CHAPTER 5

PERFORMANCE REQUIREMENTS REFINEMENT MODEL

This chapter is about the layered model which is proposed for the purpose of refining the performance requirements. This chapter includes the sub sections like, importance of the layered model, performance parameters, performance factors, proposed layered model description and finally summary.

In software development, specifying requirements completely and consistently is most important task which in turn plays an important role in success or failure of the software product. Requirements specification includes both functional and nonfunctional requirements. During the requirements analysis, most of the time, people will focus more on functional requirements compared to nonfunctional requirements. Performance requirements are the most important nonfunctional requirements to consider in software development. Performance requirements have to be specified completely and consistently in requirements specification. But in most of the development processes, performance requirements are not specified completely and consistently at requirements engineering level due to ambiguity with the information available [124]. For fast development processes like agile software process, it is often difficult to specify performance requirements completely at early stages of software development process [125,126]. Once the product is developed and tested if any performance issues arise these issues must be resolved and see that these problems cannot be repeated for next releases. Improper specification of performance requirements always causes for poor performance of the software product. So there is a great need of refining the performance
requirements. For this purpose, a seven layered model is proposed in this dissertation which uses stakeholder feedback to refine the performance requirements. This layered model includes seven layers. These are feedback report, performance failures, performance requirements and parameters, goals and sub goals, stakeholders, performance objectives and refined performance requirements. This layered model is shown in figure 5.1.

5.1 Importance of Layered Models

Layered models are used for different purposes in different contexts. For example, OSI model [127] and TCP/IP layered model [128] are used in computer networking. Software engineering was explained as layered technology with different layers’ for the flexibility of development. There are many advantages of using layered models.

Layered approach allows the developers to use the information of one layer in another layer. Continuous assessment is possible with layered models, which in turn reduces the probability of occurring errors in development. Layered approach is an economical way of developing and implementing any system be they small, simple, large or complex makes no difference. Layering of processes, procedures and communications functions gives developers the freedom to concentrate on a specific layer or specific functions within that layer. Layering allows for a more precise identification and delineation of task, process and methodology. This permits a clearer definition of what needs to be done, where it needs to be done, when it needs to be done, how it needs to be done and what or who will do it. Hence, in this thesis a layered model is proposed for performance requirements refinement. This is a generic model suits for any software application development.
Figure 5.1: General Architecture of the seven layered model for performance requirements refinement
5.2. Performance Requirements Parameters

The proposed performance requirements refinement model suits for any software application to refine the requirements. The performance parameters considered are given below.

*Response time:* It is the time required to complete an operation or action. For example, in online railway reservation system the “book ticket” activity has to be completed within ten seconds.

*Resource utilization:* It is the rate or quantity of resource consumption, such as CPU utilization rate or memory usage, including main and secondary memory. For example, the web server process shall not consume more than 1GB of the main memory.

*Throughput:* It is the amount of work accomplished in a certain amount of time period. For example, the server shall be capable of process 1,000 files per hour.

5.3. Model Description

The proposed performance requirement refinement model includes seven layers. Each layer includes some activities. The description of the model is given below

The seven layers are

1. Feedback report
2. Performance failures
3. Performance requirements and parameters
4. Goals & Sub goals
5. Stakeholders
6. Performance objectives
7. Performance requirements
1. Feedback Report

This is the first layer in the model. This layer explains about feedback report. The feedback report is the report given by the customers after using the product. During the iterative software development process, feedback will be collected at the end of each iteration. Based on this feedback collected from different people the necessary changes will be made to the iteration to achieve expected results. So it is always important to consider the feedback in software development. Once the product is developed and tested then the first version will be released. Then it will be given to different stakeholders for review which includes even customers and end users.

