4 Requirements Prioritization

Patrik Berander and Anneliese Andrews

Abstract: This chapter provides an overview of techniques for prioritization of requirements for software products. Prioritization is a crucial step towards making good decisions regarding product planning for single and multiple releases. Various aspects of functionality are considered, such as importance, risk, cost, etc. Prioritization decisions are made by stakeholders, including users, managers, developers, or their representatives. Methods are for combining individual prioritizations based on overall objectives and constraints. A range of different techniques and aspects are applied to an example to illustrate their use. Finally, limitations and shortcomings of current methods are pointed out, and open research questions in the area of requirements prioritization are discussed.

Keywords: Requirements analysis, Software product planning, Requirements prioritization, Decision support, Trade offs.

4.1 Introduction

In everyday life, we make many decisions, e.g. when buying a DVD-player, food, a telephone, etc. Often, we are not even conscious of making one. Usually, we do not have more than a couple of choices to consider, such as which brand of mustard to buy, or whether to take this bus or the next one. Even with just a couple of choices, decisions can be difficult to make. When having tens, hundreds or even thousands of alternatives, decision-making becomes much more difficult.

One of the keys to making the right decision is to prioritize between different alternatives. It is often not obvious which choice is better, because several aspects must be taken into consideration. For example, when buying a new car, it is relatively easy to make a choice based on speed alone (one only needs to evaluate which car is the fastest). When considering multiple aspects, such as price, safety, comfort, or luggage load, the choice becomes much harder. When developing software systems, similar trade-offs must be made. The functionality that is most important for the customers might not be as important when other aspects (e.g. price) are factored in. We need to develop the functionality that is most desired by the customers, as well as least risky, least costly, and so forth.

Prioritization helps to cope with these complex decision problems. This chapter provides a description of available techniques and methods, and how to approach a prioritization situation. The chapter is structured as follows: First, an overview of the area of prioritization is given (Sect. 4.2). This is followed by a presentation and discussion of different aspects that could be used when prioritizing (Sect. 4.3). Next, some prioritization techniques and characteristics are discussed (Sect. 4.4), followed by a discussion of different stakeholders’ situations that affect prioritiza-
tion in Sect. 4.5. Section 4.6 discusses additional issues that arise when prioritizing software requirements and Section 4.7 provides an example of a prioritization. Section 4.8 discusses possible future research questions in the area. Finally, Sect. 4.9 summarizes the chapter.

4.2 What is Requirements Prioritization?

Complex decision-making situations are not unique to software engineering. Other disciplines, such as psychology, and organizational behavior have studied decision-making thoroughly [1]. Classical decision-making models have been mapped to various requirements engineering activities to show the similarities [1]. Chapter 12 in this book provides a comprehensive overview of decision-making and decision support in requirements engineering. Current chapter primarily focuses on requirements prioritization, an integral part of decision-making [49]. The intention is to describe the current body of knowledge in the requirements prioritization area.

The quality of a software product is often determined by the ability to satisfy the needs of the customers and users [7, 53]. Hence, eliciting (Chap. 2) and specifying (Chap. 3) the correct requirements and planning suitable releases with the right functionality is a major step towards the success of a project or product. If the wrong requirements are implemented and users resist using the product, it does not matter how solid the product is or how thoroughly it has been tested.

Most software projects have more candidate requirements than can be realized within the time and cost constraints. Prioritization helps to identify the most valuable requirements from this set by distinguishing the critical few from the trivial many. The process of prioritizing requirements provides support for the following activities [32, 55, 57, 58]:

- for stakeholders to decide on the core requirements for the system
- to plan and select an ordered, optimal set of software requirements for implementation in successive releases
- to trade off desired project scope against sometimes conflicting constraints such as schedule, budget, resources, time to market, and quality
- to balance the business benefit of each requirement against its cost
- to balance implications of requirements on the software architecture and future evolution of the product and its associated cost
- to select only a subset of the requirements and still produce a system that will satisfy the customer(s)
- to estimate expected customer satisfaction
- to get a technical advantage and optimize market opportunity
- to minimize rework and schedule slippage (plan stability)
- to handle contradictory requirements, focus the negotiation process, and resolve disagreements between stakeholders (more about this in Chap. 7)
- to establish relative importance of each requirement to provide the greatest value at the lowest cost