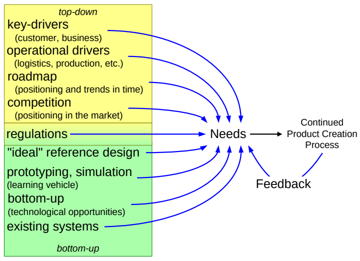
Requirements Capturing by the System Architect

-



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

Abstract

The basis of a good system architecture is the availability and understanding of the requirements. This presentation shows how a system architect can capture the requirements and how to use these requirements in the context of the product creation process.
The notion of "business key drivers" is introduced and a method is described to link these key drivers to the product specification.

version: 1.5 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, dangerouys misunderstandings can grow if some stakeholders interpret a requirement as want, while other stakehodlers 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.

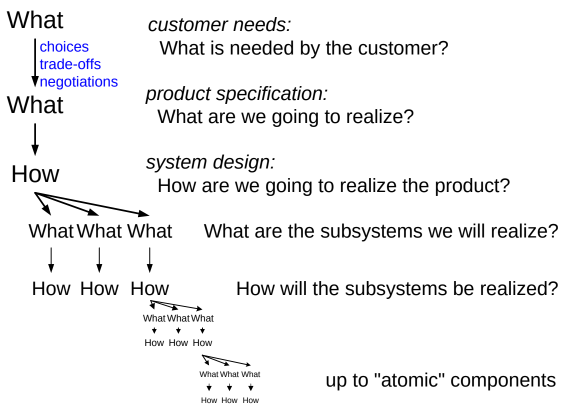


Figure 1: The flow of requirements

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.

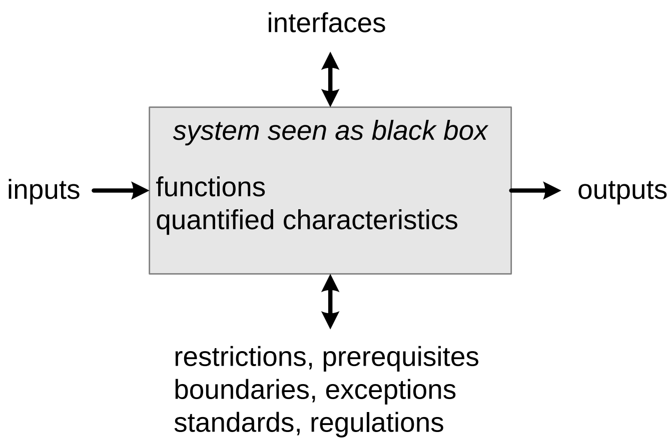


Figure 2: System as a Black Box

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

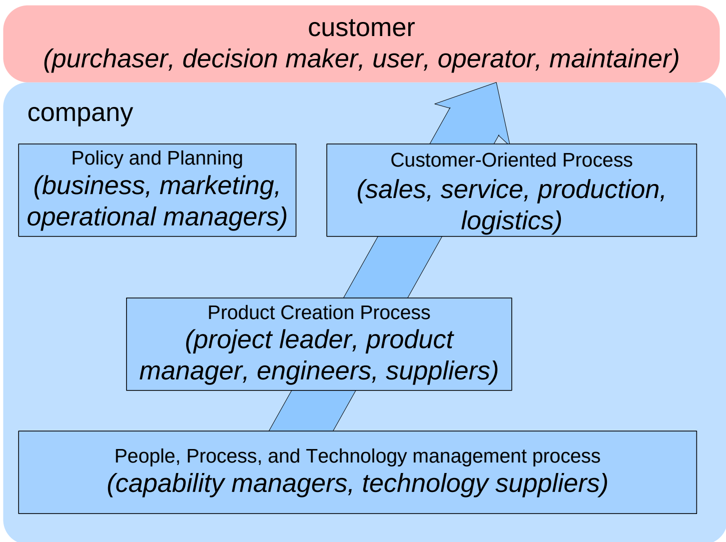


Figure 3: A simplified process decomposition of the business. The stakeholders of the requirements are beside the customer self, mainly active in the customer oriented process and the product creation process.

