ISSN:2231-0711
www.ijcset.net



# New Process Model: Yes Confident Software Development Process Model

S.Thulasi Krishna, K. Rajesh Kumar Reddy

*Dept of CSE*
*Kuppam Engineering College, Kuppam, Chittoor (Dt), AP.*
thulasi_krishna2001@yahoo.com, rajeshk535@gmail.com

**Abstract- An overview of the more common system development process models used to guide the system analysis, design development and maintenance of information. Generally we have many different techniques and methods used to software development life cycle. Project and most real word models are customized adaptations of the generic models while each is designed for a specific purpose or reason, most have similar goals and share many common tasks. This paper will explore the similarities and difference among these various models and also discovery new process models.**

**Index Terms- process model, Rup, and v process model, B&F model, F-model.**

## I.INTRODUCTION

"Software process model is an abstract representation of a software process" [1].Each process model represents a process from particular prospective, and thus provides only partial information about that process [1].The development life cycle of software comprises of four major stages namely Requirement Elicitation, Designing, Coding, Testing. A software process model is the basic frame work which gives a workflow from one stage to the next. This workflow is a guideline for successful planning, organization and final execution of the software project. A project model is chooses by keeping in view the nature of project, tools to be used and discovered by that are required [2].A software process is a sequence of tasks intended to produce a high quality software product on time and with in budget [3].A software development process is structure imposed on the development of a software product. Similar terms include software life cycle and software process. There are several models for each process, each describing approaches to a variety of tasks or activities that task place during the process, some people consider a life cycle model a more general terms and a software development process a more specific term.

## II.EXISTING SOFTWARE PROCESS MODELS

Listed below are some traditional and most commonly used software process models:

### A. Concurrent Engineering Model:

The concurrent development model sometimes called concurrent engineering model can be represented schematically as a series of frame work activities, software engineering action and task, and their associated status [4]. Provide a schematic representation of one software engineering task with in the modeling activities for the concurrent process model. The activity-modeling may be in any one of the states noted at any given time. Similarly, other activities or task can be represented in an analogous manner. All activities exist concurrently but reside in different states .its first iteration and exist in the waiting changes state. The modeling activities which existed in the none state while initial communication was completed, now makes a transition into the under development state. if, however, the customer indicates that changes in requirements must be made, the modeling activities moves from the under development states into the awaiting changes states. The concurrent process model defines a series of events that will trigger transition from state to state for each of the software engineering activities, actions, or tasks. The concurrent model is applicable to all types of software development and provides an accurate picture of the current state of a project. Rather than confining software engineering activities, actions, or task on the network exists simultaneously with other activities, actions, or tasks. Events generated at one point in the process network trigger transitions among the states.

The identified drawbacks of the process are:

- The SRS must be continually updated to reflect changes.
- It requires discipline to avoid adding too many new features too late in the project.

### B. The V-Model:

The V-Model [5] is an extension to the water model in that it does not follow a sequence model of execution rather it bends up word after the coding phase to form v-shape it bends upward after the coding phase to form v-shape.

It has the following drawbacks:

- It addresses software development within a project rather than a whole organization.
- The v-model is not complete as it argues to be as it covers all activities at too an abstract level.

### *C. Build and Fix Model:*

The evolution of software process models began its journey from the most primeval process model namely the "Build and Fix" model [8].

This model constitutes two basic steps:

1. Writing the Code
2. Fixing Problems in the code.

To give a clearer picture of its flaws, some important points are listed below:

- Does not follow any proven method.
- Its working included, first coding then moving towards other stages. Due to which the resulting product had a structure which often did not meet the requirements of the user consequently ending up in either Project termination redevelopment which was highly expensive [7].
- Due to a number of fixes, the resulting code had a poor structure also these fixes were highly expensive when addressed late in the development process. This was due to the absence of a detailed design phase before the coding phase [8].
- Not suitable for environment where changes are dynamic in nature [4] .
- The model handled lightly the testing phase after coding the basic requirements. This eventually led slipping of numerous undiscovered errors in the code; weakening its output [3].

**D.** ***Rapid Application Development Model:***

The RAD model [9] is an adaptation of the classical model for achieving rapid development using component based construction. If requirements are well understood with a well constrained project scope, the RAD process enables delivery of the fully function system. The model is considered to be incremental development model and that have emphasis on short development cycle.

Rapid Application Development has following drawbacks:

- Reduced features occur due to time boxing, where features are delayed to later versions in order to deliver basic functionality within abbreviated time.
- Reduced scalability occurs because a RAD developed application starts as a prototype and evolves into a finished application using existing component and their integration.
- RAD, for large projects, requires a sufficient number of human resources also requiring existence of components for reuse.
- Also RAD is not suitable for all types of application development [9].
- High technical risks discourage RAD use. This is because use of new technology in software is difficult in a changing global software market [9].