All these stakeholders will go through the product, they will try to use different services provided by the system. After using the different services of the product they will provide their feedback about the services and the complete product. This feedback provides valuable information for developers to know about the product and how it is working. This feedback may contain positive information or negative information about the product. Feedback reports are most important aspects in iterative development. In the first layer feedback report will be collected from different people. The feedback given by the people may include the information about the complete system or individual behaviors. The feedback report is an important document which helps the developers not only to know the performance of the system but also it guides the developers in refining the performance requirements. So the feedback report plays an important role in performance requirements refinement. In this layer feedback report will be selected.
2. Performance Failures

This is the second layer in the model. This layer is about performance failures. In this context, performance failure refers to the scenarios with which users are not satisfied when they are using the system. The services with which users are not satisfied are treated as failure cases. For example, in online shopping system placing an order should be processed within 10 seconds. If it takes more than 10 seconds to process order, then it can be considered as performance failure. The information about performance failures can be obtained from feedback reports. Once the feedback report is collected from the stakeholders, the report has to be analyzed. The feedback report may contain positive or negative feedback. The analysis includes studying the feedback and understanding the feedback report. The analysis gives results about the positives and negatives of the system. Positives means, if the customer feels happy of using the system he may give the good feedback about the system. Negative means, if the customer does not feel happy of using the system then, he may give bad feedback about the system.

This dissatisfaction or unhappiness happens with the complete system or with some individual services of the system. The performance failure may occur with different parameters like response time, throughput and resource utilization. Usually performance failures mainly occur due to improper specification of performance requirements. In this layer performance failure cases can be identified from the feedback report. In feedback reports, users can write their comments about the system. Based on these comments performance failures can be identified. Feedback reports are very important things in software development to improve the quality of the software. Again, failure cases have to be analyzed like which behavior caused for failure, what is the parameter, and reasons for
Failure may occur for different reasons like improper specification of performance requirements, workload constraints or environment constraints. So the performance failures have to be analyzed like *how much is the measure of the failure.* Sometimes the failure cases identified may be very marginal which may not cause for severe damage for the system. This type of cases can be handled with minor changes. But in some cases the measure of the failure is high which must be treated carefully and can be handled with major changes. Performance failures can be caused for different reasons. Though the performance requirements are specified properly at requirements level, if these specifications are not followed correctly both at architecture level and implementation level, then it leads to performance failures. There are different mechanisms to identify performance failures during the development process and after the software development process. During the development process quality of the product can be tracked by observing how the developers are incorporating the nonfunctional requirements and based on the intermediate results. After the product development quality of the product also can be assessed. This can be done by using the product.

Once the product is used by different people who need that, definitely they can give right feedback about the product how it is functioning and what performance it is giving. This is the useful information which is considered in this model that is the feedback report. Based on the information available in the feedback report performance requirements can be refined. In this layer, the main activity is identification of performance failures.
3. **Performance Requirements and Parameters**

This is the third layer. This layer explains about identification of performance requirements and parameters that are connected with performance failures. The only way to meet the performance targets of the software systems is to specify performance requirements unambiguously and completely. If these are specified incorrectly, definitely they lead to failure of the software systems.

Once the performance failures are identified from the feedback report, these failures have to be analyzed for things such as which scenarios are falling under failure cases, what are the constraints posed on the scenarios and what are the use cases that are related with scenarios. In this layer, particularly after identifying the performance failures, the corresponding performance requirements and parameters have to be identified with which the failures are connected. These failures may be connected with one or more performance requirements. All performance requirements must be identified which are connected with performance failure. Along with performance requirements, performance parameters also should be identified. Consider the example of the performance requirement “response time for login shall be less than 5 seconds”. Response time is the performance parameter. Identifying performance requirements is a very important task. If these performance requirements are identified then it will be easy to track the performance failures where it has occurred. This information will be useful in improving the performance requirements.

