# Hybrid Software Development Approach for Low Risk Applications

Arun Dabas
Lecturer
*Kasturba Polytechnic for Women, Pitam Pura, Delhi*
dabas99@yahoo.com

***Abstract***— **T**he software development process is a very challenging and important factor in the development of a meaningful and successful software production. The appropriate model and approach makes the software development process smooth, effective and efficient. Application of inappropriate software development model makes the software development process complicated and costly. In many applications the requirements are not known or clear to the customer and in that situation the software development process becomes even more complicated. Particularly in these types of applications a single model may not be suitable. The suitability of any single model becomes problematic where the previous experience of the developer is low and the development period is moderate. In such situations and to make the software development process more practical and successful different approach may be applied. Instead of applying a single conventional model a combination of multiple approaches may be used. The applications where the complexity and risk factor is low such models may save efforts and cost effective.

***Key Words: Software Development Life Cycle, Water Fall Model, Software Engineering, Prototype Model***

## 1.Introduction

Software development is a crucial development process. The success and failure of a software system mainly depend upon the approach adopted to develop a particular system. Prior to 1960, no specific software models were used and because of that many software projects were either failed, could not be commissioned or left unfinished in the middle of the development phase itself. The main reason of failure of these projects were unexpected increased cost, not fulfilling the requirements, absence of process, poor quality of developed product, hardware-software compatibility problems and many more. Since many software projects were not taking place successfully, software researchers observed that a formal mechanism should be in place, which may specify the standard process to be adopted for development of a software right from its conceptualization, resulting software development models were evolved.

Software Development Life Cycle (SDLC) is a mechanism which involves standard set of steps to be adopted for the development of particular software. These steps may be followed right from conceptualization to the retirement of a particular system. There are various models used in SDLC depending upon the requirement, size, nature, complexity and other main factors of the system. Some of the popular models are known as Water Fall Model, V-Model, Prototype Model, Incremental Model, Spiral Model, Rapid application Development model etc. The application of each of these models depends upon the time of development, application, previous experience, cost, risk involved in the project such as chances of failure of the project.

Waterfall model is considered to be the simplest and among the least complicated models for implementation. The limitation in the waterfall model is that the requirement should be very well understood before the start of the project. Many times it has been observed that during medium size low risk projects, customers are not very well aware about the requirements in their project and in that situation it becomes very difficult to understand the actual requirements of the customer. However in such situations, if the size of the project is small and requires less development time, then prototype model is considered. But it has its own boundaries and limitations. Prototype model expects extreme user involvement during the project development process right from the day one, which is not possible in most of the cases. Prototype model heavily depends upon the prototypes in every step of the development process which becomes a point of confusion for the customer and sometimes customer misunderstood the difference between prototype and actual software. Besides it, this model is preferred where development time is less than 120 days in complete development process, experienced and highly skilled manpower with previous experience of similar projects is preferred. Another model where prototype may be used is spiral model. This model is suitable for high risk large to very projects where the chances of failure of the project are high and the cost involved in the project is also high. High risk turnkey software projects are suitable for spiral model. There are many other factors involved in different software development models such as use of technology etc. and depending upon the type and size of the software project, an appropriate and most suitable model becomes the most crucial and challenging task. I suggest in this paper a hybrid approach method, which may be suitable for medium size low risk applications. This model may be cost effective and may be used in medium size low risk applications where mixed skilled manpower may be effectively utilized. Moreover overall dependency over prototyping may be reduced upto a reasonable level.

The next section defines motivation behind my work, and section 3 defines my model in detail, section 4 specifies proposed technology suitable for this type of model  and section 5 provides conclusion and future work.

## 2. Motivation behind This Work

The domain of software development is increasing rapidly throughout the world. In the initial years, only big organizations could afford the automation of their business model. With the time, the requirements of almost all type of organizations have been changed and to remain in the business, automation has become a mandatory tool to remain in the business. Most of the software development models are capable of developing either high risk large projects, which involve large amount of efforts and cost or small projects with almost no risk, less development cost and small development time. On the other hand there are certain organizations which have medium to small development burden. But the issue with such organizations is that, there is no specific model which may be suitable for such applications. The volume of such organizations has increased in the recent past.  The rapid increase in number of such applications which consists of low risk in their development behavior, motivated me to work on this.