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.

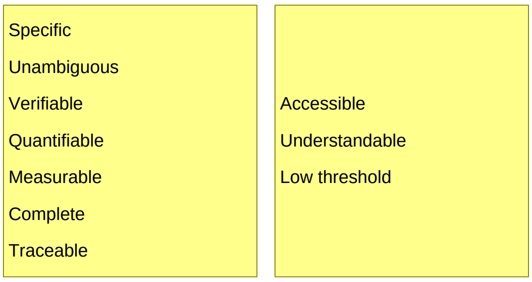


Figure 4: Requirements for Requirements

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.

6 Viewpoints on Needs

Needs for a new product can be found in a wide variety of sources. The challenge in identifying needs is, in general, to distinguish a solution for a need from the need itself. Stakeholders, when asked for needs, nearly always answer in terms of a solution. For example, consumers might ask for a *flash based video recorder*, where the underlying need might be a light-weight, small, portable video recorder. It is the architect's job, together with marketing and product managers, to reconstruct the actual needs from the answers that stakeholders give.

Many complementary viewpoints provide a good collection of needs. Figure 5 shows a useful number of viewpoints when collecting needs.

The **key-driver** viewpoint and the **operational** viewpoint are the viewpoints of the stakeholders who are "consuming" or "using" the output of the Product Creation Process. These viewpoints represent the "demand side".

The **roadmap** and the **competition** viewpoints are viewpoints to position the products in time and in the market. These viewpoints are important because they emphasize the fact that a product is being created in a dynamic and evolving world. The product context is not static and isolated.

Regulations result in requirements both top-down, as well as bottom-up. A top down example are labor regulations that can have impact on product functionality

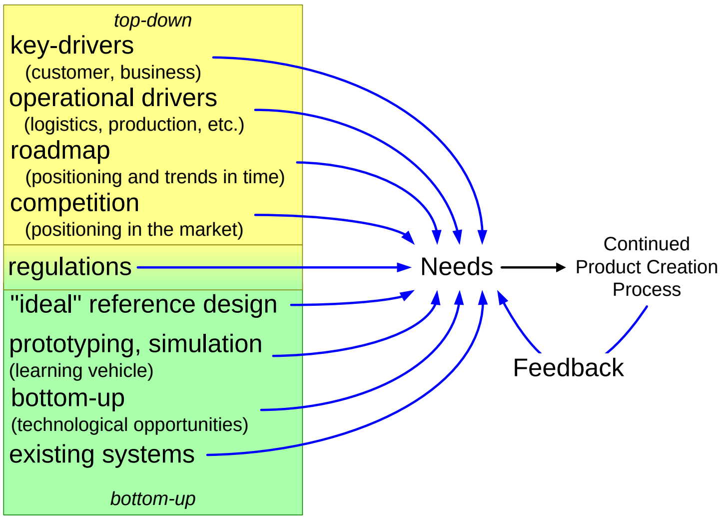


Figure 5: Complementary viewpoints to collect needs

and performance. A bottom up example are materials regulations, for instance do not use lead, that may strongly influence design options.

The **"ideal" reference design** is the challenge for the architect. What is in the architect's vision the perfect solution? From this perfect solution the implicit needs can be reconstructed and added to the collection of needs.

Prototyping or simulations are an important means in communication with customers. This "pro-active feedback" is a very effective filter for nice but impractical features at the one hand and it often uncovers many new requirements. An approach using only concepts easily misses practical constraints and opportunities.

The **bottom up** viewpoint is the viewpoint where the technology is taken as the starting point. This viewpoint sometimes triggers new opportunities that have been overlooked by the other viewpoints due to an implicit bias towards today's technology.

The **existing system** is one of the most important sources of needs. In fact it contains the accumulated wisdom of years of practical application. Especially a large amount of small, but practical, needs can be extracted from existing systems.

The product specification is a dynamic entity, because the world is dynamic: the users change, the competition changes, the technology changes, the company itself changes. For that reason the **Continuation of the Product Creation Process** will generate input for the specification as well. In fact nearly all viewpoints are present and relevant during the entire Product Creation Process.

