International Journal of Engineering Applied Sciences and Technology, 2021
Vol. 5, Issue 12, ISSN No. 2455-2143, Pages 311-316
Published Online April 2021 in IJEAST (http://www.ijeast.com)

REVIEW OF THE SPIRAL MODEL AND ITS APPLICATIONS

Dhruv Doshi
Department of IT
KJSCE, Mumbai, Maharashtra, India

Labdhi Jain
Department of IT
KJSCE, Mumbai, Maharashtra, India

Kunj Gala
Department of IT
KJSCE, Mumbai, Maharashtra, India

***Abstract*—Software Development Life cycle is the process consisting of many phases used to develop a software and for eliciting the user's requirements. There are a number of models available that are used according to the suitability of the situation. In this paper we discuss the spiral model and its application in various scenarios such as website development, mobile application development and traffic management system.**

***Keywords*—spiral model, applications of spiral model, ATMS, Prototype, MADLC, Traffic, Website development.**

I. INTRODUCTION

Based on experience with several changes to the traditional waterfall model used in major projects the wind model is a software process model that has been in progress for several years. Most of the models that are developed previously are assisted by spiral models as significant cases and notify us that for the given software condition which amalgamation of old models perfectly comply in the situation.

Spiral model was initially proposed by Boehm, which is a very important software development life cycle model. It is an amalgamation of evolutionary, waterfall and prototyping models. It is mostly used in projects which are too large or too expensive or too complicated to handle. The feature which makes the spiral model unique is its capability to handle risks. The spiral model is represented with the help of spiral but the number of loops is subjective, and depends purely upon the project. The larger the wind speed, the higher the project payload and the angular rate determines the success of the continuous phase.

As the Spiral Model complies with the action of development i.e. it contains less risk to development corporations and customers, it's extensively utilized in the code business. Spiral model is typically utilized when assessment of risk is vital and there is a limitation on the budget, projects involving medium to high-risk and in the projects with compound needs.

II. DESIGN OF SPIRAL MODEL

The concept of repetitive development with an organized, supervised part of the waterfall model instilled by the spiral model. This Spiral model is made up of a recurring process and a successive model meaning a waterfall model with a strong focus on analysis of risk. Allows additional product releases or additional filters per processor.

The Spiral model is one of the most vital types of Software Development Life Cycle. This model provides support for Risk Handling. In its representation in the following diagram, it looks like a spiral like structure consisting of several loops. The accurate number of spiral loops is not known and it may change for each project. Each spiral loop is called a Software Development Process Phase. The proper amount of steps needed for product development may change by the admin of the project based on risk determined for the project. Because the person who is managing the project accurately determines the amount of steps, hence the manager of the project plays a vital role in development of the software using the Spiral model.
The pricing of the software product can be determined by the radius of the model. Also, the development of the product so far in the present cycle can be determined by the angular attributes

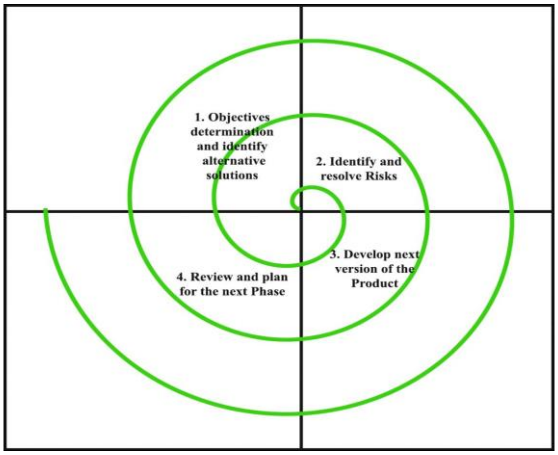
1. Objectives determination and identify alternative solutions
2. Identify and resolve Risks
3. Develop next version of the Product
4. Review and plan for the next Phase

Fig I: Different phases of the Spiral Model

Finding alternative solutions along with identifying objectives: From the clients, information regarding requirements can be gained and objectives are determined, identified and analyzed at the start of each phase. Subsequent alternative solutions are proposed at this stage in this quadrant.

Determine and solve risks: All the solutions that can possibly be obtained are estimated to identify the most optimal solution during the second quadrant. Next, the risks that are derived from the solution are determined. Using the optimal possible strategy those risks are evaluated and resolved. At the end of this quadrant, Prototype is designed with the best solution.

