

# p-model: An internal mechanism model for SDLC in Software Engineering

Saima Amber  
Department of CS&IT  
Jinnah University for women,  
Karachi, Pakistan

Saira Begum  
Department of CS&IT  
Jinnah University for Women,  
Karachi, Pakistan.

Saboohi Naeem Ahmed  
Department of CS&IT  
Jinnah University for Women,  
Karachi, Pakistan.

**Abstract:** Software development Life cycle provides a generic structure of the procedure that can be followed by software developed team to achieve their goals. With the help of software process model victorious activities of SDLC are addressed. A right software process helps software development organizations in various faces of saving in terms of time, efforts and money. Each Process model has some activities to which different stages of SDLC are performed.

In this research three different and traditional process models water fall, Prototype and Spiral model are analyzed. This research finds different sources of risks that become a major hindrance of the success of the project's completion and production of the quality products. This research proposes a new model which encompasses different activities which are useful to reduce sources of risks.

## I. INTRODUCTION

Software development is the set of interrelated activities which encompass requirement gathering, analysis, design, and implementation. These activities are interdependent to each other. Output of the previous stage is the input of second stage [1]. In order to develop high quality software system, Software development organizations meet both the customer and the real world requirement by considering all coupled portion of pros and cons of requirements, designing, software testing and implementation. The life cycle focuses on the product, defining the state through which a product passes from when it starts to be built to when software enters into operations and finally retired [2].

A software process model is a conceptual depiction of the architecture and design of the software

development process. In software development, process models are implemented to control diverse concerns associated with cost, time, and quality and changing in client's needs. The primary functions of software process model are to determine the order of stages involve in software development and to establish the transition criteria for progressing from one stage to the next. Software process model provide guidance on the order in which a project should carry out its major tasks. Many software projects have come to grief because they pursue their various development phases in the wrong ordering. Engineering processes are composed of many activities, notably the following:

- Requirements gathering
- Designing
- Coding
- Testing
- Implementation

Each software development Company adopts the best-suited process model, which facilitates the software development process and boosts the productivity of its team members[5]. This research focuses the strength and weakness of water fall model, Prototyping and Spiral model and proposed a model which includes a set of different activities to control risk created factors.

### 1.1 PROBLEM DEFINITION

Software projects fail when do not meet the criteria for success. Most of the IT projects run over budget or are terminated prematurely and those that reach completion often fall far short of meeting user expectations and business performance goals[2]. According to the Survey conducted by Standish Group,

- 84 % of all software projects do not finish on time and within budget.
- 8000 projects in US in 1995
- More than 30 % of all projects were cancelled
- 189 % over budget

Following table is extracted from [2], which gives the result from the survey of different organization.

Name	Reported year	Failures % scheduled	Budget %	Requirement %
TCS (Tata consultancy services)	2007	62%	47-49%	41%
Avande research report	2007	66%	49%	51%
ESSU(Euro pisan service strategy unit)	2007	33%	57%	33% - 45%

TABLE 1: FAILING OF SOFTWARE PROJECTS

The common sources of these situations are as given below

- ambiguous and unclear Requirements
- The clients ask for new features, just before the end of the project, and unclear requirements.
- Uncertainties throughout the development project
- Software itself is awfully complex.

All Process models share basic properties. They all consist of a sequence of phases or steps that must be followed and completed by requirement engineers, system designers and developers in order to attain some results and deliver a final product. For instance, the Waterfall model, one of the earliest SDLC models, comprises five consecutive phases and they are respect

tively: Business analysis, design, implementation, testing, and maintenance [6]. When

an organization is going to develop a product, it uses a process model to address all stages of SDLC.

This research will focus different sources of risks, which are produced by mishandling of different stages of Water fall, Prototyping and Spiral model. These models are analyzed on the basis of their strength and weakness.

### 1.2 OBJECTIVES OF THE STUDY

The main objective of this research is to find out such activities which diminish sources of risks that are related to the different stages of software development process models. Three software process models are analyzed that have different in nature therefore these models are used in different nature of projects according to the size.

In this research a questionnaires is prepared and filled by the requirement engineers and software engineers, working in different software houses. On the basis of questionnaire different activities are identified which are useful to reduce risk created factors. These activities are composed in a form of an internal mechanism model. The results of this survey will be discussed in conclusion.

