An Additional Sub Phase for Development of Software Process Model

Pawan Kumar Garg
M.Tech, Jagan Nath University
Computer science
Jaipur, India
E-mail: g.pawangarg@gmail.com

Abstract: The software development models follow fixed or iterative design and development approach. There is no scope for dynamic testing in software development process. To make the software development effective and reliable, a new approach is required. We present the new software process model which will address all the concerns and consider each phase of software development as software development process and provide an effective model for software development phases. . Main motive of this technique is to monitor security requirements and identify security threats at each phase. For developing large sized system, still there are some limitations associated with it. It doesn't have the phase where a developer or organization can impose some security to the code. So the main concern of our technique is security and trademark.

Keywords: Software Development, Process, Security, Trademark.

I. INTRODUCTION

Software development is a highly complex field with countless variables impacting the system. All software systems are imperfect because they cannot be built with mathematical or physical certainty. Software development, however, has no laws or clear certainties on which to build. As a result, software's almost always flawed or sub-optimized.

The importance of software process model for development of software product is well known, which include various steps that guide the team with common goals and strategies. Several software life cycle models or process models have appeared till now. All these models share certain characteristics. They identify stakeholder goal, specify key activities to be followed according to a certain sequence, work within time constraints and are based on what has been learned from past experiences. The evolution of software process models has played a significant role in how models have diversified over time.

The causes of the software crisis were linked to the overall complexity of the software process and the relative immaturity of software engineering as a profession. The overall success in the development of software is still not achieved because each software development process or model consider only one or few concerns and specify a phase wise abstraction for the development, but no definite approach or model is specified for the phases of software process model. In current software engineering practices, ever changing requirements during the development process for large software development is still not managed by software process models. The solution space analysis concept of software engineering is very effective, but this concept is not completely integrated to software development yet. Alternative management, a technique which is used in mature engineering disciplines is not explicitly followed in software engineering discipline.

To make the software development effective and reliable, a new approach is required. Improvement model is based on component driven development approach, which is different from component base software development. In Improvement model, each component implements a problem solving model. It includes the explicit processes for technically analyzing the problem, solution space analysis, alternative management, dynamic design specification and development and scope for dynamic testing.

II RELATED WORK

A software development process is a structure imposed on the development of a software product. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process [3]. After studying some of the most used software development process models like Waterfall model, V-process model, Prototyping model, Spiral model and RAD model, we can say that there is no concern of security and trademark. As there are a lot many models, which are being used like Code and fix model. To decide which model is more appropriate and suitable with respect to different software metrics, development time, complexity, implementation challenges, return-on-investment with minimal initial capital expenses, development cost etc. The most important aspect of a software development is its development, as it undergoes a number of development stages (software development life cycle) to reach to its final shape. The development steps, which need to be followed for developing software, project are- Project Planning, feasibility study, Requirement analysis, design etc. However the most important aspect of software development is the system design [4].

The comparison of different process models on the basis of certain factors which may influence the selection of lifecycle models. Each of the models has different response to these features. By this comparison table each process model can categorized on the basis of their features. By seeing this table the developer can easily justify the prones and cons of each software process model [5].

Model/ Feature	Waterfall Model	Prototype Model	Spiral Model	Iterative Model	Agile Model
Requirement Specification	Beginning	Frequently Changed	Beginning	Beginning	Frequently Changed
Understanding Requirements	Well Understood	Not Well Understood	Not Well Understood	Well Understood	Well Understood
Cost	Low	High	Intermediate	Low	Very High
Guarantee of Success	Low	Good	High	High	Very High
Resource Control	Yes	No	Yes	Yes	No
Cost Control	Yes	No	Yes	No	Yes
Simplicity	Simple	Simple	Intermediate	Intermediate	Complex
Risk Involvement	High	Don't say	Low	Don't say	Don't say
Expertise Required	High	Medium	High	High	Very High
Changes Incorporated	Difficult	Easy	Easy	Easy	Difficult
Risk Analysis	Only at beginning	No Risk Analysis	Yes	No Risk Analysis	Yes
User Involvement	Only at beginning	High	High	Intermediate	High
Overlapping Phases	No	Yes	Yes	No	Yes
Flexibility	Rigid	High Flexible	Flexible	Less Flexible	High Flexible

Table 1: Comparison among Software Process Models [5].

III Proposed Technique

The proposed method has the phase where a developer or organization can impose some security to the code. There can be two types of security concerns:

1. The code that is being developed can be theft during the development process and
2. The product can be copied in terms of copyright violation etc.

So, while developing a system we need to look at the concern of security as well as trademark of that particular model or system.

