CHAPTER 1

INTRODUCTION

In software development life cycle, requirements engineering is one of the most important activities [1]. The main aim of requirements engineering is to gather requirements according to customer needs. These requirements are two types. One is functional requirements and other one is nonfunctional requirements. Both are equally important in software development [2]. The clear statement of requirements always leads to success of the software project [3]. For example, if online shopping system is considered the functional requirements are search product, place order, make payment etc. There are several nonfunctional requirements like performance requirements, reliability requirements, security requirements, usability requirements etc. Among these nonfunctional requirements, performance is the most important, because the acceptance of software system depends on how well the software system satisfies performance requirements. For example, if the response time of online shopping system is more, a user may get annoyed in using the system and may prefer to shop with its competitor [4].

In one of the major projects (NASA), the study of the interdependence of the earth’s eco systems, the effect of not satisfying the performance requirements is reflected. Launching of satellite is delayed by NASA [5] considerably around eight months due to the software used for generating satellite schedules has unacceptable high response time. Hence, not satisfying performance requirements may lead to delay in the project launching, increased cost, rejection of the product and degrades the image of the
company [6]. In addition, customer relationship may get damaged. In business point of view, long term good customer relationship is important in software industry [7]. So the performance requirements have to be specified completely and consistently in system requirements specification. If these are not specified correctly in the system requirements specification for any reason, at least they have to be refined for next version of the product. In the literature very few methods are proposed for performance requirements identification and refinement. The research work carried out about performance requirements identification can be found in [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]. The research works carried out about performance requirements refinement can be found in [24, 25 26, 27, 28]. These methods are not much effective in identification and refinement of performance requirements because the methods proposed are developer oriented rather than user oriented, application specific and not detailed. Hence, there is great need to focus on performance requirements as they play an important role in success of any software product. Therefore, this thesis focuses on performance requirements.

1.1 Importance of Performance Requirements

There are thirteen nonfunctional requirements listed by IEEE std-830 -1993 [29]. One among those thirteen nonfunctional requirements is performance requirements. Performance requirements are the most important nonfunctional requirements that contribute maximum to the success or failure of the software product. Poor performance is always causes for failure of software project [30]. The main cause of poor performance is inadequately specified performance requirements. Performance requirements may be
vaguely written, or might not even have been written at all by the time the project is close to completion [23]. Poorly written performance requirements always lead to increased project cost. They create confusion among the developers and testers who are responsible for developing the product with specified performance. Sometimes confusion can also be raised when performance requirements are written for unwanted reason or when it conflicts with other performance requirements. Like functional requirements, performance requirements should exhibit the properties like unambiguity, completeness, correctness, traceability, and verifiability [31]. They must be written in measurable terms, expressed in correct statistical terms, and written in terms of one or more metrics that are informative about the problem domain [32]. They must also be written in terms of metrics suitable for the time scale within which the system must respond to stimuli. In addition, the requirements must be mathematically consistent. All performance requirements must be linked to business, regulatory, and engineering needs [33, 34, 35].

Since performance problems often have their roots in poor performance requirements specification, the early establishment of performance requirements for a new system is crucial to the project’s success [36]. Hence, this thesis focuses on performance requirements. As part of this thesis, two different layered models are proposed to identify, specify, validate, and refine the performance requirements. These models do not require architectural design information and enables to include the performance requirements in Software Requirements Specification [SRS] during the requirements engineering phase. Proposed models in this thesis are useful in estimation of the project cost and schedule before starting of the design as they are not depending on the architectural design.
1.2 Motivation

The main reason for identification of performance requirements, criticality assessment of performance parameters, and refinement of performance requirements is to produce the product with acceptable performance. Performance plays crucial and important role in wide range of applications varying from e-commerce application to safety-critical systems. Performance risks or problems always occur with improper performance requirements specifications. Performance requirements are most important to consider among all the nonfunctional requirements.

For example, consider online reservation system, when people are using online railway reservation system, it is observed that the response time of “search train” activity or “book ticket” activity or “make payment” activity is more than expected [37]. Due to the poor response time people may not get the tickets within the time required for specific trains. This is all what happens with the poor performance of railway reservation system. Similarly, if it is online trading system, responding to the requests placed by the user or system within the given time is more important, where response time plays an important role. There is a chance of trading loss, if the system is not responded for the user request with in the time. These are two examples which explain the importance of performance. So, there is great need of developing the software systems with good performance. To develop the system with good performance, all the performance requirements have to be identified and incorporated into performance requirements specification. Proper performance requirements specification always leads to good product with acceptable performance [38].
For this purpose, few approaches are there for performance requirements identification. These are goal and scenario oriented approaches, use case maps approaches and stochastic probes. But these are not much effective in identifying the performance requirements. Because the results obtained from these approaches are not detailed and very specific to application. For example the performance requirements identified by use case maps approach are response time and space, which are represented as high level performance requirements. This may cause for difficulty in understanding the performance requirements. Therefore, in this thesis a five layered model is proposed to identify performance requirements. This model includes set of conceptual rules which are helpful in identifying all possible performance requirements, moreover it is a generic model suits for any application.