### 1.3 PROBLEM RESOLUTION:

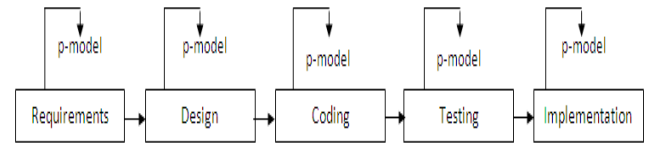


Figure a: p-model working with water fall

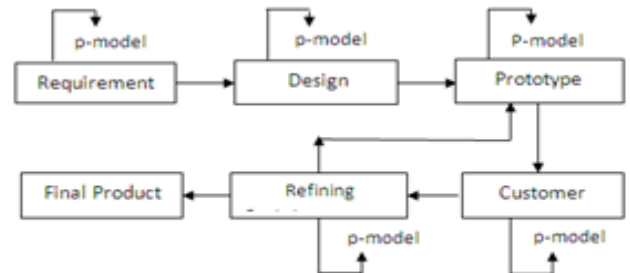


Figure b : p-model working with prototyping

### 2.0 LITERATURE REVIEW

Software process models are used to organizing a structured set of activities to develop software systems. A Programming process model is an abstract representation to describe the process from a particular perspective. There are numbers of general models for software processes, like: Waterfall model, Evolutionary development, Formal systems development and Reuse based development, etc.

It presents a description of a process from some particular perspective as[3]:

Specification – defining what the system should do;  
 Design and implementation – defining the organization of the system and implementing the system;  
 Validation – checking that it does what the customer wants;

Evolution – changing the system in response to changing customer needs.

Software development is address by means of Software process and softwarerisks are involved in every phase of software development life cycle. Software projects have a high probability of failure therefore effective software development means dealing with risks adequately. Risk can be defined as the probability of suffering harm or loss. There are two things involved in the risks i.e. uncertainty and Loss. Some of the risks are Dynamic and some are Static. In Dynamic there might be some profit or loss associated with risk. And in Static only Loss is associated with the risk[4].

### 3.0 PROCESS MODELS

#### 3.1 Waterfall model

Waterfall model is a sequential process of software development. In this model the production cycle progresses sequentially, from one stage to the other. This mode includedifferent phases of software development in which requirement gathering, analysis, design, coding, testing & debugging, installation and maintenance are presented. In this sequentially structured approach, the development team goes ahead to the next stage of development, only after the previous stage is fully accomplished. The belief that drives this kind of software development model is that considerable time spent in initial design effort corrects bugs in advance. Software development companies, adopting this model, spend a considerable amount of time in each stage of development, till all doubts are cleared and all requirements are met.



Figure c: Waterfall model

##### 3.1.1 When to use Water fall model:

According to [7],this model works best when

1. The Software requirements are well understood
2. The nature of the software development involves contractual agreements.

#### 3.2 Prototype model:

A prototyping is a development approach to create a working model of software from the beginning with limited functionality. It acts as a sample to test the development process. By using this sample, developers try to understand user requirements and build a better final product. User is involved throughout the development process, which increases the likelihood of user acceptance of the final implementation [5].

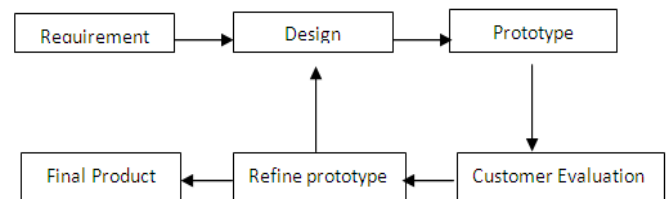


Figure d : Prototype model

##### 3.2.1 When to use Prototype model

- Prototype model should be used when the desired system needs to have a lot of interaction with the end users.
- Online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model.
- They are excellent for designing good human computer interface systems.

#### 3.3 SPIRAL MODEL

The spiral model is risk driven process model and used to guide multi stakeholder concurrent engineering of software system. It has two unique features. one cyclic approach for incrementally growing a system's degree of definition and implementation while decreasing its degree of risk .the other is a set of anchor point milestone for ensuring stakeholder commitment of feasible and mutually satisfactory system solution .