**E.** ***Waterfall Process Model:***

The Classical Life Cycle or the Waterfall Process Model [4] was the first process model to present a sequential framework, describing basic stages that are mandatory for a successful software development model. It formed the basis for most software development standards and consists of the following phases: Requirements elicitation, Designing, Implementation and Testing. The model restricted software engineers to follow a Sequential order moving from one stage to the next only after the completion of the former.

Listed below are some flaws:

- Rigid design and inflexible procedure.
- Restricting back & forth movement from a later stage to a former one.
- When new requirements surface accommodating those with existing ones become difficult due to restrictions in looping back to prior stages.
- Waterfall Model faced ***"inflexible point solutions"*** which meant that even small amendments in the design were difficult to incorporate later in design phase.
- As the requirements were frozen before moving to the design phase, using the incomplete set of requirement, a complete design was worked on. Such an approach worked normally well for a small project requiring average amendments. In case of a large project, completing a phase and then moving back to reconstruct the same phase, incurred a large overhead.
- Once a phase is done, it is not repeated again that is movement in the waterfall goes one to the next and the vice versa is not supported. Deadlines are difficult to me in case of large projects [7].

***F. Rational Unified Process:***

The RUP [1] provides dynamic, static and practice perspectives of a product. The RUP provides each team member with the guidelines, templates and tool mentors necessary for the entire team to take full advantage of the best practices. The software lifecycle is broken into cycles, each cycle working on a new generation of the product. This phased model identifies four discrete phases:

- Inception phase
- Elaboration phase
- Construction phase
- Transition phase

The identified drawbacks of the process are:

- Each phase has a milestone which needs to be satisfied for the next particular phase to start.
- If the respective milestone of the particular phase is not satisfied the entire project might get cancelled or re-engineered before proceeding further.
- The satisfaction criteria of a particular milestone has its own constraints and are not listed specifically [6].

**G. The Formal model:**

The formal methods model encompasses a set of activities that leads to formal mathematical specification of computer software formal methods enable a software engineer to specify, develop, and verify a computer-based system by applying a rigorous. When formal methods are used during development, they provide a mechanism for eliminating many of the problem that are difficult to overcome using other software engineering cardiogram

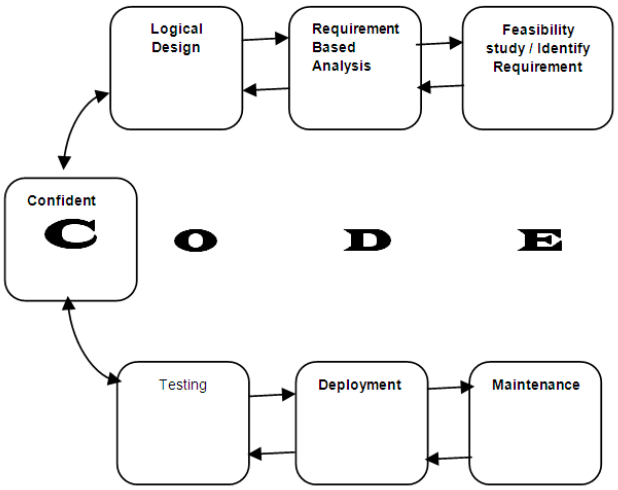
These identified drawbacks of the process are:

- The development of formal models is currently quite time consuming and expensive
- It is difficult to use the models as a communication mechanism for technically unsophisticated customers.

## III. CONFIDENT SOFTWARE DEVELOPMENT PROCESS MODEL

The Confident process model which we have proposed has seven phases, namely; Feasibility study/Requirement, Requirement Based Analysis, Logical Design, Confident Code, Logical Testing, Implementation & Deployment, and Maintenance. It is a flexible model not restricting the developers enabling them to move both Front and back from any given stage to any other stage during its lifecycle. Each phase is further divided into sub phases, each specifying a criterion which has to be met to move to the next phase. This criterion also points out which phase to back track in case of failure. The model starts with the Feasibility Study/ Requirement and Analysis phase in which requirements are gathered as well as validated by the user. Next phase inline is the Logical Design, We will design our software .Next phase Confident Code, we will write code confident manner. Next phase Logical Testing phase in which
we will testing software after coding. Next phase , Deployment ,it's last phase ,our software and later on it is tested and is validated by the user. But meanwhile if some of the requirements changes or new requirements surface during this phase, then we can loop back to requirement phase to accommodate new requirements with the existing one. Following this phase is the Implementation and Deployment phase which ultimately leads to Maintenance Phase.

