ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 8, Issue 8 , August 2021

An Enhanced virtual multi-variant behaviour based Interdisciplinary software Project Development model or methodology with a novel improvised Prototype model as a sub model

Shivankur Thapliyal, Renu Bahuguna

MCA, GBPIET Pauri, Uttarakhand, India, Assistant Professor: Doon Institute of Engineering and Technology, Rishikesh Dehradun, Department: Computer Science and Engineering
M-TECH , GBPAU Pantnagar, Uttarakhand, India, HOD-CSE: Doon Institute of Engineering and Technology, Rishikesh Dehradun, Department: Computer Science and Engineering

ABSTRACT: In this paper, we proposed a virtual multidisciplinary model for Software Project development process because Software Project Development model play's a major and significant role behind the deployment of each software product. Various Evolutionary and Incremental process models also accommodated with the traditional approach of software development process , but here In the today's information age context , Software project development processes are also a very crucial or significant and multidisciplinary in nature, so the need of the hour that we have a strong multidisciplinary software project development model behind the deployment of any potential scientific real time based simulated automated software systems. In the present scenario the development of any Real time or scientific applications based simulated software are very typical in nature , so the software development process model , which works on traditional orthodox methodology are not much sufficient to deploy these type of potential and strong software systems because the growth of software engineering research issues are dynamically enhanced with full fledge of advanced superior potential mechanisms and also contains various multi varieties and multidisciplinary in nature with enough of complex and typical behaviour. So In this research work we focused those areas which are more crucial and very critical in nature. We proposed a generalized virtual multidisciplinary software project development model, which concerns all critical areas and mechanisms and also tracing the way of development process of any software systems from requirements gathering to deployment stage. This model also applied for Soft Real Time or Scientific Applications or simulation based software because it's contains a very broad spectrum, which explore each aspects of software development processes at each stages. We also known that the process of software development are not adhoc in nature , each process which separately run on each stages are deploy a software module , which is not a whole software product but a part of it , also dependent on the another module , which accomplished on next stages , so these all stages respective software project development process are linearly or cyclic in nature based on the nature of software, here we try to deploy a multidisciplinary model for these type of potential and strong software systems.

KEYWORDS: Software Development model, Software Engineering models, Interdisciplinary Software Development models, Prototype models , Software Project Development Process.

I. INTRODUCTION

The recent trends of Software Engineering issues [1] are still very challenging and typical, In today's information age where the role of software systems [2] or applications are multidisciplinary in nature. Behind the development of Software systems a strong potential software project processes [5] have to be included, which provides some fundamental background knowledge based information for developing a kind of knowledge based software systems [5]. In the processes of software project development many traditional software process model have to be proposed previously by many organisations and researchers. But to concern the multidisciplinary nature of software system, a

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 8, Issue 8 , August 2021

multidisciplinary software project development model [7] is the basic needs of software information age. Our research work totally dependent on this concept that how we make a software project process development model [3], multidisciplinary in nature. There are various proposed software project development models are existed, but after the detailed analysis many findings prove that some models are linearly in nature, and some are cyclic in nature, but many time after the proper utilization of these software models doesn't released a satisfactory software product. There are various software failures [4] comes into existence at recent times , the main reason of this failure includes that does not proper utilization of software project process development model or the model which would adopt for software project development are doesn't give a satisfactory results or it's doesn't multidisciplinary in nature. So here we will try to optimize these failure rates of software system by using the newly proposed multidisciplinary software project development model. A generalized methodology for development software system, SDLC provides a strong mechanism for software development. Here our models also adopts some significant phases of SDLC [4] but the internal mechanisms and the way of developing methodologies are totally differ than all other software project process development models [3]. There are another very strong potential methodology for software development which is the agile methodology, where the iterative development processes have to be adopt for software development. In the Agile methodology [11] each phases of software project development user interaction have to possible to established satisfaction between the stakeholders and the developers. But here we adopt some mechanisms of Agile [11] that maximum user interactions also possible in all phases of software project development but here the role of the developer or stakeholders are changed and it's play a multidisciplinary role in this model.

II. SOME PREVIOUSLY PROPOSED METHODOLOGY FOR SOFTWARE DEVELOPMENT PROCESS

There are various software development models which are responsible to deploy a potential software product. Some of these methodologies are :

A. SDLC (Software Development Life Cycle) [11,13]: There are multiple phases which are existed in SDLC, some of these phases are as follows:

A.1 Requirements Gathering: In these phase we gathered user requirements for software development. A Requirements Engineering [13] have to be applied on the process of gathering the user requirements. Requirements engineering covers the 4 aspects to categorized the process of requirements gathering, which are Requirement Elicitation [13], Requirements Analysis [13], Requirements documentation [13] and Requirements Review [13]. After covers these 4 aspects a well-defined Software Requirement Specification (SRS) [15] have to be modelled or designed. A SRS [15] is a framework, which contains straightforward requirements in a very well defined abstract and concrete/concise manner. A well-defined SRS [15] contains following things: Correct, Unambiguous, Complete, Ranked for important and/or stability, Verifiable, Modifiable and Traceable. A SRS [15] document always checked for Completeness & consistency, Conformance to standards, Requirements conflicts, Technical errors, and Ambiguous requirements. After gathering the user requirements, the next stage will require to do detailed feasibility analysis of these requirements.

A.2 Detailed Feasibility Analysis: This stage required a detailed Feasibility Analysis for make up the software correspondence to user requirements. A detailed Feasibility Study would established in this phase of SDLC [11]. The detailed Feasibility Study [21] categorized into 10 parts , which are as follows:

- ✓ Technical Feasibility Study
- ✓ Managerial Feasibility Study
- ✓ Economic Feasibility Study
- ✓ Financial Feasibility Study
- ✓ Cultural Feasibility Study
- ✓ Social Feasibility Study
- ✓ Safety Feasibility Study
- ✓ Political Feasibility Study
- ✓ Environmental Feasibility Study
- ✓ Market Feasibility Study

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 8, Issue 8 , August 2021

After analysis detailed feasibility study [21] a detailed documentation preview would accomplished in the next phase.