4. **Goals and Sub Goals**

This is the fourth layer in the model which explains about the goals and sub goals.
Goals and sub goals are referred as functionalities and sub functionalities of the system. It is also important to identify the goals and sub goals after identifying the performance requirements and parameters from performance failure. The corresponding goals and sub goals connected with these performance requirements must be identified. Unless these goals and sub goals are identified, it is difficult to refine the performance requirements. Because the performance requirements may differ from scenario to scenario similarly form goal to goal. This is one of the important activities in this model, because, to refine the performance requirements, it is also necessary to know the corresponding goals and sub goals on which these performance constraints are posed.

One should able to identify clearly what is performance failure, the corresponding performance requirement and corresponding goals and sub goals. It is not compulsory to have sub goals for each goal. Some goals may or may not have sub goals. For example in the college information system, if “login” is a goal, then the “student login”, “faculty login” and “parent login” are the sub goals. The performance constraints on each sub goal may differ with other sub goals. So the goals and sub goals may not have same performance requirements. Each goal and each sub goal may have their own performance constraints according to user needs. The performance failure may occur with goals or individual sub goals of the system. So it is very important to know the corresponding goals and sub goals of the system connected with performance failure. If these goals and sub goals are identified, with respect to performance failures, then, it is easy to make changes or propose the new requirements for the product. So in this layer goals and sub goals can be identified that are connected with performance failures.
5. Stakeholders

This is the fifth layer in the model. This layer is about communicating the information of performance failures, performance requirements and corresponding goals and sub goals with stakeholder for the sake of performance requirements refinement. Once the performance failures, performance requirements, and corresponding goals and sub goals are identified, the information about the performance failures has to be communicated with stakeholder. It is always important to make the information available about the performance failures to the stakeholders. So that, they can understand about the performance failure and where exactly the problem has occurred and what exactly the problem is.

Once the problem is understood by the stakeholder, then he can suggest some changes to the performance requirements specification or he can give some advices to meet the performance targets of the system. Communicating with stakeholder is an important task in software development [130]. The information provided in the feedback report has to be discussed with all stakeholders especially with clients, customer and end users. Particularly the negative comments have to be discussed which are given in the feedback report. The main purpose of communication is to get the information from stakeholders about performance requirements refinement. To refine performance requirements, the performance objective has to be reestablished. Reestablishing the objectives involves redefining the performance requirements or modifying the performance requirements or proposing the new performance requirements.
6. Performance Objectives

This is sixth layer in the model. This layer explains about reestablishing the performance objectives. Usually performance objectives specify the quantitative criteria to evaluate the performance characteristics of the system. The performance objectives can be expressed in three ways that is by constraints on resource usage response time, or throughput. From a user perspective, the response time is described as the number of seconds required to respond to a user request for web application. If it is real time system, the response time is described as the amount of time required to respond for a given external event. Throughput is specified as the number of events or transactions to be processed per unit of time. These are the three important parameters to be considered at the time of reestablishing the performance objectives. Once the failures are identified with respect to performance, the objectives have to be verified which are established previously.

The objectives of performance requirements failures have to be reestablished to meet the user needs. According to the user feedback, the performance objectives can be changed. In this reestablishment process many things will be considered based on the feedback provided by the customers. Based on the feedback reports the performance measures have to be reestablished where ever it is necessary.

7. Performance Requirements

This layer is seventh layer in the model. This layer explains about the refinement of performance requirements. To refine the requirements, the objectives have to be reestablished. Reestablishing the objectives is the process where the objectives of the performance will be modified based on the information provided in the feedback reports
according to user needs. The performance objectives are reestablished in the sixth layer. In reestablishing the objectives, some changes will take place quantitatively. These changes have to be applied for original performance requirements specification. Refine the requirements by using quantitative measures specified for all performance requirements.

5.4 Summary

In this chapter the proposed seven layered model for refinement of performance requirements is explained. This model includes seven layers. Each layer includes different activities. Each layer has its own importance. This layered model is used for refining the performance requirements based on the users’ feedback. This model is simple to understand and easy to use.