Figure 1:




### A. Identify *Requirements & Requirement Based Analysis(IR&RBA):*

First, Feasibility study, System Based Requirement and Model Based Requirement, Identify requirements and are divided into "Must have" and "Should have" requirements as shown in fig. 2. The "Must have" requirements are those which are most important to the system and must be the part of the system. These mandatory requirements are expected to be implemented in the first version itself. While "Should have" requirement are those which have lesser priority owing to trivial implementation their they are delayed till later versions."Should have" requirements are equally important and must not be left out in later versions. We have divided requirements to have more and more focus on the "Must have" requirements, and also to make our work simple and easier to carry out. The requirements are then initially Verification and validated by the user and we are left with the refined requirements only. Initial Verification and Validation by the user will ensure whether or not the developer is going in the right direction and that all the concerned parties are committed with the requirements being gathered. The refined requirements are also revalidated again by the user in the Final Validation step to get the clear and obvious picture of the user requirements. Then all the requirements i.e. "Must have" and "Should have" are integrated to have the overall scenario of requirements for the software.

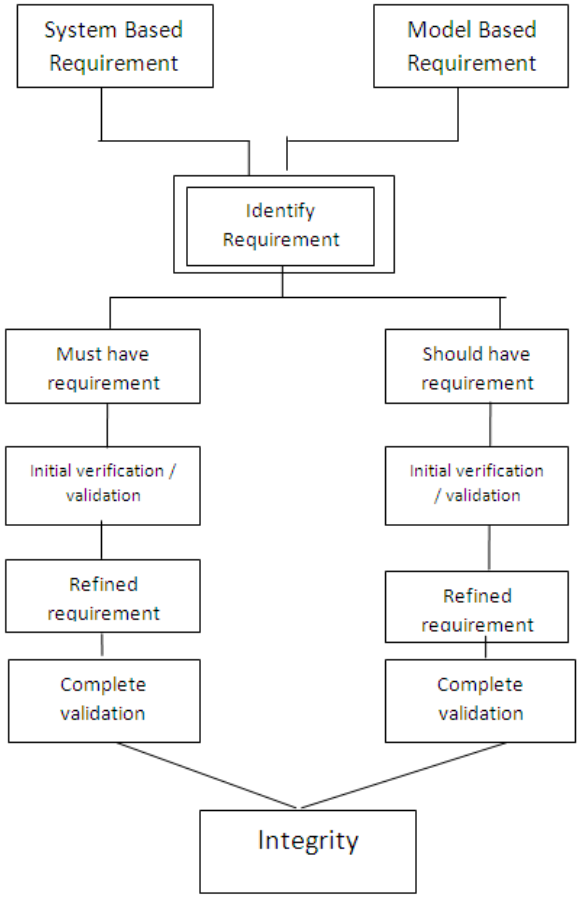



Figure 2: Identify Requirement &Requirement Based Analysis

### B. Logical *Design with Validation:*

Beginning of Design phase is marked by the existence of an Initial Architectural Model. Architecture Model consists of all basic modules that our system will have. The next step is the Risk Analysis and Risk Resolution of the Architectural

Model in the Designing phase. A risk [10] is a potential for loss or damage to development of the software from materialized threats. Risk Analysis attempts to identify all the risks and then quantify the severity of the risks. If it occurs, it exploits vulnerability in the security of a computer based system. Here, different risks involved are identified and resolved ensuring secure development and advancement of the design towards its deliverable version. This step leads to achievement of a Design. Following this step is the Model Validation step. Here the identified Architectural Model is validated ensuring all basic modules have been covered. The validation's major objective is to ensure that all concerned parties are equally satisfied by the output. Once the design is validated and is found to be up to the mark, next step is to give a more detailed complete Design Fig 3.

Each Validation sub phase in Design phase commands fulfillment of a certain criteria. Criteria fulfillment allows moving downward/ forward in the flow towards the next phase whereas criteria non-fulfillment restricts moving forward and loops back to the phase decided by the criteria. The flow works its way all the way back towards the phase it looped back from. The criteria fulfillment is tested again and next decision is taken accordingly.