A.3 Analysis: In the analysis phase a detailed documentation would accomplished, various diagrams have to be prepare to draw a detailed layout corresponding to software systems [22] . Some of these diagrams [24] are as follows :

- Structure Diagrams
  - ✓ Class Diagram
  - ✓ Component Diagram
  - ✓ Deployment Diagram
  - ✓ Object Diagram
  - ✓ Package Diagram
  - ✓ Profile Diagram
  - ✓ Composite Structure Diagram
- Behavioural Diagrams
  - ✓ Use Case Diagram
  - ✓ Activity Diagram
  - ✓ State Machine Diagram
  - ✓ Sequence Diagram
  - ✓ Communication Diagram
  - ✓ Interaction Overview Diagram
  - ✓ Timing Diagram

Various diagrams have to prepare according to needs of software development process. Data Flow diagram and Flow charts will also prepares in these stage of software development.

A.4 Design: In these phase of software development [11] the activities of development of source code will occur. A System Administrator must have a sufficient knowledge of programing skills and technical ability. All measurement matrices such as Halstead Software Science matrices [31] and finding the Cyclamate Complexity [32] and Function Point Analysis (FP) [32] would also occur in these phase. After the development of Source code a detailed documentation would also prepare corresponding to software development [11] . Risk Analysis [39] are also sone in this phase of SDLC [11] , following steps would taken to analyse Risk factor in software design:

- **Risk Management**
  - Risk Assessment
    - ✓ Risk Identification
    - ✓ Risk Analysis
    - ✓ Risk Prioritization
  - Risk Control
    - ✓ Risk Management Planning
    - ✓ Risk Monitoring
    - ✓ Risk Resolution

A.5 Implementation : Testing the software are the most significant phase of software process development model [11] after designing the detailed functionality of software product through the well-defined source code , but verification and validation are still a very challenging task for software development process [11] , so here are the various software testing strategies which is fundamentally most significant for any software design. Generally we divided testing [21] into some broad spectrum , which are as follows:

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 8, Issue 8 , August 2021

- Accessibility testing
  - ✓ Acceptance testing
  - ✓ Black box testing
    - Boundary Value Analysis (BVA)
    - Equivalent Partitioning
- End to end testing
- Functional testing
- Interactive testing
  - ✓ Integration testing
  - ✓ Load testing
- Non-functional testing
  - ✓ Performance testing
- Regression testing
  - ✓ Sanity testing
  - ✓ Security testing
- Single user performance testing
  - ✓ Smoke testing
  - ✓ Stress testing
  - ✓ Unit testing
  - ✓ White-box testing

A.6 Post Implementation and Maintenance: After releasing Software product an adequate maintenance would require with appropriate time stamp, because in the today's information age various technological advancements would occur. As the technological perspective and technological advancements will increasing abnormally, a proper maintenance of software product would require that our software systems adopt the dynamically behaviour of ever changing environment of technology and platforms. Some of the various types of maintenance [34] , here are as follows:

- ✓ Preventive Maintenance.
- ✓ Condition-Based Maintenance.
- ✓ Predictive Maintenance.
- ✓ Corrective Maintenance.
- ✓ Predetermined Maintenance.
- ✓ Gaining Maintenance Knowledge with Interplay Learning.

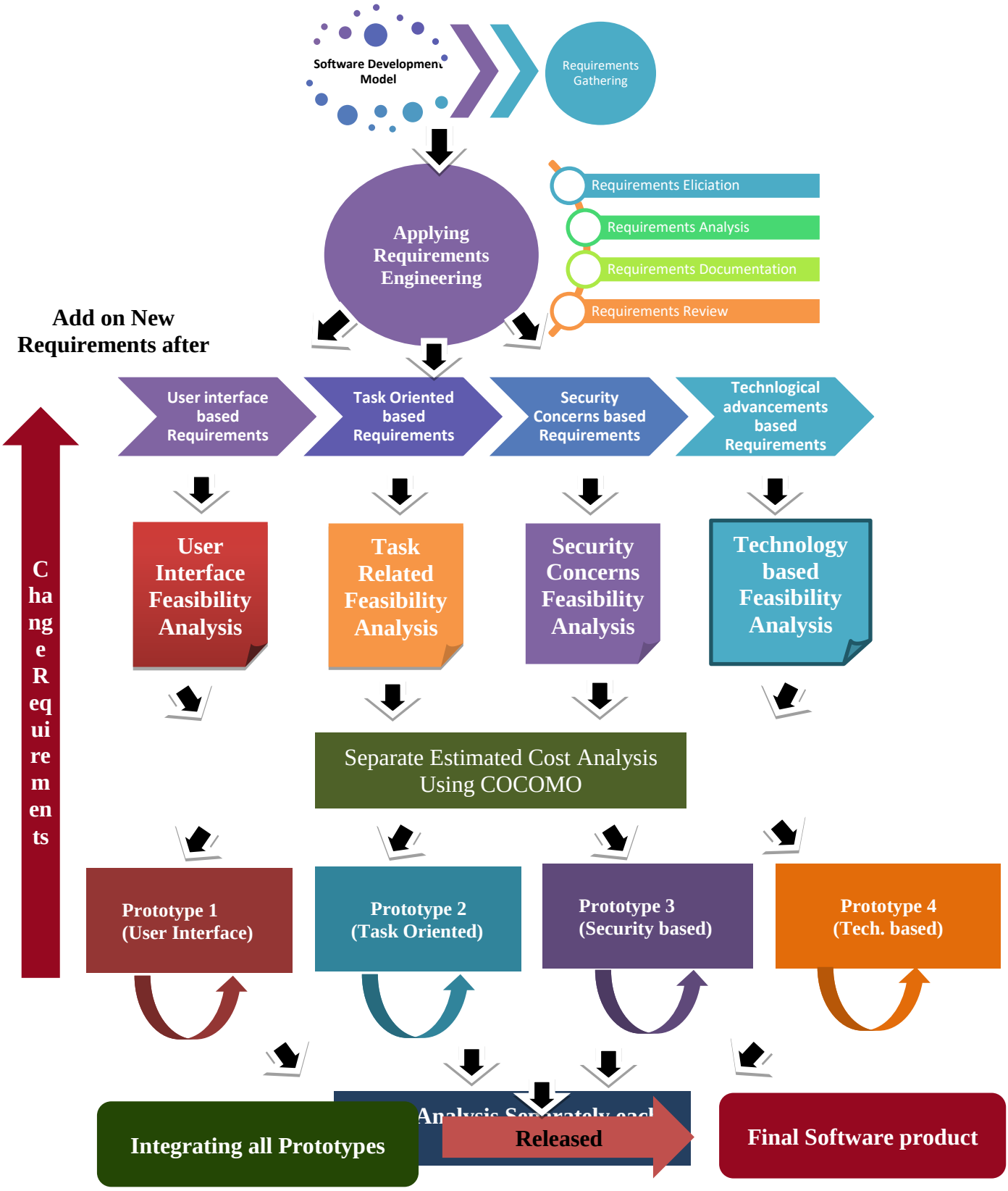
B. Agile Methodology [21]