Phases added in proposed Technique are as follows:

1. A sub phase named **security check task** can be added along with the phase of coding and testing (construction) that can be looked like the following:

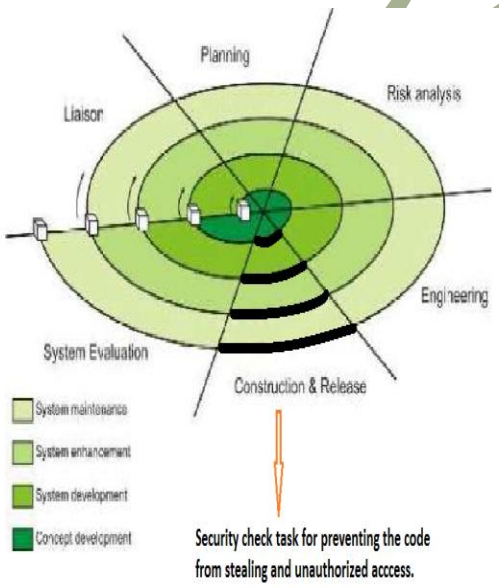

Figure 1: The dark black lines are showing the security check task along with the coding, testing and releasing processes. This security check involves the strategies to make the code secured from stealing and unauthorized access.

2. Suppose some organization gets an order to develop a large scaled system and after doing the enough amount of feasibility study, if the organization finds it feasible, the organization starts the development. But before starting the development the organization can think for converting the project into product i.e. they will install the same project at different organizations with slight difference according to the particular needs of the customer. Now, if the organization really thinks that the making of the project will be a big explosion in terms of success and money, they can really go for getting the project (ultimately product) trademarked and registered. This can be shown like the following figure:

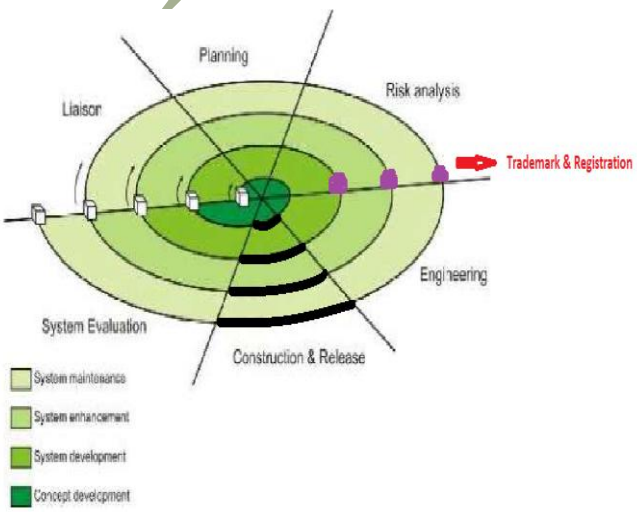

Figure 2: Strategy for getting the project converted into product. Have the product trademarked or registered before releasing it in the market.

IV Result

By the proposed method the concern of security and trademark has been added to software process model. By

applying these techniques to software process model all the testing techniques can be ordered and their relationship can be modeled. Furthermore, this security model can be synchronized with the software engineering model and resulting model will be secure software engineering model for software development. However, their use has not been widely adopted due to the mathematical nature of the languages.

V CONCLUSION

After completing this research, it is concluded that:

1. There are many existing models for developing systems for different sizes of projects and requirements.
2. These models were established between 1970 and 1999.
3. Waterfall model and spiral model are used commonly in developing systems.
4. Each model has advantages and disadvantages for the development of systems , so each model tries to eliminate the disadvantages of the previous model.

Finally, some topics can be suggested for future works:

1. Suggesting a model (Improvement Model) to simulate advantages that are found in different models to software process management.
2. Making a comparison between the suggested model and the previous software processes management models.
3. Applying the suggested model to many projects to ensure of its suitability and documentation to explain its mechanical work.

Acknowledgement

I would like to thanks all my faculty of Jagan Nath University Jaipur for their encouragement.

REFERENCES

[1] Boehm B, "A Spiral Model of Software Development and Enhancement", ACM SIGSOFT Software Engineering Notes", "ACM", 11(4):14-24, August 1986.

[2] Ian Sommerville, "Software Engineering", Addison Wesley, 7th edition, 2004.

[3] CTG. MFA – 003, "A Survey of System Development Process Models", Models for Action Project: Developing Practical Approaches to Electronic Records Management And Preservation, Centre for Technology in Government University at Albany / Suny, 1998.

[4] Steve Easterbrook, "Software Lifecycles", University of Toronto Department of Computer Science, 2001.

[5] Nabil Mohammed Ali Munassar1 and A. Govardhan, "A Comparison between Five Models Of Software Engineering", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 5, September 2010.