A Review on Spiral Model with Symphony .NET

Sonia Thind  
M. Tech Student  
Department of Computer Science & Engineering  
UIET, KUK, India

Karambir  
Assistant Professor  
Department of Computer Science & Engineering  
UIET, KUK, India

Abstract

In a software industry, many Software development life cycle (SDLC) models are used with a sequence of activities for system designing and maintaining system information. These models are used for the development of large scale software system like waterfall, spiral, prototype and rapid application development model. An empirical study was conducted in 2012 that proposed the many factors for the selection of any software life cycle model. Team size was considered as one of the most important factor in this study. The inter arrival time of software projects in a software industry is unsystematic and allocating the number of workers and resources to these projects sometimes results in most important deficiencies like late delivery of the project, budget overruns and disappointed client. Therefore this review will purpose a model, for the Spiral development process with the use of a simulating tool (Symphony.NET). This model supports in make best use of the development processes by keeping all employees busy, all the time for the software development. For future work, other development models like Prototype, Incremental etc. will be simulated, in order to decide which model is more effective in performance and timely delivery in the software industry.

Keywords: Simphony.NET, Software Engineering, SDLC models, Spiral model, Simulation and modelling

I. INTRODUCTION

By using different development methodologies computer software and information systems has been developed. Software development methodology is a framework of activities that are used to plan, construct and control the procedure of development of an information system [1]. This approach is known as Software Development Life Cycle (SDLC). In effect, these SDLC models has been studied and investigated by many researchers and practitioners in the world, and many models have been proposed, each with their own acknowledged strength and weakness. Models like Waterfall, spiral, incremental, rational unified process (RUP), rapid application development (RAD), agile software development, and rapid prototyping are few to mention as successful models. In order to attain some results and deliver a final product a sequence of phases or steps of these models must be followed and completed by system designers and developers. However, it is a dilemma and confusion in assigning the appropriate number of resources for every single phase of the SDLC. Accomplishing full productivity with best resources like workers, number of expenses and time is a difficult job for project managers and directors in software industry. In that case, in order to complete a specific task or phase of a project in the software industry it is necessary to find the optimal number of resources.

In order to fulfil a certain project of a certain scale simulation for the SDLC is needed. Simulation helps in estimating the appropriate number of resources. Relatedly, a computer simulation is a computer program that tries to simulate an abstract model of a particular system. In practice, simulations can be employed to discover the behaviour, to estimate the outcome, and to analyse the operation of systems [2].

Therefore, this paper proposes a review on a model, for the Spiral development process with the use of a simulating tool (Symphony.NET) [3]. This model supports in make best use of the development processes by keeping all employees busy all the time. Different stakeholders which are involved throughout the complete development process are simulated by using this model. In order to design and develop they include the software solutions such as the employees like programmers and designers; the different spiral phases and activities. This simulation model measures the utilization of various resources during every task and phase.

II. SPIRAL MODEL

The Spiral model proposed by Boehm [4] is an evolutionary process model that includes the iterative activities of prototyping, with the organized and controlled characteristics of the waterfall process model. This model also includes the potential of Rapid Application Development (RAD) model, which helps in the development of increasingly, more complete-versions of the software project. The risk-driven nature of the spiral model is used to direct the stakeholders in the Software industry. This model develops the software in a sequence of evolutionary iterations. Throughout the initial iterations, the release of the product might be a prototype or a paper model. But, during advanced iterations, increasingly more complete-versions of the software projects are produced.
As shown in fig. 2.1, this model is divided into a set of frame-work activities. Each segment of the spiral path is represented by these activities. As the process of the model begins, the software team performs tasks that are implied by the circuit around the Spiral in a clockwise direction. This model is a realistic approach for the development of the large scale projects. With the progress of the software evolution process, the customer & the developer better react & understand to risks at each level of the Spiral process.

III. MOTIVATION & PROBLEM STATEMENT

A survey [5] conducted in 2012; in Indian software industry inspired us. This survey was focused on many empirical factors which has the capability of influencing the software project industry. These factors play an important role in the selection of software process model for the particular software product. Many types of software projects (small scale, medium scale and large scale) per day arrive in a software industry; therefore using the optimal resources in order to achieve the maximum productivity is the goal of any software development team. The inter arrival time of software projects in a software industry is unsystematic and allocating the number of workers and resources to these projects sometimes results in most important deficiencies like late delivery of the project, budget overruns and disappointed client[6].