Spiral is a meta-model as it comprises of other models of SDLC. Waterfall and Prototype approaches are used in it. To address SDLC's stages

systematically, it works over the loops (waterfall approach is used) and at the same time a prototype is made and show it to user after completion of various phase (working of prototype model). These are the ways to reduce various risks

### 3.3.1 When to use spiral model

1. Project is complex.
2. Risk factors are too high.

Process Models	Strength	Weakness
<b>Water fall</b>	Simple and easy to use Easy to manage due to the rigidity of the model. One phase is completed at one time. Each phase has specific deliverables.	When an application is in the testing stage, it is very difficult to go back and change something that was not well. High amounts of risk and uncertainty.
<b>Prototyping</b>	User is involved at each iteration. Useful in small sized projects. Development time is reduced i.e development cost is reduced. Quantifiable user feedback. Results in highly user satisfactions.	Too much user involvement. Not suitable for the large sized project. Project management difficulties. Increase pressure to the developers
<b>Spiral</b>	Risk management is performed. Changes that are introduced later in any stage, can be handled. As the project goes on, the project estimates in terms of schedule and cost becomes more and more realistic. This will result in the completion of a loop in spiral. Suitable for the projects where risk factors is high.	Cost involved in this model is usually high. Expertise is required to evaluate and review the project on timely basis.

TABLE 2 : STRENGTH AND WEAKNESS OF THREE MODELS

### 3.4 P- model: An internal mechanism

It is a set of activities which are performed as an internal mechanism with process models to reduce the risk creating factors associated with each model. This model adds functionalities with well-known process models and can be used to reduce the weakness of models. This model represents an internal mechanism of software development as the following table[3] represents.

Research shows that SDLC mainly address five major stages of requirement, designing, coding, testing and implementation. If error are not fixed in one stage, it can be propagated to the later stage. Cost of reworking and fixing is getting too high and in many cases in time or in budget project delivery is used to getting failure. The main cause of this failure is to avoiding different risk, related to the different stages of software development methodology or process. Neglecting of these risks has profound impact on the development of software. One of the worse results is the cancellation of project.

Risk is defined as a combination of two factors: probability of malfunctioning (failure) and the consequence of malfunctioning (severity)[9].

$$\text{Risk} = \text{Probability of undesired event} * \text{Severity of undesired event}$$

Most of the risk assessment methodologies are used to estimate in the later stages of the software life cycle, typically from design models or code. As a result these methodologies can identify risks but have limited capability in preventing these risks from occurring.

Software requirement risk addresses the possibility of suffering a loss of any functional or non-functional requirement of the software system. It is a well-known fact that it is more feasible to make changes to the software system under development in the early stages of the software development cycle[11]. For the detailed calculation of risk factors, [14] can be concerned.

The main focus of proposed model is to put a check on risk factors therefore it include risk management with waterfall model as table [3] shows. Although this parameter is not a part of water fall model. The table described below shows the activities involved in each model

Process model	foundations of risks	Activities of p-model	Impact of P Model
Waterfall	Users requirements change continuously Stages are completely separated. Verification is performed only testing stage.	Strong Review techniques should be used and performed at each stage. Internally, verification is the part of requirement gathering stage. Internally risk management should be applied at each stage before completing the stage. Ensure that all problems, bugs, and risks related to specific stage are discovered in that stage. These should be fixed and don't transfer into later stages. Activities related to QA should be performed internally.	By using Strong review you can overcome on the risk. Review meeting with internal management is also providing the way to overcome on risk..
Prototyping	Highly user's involvement Not right	User/customer representative	By having knowledgeable

	customer representative Large project 's sizing Large team sizing Users requirements change continuously change in project's scope.	ve should have adequate knowledge about system to be developed. Ensure that project size and team size is small. Internally risk management should be applied at each stage . Ensure that all problems, bugs, and risks related to specific stage are discovered in that stage. These should be fixed and don't transfer into later stages. Agreed set of SRS should be sign off.	customer and short project size, p-model can be a successful process model specially by applying bug fixing techniques in each stage.
Spiral	It uses prototyping as risk reduction therefore dependencies on human factor are too high. Detailed risk management [8]. Risk related	User/customer representative should have adequate knowledge about system to be developed Strong Review	The way a customer view/understands his problems is different from the point of an analyst.

