



PROPOSED HYBRID SOFTWARE MODEL IN THE SOFTWARE DEVELOPMENT AURA

Dr. Jayant Shekhar
Director

Subharti Institute of Technology and Engineering
Meerut, India

Nitika Singh Yadav
M.Tech (CSE)

Subharti Institute of Technology and Engineering
Meerut, India

Abstract - The orientation of enlargement scientific high-tech convolution of the various systems paired with the need of repetition and foreseen process methodologies have driven system developers to establish new system development models. The essence of this paper analyzes some methodologies that could have result in successful completion of proposed model. This paper explores proposed model peculiar advantages, disadvantages. This paper also explains domestication of the proposed model and the common elements in the process. Finally, this paper suggests a new compound software development model which meets the strength of present scenario. The recommended method can be utilized in the software industry, particularly in the business sectors that deals with large scale projects. The main objective behind this research is to design development model that could meets the needs of different systems and eliminates the defects presented in the previous development models. The present research introduces a model “hybrid model” which combines the features of the five development model i.e waterfall model, iterative model, prototype model, spiral model, agile model. The introduced model in this research has the advantages and some features of the previous models with some adjustments and because of this it avoids and overcomes many software problems that exist in the previous models. Thus, the new proposed model is a concatenation of various models, which is relevant to most software programs and systems.

KEYWORDS – Proposed hybrid model, software management process, software development life cycle, software development processes

I. INTRODUCTION

As the name implies “hybrid” which already means that hybrid development model is a combination of other software development life cycle models. Typically, the way of working of this model is define by development team which will adopt the best features of heterogeneous development method. In order to design a hybrid project we can use both viewspreliminary placing and planning board which is available at the same instant of time by using the same data. For hybrid projects we can execute fundamental requirements by the classical way using a hierarchy of features , enterprise requirement and system essentials.. There are many different compound approach that attempt to apply feedback mechanisms to the acceptable model so that scientific and practical imperfection in the original design could bring to light during development that can be more quickly consolidated .So, enters the hybrid SDLC that considers advent way for all or fractions of the project. The hybrid SDLC is provided as a layout to help project managers and business analysts in development of the own hybrid SDLC by the way of using enterprise process and decision model. Project manager and business analysts collaborate by selecting the best way for the system development SDLC on a project. Model will control the entire process based upon superiority and importance and this waythis model will suit the project size and type which will match the organizations environment for development. It is easy to adopt due to adjustability of proportion. Their development will be monotonous and quick as we follow only applicable process cycle. Rather, than adopting a pure SDLC approach generally we go for hybrid model which containsessentials of those models together.



II. PROPOSED HYBRID MODEL

Here is the incentive to design the introduced model which imitate the advantages of the previous different models found in software process projects is to be confirm of its capability and capacity to show how this model works.. The acute importance of this engineering and their relationship in order to develop various software. This model is a collective association of these models:-

1. Waterfall model
2. Iterative model
3. Prototype model.
4. Spiral model
5. Agile model

III. VARIOUS PHASES OF HYBRID MODEL

1. **PLANNING**-Planning usually refers to how the resources are being allocated in order to live with a fixed economic budget. Time span is also given a second thought. Planning is the art of organizing the necessary activities in time, space and across operatives in order to optimize production, long term goals, customer's satisfaction.
2. **REQUIREMENT**-Stabilize the components for building the systems, including the hardware requirements, software tools. It also, involves the prospect for software utility and identifies which system fundamentals usually affects the software..
3. **DESIGN**-Determines the software structure of a system to meet the specific preliminary. The design usually defines the major components and the interaction amongst these components but it does not define the structure of each component. Rather it also defines the architectural design.
4. **INTEGRATION**- It is define as a part of bringing all the pieces together into a special testing environment, than checks for errors, bugs and interoperability. Integration combines different modules in one system rather than creating any negative effect on the rest of the system components.
5. **DEPOSITION**- The declining of software is due to performance front which occurs as the time span. It may go completely outmoded or may need extreme progress. Therefore the exacting helps to eliminate a major portion of the system that may arises .This phase also include archiving of data and required software components , closing

down the system, planning deposition activity and abort system at convenient end of system time.

6. **DEPLOYMENT**:- It means sending the system after completing to customer for using and working on and showing the problems based on its use for the first time
7. **TESTING**- It helps to Determines whether the software is to ready meet the specified requirements and finds any error presents in the code. Execution of software is performed in this phase in order to find the defects. Exercise new code in combination with code that already has been coordinated.
8. **BUSINESS MODELLING**- A business process is a collection of related structured activities or tasks that produce a specific service or product (serve a particular goal) a particular customer. This process can be decomposed into several sub-processes , which have their own attributes, but also contribute to achieving the goal of a super-process
9. **RISK ANALYSIS**:- It includes all the expected risk involves and suggests all the necessary activities to reduce such risks.
10. **CONFIGURATION**-Configuration management is the task of tracking and controlling changes in the software. If something goes wrong configuration can determine what was changed and who change it. It is generally understood to cover changes typically made by a system administrator.
11. **IMPLEMENTATION**-It involves source code, database, user documentation, testing. In this the real code is being written.

The tractability of this hybrid model is that you can take any phase to give a start which can be applied to small, medium, and large projects.

