiral Model is mainly that the user and customer develop a good level of understanding between each other. The Spiral model uses prototyping as a risk reduction mechanism. Some of the Disadvantages Spiral model is

- i) The Spiral model is used for creating a large product.
- ii) At times becomes complex to use

As we have seen both the models that forms the base for **Chakra**. Let us now have a look at Chakra

Chakra:

Chakra is the name of the Model that we have developed, Chakra is a name taken from Sanskrit language meaning wheel or round. As mentioned earlier our model which is derived from Spiral model and the Prototype model. Since there were some practical problems with the above mentioned 2 models. We created a model that takes all the positives of these two models and leaves the negatives, by doing this we landed into a model which we call as **Chakra**.

Chakra, if needed can also be called as Spiral-Prototype model, or can be considered as a solution for the flaws of the Spiral model. **Chakra** takes the demonstrating concept mixes it with the spiral movement of the spiral model.

The basic definition of Spiral Model said that it is suitable for creating large sized products. So we decided to create a model which looked like spiral model but can be used to create even small sized product.

Prototyping is normally used as a part of the model here also we use it as a part of it.

There may be chances that the user may vary his requirement in between the implementation of the project so the developer has to start again from scratch. To avoid this we went for spiral model.

Chakra also concentrates mainly on the changing desires of the user, so it uses the spiral technique to communicate with the user after each and every spiral or phase of the software development.

Difference between Chakra and other models:

This section tells how chakra is different from its base models.

i) Chakra Vs Spiral Model:

- a) The main difference between our model and spiral model is, Spiral model concentrates only on the large size products but our model can develop product of any size.

b) There were times when spiral model was not able to provide a clear-cut picture of how a software looks to the user. Chakra gives a picture for the user and creates an interface which allows the user to work with.

c) In spiral model Prototyping was used only for Risk Management but here it plays equally important role as spiral model.

ii) Chakra Vs Prototype Model:

a) There are two types of prototyping available they are (a) Evolutionary Prototype (b) Throwaway Prototype. But our model uses only evolutionary prototype.

b) In Prototype model we create a prototype which is shown to the user if the user is satisfied the prototype is thrown away and the software is created. Thus the time taken for creating the prototype becomes a prime factor. But it is not the case in our model.

The Figure Showing the Chakra Model is Given below.

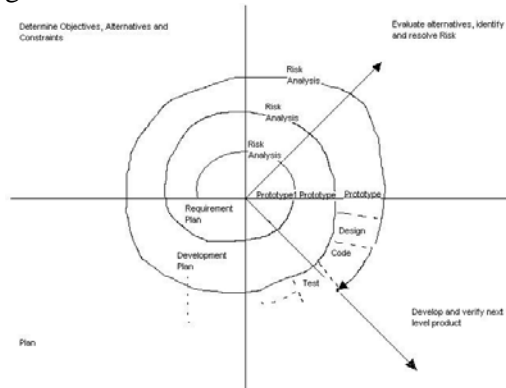


Figure 3: Chakra model

Working Of Chakra:

The Working of Chakra is quite similar to how a Spiral model works. It starts with the innermost Spiral which is the initial phase of the product development. Then before entering into the second spiral the output of the first spiral is again discussed with the User, This process continues till the final spiral of the product development process comes. Thus we can say when the developer and the customer sit to discuss about the next phase they will have the output of the previous phase. So the developer can easily track whether the user is satisfied with the project or not.

The main difference between working of Chakra and Spiral model is the usage of pipelining concept which was not present in the Spiral model.

Next is the usage of Prototyping model in the Chakra. As told earlier the prototyping is used in Spiral model as a Risk Management but here we use it to give the output. After every model the output what the user gets is not the prototype but it is the real product. This further reduces the time involved in development of a product.

Boons and Banes:

The pros of this model are many in short it can be said that it has all the advantages of the Spiral and Prototype model, in addition to that it has met with some of the limitations of these two models, Some of them are a) The time is reduced considerably b) The cost will be reduced. C) Since we are giving the actual product for the user to work with the success rate of the product is very high.

Some of the Disadvantages

- i) The User has to be technically sound to understand the requirements clearly and tell it to the developer.
- ii) It can be a bit time consuming while developing when compared to other models.

The Second disadvantage is only limited to the design and implementation phase but when the overall phase of the product development is considered the time will be less than other model.

Conclusion:

This paper has proposed a new life cycle model for developing a Software. Our main objective before the start of the project was to propose a lifecycle model that will work much faster than the existing model. Although this paper has taken some basic concepts from old models we are sure that the small modifications this paper has made will be able to curb the timing constraint.

Reference:

- [1] “Software Engineering-A Practitioner’s Approach” By Roger S. Pressman, McGraw-Hill International Edition.
- [2] ”An Integrated Approach to Software Engineering” By Pankaj Jalote.
- [3] “Software Engineering” By Ian Somerville.
- [4] Web reference www.mhhe.com