Upgrade the Product to the up-coming version: The attributes that are identified are validated and enhanced by testing in the third quadrant. The succeeding version of the software is made available by the end of this quadrant

Review and plan for next Section: Clients check and analyze the new upgraded version of the software in the fourth quadrant. Finally, the next section is planned.

III. HANDLING OF RISK IN SPIRAL MODEL

Risk is any unfavorable event that could have effects on fruitful conclusion of a project. The most vital attribute of this model is organizing and dealing with these risks that are not known after the development has begun. Such risk solving is easily facilitated by creating a specific type of prototype. The spiral model helps in coping from such risks by giving a build-up model at every step of development software.

The Prototyping Model also has help for management of risk. But for that, the risks must be fully determined before the work of development of the software project can begin. But in the case of real project risk can be identified after the s development work has already been started. So, in this case, we can not use the Prototyping Model. Product features are d analyzed, and the present risks are identified and resolved initially at each phase of the spiral model. Hence, this model is much more workable collated with different models.

IV. APPLICATIONS OF SPIRAL MODEL

A. *Spiral Model in Web Based Applications*

We have been using the waterfall model for many years now for web applications, but the problem with using that model is that it is a very long process, and hence takes a lot of time to be delivered to clients. The alternative to the waterfall model that counters the problems faced earlier is the spiral model.

When we follow the spiral model for designing a web application we will find out that the process is flexible, agile, easily maintained and more suitable. The spiral model segregates the full process into various modules and each module is provided to the customer because of which they can utilize from the beginning stages. The proposed model is also fruitful because if the client requirements change in the middle we can easily implement them as we develop the project in stages.

Like the name suggests all the activities of the project are organized in the shape of a spiral. The budget of the project developed till any given point of time can be determined by the radius of the spiral and the angular attributes helps us to determine the progress made in each cycle. There are four phases in a spiral model and each process passes through all phases, with each increment in the spiral there is a refinement.

The four phases for website development will be:

Requirement Analysis and Objective Identification.- This step involves the identification of objectives and the alternatives that can be used for the accomplishment of those objectives, we also study the risks, uncertainties and constraints involved with each of them.

Design Phase - In this phase we mitigate the risks. We do so by investigating the alternatives, then designing and prototyping them, followed by stimulation and finally reducing the risks associated. If the risks still pertain, then detailed prototyping is done before moving to the next phase.

Building or Coding Phase - In this step we follow a basic waterfall or incremental model whichever is appropriate to code, and develop the website followed by code, acceptance testing and the website is then finally hosted. In the subsequent spirals, when we have a better understanding and clarity of the requirements, each build is given a corresponding version number.

Testing and Risk Analysis - Each module or spiral ends with this testing and risk analysis phase in which the work done by now is reviewed by the clients, whereas the progress, merits, status and risks of the process are evaluated by the team. This phase of the spiral model is common with most other models as well. In this phase the plan of the next phases is also decided.

Advantages of this model in website development are as follows:

- Changes can be made as per the end users feedbacks an opinions
- Flexibility helps in easy changes in layout, design, content, etc.
- It takes in consideration all the possible risks of various alternatives of the requirement.

There are many advantages but there are still some drawbacks which are as follows:

- It can only be used when a website development project can be broken down into multiple modules.
- It is a complex method and needs a skilled set of people to work on it.

The demand for website development is increasing in this digital era, and hence a suitable model is needed for easy development and maintenance of the project, hence the spiral model with some aided modern technology is the perfect solution to do so.

B. *Development of Advanced Traffic Management System using Spiral Model*

Due to the rapid economic development the problems of transportation become serious especially in developing countries. Intelligent Transportation System (ITS). Modern techniques like transduce, computer and information technologies can combine to provide an error-free, well organized and real time management. Intelligent transportation systems strive for the improvement of efficacy, harmlessness of transportation, ecological environment and use of energy by making humans, vehicles, and roads common.

The largest cardinal system of the Intelligent Transportation system is the ATMS i.e Advanced traffic management system, which uses deployment, computer and Information Technology to provide sensitive knowledge to system admins to increase vehicle performance and safety and reduce traffic congestion and pollution.

Because of the swift changes in the process of designing and scheming, intelligent transportation system application, a modern model which is called a spiral model, is used in order to beat the disadvantages of the conventional waterfall model. This model allows us to assist in any suitable integration of simulation guidance, focus on placement, focus on automated transformation, or other approach to system development that proves to be beneficial. This feature can be used for flexible scenarios in the construction process while developing its ITS systems.