## 3. Methodology

The software development model is a procedural mechanism in which pre-defined steps are to be followed. As mentioned in section 1, introduction part, there are several models used in software development process depending upon the requirement. There are instances when the customer is not much conversant about the requirements and the actual business process of that organization is complex enough to identify a suitable model. Moreover, a single model is not able fulfill the compete software development process requirements. In that case a hybrid model plays a crucial role. The software development mechanism is continuously changing very fast mainly after 1960. Emerging techniques are coming up some are need based and some are technology based. Enhancement in the capability and maturity of software development process, the cost, time, efforts are reducing, quality of developed product is increasing.

The schematic diagram of the proposed model is shown in Fig. 1. Initially a detailed model comprising broad functionality is constructed to understand and make clarity of the proposed system as a whole.

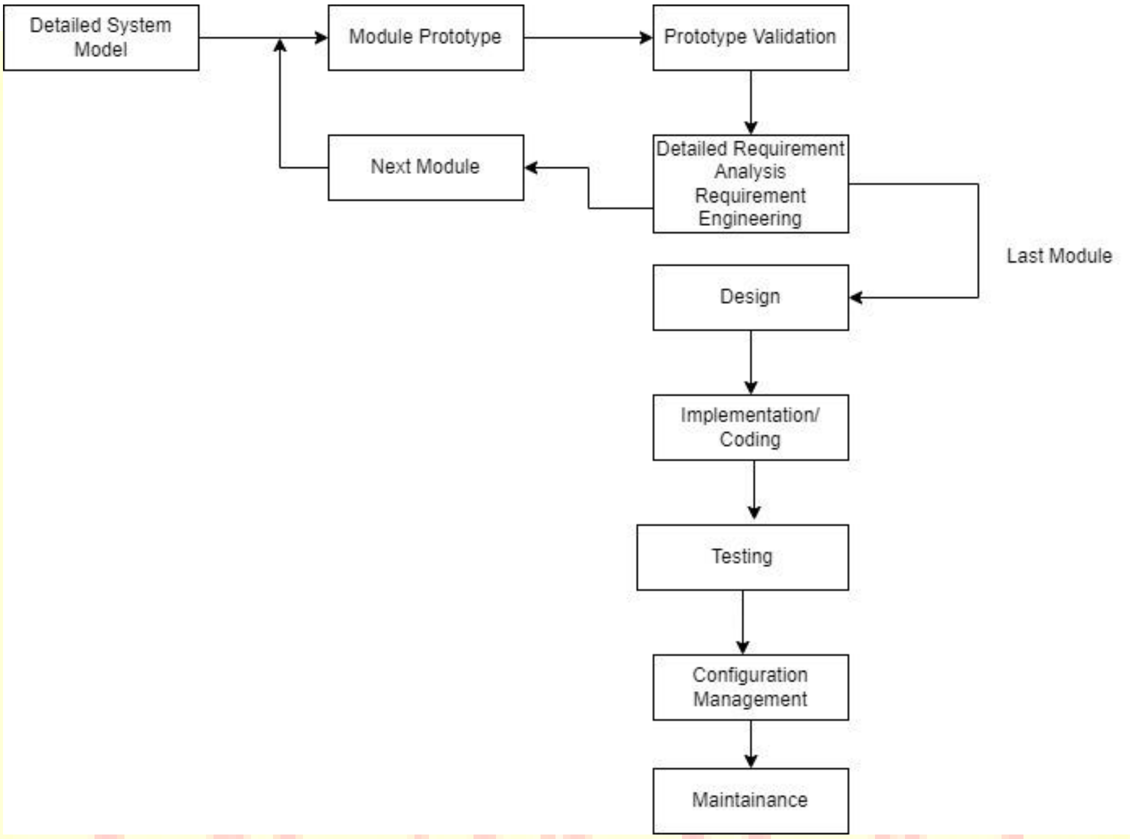

Fig. 1 Schematic Diagram of the Proposed Model

This detailed system model would be the complete broad description of the system under consideration comprises placement and connection of all the modules in the system. Schematic diagram would be very helpful and minimize the understanding gap between the requirements team and the customer. Once this broad schematic diagram is agreed and approved by both the parties, the work on the module level starts. The prototype of each module is constructed in details, on completion, the module is validated. The detailed requirement analysis is performed and documented in the form of SRS in accordance with IEEE 830. This process is repeated and continues for all the modules in the system. After this step the process of waterfall model starts and all the processes of waterfall model are executed step by step. Requirement document would record all the requirement details of the system under development and it would further used as a reference document for design purpose. The design document is the reference document for coding and/or implementation. The coding part followed by testing, configuration management and maintenance. These processes are described in detail, in the end of this section.

In my work, I have adopted a hybrid approach. The complete system development process has been segmented into two stages, stage I and stage II.The initial or preliminary stage emphasize on the understanding and intent of the whole system. The development process has been segregated in 2 stages, prototype segment stage one and waterfall segment stage 2.