This methodology is very much similar to incremental process model but one main thing that differs it from another model is that it's an iterative in nature. Each software modules will get through each iterations, generally called increments. Each increments user presence make possible to obtain the desired feedback related to that module, and this process is cyclic in nature. Firstly we gathered some requirements and initially we take some requirements and prepare a module related to that and obtains desired feedback by make user presence available after the development of each increments, user will provide some necessary adequate feedback according to that module , if some changes require then the developer easily modified that module , because this process is cyclic in nature so this process repeated several times , until the final software have been deployed to the stakeholders.

C. A NEWLY PROPOSED VIRTUAL MULTI VARIANT GENERALIZED INTERDISCIPLINARY SOFTWARE PROJECT DEVELOPMENT MODEL.

The Diagrammatic representation of this model is as follows:

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 8, Issue 8 , August 2021



ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 8, Issue 8 , August 2021

Now each phase elaborate with some steps:
Step 1: Firstly we gathered user requirements, for gathering user requirements in appropriately manner firstly we will apply Requirements Engineering, using this approach, its must follows these activities, which would conduct on these four phases of Requirements Engineering, and these 4 phases are as follows: Requirements Elicitation, Requirements Analysis, Requirements Documentation and Requirements Review. Now after gathering user requirements through the use of Requirements Engineering, we categorized or partitioned these requirements in the following 4 phases: which are as follows:

- User Interface based Requirements
- Task Oriented based Requirements
- Security Concern based Requirements
- Technological advancements based Requirements

C.1 User Interface based Requirements: In this phase some user interface based requirements have to be placed ,now how we know that which requirements are based on user interface so we study the broad spectrum of user interface based requirements , In this requirements we further categorized user interface based requirements into some phases which are as follows:

- ✓ User Groups
- ✓ Content Presentation
- ✓ User Instructions
- ✓ User Navigation
- ✓ Data Manipulation
- ✓ Saving and Restoring work

All Requirements which is related to these categorizes, those requirements must place into the category of User Interface based Requirements.
The diagrammatic representation of user interface based requirements are as follows:

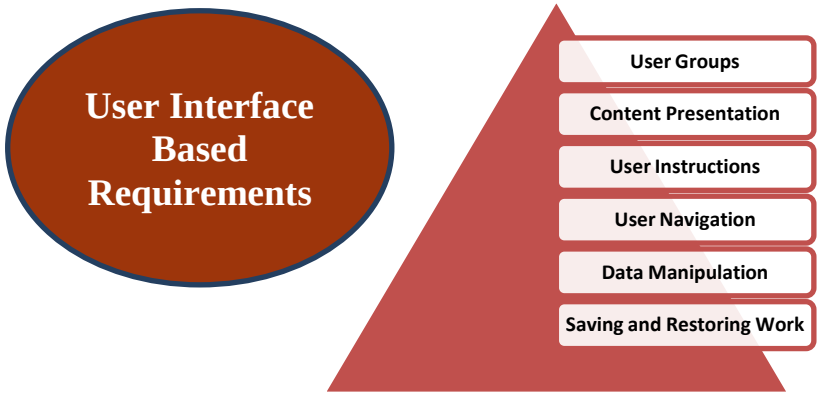


Fig a: User Interface based requirements

C.2 Tasks Oriented Based Requirements: In this phase , all focused upon the task of the software systems, Requirements which are related or based on the main task of the software systems we placed those requirements into the Task Oriented based Requirements. Now a one major question will arise that how we know that , which requirements comes under the category of this phase , now we analysis a detailed study related to those requirements , which belongs to the main tasks of the software and expand a broad spectrum of task oriented based requirements , Now some sub categories of task oriented based requirements are as follows:

- ✓ Stakeholder Tasks
- ✓ Data Collection and Pre processing

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 8, Issue 8 , August 2021

- ✓ Task Description
- ✓ Domain data (UML, Class diagram , ER Diagram)
- ✓ System Function , functions description
- ✓ Interaction data , UML class diagram
- ✓ UI data , Virtual window
- ✓ Modularization

So these are some sub categories of task oriented based requirements , user requirements which are based upon those sub categories , we placed those requirements into these sub categories.
The diagrammatic representation of these task oriented requirements phase are as follows:

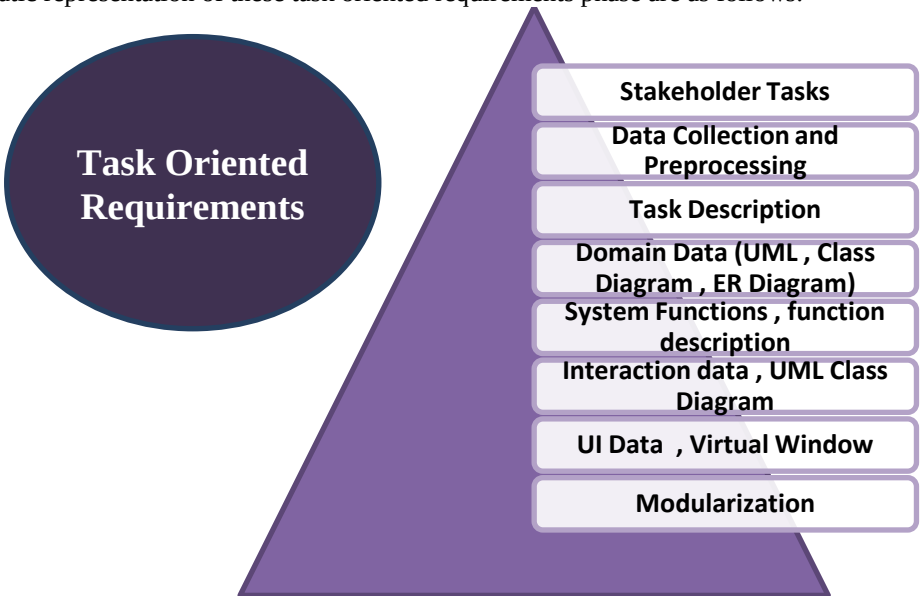


Fig b: Task Oriented based requirement

C.3 Security concern based Requirements: Requirements those are directly related to the security aspects of software systems, we placed those requirements into this phase of software requirements model. Generally it's the most significant phase of software Requirements gathering , because the behaviour of the software are totally dependent on the security aspects of the software , so requirements which are related to security aspects are carefully observable or identified , but a one major question will arise that how we know that which requirements belongs to the security aspects of software now we also do the detailed analysis and brief study and expand the broad spectrum of Security Concern based Requirements and divided this phase into some subcategories and these subcategories are as follows:

- ✓ Access Control
- ✓ Security Policy
- ✓ Non Repudiation
- ✓ Physical Protection
- ✓ System Recovery
- ✓ Attack Detection
- ✓ Boundary Protection

Requirements those belongs to these sub categories we placed those requirements into this phase of software requirements model.
The diagrammatic representation of this phase are as follows:

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 8, Issue 8 , August 2021

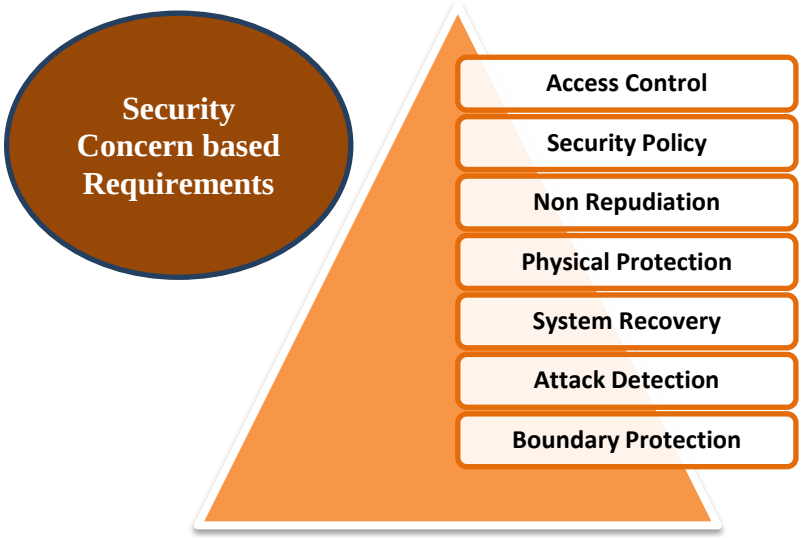


Fig c: Security Concerns based requirements

C.4 Technological advancements based Requirements : In this phase we placed those requirements which are directly inter-related to the technology trends of software development , such as the current technology , which the software adopt to do some tasks , and the platform or environment where the software systems performs it's functionalities and operating system related requirements and the portability of the software systems. But we also explore this phase and make up the broad spectrum of this phase , which are as follows:

- ✓ Implementation a Cloud based Data Storage.
- ✓ Strengthen Cyber Security Policies
- ✓ Platforms / Environment Interfaces
- ✓ Technological Pedagogical Knowledge (TPK)
- ✓ Current Trends of Technology , Big Data , IOT
- ✓ Knowledge based system for eco design
- ✓ Operability and Reliability

Requirements those comes under the sub categories of technological based requirements we placed those requirements into this phase.
The diagrammatic representation of this phase are as follows:

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 8, Issue 8 , August 2021

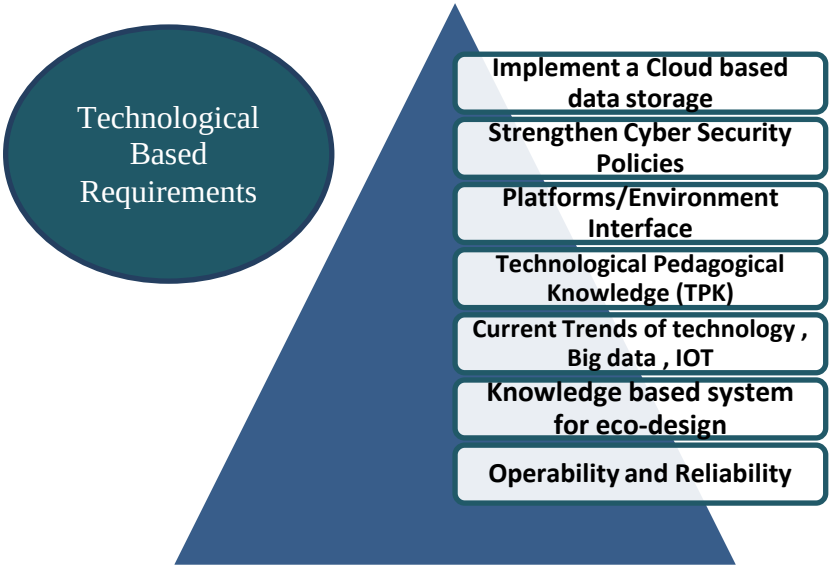


Fig d: Technological based requirements

Step 2: After categorization of requirements into four respective partitions such as user interface based requirements, task oriented based requirements, security concern based requirements, technological advancements based requirements, the next process is to do the detailed analysis of feasibility study corresponding to these requirements separately and at the end a cumulative feasibility analysis have to be done. There are various categories of feasibility study, in a broad spectrum feasibility study are divided into several sub parts, some of them are as follows.

- ✓ Technical Feasibility Study
- ✓ Managerial Feasibility Study
- ✓ Economic Feasibility Study
- ✓ Financial Feasibility Study
- ✓ Cultural Feasibility Study
- ✓ Social Feasibility Study
- ✓ Safety Feasibility Study
- ✓ Political Feasibility Study
- ✓ Environmental Feasibility Study
- ✓ Market Feasibility Study

Some of the most significant and generalized feasibility study, which are must require for any software development process are as follows:

- ✓ Technical Feasibility
- ✓ Economic Feasibility
- ✓ Legal Feasibility
- ✓ Operational/Behaviour Feasibility
- ✓ Schedule Feasibility

The diagrammatic representation of feasibility analysis are as follows:

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 8, Issue 8 , August 2021

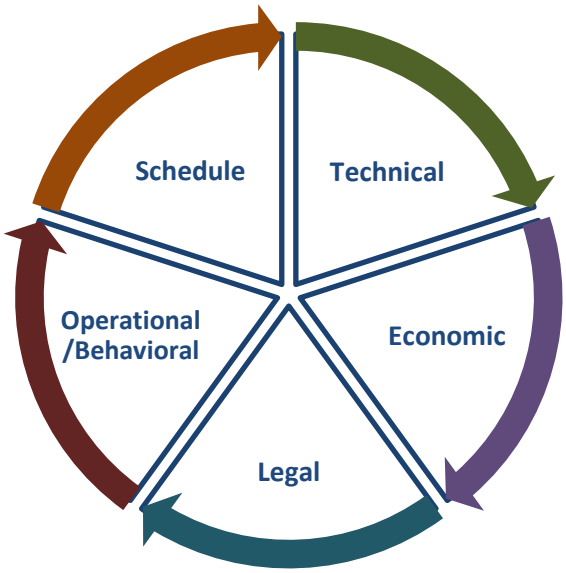


Fig e: Types of Feasibility Study

In this phase of software development model Cost analysis [32] is an important and significant factor. There are various cost analysis models but COCOMO [32] is a very reliable model to find out the estimated cost of software systems. There are various flavours of COCOMO [32] to find out the estimated cost some of them are as follows:

- ✓ Organic
- ✓ Semi-detached
- ✓ Embedded

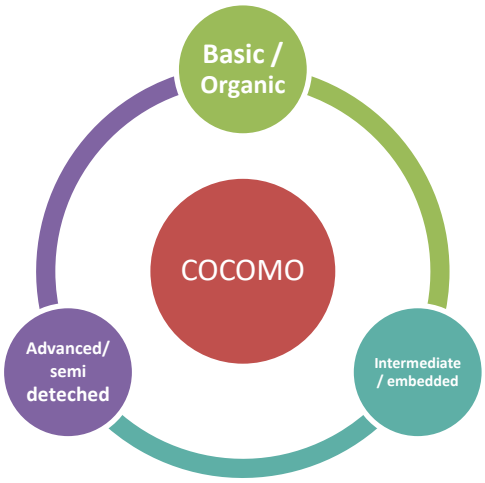


Fig g: Constructive Cost model (COCOMO) for Cost analysis

The graphical representation of COCMO model with respect to time and size are as follows:

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 8, Issue 8 , August 2021

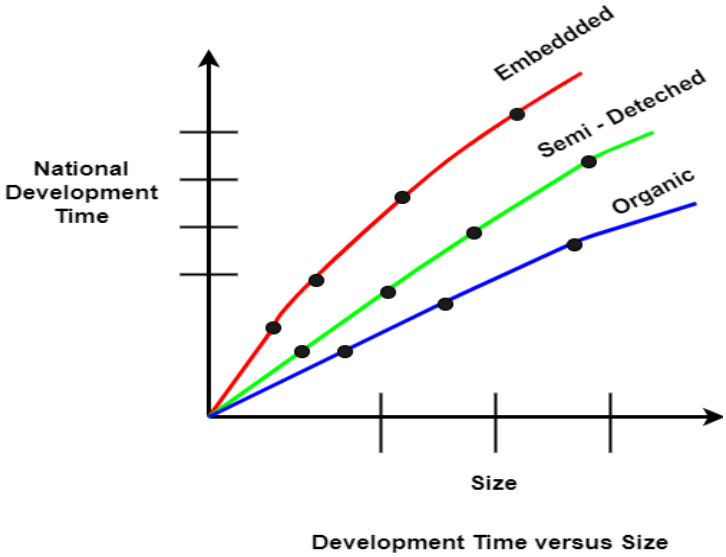


Fig h: Graph of COCOMO model

** Cost Analysis also done with separately corresponding to all 4 phases of this software project development model.

Step 3: After Analysis all the above phases, the next phase is to design it. A separate prototype have been built corresponding to requirements. In this software project development model, the traditional model have not been used, it's a newly proposed prototype model in a broad spectrum. In the agile methodology , we never known that how many iterations will occur it's totally dependent on the size of the software product , but in this multi variant Interdisciplinary software project development model , we partitioned it or we can say that the total number of iteration are only 4 , which prioritized in these four categories , such as user based requirements, task oriented based requirements , security concern based requirements and technological advancements based requirements , so because in the agile methodology software building process becomes very slow in manner because , the number of iteration would be drastically increase , when the software reach it's peek stage points or develop to an end , so we don't know that how many iterations will require to building any software project , and this is the reason that maximum time user interactions becomes possible , when each iteration deployed , this also becomes software development process very slow , so here we optimized these thing , we generalized these model into 4 phases and this is the reason , it's model working mechanism are very fast in nature because the number of iterations are optimized and maximum time the number of user interactions also becomes possible very low corresponding to software development time. We proposed a prototype model in a broad spectrum are as follows:

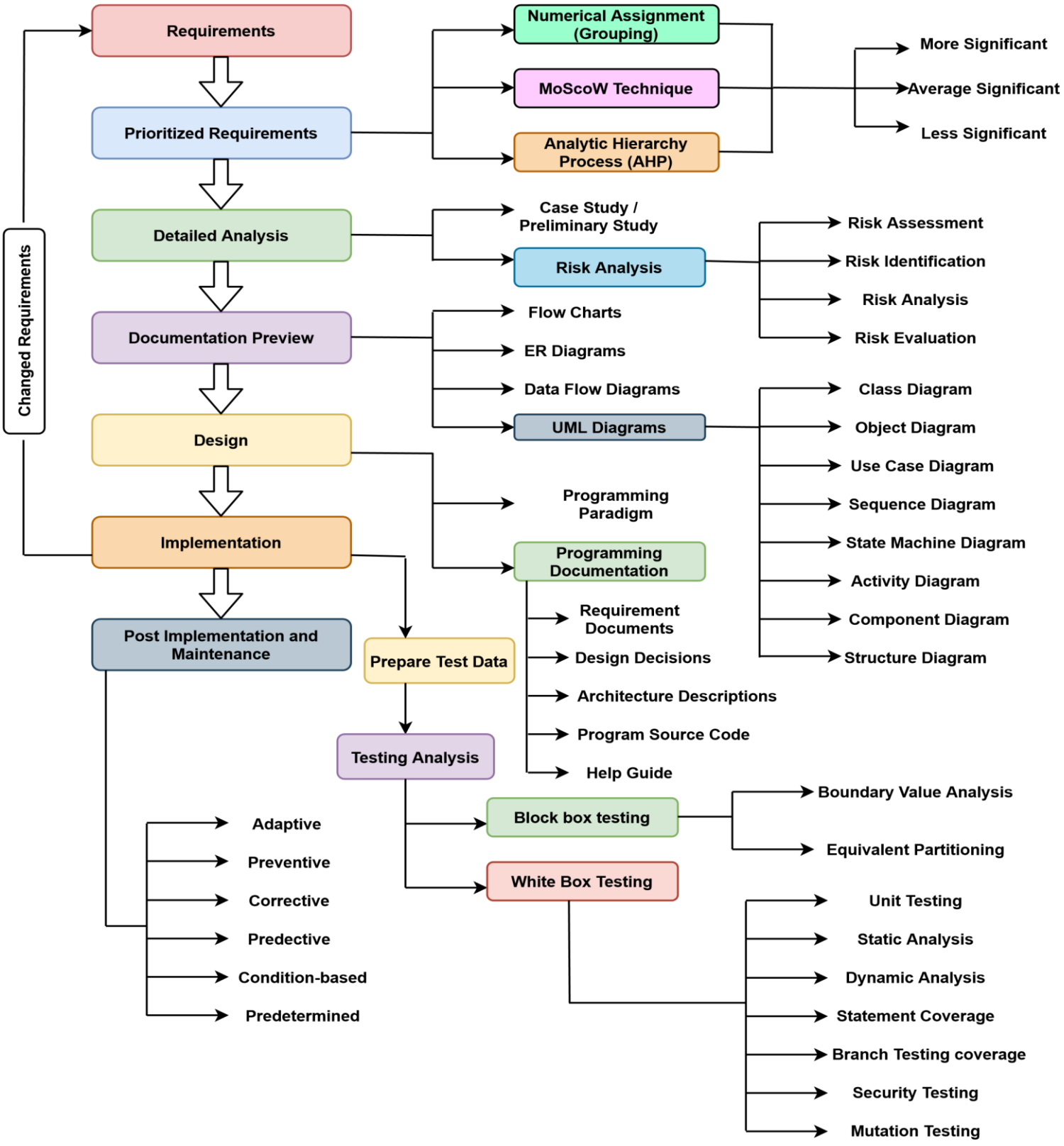
III. A newly proposed Prototype model in the broad spectrum

A newly proposed Prototype model in the broad spectrum
The diagrammatic representation of this newly proposed prototype model are as follows:
This model contains 7 stages for software project development and exploring each stages in the broad fashion to do the detailed analysis and to understand the concept of software project development process. These 7 stages are as follows:

- ✓ Requirements
- ✓ Prioritized Requirements
- ✓ Detailed Analysis
- ✓ Documentation Preview
- ✓ Design
- ✓ Implementation
- ✓ Post Implementation and maintenance

The main advantageous factor of this prototype model is that this model also contains the Risk Analysis Factor and also contains the documentation preview.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 8, Issue 8 , August 2021



ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 8, Issue 8 , August 2021

Generally traditional prototype model doesn't provide the mechanism of Risk Analysis, but this prototype model also contains the Risk Analysis Factor and will also explore each stages in a broad way. That's why it's a very reliable model to develop or too modelled or to design any software product. An another key factor regarding this prototype model is that it's also provide an adequate maintenance mechanisms, sometimes if user requirements have to be changed then after add on the new requirements, it's doesn't impact on other stages of this prototype model , generally we can say that it's also contains a strong adaptability features.