Spiral Model for ATMS:

The spiral development model consists of 3 circles:

1. System concept development circle
2. System prototype development circle
3. System engineering development circle

These three circles begin with three step requirements which are the requirements for a conceptual development plan, the initial requirements for the design of the model development circle and the details for the construction of the details of the engineering development circle.

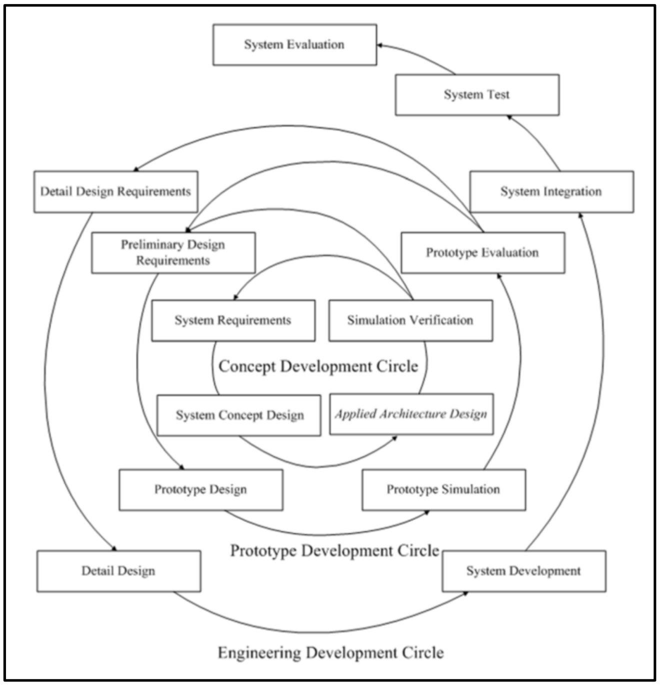

Fig II: Spiral Development Model for an ATMS Based on Prototype

System concept development circle:

This circle has four steps, the first is the requirements of the program, the second is the formation of the concept of the program, the construction of the buildings used for the third time and the validation of the fourth simulation. To use the concept development circle, if the results of the system verification steps are incorrect, the designers can amend the system requirements step. In this concept development circle, the applied architecture step is very important which is illustrated in the next section.

System prototype development circle:

This circle consists of 4 steps which are Prototype design requirements, Prototype design, Prototype simulation and Prototype evaluation. Based on the results of the concept development circle in the previous step, the prototype of the advancement traffic management system is designed in the prototype development circle. The prototype simulation platform can be used to develop the prototype. To test the characteristics of the prototype, this platform included necessary hardware and software.

System engineering development circle:

This circle consists of 5 steps which are Detail Design Requirements, Detail design, System development, System Integration, System test and System Evaluation. From the engineering development circle if the naming requirements are not determined then the next 5 steps will be taken to complete the development of the program.

In the whole system development process, the applied architecture step plays a vital role. Simulation models can be developed based on structural models prior to system development. As a result, until users and developers are satisfied with the program, the functions and functionality of the program can be reviewed in the simulation area. Using this prototype, dynamic performance and real time can also be tested. The cost of the system can be effectively reduced and the quality of system design can be improved with the help of the wind turbine development model developed.

ATMS Applied Architecture:

At the present stage the system is built mainly upon an intelligent camera system. This covers intersection, squares, urban arterial roads and highways. ATMS applications can be improved greatly with digital image processing. If the financial input improves in the future more equipment and functions can be added to the system.

Main Component Modules of ATMS:

Intelligent Camera System: Detection of the vehicles, apprehend the motion graphics in the observable scope, observe the license number plates.

Intelligent Camera Access Network: Connection with 4th Generation mobile technology is possible.

Intelligent Camera Server: Controls intelligent camera system and when the camera nodes access other ATMS module nodes then act as a camera system server.

ATMS Communication System: Setting up communication between cameras on the field is possible by 4G getaways and various other communication can be set up by networks of traffic communication.

ATMS Management Center: The ATMS Management center consists of several sections such as specialized referring systems, message, databases and web servers. The management center plays a principal role in ATMS.

ATMS Payment Gateways: Offline and real-time payments can be processed and communication with the bank can be set up for fee settlement.