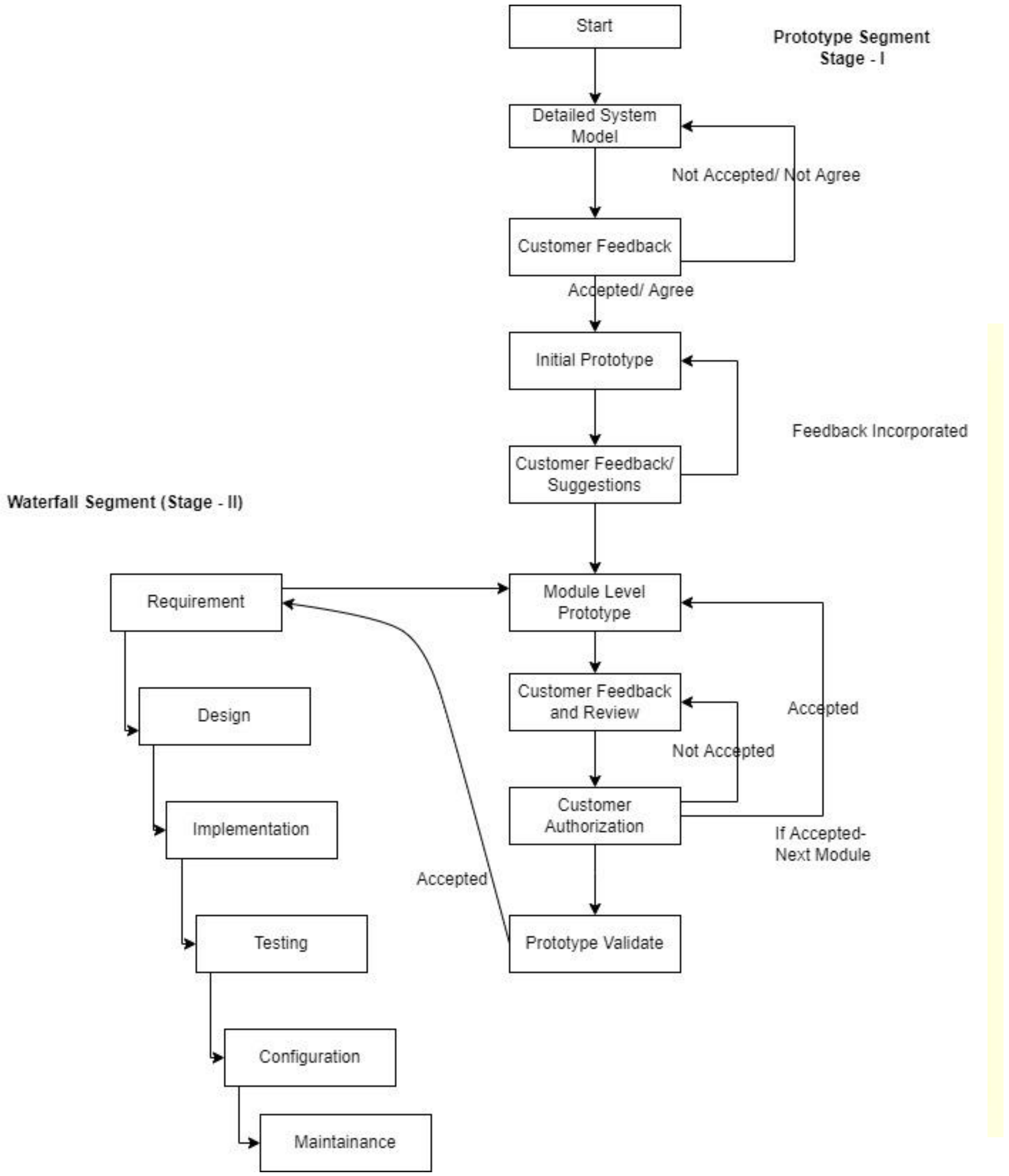

Fig. 2 Detailed Diagram of Proposed Model

Since my proposed model is mainly meant for such applications where the requirement is not clear or the customer is not aware about its requirements and the risk factor is moderate to low. In my model, the software development process is initiated with stage I prototype segment. After extensive study of existing manual business model of the organization in details , a preliminary detailed prototype to be constructed, the prototype shall be broadly comprise all the models of the system for better understanding and feedback. The purpose of this prototype is to understand the expectations and response of the customer or the organization. Once the basic prototype model is constructed, the customer or the organization shall be provided with enough time to understand and review the prototype model which has been provided to it for response. The review and feedback of the model at this stage is one of the basis of the successful and meaningful execution of the project. The customer understands the complete system in a broader perspective and on the basis of its understanding and the feedback, the model maybe either accepted or may be sent back to the requirement team for further changes. This process continues in the form of an iterative manner until the customer understands the whole system and expectations out of it. This part of the development process is quiet crucial. If there is any doubt or ambiguity in understanding or feel of the system, it may create misunderstanding or confusion like situation and deviate from expectations of the final product and the chances of its success may be affected drastically.

Once the system model is understood by the customer and the customer accepts, agrees and approves the complete system functionality as a whole, in the form of a complete system. After this process is over, the analysis of module level prototype starts. In this phase one model is a conceptualised, prepared in detail and sent to the customer for