The following figure shows the proposed model:-

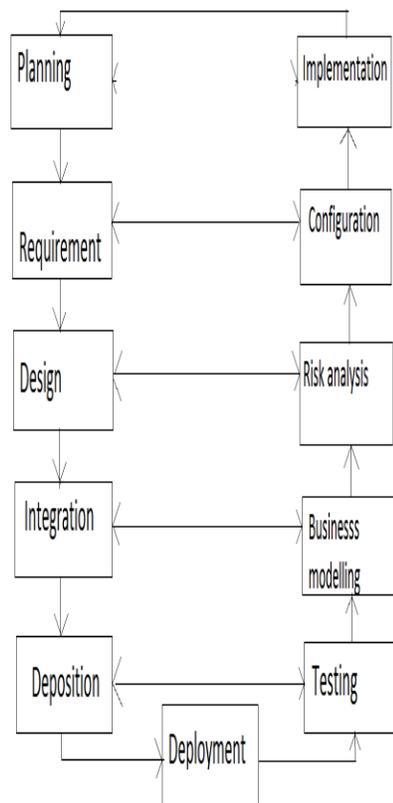


Fig:-1

IV. ADVANTAGES

1. Brighten innovation and compliance design.
2. Provides better Environment to resolve imprecise objectives.
3. May generate particularization for the production applications.
4. Clients are constraint to take active part in the fundamental definition process.
5. Explicitly interface operative cost and development times.
6. Immediate user evaluation with good conclusions.
7. Produces business value and product marketing in the development life cycle.
8. Better use of resources.
9. Deliver partial versions to smooth the introduction of the new product in the clients organization.
10. High amount of risk analysis.

V. COMPARISON TABLE OF PROPOSED MODEL WITH OTHER SOFTWARE DEVELOPMENT MODELS

PROPOSED MODEL	OTHER SOFTWARE MODELS
Frequent adaptation to small, medium and large projects.	Has not been widely used for complex projects.
Unrestricted applicability	Limited applicability.
Developers have to be competent in risk analysis and risk resolution.	Only applicable in spiral model.
Focus on planning phase and risk management.	Risk analysis is only in spiral model.
Identifies end point for each phase.	Not yet proven beyond all doubt.
Disciplined approach.	Delivered product may not meet client's needs.
Best suited for Reusing series of similar product and future products.	Cannot be easily reused.

VI. CONCLUSION

On completion of this research paper, it has been concluded that proposed model has advantages over the other already existing SDLC models of systems so each model provides efforts to eliminate the disadvantages of the previous existing models. SDLC steps are useful to create the proposed software that meets a business need. Idea for completing this research has been borrowed from previous models. This model has the potential to provide a straight forward structured approach in the software development. This paper gives a comparison analysis between the proposed model and other software development models. The Hybrid model is dependent on the five development models: Waterfall, Iterative, Prototype, Spiral, Agile model.

VII. REFERENCES

1. Barry Boehm edited by Wilfred J. Hansen, "Spiral Development: Experience, Principles, and Refinements", 2000.



2. Ian Sommerville, "Software Engineering", Addison Wesley, 7th edition, 2004.
3. Steve Easterbrook, "Software Lifecycles", University of Toronto Department of Computer Science, 2001.
4. Research Topics in Software Engineering Unaided Curricular Paradigms as Compute, "Paradigms of Computation (UCPC)Minho, FEUP July 23, 2009
5. Roger Pressman, Software Engineering: A Practitioner's Approach, Sixth Edition, McGraw-Hill Publication.
6. A. M. Davis, H. Bersoff, E. R. Comer(1988), "A Strategy for Comparing Alternative Software Development Life Cycle Models", journal IEEE Transactions on Software Engineering ,Vol. 14, Issue 10,1988.
7. B. Boehm(2002), "Get Ready for Agile Methods, with Care," IEEE Computer, vol. 35, no. 1, 2002.
8. Klopper, R., Gruner, S., &Kourie, D. (2007), "Assessment of a framework to compare software development methodologies" Proceedings of the 2007 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries, 56-65. doi: 10.1145/1292491.1292498.
9. Roger Pressman, titled "Software Engineering - a practitioner's approach".
10. SanjanaTaya, Shaveta Gupta, "Comparative Analysis of Software Development Life Cycle Models".
11. Karlm, "Software Lifecycle Models", KTH 2006.
12. Rlewallen, "Software Development Life Cycle Model" , 2005
13. Brooks, F. P. (1995). The mythical man-month. Reading, MA: Addison-Wesley Gall, D. M., Gall.
14. P. J., & Borg, R. W. (2003). Educational research: An introduction. Boston, MA: Allyn and Bacon
15. Glesne, C. (2006). Becoming qualitative researchers: An introduction. Boston, MA: Allyn and Bacon.
16. Grunbacher, P., Hailing, M., Biffi, S., Kitapci, H., & Boehm, B. (2004). Integrating collaborative processes and quality assurance techniques: Experiences from requirements negotiation. Journal of Management Information Systems, 20(4), 9-29.
17. Jones, C. (1997). Applied software measurements: Assuring productivity and quality. McGraw Hill.
18. Parrish, A., Smith, R., Hale, D., & Hale, J. (2004). A field study of developer pairs: Productivity impacts and implications. IEEE Software, 21(5), 76-79.
19. Summerville, I. (2004). Software Engineering. Boston: Addison-Wesley.
20. Takeuchi, H., &Nonaka, I. (1986, January-February). The new new product development game. Harvard Business Review, p137-146.
21. Watson, R. T., Kelly, G., Galliers, D., &Brancheau, C. (1997). Key issues in information systems management: An international perspective. Journal of Management Information Systems, 13(4), 91-115