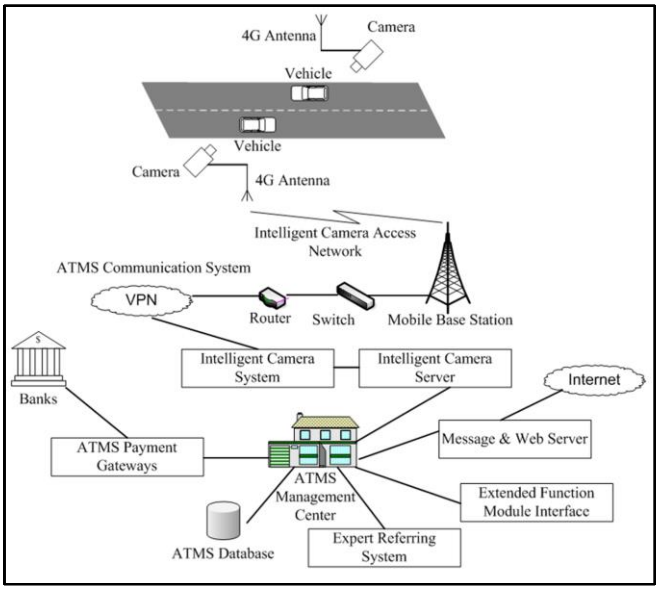

Fig III: Overall Applied Architecture of ATMS

With the event within the streams of technological imagination process and different wireless communication systems, the preparation of such a system will be administered in a low-priced and productive approach.

C. *Use of Spiral Model in Mobile Application Development Life Cycle:*

Use of Spiral Model in Mobile Application Development Life Cycle

Due to the huge development in the technical market, the use and demand of mobile phones has increased. This resulted in software firms shifting from the customary way of using computers to the new mobile technology. Since web applications are being replaced by mobile applications, it results in the augmented demand of these mobile applications. Mobiles are becoming a necessity for humans and a vital part of their lives. Compared to desktop and web applications, the development of mobile applications is a bit more complex. In mobile application development we need to take under deliberation aspects like cross-platform development, the way of interaction with users and memory usage (since low memory space is a common issue in mobiles).

Mobile Application Development Life Cycle (Khandelwal, Tyagi 2015) is a term coined first in 2015. Mobile Application Development Life Cycle (MADLC) is a combination of traditional Prototype Model and Spiral Model. We will be mainly focusing on the implementation of spiral models in MADLC.

The spiral model, first put forward by Boehm, is a transformative software system. Using this spiral model, the software is developed in a sequence of growing releases. The maturity model can be tested as a risk-driven generator for software projects.

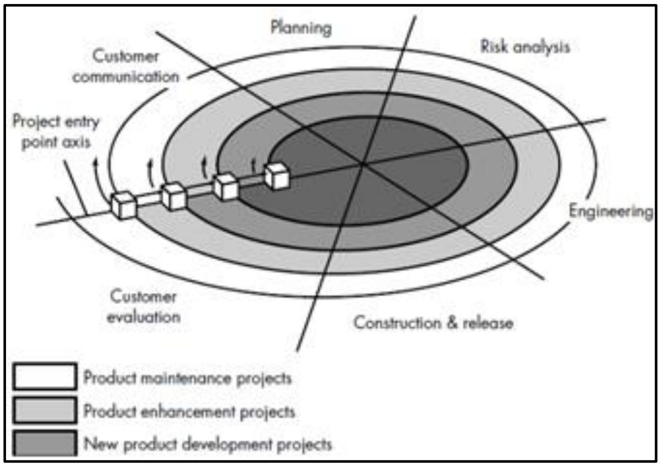

Fig. IV: Shows a spiral model which consist of 6 task regions

1. Customer Communication: this task is required so that an effective communication is established between customer and developer.

2. Planning: this task is required so that resources, timelines, and various project related information can be defined.

3. Risk Analysis: this task is required so that both management and technical risks can be assessed.

4. Engineering: this task is required so that one or more representations of the application can be made.

5. Construction and release: this task is required to install, construct and provide user support.

For the best and optimal results, MADLC combines both Spiral model and Prototype model and results in the mentioned parts of SDLC:

1. Customer Communication: in this phase, requirements are gathered by communicating with the customer. The development team gets to know new ideas from the customer and evaluates the same.

2. Planning: in this phase, the development team strategizes and plans how to implement the requirements and customer's ideas. This planning must be done keeping in mind the available time for this project.

3. Construction of prototype: in this phase, a rough model is created so that the customer gets a visualization of how the final product will be like. Customer feels relieved seeing that his requirements are being implemented in the prototype.

4. Evaluation of the prototype: Customer will evaluate the prototype and may even refine the requirements. The prototype might go through iterations of tuning until the prototype is confirmed by the customer.

