# Hybrid Model for Software Development Life Cycle

Nabil Mohammed A. munassar[+] A. Govardhan[++]

## Abstract

**This research deals with a vital and important issue in the world of computers. It is concerned with the software management processes that examine software development through development models, which are known as the 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, the 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 existed in the previous models. Thus, the newly proposed model is an integrated model, which is relevant to most software programs and systems.**

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

## 1. Introduction

The existence of a number of software development models in software engineering, because of their different methods of use and various applications, has created different types of problems. These problems could only be realized and experienced by system engineers while developing software programs. These problems may lead to an increase in cost and effort during the development of these systems. One of these problems can be noticed in the fact that some of these development models do not include risk analysis and plans. Another problem is their being limited only to large projects. Therefore, this research aims to develop a model that can be applied to all projects (small, medium, and large) taking into account the factors of cost and effort, especially in small and medium projects.

## 2. Objectives

1. Designing a proposed model that imitates 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 and to show how it works.

[+] **Ph.D Student of Computer Science & Engineering, Jawahrlal Nehru Technological University, Kuktapally, Hyderabad- 500 085, Andhra Pradesh, India, Nabil_monaser@hotmail.com.**

[++]**Professor of Computer Science & Engineering, Principal JNTUH of Engineering College, Jagityal, Karimnagar (Dt), A.P, India. govardhan_cse@yahoo.co.in.**

## 3. The Motivation for Choosing this 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 software engineering projects, and its relationship to the development of various software.
3. Studying different models for the purpose of finding out their 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"

The 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: the 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 the System Development Life Cycle (SDLC)

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

1. **Planning**: Contains the planning of necessary tasks to define the resources and timelines, and making plans for the system development process and other information that is 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 the system.
   - Undesirable characteristics: determine which system behavior is unacceptable.

   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 the software. In addition to this, the 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).

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

4. **Implementation**: This transforms the design into formula which computers can read. Also, it structures the system components by using one of the programming languages according to its planning.[1]
5. **Integration development**: This connects the different components of a system into one community and subsequently forms one formal system. In this phase, the different parts should complete each other without any negative effect on the rest of the system components. [2]
6. **Deployment:** This means sending the system after complete to the customer for use and work in order to show the problems based on its use for the first time.[1]
7. **Testing**: The testing process starts with the first phase of any system, namely "planning'.' It includes the planning test, requirements collection test, design, representation and integration.[3]
8. **Maintenance**: Software undergoes some modifications after handing it over to the customer. These modifications occur because of the difficulties faced. Other reasons may be the necessity of adjusting to the changes in the external environment or to the customer's requests to make improvements 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]

It is known that the process of developing medium and large systems starts with the planning phase. In this phase, system requirements, like human cadre, time, materials and costs are determined. Small projects, on the other hand, usually start with the designing or programming phase, in which the plan is tested to find out whether it is appropriate to apply it to 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 the requirements of collection and analysis. Through this phase, the requirements are gathered by contacting customers, then analyzed based on customers' decisions whether such requirements fulfill the system objectives or not (taking into account the existing risks). The designing phase comes after the requirement phase. Here, the system is primarily designed in papers with algorithms and diagrams showing system progress. In this phase, the design is also tested to make sure of its appropriateness to the requirements. The system representation phase follows all the previous phases. This phase is an implementation for the design of computer devices according to certain specifications shown in the plan. In addition, the implemented programming parts are tested for the purpose of assuring their appropriate application. The integration phase immediately follows the 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 that they are 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 for 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]

The hybrid model is characterized by its appropriateness to all small, medium and large systems and its flexibility in this model, one can start with any phase, whether planning, requirements collection and analysis, designing, or programming representation in accordance with the system type and complexity.