understanding, review and feedback. if any further modification or changes are required, the model level prototype is reviewed again and necessary changes would be initiated. This process continues until the prototype module is not accepted by the customer. When the customer agrees with the prototype model Shown to him and accept it, the accepted prototype would be validated and ensured that the prototype is complete in all respects eventually this prototype may be considered complete and ready and may be used for next step for further processing.

Once a prototype module is completed, accepted and validated in all respects, it is then sent to waterfall segment stage II. The stage II comprises the steps of a conventional waterfall model. When an accepted and validated module is received from prototype segment stage I, as an initial step, it is rigorously passes through the requirement analysis process. The requirements are gathered thoroughly and different requirement engineering operations such as requirement analysis, specification and validations are performed. Similarly this process continues for all the modules developed in prototype segment stage I. When all the requirement analysis of all the modules is completed, the requirements are formally documented and recorded in the requirement document. This requirement document shall be used further as an input for detailed design document, based on the requirement document as an input, a detailed design document shall be prepared. Subsequently the design document shell work as a reference and input document. Based on this document the coding and implementation process shall be initiated. These steps would be followed by other conventional processes of waterfall model such as:

Testing - once the coding process is in process, the testing start simultaneously with development phase. The aim of the testing at this stage is to identify and eliminate the bugs / errors in the software under development at the earliest. The quality of any product depends upon the quality and the testing methodology.