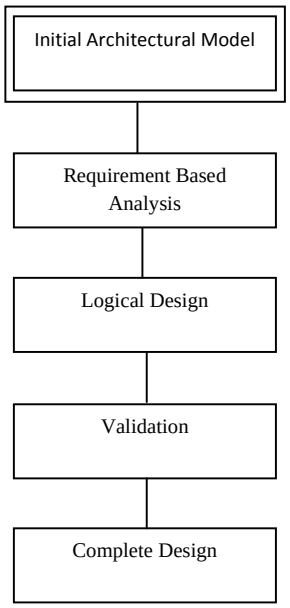



Figure 3: Logical Design & Testing

**C. Code *&Deployment:***

The first sub phase of code & Deployment is Coding. Next this coding is analyzed for Risks (if any) and their Resolution. Risk Analysis and Resolution (if any) like technology risk, platform risk, programming language risk, and last but not least the cost risk. Then, whole system is tested; here the System Testing is involved. System testing of software is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. [16] Now, the software is validated and verified by the concerned authority. After which the system is deployed. Deployment should be interpreted as a general process that has to be customized according to specific requirements or characteristics.

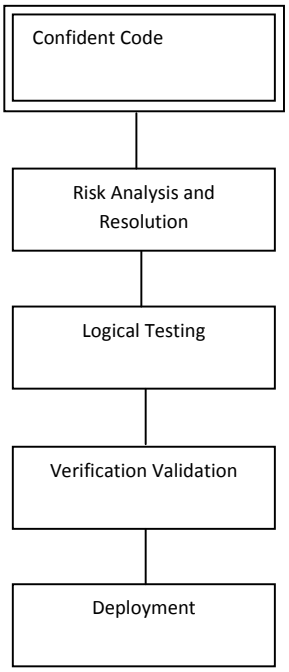



Figure 4: Confident Code & Deployment

**D. *Maintenance:***

Maintenance refers to modification in the product after delivery, to correct errors or to improve the performance. Maintenance phase in our model may include many activities like creation of management plan, analysis of the problem process. It includes other activities like coding of modification, checking whether the software conforms to particular environment or not whatever is needed.

## IV. CONCLUSION

Waterfall assumes all upcoming requirements to be frozen and to be accommodated in later versions of the product. In Confident Model, we haven't frozen requirement phase, one can move easily from design phase to requirement phase if a new requirement(s) surfaces. We can easily incorporate small to large changes in any phase. Unlike Evolutionary Model, Confident Model does not include users involvement at every step rather users are involved only when there are critical assessments to be made. In prototype model, the user is provided with the demo of the system. The user suggestions and feedback are used to incorporate new requirements and correct existing ones. This ultimately ends up in not knowing in advance how long it will take to complete a project.

## REFERENCES


[1] Ian Somerville, "Software Engineering", 8th Edition, 2006, pp. 89.

[2] R.S. Pressman, "*Software Engineering, A Practitioner's Approach*",5th ed. New York: McGraw-Hill, 2001, pp. 26.

[3]"Software project management a Real Word Guide to success "person Education, pp.25

[4] Roger S.Pressman" Software Engineering, a practitioner's Approch"6th Ed New York: McGraw-hill 2005 pp 88.

[5] Jeff Tian, Southern Methodist University, Dallas, "Software Quality Engineering", IEEE, Computer Society Publication, Willy Inter Science, 2005.

[6] P. Kruchten, "*Rational Unified Process Best Practices for Software Development Teams"*, Canada: rational Software, 2001

[7] B.W. Boehm, "*Anchoring the Software Process*", IEEE, IEEE Software, vol. 13, Issue 4, July 1996, pp. 73-83

[8] E. Carmel, S. Becker, "*A Process Model For Packaged Software Development*", IEEE, IEEE Transaction on Engineering Management, vol. 42, Feb 1995, pp. 50-58.

[9] R.S. Pressman, "*Software Engineering, A Practitioner's Approach*", 5th ed. NewYork: McGraw-Hill, 2001, pp. 34.

[10]www.onestoptesting.com/risk Analysis/, accessed on 25, Nov. 2009. [11]"http://en.wikipedia.org/wiki/System_ Testing" accessed on 26, Nov. 2009.


## AUTHORS:

**S.THULASI KRISHNA** received the B.Tech. Degree in Computer Science and Engineering from Jawaharlal Nehru University ,Hyderbad,India in 2005, M.E from Sathyabama University Chennai ,and Ph.D pursing from Rayalaseema University ,Kurnool. He joined as Asst.professor in VIST Engineering College in august 2005 ,Hyderabad. He was worked as Asso.Professor in Vidyanikethan Engineering College Tirupathi. and currently working as Asso.Professor in Kuppam Engineering College. He is member of International Association of Engineering.

**K. Rajesh Kumar Reddy** received the B.Tech. Degree in Computer Science and Engineering from Jawaharlal Nehru University, Anantapur in 2009, M.Tech from Jawaharlal Nehru University, Anantapur in 2011 and currently working as Assistant Professor in Kuppam Engineering College. He is member of International Association of Engineering.