	to cost and schedule increases as spiral size increases.	techniques should be used and performed at each stage. Ensure that all problems, bugs, and risks related to specific stage are discovered in that stage. These should be fixed and don't transfer into later stages. Activities related to QA should be performed internally.	Analyst having experience and domain knowledge can better understand or analyze a specific problem. User can only help to confirm whether it is going in right or wrong direction. QA testing along with risk management improve significant results. QA should perform integration testing of different modules of the system. P-model is giving pre-risk information. The cost in spiral model is problematic parameter for software engineer. P- model
--	--	---	---

			calculate risk before going in the next SDLC phase.
--	--	--	---

TABLE[3]: IMPACT OF P-MODEL ON THREE PROCESS MODELS

4. CONCLUSION AND FUTURE WORK

The main focus of this research is to reduce risk creating factors which are associated with all process models (although discuss three models). In this research, we prepare a questionnaire and performed a survey from the different software houses of Karachi. This survey helped us to understand the working for software development. On the basis of this survey, P-model is derived. P-model works with process models and perform different activities which reduce risk factors and make it possible to deliver project within time and in budget.

According to the survey,55 % of the software developers agree that spiral model is expensive in terms of costing and timing.73% agree that water fall will provide easiest way to develop a software if p-model work together. In future p-model will be evaluated and associated with other process models.

REFERENCES:

[1]. waterfallvs v-model vs agile: a comparative study on sdlc by S.Balaji Computer Science Dept., Gulf College Muscat, Sultanate of Oman. Dr.M.SundararajanMurugaiyan Computer Science Dept., Government Arts College Chennai, TN, India.

[2].Software Process Models and Analysis onFailure of Software Development Projects RupinderKaur, Dr. JyotsnaSengupta

[3].A Comparison Between Five Models Of Software Engineering Nabil Mohammed Ali Munassar1 and A. Govardhan2 [4]. Risk Analysis of Various Phases of SoftwareDevelopment Models Tahir Abdullah Department of Computer Science, University of Agriculture Faisalabad, Pakistan E-mail: tahir\_wains@hotmail.com

[5].A Comparative Analysis of Different types of Models in Software Development Life Cycle Ms. Shikhamaheshwari1Research Scholar in Dept.of C.S.E S.V.I.T.S, Indore(M.P)



[6]. A Simulation Model for the Waterfall Software Development Life Cycle Youssef Bassil; LACSC – Lebanese Association for Computational Sciences

[7].Spiral Model - An Improvement over Waterfall Model ChetanaSamal University of Houston- Clear Lake Chetana\_samal@hotmail.com

[8].A Review of Risk Management in Different Software Development Methodologies HaneenHijaziHashemite University Zarqa, Jordan; ThairKhdour,AlBalqa Applied University Salt, Jordan; AbdulsalamAlarabeyyat Al Balqa Applied University Salt, Jordan

[9]. Software Requirements risk assessment Using UML by K.Appukkutty, HanyH.Ammar, Katerina Goseva Popstajanova Lane Department of Computer Science, West virgina University Morgantown

[10].Process Models in Software Engineering Walt Scacchi, Institute for Software Research, University of California, Irvine February 2001

Revised Version, May 2001, October 2001 Final Version to appear in, J.J. Marciniak (ed.), Encyclopedia of Software Engineering, 2nd Edition, John Wiley and Sons, Inc, New York, December 2001.

[11]. A methodology for measuring the risk associated with a Software Requirement Specification by DrTrevorMoores.

[12]Design Paradigm and Risk Assessment of HybridRe-engineering with an approach for development of Re-engineering Metrics Sandhya Tarar<sup>1</sup>, Dr. Ela Kumar<sup>2</sup> Research Associate<sup>1</sup>, Professor and Dean<sup>2</sup> School of ICT, Gautam Buddha University, Greater Noida, India

[13]. Reducing Software Failures: Addressing the Ethical Risks of the Software Development LifecycleBy Don Gotterbarn

[14]. Determination of Risk during Requirement Engineering Process by Saima Amber, Narmeen Shawoo Bawany and Saira Begum was included in Journal of Emerging Trends in Computing