Step 4: At the end integrating all respective prototypes which deployed through each separated user requirements phase and deployed or to release a final software product. Now a one main scenario that if the user requirements have to be changed after releasing the prototype or generally we can say that if this software product require some maintenance then this newly required requirements add on to the requirements phase to select their respective categories after analysis their behaviour and again repeated all steps from steps 2 to end until the desired software product item have to be achieved.
So these are the newly proposed virtual multi variant interdisciplinary software project development model.

IV. RESULT ANALYSIS OF NEW IMPROVISED PROTOTYPE MODEL

- ✓ **Based on the characteristic of requirements :**

Requirements	**Waterfall**	**Prototype**	**Iterative Enhancement**	**Evolutionary Development**	**Spiral**	**RAID**	**New improvised Prototype**
Are requirements easily understandable and defined?	Yes	No	No	No	No	Yes	Yes
Do we change requirements quite often?	No	Yes	No	No	Yes	No	Yes
Can we define requirements early in the cycle?	Yes	No	Yes	Yes	No	Yes	No
Requirements are indicating a complex system to be built	No	Yes	Yes	Yes	Yes	No	Yes

- ✓ **Based on the status of development team:**

Requirements	**Waterfall**	**Prototype**	**Iterative Enhancement**	**Evolutionary Development**	**Spiral**	**RAID**	**New improvised Prototype**
Less experience on similar projects?	No	Yes	No	No	Yes	No	No
Less domain knowledge (new to the technology)	Yes	No	Yes	Yes	Yes	No	No
Less experience on tools to be used	Yes	No	No	No	Yes	No	No

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 8, Issue 8 , August 2021

Availability of training if required	No	No	Yes	Yes	No	Yes	No

✓ **Based on User Participation:**

Requirements	**Waterfall**	**Prototype**	**Iterative Enhancement**	**Evolutionary Development**	**Spiral**	**RAID**	**New improvised Prototype**
User involvement in all phases	No	Yes	No	No	No	Yes	Yes
Limited user participation	Yes	No	Yes	Yes	Yes	No	No
User have no previous experience of participation in similar projects	No	Yes	Yes	Yes	Yes	No	Yes
Users are experts of problem domain	No	Yes	Yes	Yes	No	Yes	Yes

✓ **Based on type of project with associated Risk:**

Requirements	**Waterfall**	**Prototype**	**Iterative Enhancement**	**Evolutionary Development**	**Spiral**	**RAID**	**New improvised Prototype**
Project is the enhancement of the exiting system	No	No	Yes	Yes	No	Yes	No
Funding is stable for the project	Yes	Yes	No	No	No	Yes	Yes
High reliability requirements	No	No	Yes	Yes	Yes	No	Yes
Tight project schedule	No	Yes	Yes	Yes	Yes	Yes	Yes
Use of reusable components	No	Yes	No	No	Yes	Yes	Yes
Are resources (time, money, people etc.) scare?	No	Yes	No	No	Yes	No	Yes

V. FUTURISTIC SCOPE

This model contains a multidisciplinary behaviour, adopt this feature, it's become a versatile model at all aspects of software development process and works reliably at any situation, which is either worst or best. This model explore a broad spectrum of each aspect of software development and focus on the minor to major process equivalently. This model uncovers most of the software development areas, which is either major or minor and Risk Analysis is the main factor of this model, exploring each aspects of Risk management activity. This model comes in a varieties of ways. This model also contains a newly proposed software development prototype model, which is much advanced than previously developed traditional approach based prototype model. This model is a

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 8, Issue 8 , August 2021

miles stone for complex and typical software development process. Most Real time systems and simulation based systems software, which are more complex in nature with respect to process are easily developed to follow the all necessary protocols and stages of these models at a very reliable manner. This model are very fast in execution , because time consuming features of software development model such as the many time user involvements, retesting, requirements changing , increasing software failures are totally avoided by these model. This model also possible user involvement but not in the maximized way. Many critical software project of medical sciences and scientific software are successfully build up by follow this model. This model proposed in a broad spectrum way, each stage fully uncovered with each aspects in a very detailed manner so the error rate and some missing features are less optimum. Maintenance are also the main key feature of this model , all types of maintenance are fully described and their uses at each steps , which make this model also reliable and versatile after the implementation phase that is post implementation and maintenance. This model have capability to deploy a sustainable satisfactory and technological enhanced reliable software product.

VI. CONCLUSION

There are various existing models which based on some traditional approaches or orthodox concepts, but here we developed or modelled a multidisciplinary software project development model along with an improvised detailed prototype model, generally these new improvised software development models covers broad spectrum of each stages and explore it into a very detailed conceptual method. Some software Project Development models or methodologies are such as SDLC and the Agile methodology are also a very versatile model for software developments, but here this model also contains some pros and cons, and we improvised a new software project development model which overcome the barriers of previously developed traditional approach based software systems. So this new improvised models have also a very fast time execution , It's gives a superior productivity and utilization , because in the Agile methodology there are multiple increments , and to deploy these increments to user and to obtained the desired and appropriate feedback from user. But it's a most time consuming tasks, so here we partitioned these all user requirements into following categories, such as: User Interface based requirements, Task oriented based requirements, Security Concern based requirements, and at the last platform based requirements. So these are the 4 partitioned and we prioritized any requirements to these 4 phases. We also proposed a prototype model in a broad spectrum. Now using these 4 requirements, feasibility analysis study conducted with separately at each stage. After feasibility analysis we draw out an idea for prototype model. In this phase of feasibility analysis, we estimated the cost of each respective requirements. Now a one main thing that how we estimated the cost of software project development. So here we use the Constructive Cost model (COCOMO), which provides much reliability for to find out the estimated cost of each chunks or Requirements, corresponding to Requirements engineering and all partitions. This model comes into the 3 stages, such as: Organic model, Semi-detached model, and embedded mode, using these modes, we can easily calculate the estimated cost of each 4 requirements. This model will perform a very reliable work in all platforms successfully such as some real time software systems, simulated software systems, which are very critical in the present view of situation, will also made up through these improvised models. Because we explore each stages in a broad spectrum way, and at the last we will integrate each prototype respective to all requirements partitioning categories and finally after to perform these all activities, finally deploy the official software project.