Configuration - the configuration phase basically caters the installation process of the developed software at the clients location / locations.

Maintenance - the maintenance of software is a continuous process, it starts immediately after installation and lasts until the software is retired.

Using this hybrid model both the stages plays a crucial role. Stage I, is one of the best possible method to understand a system under development and waterfall model make that requirement, design and implementation process simple, efficient and cost effective.

## 3. Conclusion and Future Work

I have tried to develop a software development approach for low cost, low risk and moderate size applications. The model may be suitable where the understanding about the requirements of the customer very less or no understands at all. This model may be suitable for the development team having no previous experience of executing the similar project of the similar domain. This model is a combination of two widely used software development models viz. prototype model and waterfall model. This integrated software development model shall provide flexibility, better understanding between the developer and the customer resulting more robust developed product satisfying complete needs of customer. This model shall be helpful in reducing the total cost, efforts and reduced development time. This model will also reduce interaction and customer input dependency in the project development. The proposed model is suitable for implementation on most of the software development tools.

In future, the integration of other models may be applied for medium to moderately large projects with medium risk low cost projects. The involvement of customer may be reduced further. The model may be modified and extended to generic applications for common solutions such as ERP, Human Resource Management and other cost effective solutions sold over the counter.

## References

[1] Ian Sommerville https://doi.org/10.1145/234313.234420

[2]Fred van den Bosch,John R. Ellis,Peter Freeman,Len Johnson,Carma L. McClure,Dick Robinson,Walt Scacchi,Arndt von Staa,Leonard L. Tripp Evaluation of software development life cycle: methodology implementation https://dl.acm.org/toc/sigsoft/1982/7/1

[3] Roger S. Pressman, Software Engineering : A Practioner's Approach , Mc Graw Hill

[4] https://dl.acm.org/toc/tosem/2002/11/4

[5] GENNAROCOSTAGLIOLA, VINCENZO DEUFEMIA, and GIUSEPPE POLESE Universit´adiSalerno, AFramework for Modeling and Implementing Visual Notations With Applications to Software Engineering, ACM Transactions on Software Engineering and Methodology, Vol. 13, No. 4, October 2004.

[6] BALASUBRAMANIAM RAMESH , TAREK ABDEL-HAMID, Modeling Dynamics in Agile Software Development LAN CAO Old Dominion University Georgia State University and Naval Postgraduate School,ACMTransactionsonManagement,InformationSystems,Vol.1,No.1,Article5,Publicationdate:December2010.

[7] Rikard Lindell, Crafting interaction: The epistemology of modern programming, https://doi.org/10.1007/s00779-013-0687-6

[8] Boehm B (2006) A view of twentieth and twenty first century software engineering. In: proceedings of the 28th international conference on software engineering (ICSE `06). ACM, New York, NY, USA, 12---29

[9] Paul Dourish, RankXeroxResearchCentre, DevelopingaReflectiveModelof CollaborativeSystems, ACMTransactionsonComputer-HumanInteraction,VOI2,No.1,March1995Pages40-63

[10] Elske van der Vaart1,2 1Theoretical Biology University of Groningen Kerklaan 30, 9751 NN Haren The Netherlands elskevdv@ai.rug.nl, Rineke Verbrugge2 2Artificial Intelligence University of Groningen P.O. Box 407, 9700 AK Groningen The Netherlands rineke@ai.rug.nl, Agent-based Models for Animal Cognition: A Proposal and Prototype, AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 2May 2008, pp 1145–1152

[11] Soichiro Hidaka, Zhenjiang Hu, Hiroyuki Kato, Keisuke Nakano Towards a compositional approach to model transformation for software developmentSAC '09: Proceedings of the 2009 ACM symposium on Applied ComputingMarch 2009, pp 468–475https://doi.org/10.1145/1529282.1529383

[12] Ben Livson Future software development management system prototypeCSC '88: Proceedings of the 1988 ACM sixteenth annual conference on Computer scienceFebruary 1988, pp47-56 https://doi.org/10.1145/322609.322615