Sometimes, the specification of performance requirements may not be done correctly at requirements engineering level due to wage information provided by customers or improper knowledge of requirements analysts [39]. Due to fast development schedules, very little time is given for requirements engineering in agile development [40]. In these cases the product developed with improper specification leads to failure of the product with respect to performance. So to produce the product with acceptable performance, the performance requirements specification has to be refined for next release. For this purpose, a seven layered model is proposed in this thesis, which includes a refinement procedure and rules. By using the procedure and rules, performance requirements are refined to the extent possible.
There are some problems associated with performance, these performance problems may lead to performance failures. Performance failures can have many negative consequences. These include [30]:

- **Damaged customer relationship:** Once the poor performance is encountered during the usage of the product, the developer fixes the problem. Later users likely to continue to tag the poor performance to the rectified product and also affect the future products sale.

- **Business failures:** More time is required to complete important and key tasks because of poor performance. More people are required to complete it in the same time.

- **Last income:** There may be a loss of revenue due to late delivery.

- **Additional project resources:** To rectify poor performance related issues additional resources are allocated which increases the project cost.

- **Reduced competitiveness:** Tuning and rectifying poor performance issues take more time and leads to late delivery of the product which misses the market window.

- **Project failure:** Tuning or redesign is impossible for some software systems to meet performance targets late in the process. Sometimes the projects may get canceled.
1.3 Contributions of the Thesis

As the performance requirements are gaining more importance in the software development, in this dissertation three issues have been addressed. First one is performance requirements identification, second one is criticality assessment of performance parameters and third one is refinement of performance requirements.

- A layered model is proposed for performance requirements identification. This model includes five layers. These are stakeholders, goals, sub goals, performance requirements and performance parameters. First layer is about stakeholders who are there in the system. Stakeholder is a person who directly or indirectly connected with system [41]. For example developer, analyst, project manager, customer, end user etc. All stakeholders’ views must be considered to identify the performance requirements because different people may visualize the system in different ways. Due to this there is a possibility of getting more number of requirements from different people. Second layer is about goals. In this layer all possible goals from stakeholders are identified. Goals are nothing but functional requirements or use cases. Third layer includes sub goals. In this layer all goals are decomposed into sub goals. Forth layer is about performance requirements. After decomposing the goals into sub goals then performance requirements are identified for corresponding sub goals. Fifth layer is performance parameters. After identifying the performance requirements the corresponding parameters are identified. As part of this model set of rules are proposed which are useful in identifying the
performance requirements. Using this approach all performance requirements are identified.

- Four metrics are proposed to estimate and assess the performance parameters criticality. They are MGPP_RT, MGPP_ED, MGPP_RU, and MGPP_TP. They are validated analytically using Weyuker’s properties.

- Another layered model is proposed for performance requirements refinement. This model is used to refine the performance requirements based on the user feedback reports. The feedback reports given by users about performance of the system are useful for understanding the factors, contributing to unacceptable performance. The information obtained from the feedback report can be useful for the development team in making knowledgeable requirements decisions. This model includes seven layers. These are feedback report, performance failures, performance requirements and parameters, goals and sub-goals, stakeholder, performance objectives and performance requirements. As part of this model nine rules are proposed in sixth layer that is performance objectives. These rules are used to reestablish performance objectives. These objectives that are reestablished are useful to refine performance requirements by defining the new performance requirements or modifying the existing requirements.

These two models are used to identify and refine the performance requirements. To evaluate the ability of these models and proposed metrics, three case studies have been considered for first layered model and two case studies have considered for
second layered model. The description of the models and the case studies results are presented in this dissertation.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows.

Chapter 2 presents the literature survey about requirements engineering, functional requirements, non-functional requirements and performance requirements.

Chapter 3 describes a layered model for identification of performance requirements. This model is applied on three case studies and results are presented in this chapter.

Criticality assessment of performance requirements parameters were discussed in Chapter 4. It also describes the proposed metrics and validation of metrics.

Layered model for performance requirements refinement is presented in Chapter 5. It describes about the different layers in the proposed model.

Chapter 6 presents the performance requirements refinement with feedback reports, and results of the case studies.

Chapter 7 concludes the dissertation with contributions of the research work and also presents the future work.