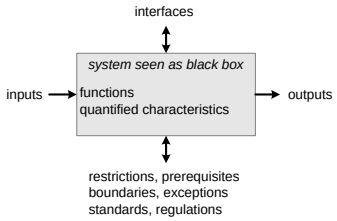
Fundamentals of Requirements Engineering

-



Gerrit Muller
University of South-Eastern Norway-NISE
Hasbergsvei 36 P.O. Box 235, NO-3603 Kongsberg Norway
gaudisite@gmail.com

This paper has been integrated in the book "Systems Architecting: A Business Perspective", http://www.gaudisite.nl/SABP.html, published by CRC Press in 2011.

Abstract

Requirements engineering is one of the systems engineering pillars. In this document we discuss the fundamentals of systems engineering, such as the transformation of needs into specification, the need to prescribe *what* rather than *how*, and the requirements when writing requirements.

version: 0.1 status: concept September 9, 2018

1 Introduction

The basis of a good system architecture is the availability and understanding of the needs of all stakeholders. Stakeholder needs are primary inputs for the system specification. The terms *requirements elicitation*, *requirements analysis*, and *requirements management* are frequently used as parts of the Product Creation Process that cope with the trandormation of needs into specification and design.

2 Definition of Requirements

The term requirement is quite heavily overloaded in Product Creation context. *Requirement* is sometimes used non-obligatory, e.g. to express wants or needs. In other cases it used as mandatory prescription, e.g. a must that will be verified. Obviously, dangerous misunderstandings can grow if some stakeholders interpret a requirement as want, while other stakeholders see it as must.

We will adopt the following terms to avoid this misunderstanding:

Customer Needs The term *Customer Needs* is used for the non-mandatory wishes, wants, and needs.

Product Specification The term *Product Specification* is used for the mandatory characteristics the system must fulfill.

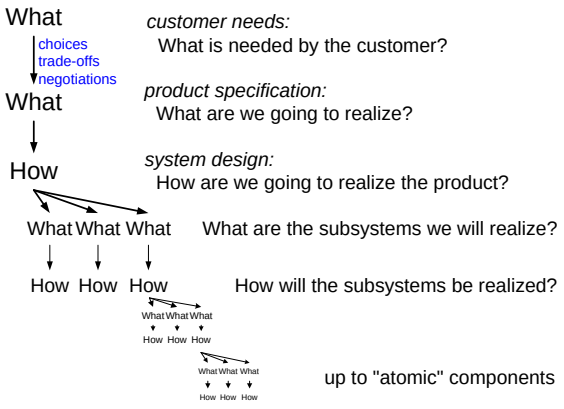


Figure 1: The flow of requirements

In the system engineering world the term *Requirements Management* or *Requirements Engineering* is also being used. This term goes beyond the two previous interpretations. The requirements management or engineering process deals with the propagation of the requirements in the product specification towards the requirements of the atomic components. Several propagation steps take place between

the product specification and atomic components, such as requirements of the subsystems defined by the first design decomposition. In fact the requirement definition is recursively applied for every decomposition level similar to the product specification and subsystem decomposition.

Figure 1 shows the requirements engineering flow. The customer needs are used to determine the product specification. Many choices are made going from needs to specification, sometimes by negotiation, sometimes as trade-off. Often the needs are not fully satisfied for mundane reasons such as cost or other constraints. In some cases the product specification exceeds the formulated needs, for instance anticipating future changes.

Figure 1 also show the separation of specification, *what*, and design, *how*. This separation facilitates clear and sharp decision making, where goals *what* and means *how* are separated. In practice decision are often polluted by confusing goals and means.

An other source of requirements is the organization that creates and supplies the product. The needs of the organization itself a nd of the supply and support chain during the life cycle are described in this chapter as *Life Cycle Needs*.

3 System as a black box

One of the main characteristics of requirements in the product specification is that they describe *what* has to be achieved and not it how this has to be achieved. In other words, the product specification describes the system as *black box*. Figure 2 provides a starting point to write a product specification.

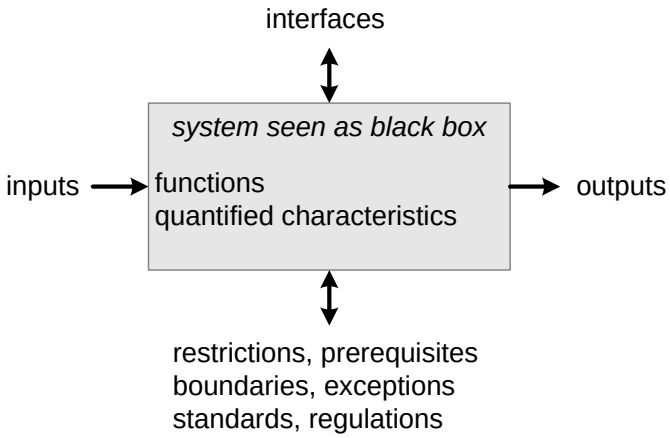


Figure 2: System as a Black Box

The system is seen as black box. What goes into the box, what comes out and what functions have to be performed on the inputs to get the outputs. Note that the functions tell something about the black box, but without prescribing how to realize them. All interfaces need to be described, interfaces between the system and humans as well as interfaces to other systems. The specification must also quantify desired characteristics, such as how fast, how much, how large, how costly, et cetera.