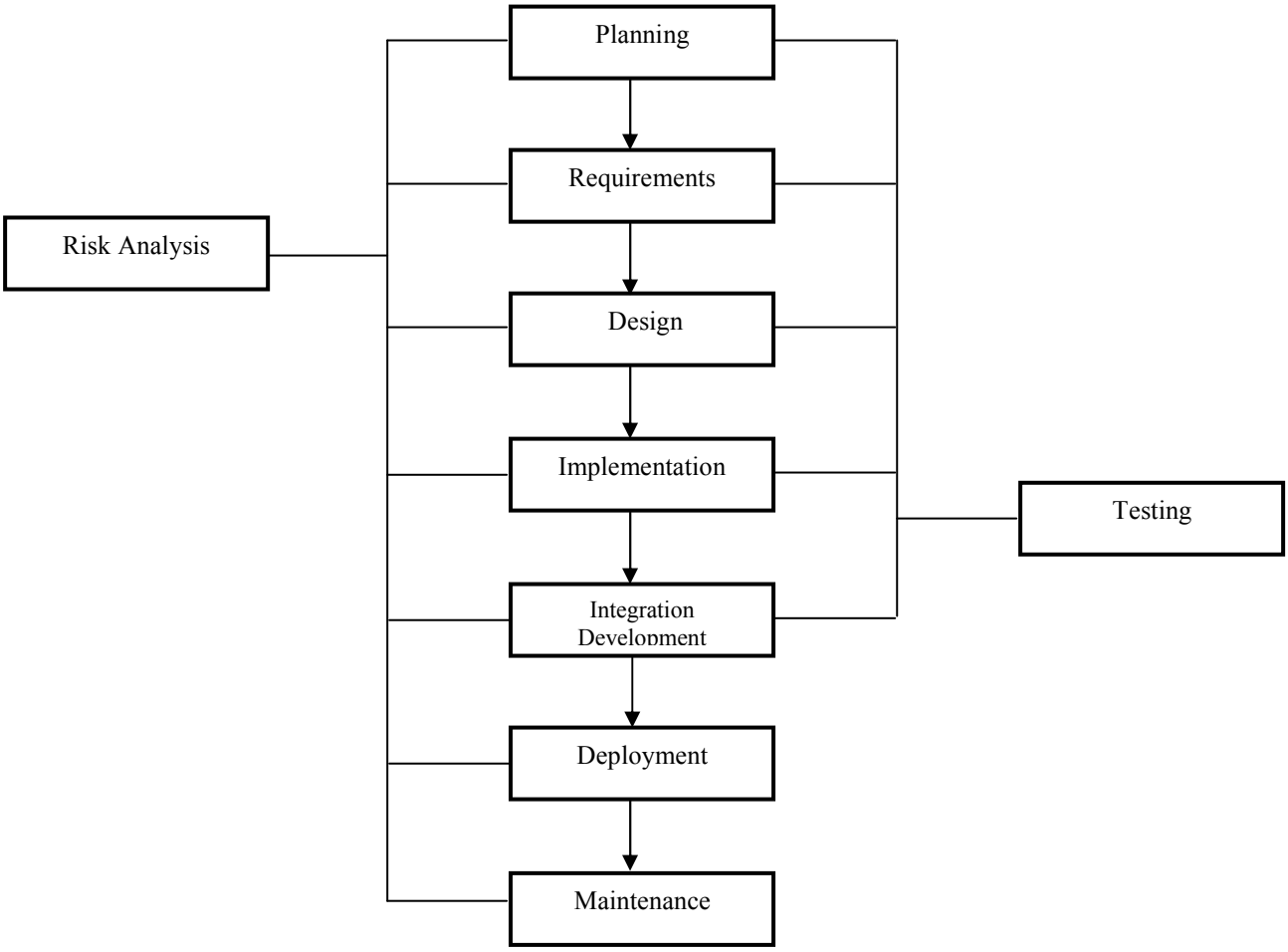


**Fig. (1): The Primary of Hybrid Model in SDLC**

## 6. Details of the Hybrid Model in the System Development Life Cycle

The detailed figure of the Hybrid model explains its operation mechanism regarding system development methods of 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 risk analysis and those of requirement 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 others 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 for 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 Figure 2 shows the connection between the various testing processes and the different development phases, except two phases, which are deployment and maintenance. There are connected to 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 design used. The second kind is low-level design, which focuses on the programming design for systems and their technical aspects, which serve the system design. Moreover, it is possible to start with ant phase of the system development phases in this model, and it includes all different kinds of projects.

