HYBRID MODEL FOR SOFTWARE DEVELOPMENT PROCESSES

Nabil Mohammed Ali Munassar* and A. Govardhan**

*PhD Student of Computer Science & Engineering, Jawaharlal Nehru Technological University, Kukatpally, Hyderabad- 500 085, Andhra Pradesh, India. Nabil_monaser@hotmail.com

**Professor of Computer Science & Engineering, Principal JNTUH of Engineering College, Jagityal, Karimnagar (Dt), Andhra Pradesh, India. govardhan_cse@yahoo.co.in

ABSTRACT

This research deals with a vital and important issue in computer world. It is concerned with the software management processes that examine the area of software development through the development models, which are known as software development life cycle. The main objective of this research is to design a development model that meets the needs of different systems and eliminates the defects presented in the previous development models. The present research proposes a model, “Hybrid Model” which combines the features of the five common development models: waterfall, iteration, spiral, v-shaped and extreme programming. The proposed model in this research has the advantages and some features of the previous models with some modification. Because of this, it avoids and overcomes many software problems that exist in the previous models. Thus, the new proposed model is an integrated model, which is relevant to most software programs and systems.

Keywords: Software Management Processes, Software Development Processes, Development Models, Software Development Life Cycle, Hybrid Model.

2. OBJECTIVE

1. Designing a proposed model that imitate the advantages of the previous different models found in software process management.
2. Applying the new proposed model to a number of projects to be sure of its adequacy to show its way of working.

3. THE MOTIVES TO CHOOSE RESEARCH TOPIC

There are a number of motives behind choosing the research topic, and they are as follows:
1. Software engineering plays a very important role in developing and building programs and they are considered fundamental elements in designing projects.
2. The extreme importance of this model in dealing with many projects of software engineering, and their relationship to develop various software.
3. Studying different models for the purpose of finding out points of strengths and weaknesses, advantages and disadvantages.
4. Constructing a model that includes or comprises more than one point of the strengths existing in the previous software engineering models.

4. THE IDEA OF THE PROPOSED MODEL "HYBRID MODEL"

Software process model is a simplified representation of a software process, presented from a specific perspective[1]. There are numbers of general models for software processes, like: Waterfall model, Evolutionary development, Formal systems development and Reuse-based development …etc. Software development models are collected to become one model as in figure (1), and these models are:
1. Waterfall model.
2. Iteration model.
3. V-shaped model.
4. Spiral model.
5. Extreme model.
5. THE PRIMARY HYBRID MODEL IN SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)

Figure (1) includes the main parts of systems development processes and the sequence mechanics of these parts or phases in a logical arrangement which insure the suitability of this model to different phases of development systems, either small, medium or large. The primary shape of Hybrid model includes the following processes:

1. Planning: Contains planning of necessary tasks to define the resources and timelines, and making plans for system development process and other information that are related to the project. [6]

2. Requirements: Three types of requirements can be determined in this phase:
   - Abstract functional requirements: determine the system's functions in idealistic methods.
   - System properties: define the non functional requirements for system.
   - Undesirable characteristics: determine the unacceptable system behavior.

   These requirements must define and determine the overall organizational objectives of the system.[7]

   System objectives:
   - Functional objectives.
   - Organizational objectives.

   In this phase, the systems engineer should understand the information range for software. In addition to that functions, behavior, and performance should be considered. At the end of this phase, software and system requirements should be documented and revised with the customer.[4]

3. Design: A Process that has many steps. It contains four important parts:
   - Data structuring.
   - Software building.
   - Data representation.
   - Processes details (algorithms).

   Requirements design process is transformed to representation for software which can be evaluated through good quality, before starting the following phase, i.e., "Implementation". Through this phase, the design is documented to become a part of software collection.[5]

4. Implementation: It transforms the design into formula which computer can read. Also, it structures the system components by using one of the programming languages according to its planning.[11]

5. Integration development: It connects the different components of a system into one community and subsequently forming one formal system. In this phase, the different parts should complete each other without any negative effect on the rest of system components.[2]

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.[1]

7. Testing: Testing process starts with the first phase of any system, namely "planning". It includes planning test, requirements collection test, design, representation and integration.[3]

8. Maintenance: Software undergoes some modifications after handing it over to customer. These modifications occur because of difficulties faced. Another reason may be software the necessity of adjusting to the changes in the external environment or customer's requests to do improvement in performance or function. [7]

9. Risk Analysis: This phase includes all development phases starting with planning processes and finishing at maintenance processes. It lists all expected risks and suggests all the necessary activities to reduce such risks. [8]

Figure 1: The Primary of Hybrid Model in SDLC

It is known that the process of developing medium and large systems starts with planning phase. In this phase, system requirements like human cadre, time, materials and costs are determined. Small projects, on the other hand, usually starts with designing or programming phase in which the plan is tested to find out whether it is appropriate to apply the project.
Moreover, this phase is used to assess project risks that are related to programming, human and material factors. This phase is followed by requirements collection and analysis. Through this phase, the requirements are gathered by contacting with customers, then analyzed based on customers’ decisions whether such requirements fulfill system objectives or not (taking into account the existing risks).

Designing phase comes after requirement phase. Here, the system is primarily designed on papers within algorithms and diagrams showing system progress. In this phase, the design is also tested to make sure of its appropriateness to the requirements.