5. Engineering: in this phase, one or more representations of the applications are built.

6. Risk analysis: in this phase, the tasks required to evaluate both management and technical risks are included in risk analysis.

7. Construction: in this phase, the initial prototype is coded and last assessment is done by the team.

8. Testing: in this phase, using simulators, the application built is teste at various levels and step by step. Testing should also be done on multiple devices. This is one of the most important phases in MADLC.

9. Release: in this final phase, the application is tested as well as evaluated by the customer and then it is ready for deployment.

Results after implementing the model:

MADLC development model helped the mobile developer to strategize and implement an application involving numerous functionalities in an efficient way and in a timely fashion. This combination of Spiral model and Prototype model will assist mobile application developers in developing applications for mobile phones in a simpler and supple way. This life cycle also focuses on different aspects of mobile applications such as integrated functionality, physical interface, display indicators, memory efficiency, platform development and continuous performance.

V. CONCLUSION

The spiral model has become one among the most vital types of SDLC. This is because it gives a base for management of risk. It can also be inferred that the model is flexible since the exact number of loops in a spiral model are unknown and can be changed according to different projects. The project manager forms an integral part of development using the spiral model since he has to determine the number of phases.

Merits	Demerits
Increases user involvement by providing the user with some end product after the end of every phase.	It is a complex method that requires very skilled staff.
Useful for complex, large and high budget projects	Since spiral models are expensive, it is not suitable for small scale projects.

Changes can be made in the website or mobile application based on the end user's feedback or reviews.	It can only be used in web and application development projects which can be broken into modules.

Table I. Merits and demerits of spiral model

It may not be easy to determine when a spiral model is ready to use, but it is most effective for high-risk projects. Clients can be helped by the prototype solution as it helps to recognize the end product and its merits. So when the customers are not sure about their needs asd requirements they can opt for a spiral model.

VI. REFERENCES

[1] A. Khandelwal, G. Tyagi, (2015) "Review Paper on Suitability of Traditional Prototype Model and Spiral Model used for Mobile Application Development Life Cycle", International Journal of Engineering Research & Technology, 2015, Volume 3, Issue 31

[2] J. Xie, Ma Y., Yuan L., Liu Y. (2015) "A Spiral Development Model for an Advanced Traffic Management System (ATMS) Architecture Based on Prototype", International Journal of Science Technology and Society, January 2015, Vol. 3, No. 6,pp.304-308., DOI:10.11648/j.ijsts.20150306.15

[3] P. Dholakia ,D. Mankad (2013)"The Comparative Research on Various Software Development Process Model" International Journal of Scientific and Research Publications, Volume 3, Issue 3, March 2013 Edition [ISSN 2250-3153] pp. 1-4

[4] Dr. S. T. Bhosale,VPIMSR, Sangli, Shivaji University, Kolhapur (2014) "Spiral Model: Applications in Web based Applications" IPASJ International Journal of Computer Science (IIJCS) Volume 2, Issue 6, June 2014, pp. 1-4

[5] W. Boehm, TRW Defense Systems Group, "A Spiral Model of Software Development and Enhancement," IEEE 2018, DOI:10.1109/2.59

[6] B. Boehm and P. Bose, "A Collaborative Spiral Software Process Model Based on Theory W", IEEE 1994, DOI: 10.1109/SPCON.1994.344423

[7] Luma-Osmani S., Arifi G., Idrizi F., 2014, "Choosing the Most Suitable Model for Developing a Software", Sixth International Conference on Computational Intelligence, Communication Systems and Networks. DOI: 10.1109/CICSyN.2014.30

[8] Tanwar Divya, 2016, "SOFTWARE DEVELOPMENT MODELS AND THEIR REAL ASPECTS", IJARSE Vol. 5, Issue No. 03 (Pg. 412-421)

[9] Henke H.A., 1994, "Spiraling Up and Down the Spiral Development Staircase", IPCC. DOI: 10.1109/IPCC.1994.347490

[10] YAMAMICHI N., OZEKI T., YOKOCHI K., TANAKA T., 1996, "The Evaluation of New Software Developing Process based on a Spiral Modeling", IEEE. DOI: 10.1109/GLOCOM.1996.591987

[11] Haraty R. and Hu G(2018), "Software process models: A review and analysis", International Journal of Engineering & Technology, DOI: 10.14419/ijet.v7i2.29.13206.

[12] Ratnmala R. Raval , Haresh Rathod, 2013 "Comparative Study of Various Process Model in Software Development" DOI: 10.5120/14263-2363