Prerequisites and constraints enforced on the system form another class of information in the product specification. Further scoping can be done by stating boundaries and desired behavior in case of exceptions. Regulations and standards can be mandatory for a system, in which case compliance with these regulations and standards is part of the product specification.

4 Stakeholders

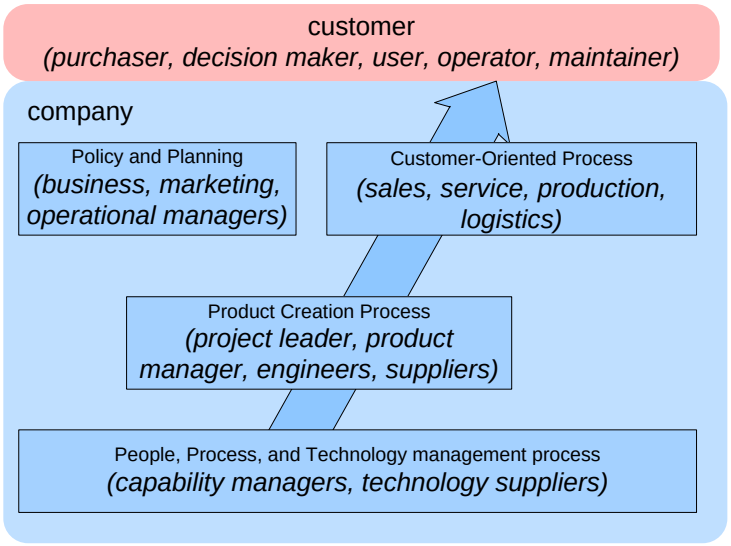


Figure 3: A simplified process decomposition of the business. The stakeholders of the requirements are beside the customer self, mainly active in the customer oriented process and the product creation process.

A simplified process model is shown in figure 3. The stakeholders of the product specification are of course the customers, but also people in the Customer Oriented Process, the Product Creation Process, People, Process, and Technology Management Process, and the Policy and Planning Process. The figure gives a number of examples of stekholders per process.

The customer can be a consumer, but it can also be a business or even a group of businesses. Businesses are complex entities with lots of stakeholders. A good understanding of the customer business is required to identify the customer-stakeholders.

5 Requirements for Requirements

Standards like ISO 9000 or methods like CMM prescribe the requirements for the requirements management process. The left side of Figure 4 shows typical requirements for the requirements itself.

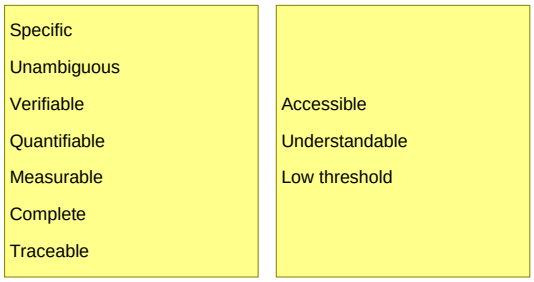


Figure 4: Requirements for Requirements

Specific , what is exactly needed? For example, The system shall be *user friendly* is way too generic. Instead a set of specific requirements is needed that together will contribute to user friendliness.

Unambiguous so that stakeholders don't have different expectations on the outcome. In natural language statements are quite often context sensitive, making the statement ambiguous.

Verifiable so that the specification can be verified when realized.

Quantifiable is often the way to make requirements verifiable. Quantified requirements also help to make requirements specific

Measurable to support the verification. Note that not all quantified characteristics can also be measured. For example in wafer steppers and electron microscopes many key performance parameters are defined in nanometers or smaller. There are many physical uncertainties to measure such small quantities.

Complete for all main requirements. *Completeness* is a dangerous criterion. In practice a specification is never complete, it would take infinite time to

approach completeness. The real need is that all crucial requirements are specified.

Traceable for all main relations and dependencies. *Traceability* is also a dangerous criterion. Full traceability requires more than infinite time and effort. Understanding how system characteristics contribute to an aggregate performance supports reasoning about changes and decision making.

Unfortunately, these requirements are always biased towards the formal side. A process that fulfills these requirements is from theoretical point of view sound and robust. However, an aspect that is forgotten quite often, is that product creation is a human activity, with human capabilities and constraints. The human point of view adds a number of requirements, shown at the right hand side of Figure 4: accessibility, understandability, and a low threshold. These requirements are required for **every** (human) stakeholder.

These requirements, imposed because of the human element, can be conflicting with the requirements prescribed by the management process. Many problems in practice can be traced back to violation of the human imposed requirements. For instance, an abstract description of a customer requirement such that no real customer can understand the requiremnts anymore. Lack of understanding is a severe risk, because early validation does not take place.

References

[1] Gerrit Muller. The system architecture homepage. `http://www.gaudisite.nl/index.html`, 1999.

History

Version: 0.1, date: August 6, 2010 changed by: Gerrit Muller

- changed logo.

Version: 0, date: June 28, 2010 changed by: Gerrit Muller

- created by refactoring Requirements.