<table>
<thead>
<tr>
<th>S.no.</th>
<th>Factor Name</th>
<th>S.no.</th>
<th>Factor Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Duration of project</td>
<td>9.</td>
<td>Required reliability level</td>
</tr>
<tr>
<td>2.</td>
<td>Nature of project</td>
<td>10.</td>
<td>Available technology and tools</td>
</tr>
<tr>
<td>3.</td>
<td>Size of project</td>
<td>11.</td>
<td>Application area understanding level</td>
</tr>
<tr>
<td>4.</td>
<td>Complexity of project</td>
<td>12.</td>
<td>User requirement understanding level</td>
</tr>
<tr>
<td>5.</td>
<td>Project team size</td>
<td>13.</td>
<td>Type and level of expected risks</td>
</tr>
<tr>
<td>6.</td>
<td>Customer involvement in project</td>
<td>14.</td>
<td>Experience of the developer</td>
</tr>
<tr>
<td>7.</td>
<td>Man – machine interaction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>Product versions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In practice, there are many software development problems and shortcomings that are faced by the developers regularly. These problems results in delay of project, cost overruns and dissatisfied the clients. A detailed investigation [7] showed that many projects do not deliver in budgets, at proper time and as expected. The reasons behind that is the improper and unintelligently assigning of the resources to the activities of the SDLC model. For this reason, some phases of the SDLC model may be delayed, while other may stay idle. This produces a bottleneck between the arrival and the delivery of the software projects, which causes following problems:

- Project budget overruns
- Late delivery of the project
- Dissatisfied clients

IV. THE SPIRAL MODEL WITH SIMPHONY.NET

This simulation model [8] can be implemented by using the Spiral SDLC model phases with a simulator called Simphony.NET. Simphony.NET consists of an adequate working frame work of scenarios [9] and a foundation library to control, manage, and create the simulation entities. The Spiral model can be developed using Simphony.NET by using many modelling elements like...
task, resource, queue, probability branch, release, counter and capture. This simulation model will be useful in keeping all employees busy all the time during the simulation of the spiral model development phases.

Generally, the simulation process with Simphony.NET tool consists of the following steps:
1) First of all, run the simulation model, and then examine the results,
2) Find out if there are any changes required, to be made to the result of the simulation model,
3) Repeat the steps as much as it takes to reach the optimal results.

Technically, the simulation process of the Spiral model consists of the following steps:
1) First of all, divide the Spiral model into different phases,
2) Then understand the concept that lies behind every phase,
3) Define the tasks, entities, work flow and the resources of different phases,
4) Simulate every phase of the model and then record the results,
5) Integrate the whole phases of the model together, simulate the whole system, and then again record results.

In this model a particular resource with a file can be binded by the capture element to a particular task. For this model it is necessary to assume that each phase upon completion will be subjected to some probability of having an error. There are many probabilistic branch modelling element exist in the simulator, whose purpose is to simulate the model’s error probability that a Spiral model development phase might exhibit after completion. The software projects are divided into three types: small scale projects, medium scale projects and large scale projects. For small scale projects, one core of the probabilistic branch will denote 10% probability of error and another core of the probabilistic branch will denote 90% probability that the project will not show any of error. For medium scale projects, one core of the probabilistic branch will denote 20% probability of error and another core of the probabilistic branch will denote 80% probability that the project will not show any of error. Similarly, for large scale projects, one core of the probabilistic branch will denote 30% probability of error and another core of the probabilistic branch will denote 70% probability that the project will not show any of error after the completion of every phase of the model.

These probabilistic branches will simulate the recursive property of the spiral model to loop over the previous task, if an error will be detected in the present task. Moreover, there exists another probabilistic branch element having three core elements. This branch will exists at the very beginning of every software project development cycle. First core branch of the probabilistic branch will denote 0.7 probabilities i.e. 70% of the incoming software projects will be of small scale, second core branch will denote 0.25 probabilities i.e. 25% of the incoming projects will be of medium scale and the third core branch will denote the 0.05 probability, i.e. 5% of the incoming projects will be of large scale.

The spiral model using symphony.NET tool will look like the fig. 2 given below. This spiral simulation model starts with an entity element. This entity element sets the number of incoming software projects. A counter element in the starting of the simulation model counts the number of projects being received and ends with another counter element that counts the number of projects being delivered. There are total 50 incoming projects arrives in the software industry, out of which there are 35 small scale projects, 10 medium scale projects and 5 large scale projects received in a software industry. These projects arrive randomly and the inter-arrival time between the projects is assumed with a triangular distribution. The lower limit is 30 days, upper limit is 40 days and the mode is of 35 days.

After running the Spiral simulation model many times, the result will be obtained. The result of the model will show that the total no. of projects received is equal to the total number of projects delivered. In this way, this simulation model will help in determine the optimal number of resources that are required to maximize the productivity in a software industry.

![Fig. 2: Simulation model for the Spiral SDLC](image-url)
V. CONCLUSION

This review will propose a model for the Spiral development process with the use of a simulating tool (Symphony.NET). In this model all the phases of the spiral model can be simulated. The model will support in make best use of the development processes by keeping all employees busy all the time. With the use of this model, the number of optimal resources can be easily and accurately calculated. As future work, other SDLC models like Prototype, Incremental etc. will be simulated, in order to determine which model is more effective in performance and timely delivery in the software industry.

REFERENCES