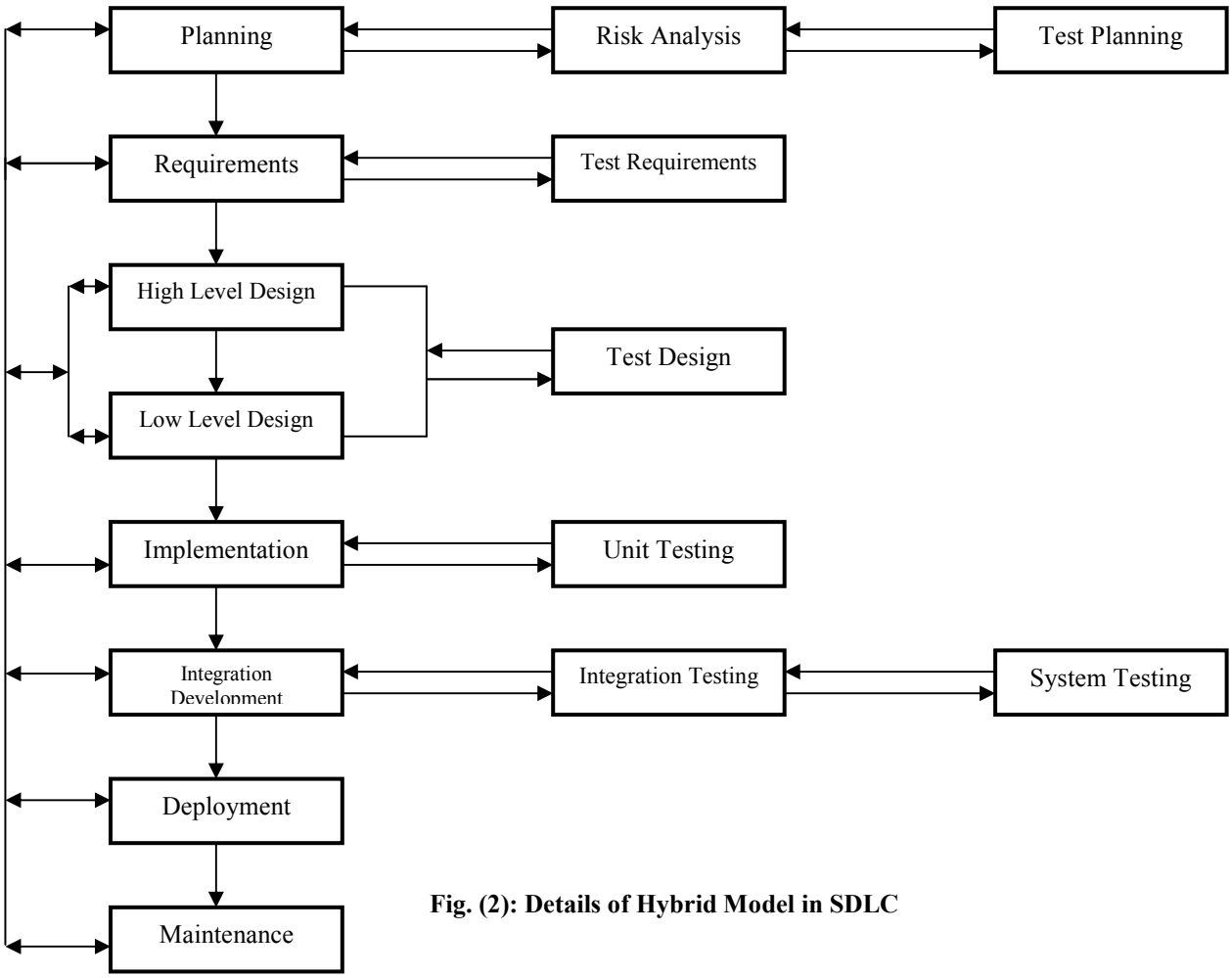


**Fig. (2): Details of Hybrid Model in SDLC**

# 7. Comparison Between the Hybrid Model and Spiral Model

The models can be compared as shown in Table (1) below.

**Table(1): A Comparison between the Hybrid Model and Spiral Model.**

| Hybrid Model | Spiral Model |
|---|---|
| This model works with small, medium and large projects. | A good model for larger and critical projects. |
| Identifies the end point for each phase. | Frequent and overlapping phases. |
| Focuses on the planning phase and risk management. | Focuses on risk management. |
| Easy to understand and apply, especially with small and medium projects. | Experience is required for its application. |
| Depends on the phases of risk analysis and tests to reveal the success of the project. | Depends on the concept of repetition to produce more than a prototype.[8] |

# 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 is comprised of 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 in 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 traditional methods and advanced software engineering such as Object-Oriented System Development.

## 9. References

[1]. Lan Sommerville, "Software Engineering", Addison Wesley, 7th edition, 2004.

[2]. Karlm, "Software Lifecycle Models", KTH,2006 .

[3]. Lifecycle Models , National Instruments Corporation , 2006.

[4]. Steve Easterbrook, "Software Lifecycles", University of Toronto Department of Computer Science, 2001.

[5]. Rlewallen, "Software Development Life Cycle Models" , 2005.

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

[7]. Roger S Pressman, "Software Engineering: A Practitioner's Approach", 7th edition, McGrawHill, 2009.

[8]. B. W. Boehm, " Classics in Software Engineering", Yourdon Press -Upper Saddle River, NJ, USA, Pages: 323 – 361, ISBN:0-917072-14-6, 1979.