VII. ACKNOWLEDGEMENTS

We are very much grateful to all respected professors of DIET Rishikesh for their kind help, lasting encouragement, valuable suggestion throughout the entire period of our project work. We are highly indebted to his astute guidance, sincere support and boosting confidence to make this Research successful.
The acknowledgment will be incomplete if we fail to express our obligation and reverence to our family members and friends whose moral support is great factor in doing this research.

VIII. FUNDING

This study was not funded by any other profitable or non-profitable organisations.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 8, Issue 8 , August 2021

IX. CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

[1] Y.S.Zhang, X. Li, Software Development Models: a Survey, Journal of Computer Engineering and Applications, issue 3, pp.109-110, 2006.
[2] J.J.Yu, A Short Course in Software Engineering, Tsinghua University Press, 2015.
[3] Information on https: //www. cnblogs. com/wintersun/p/6828400. html, 2017.
[4] S.M. Zhu, Software Testing (Second Version). Posts &Telecom Press, 2016.
[5] X.H. Liu, Software Engineering and Project Management, Peking University Press, 2009.
[6] S.J. Wu, Improvement on Quality Testing Project of SI Software Based on Agile Scrum Mode, MASTER'S THESIS of Southwest Petroleum University, 2015.
[7] Y.M. Du, S.X. Li, Estimation Process Model for RUP Project, Journal of Computer Science, vol. 40, issue 6, pp.21-26, 2013.
[8] J.X. Xia, Z. Liu, X.B. Liu, Y.Song and J.J.Yuan, Incremental Story Iteration Model Based on Rapid Application Development, Journal of University of Shanghai For Science and Technology, issue 6, pp.578-583, 2014.
[9] R.M. Zhang, D. Yang and J. Li, Design of requirements negotiation tool based on WinWin theory, Journal of Computer Engineering and Applications, issue 1, pp.100-104, 2009.
[10] Michael Negnevitsky, Artificial Intelligence: A Guide to Intelligent Systems (Third Version), China Machine Press, 2012.
[11] X.Zou, Method of Construction: Modern Software Engineering, Posts & Telecom Press, 2018.
[12] Z. Luo, S. Q.Yuan, J. L.Yuan and L. Li, Software Engineering. Posts &Telecom Press, 2017.
[13] J.J. Yu, Research on designing and achievement on RUP improvement model of instructional software, Journal of E-education Research, issue 4, pp.76-81, 2012.
[14] Z.L. Gu, Research on development model of multi-media teaching software base on CSCW, Journal of Computer Engineering and Applications, issue 9, pp.1628-1630, 2006.
[15] S.J. Chen, Inverted A-Model for Stable Software Development, Journal of Software, vol. 37, issue 12, pp.07-11, 2016.
[16] J.J. Ma, Three Triple Iterative Model Based on Agile, Journal of Electronic Technology & Software Engineering, issue 6, pp.52-54, 2017.
[17] B.Lu, L.Liu, J.R. Li and L.M.Jiang, Artificial Intelligence and Its Application, Tsinghua University Press, 2017.
[18] Y.L. Si and W.H. Liu, Research of a New Software Development Cooperative Model, Journal of Microelectronics & Computer, issue 5, pp.73-76, 2012.
[19] D.Q. Xie, Double Iteration Model of Agile Software Development, Journal of Computer Applications and Software, vol. 29, issue 6, pp.176-178, 2012.
[20] Y. Wu, J.Y. Qian and Y. Liu, MDA-Based Component Employer Method Research and Implementation, Journal of Control & Automation, issue 27, pp.198-200, 2010.
[21] Z.Q. Lin, B. Xie and Y.Z. Zou, Intelligent Development Environment and Software Knowledge Graph, Journal of Computer Science and Technology, vol. 32, issue 2, pp.242-249, 2017.
[22] www.ijcsi.org/papers/7-5-94-101.pdf
[23] Ian Sommerville, "Software Engineering", Addison Wesley, 7th edition, 2004.
[24] United States, Navy Mathematical Computing Advisory Panel (29 June 1956),Symposium on advanced programming methods for digital computers, [Washington, D.C.]: Office of Naval Research, Dept. of the Navy, OCLC 10794738.
[25] Benington, Herbert D. (1 October 1983). "Production of Large Computer Programs"(PDF). IEEE Annals of the History of Computing (IEEE Educational Activities Department) 5(4): 350– 361. doi:10.1109/MAHC.1983.10102. Retrieved 2011-03- 21.
[26] Royce, Winston (1970), "Managing the Development of Large Software Systems"(PDF), Proceedings of IEEE WESCON 26 (August): 1–9.
[27] Larman, Craig (June 2003). "Iterative and Incremental Development: A Brief History" (PDF). Computer 36 (6): 47–56.doi:10.1109/MC.2003.1204375. ISSN 0018-9162.
[28] DOD-STD-2167 Defense Systems Software Development (04 JUN 1985) on everyspec.com.
[29] Rlewallen, "Software Development Life Cycle Models", 2005 ,http://codebeter.com.
[30] Karlm, "Software Lifecycle Models', KTH,2006 .
[31] Kevin Forsberg and Harold Mooz, "The Relationship of System Engineering to the Project Cycle," in Proceedings of the First Annual Symposium of National Council on System Engineering, October 1991: 57–6.
[32] Boehm B, "A Spiral Model of Software Development and Enhancement", ACM SIGSOFT Software Engineering Notes, ACM, 11(4):14-24, August 1986.