System representation phase follows all the previous phases. This phase is an implementation for the design on computer device according to certain specifications shown in the plan.

In addition, the implemented programming parts are tested for the purpose of assuring their appropriate application.

Integration phase immediately follows system representation phase. This phase is important as all implemented programming parts are integrated and tested for two purposes. The first one is to ensure their being free of any faults and the second is to decide whether they fulfill all system requirements. This phase should be completely implemented before loading the system to customers.

The last phase is spreading and loading the system on customer’s devices. This phase is one of the most critical and important phases and should be considered according to plan and risks analysis. [1][7]

Hybrid model is characterized by its appropriateness to all small, medium and large systems and its flexibility to start with any phase either planning, requirements collection and analysis, designing or programming representation in accordance to system type and complexity.

6. DETAILS OF HYBRID MODEL IN SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)

The detailed figure of Hybrid model explains its operation mechanism regarding system development methods in its different kinds as shown in Figure (2). This figure explains the connection between the planning phases and the phase of analyzing the expected risks in the project. It also tests the plan on the basis of these risks. Furthermore, this figure shows the connection between phases of planning and risks analysis and those of requirements collection, design, implementation and finally the integration of the programming parts, system deploying and maintenance. The proposed Hybrid model has the following advantages:

- Advantages:
  1. Easy to understand and implement.
  2. Reinforces good habits: define-before-design, design-before-code.
  3. Identifies deliverables and milestones.
  4. Works well on mature products and weak teams.
  5. Simple and easy to use.
  6. Each phase has specific deliverables.
  7. Higher chance of success over the waterfall model or other due to the development of test plans early on during the life cycle.
  8. Works well for small projects where requirements are easily understood or complex projects.
  9. High amount of risk analysis.
  10. Good for large and mission-critical projects.
  11. Produces good team cohesion.
  12. Emphasizes final product.
  13. Iterative.
  14. Test based approach to requirements and quality assurance.

The model figure shows the connection between the various testing processes with the different development phases, except two phases which are, deployment and maintenance which have connection with the previous development phases and the risk analysis processes. Also, the design phase is divided into two phases: the high-level design which centers on the primary sides of system design like, charts, algorithms according to the kind of used design. The second kind is low-level design which focuses on the programming design for system and its technical aspects which serve the system design. Moreover, this model explains the possibility of starting with any phase of system development phases to include all different kinds of projects.

Figure 2: Details of Hybrid Model in SDLC.
7. COMPARISON BETWEEN THE HYBRID MODEL AND SPIRAL MODEL
Models can be compared as shown in Table (1) below.

<table>
<thead>
<tr>
<th>Hybrid Model</th>
<th>Spiral Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>This model works with small, medium and large projects.</td>
<td>A good model for larger and critical projects.</td>
</tr>
<tr>
<td>Identifies the point of end for each phase.</td>
<td>Frequent and overlapping phases.</td>
</tr>
<tr>
<td>Focuses on the planning phase and risk management.</td>
<td>Focusing on risk management.</td>
</tr>
<tr>
<td>Easy to understand and apply, especially with small and medium projects.</td>
<td>Needs to be experienced in its application.</td>
</tr>
<tr>
<td>Depends on the phases of risk analysis and the test to the success of the project.</td>
<td>Depends on the concept of repetition to produce more than a prototype.[8]</td>
</tr>
</tbody>
</table>

8. CONCLUSION AND SUGGESTIONS FOR FUTURE WORK
8.1. CONCLUSION
Based on the designed model, it has been concluded that:

1. The Hybrid model is dependent on the five development models: Waterfall, Spiral, Iteration, V-shaped and Extreme Programming Models.
2. The Hybrid model compromises the strengths of the five models mentioned above, but has the ability to deal with small, medium and large projects.
3. The Hybrid model is divided into two parts: primary and detailed, the first part deals with the main processes and the second part deals with how these processes work.

8.2. SUGGESTIONS FOR FUTURE WORK
1. A comparison between the Hybrid model with the other five development models in terms of cost and dealing with the risks according to specific criteria.
2. Applying the Hybrid model to a large number of different projects regarding requirements and needs.
3. Developing the Hybrid model to include the traditional methods and advanced software engineering such as the Object-Oriented System Development.

REFERENCES

Nabil Mohammed Ali Munassar: was born in Jeddah, Saudi Arabia in 1978. He studied Computer Science at University of Science and Technology, Yemen from 1997 to 2001. In 2001 he received the Bachelor degree. He studied Master of Information Technology at Arab Academic, Yemen, from 2004 to 2007. Now he Ph.D. Student 3rd year of CSE at Jawaharlal Nehru Technological University (JNTU), Hyderabad, A.P., India. He is working as Associate Professor in Computer Science & Engineering College in University Of Science and Technology, Yemen. His area of interest include Software Engineering, System Analysis and Design, Databases and Object Oriented Technologies.

Dr. A. Govardhan: received Ph.D. degree in Computer Science and Engineering from Jawaharlal Nehru Technological University in 2003, M.Tech. from Jawaharlal Nehru University in 1994 and B.E. from Osmania University in 1992. He is Working as a Principal of Jawaharlal Nehru Technological University, Jagityal. He has published around 108 papers in various national and international Journals/conferences. His research of interest includes Databases, Data Warehousing & Mining, Information Retrieval, Computer Networks, Image Processing, Software Engineering, Search Engines and Object Oriented Technologies.