7 Reference Architecture and Key Drivers

A system architect must look at the product from multiple complementary viewpoints. Figure 6 shows 5 useful views for a reference architecture.

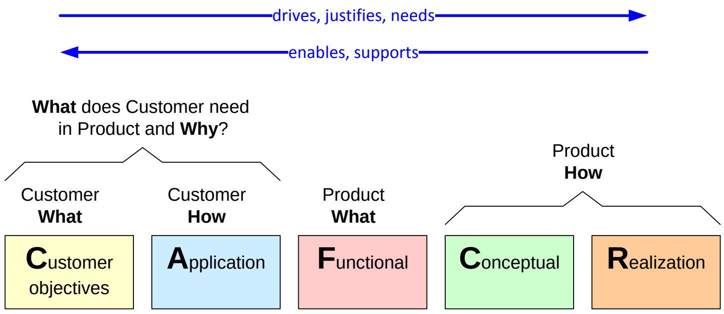


Figure 6: A Reference Architecture views the architecture from 5 viewpoints

The business architecture is the architecture of the business of the customer, in relation with the product. Typically it will describe the flow of information or goods, the business processes and the related roles.

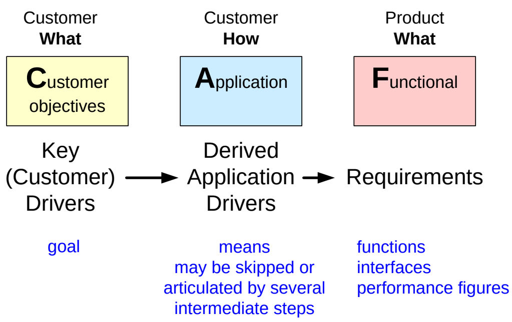


Figure 7: The mapping of Key Drivers via derived application drivers on requirements

A very powerful means to capture requirements is to describe the essence of the business in terms of *Key Drivers*. These drivers must be recognized and understood by the customer, which means that these drivers should be expressed in the language of the customer. A maximum of 5 Key Drivers is recommended to maintain focus on the essence, the name is on purpose **Key** driver. The key drivers are one aspect of the business architecture. Figure 8 shows a method to define key drivers. Figure 9 shows some recommendations with respect to the definition of key drivers.

• Define the scope specific.	in terms of stakeholder or market segments
• Acquire and analyze facts	extract facts from the product specification and ask why questions about the specification of existing products.
• Build a graph of relations between drivers and requirements by means of brainstorming and discussions	where requirements may have multiple drivers
• Obtain feedback	discuss with customers, observe their reactions
• Iterate many times	increased understanding often triggers the move of issues from driver to requirement or vice versa and rephrasing

Figure 8: Method to define key drivers

• Limit the number of key-drivers	minimal 3, maximal 6
• Don't leave out the obvious key-drivers	for instance the well-known main function of the product
• Use short names, recognized by the customer.	
• Use market-/customer- specific names, no generic names	for instance replace "ease of use" by "minimal number of actions for experienced users", or "efficiency" by "integral cost per patient"
• Do not worry about the exact boundary between Customer Objective and Application	create clear goal means relations

Figure 9: Recommendations when defining key drivers

Key drivers can be mapped on derived application drivers. Which application activities are done to enable the key driver? The derived application drivers must also be expressed in customer language. The explicit description of application drivers will also ease the job of modelling the application domain.

The derived application drivers are implemented or supported by features or functions of the product. This means that the derived application drivers can be translated into customer requirements of the product.

From point of view of requirements engineering the customer requirements are used as input to produce a product specification, which controls the entire product creation process. The design of the system will result in a technical architecture, with amongst others a decomposition in subsystems and function allocation. The technical architecture is finally mapped onto an implementation. The relation between requirements at the functional architecture level, the technical architecture level and the implementation is managed by the requirements management process.

Approaching the requirements definition in this way enables the architect to understand a technical feature in relation with the key driver from the customer business. Any feature that cannot be related back to a key driver is suspect: either it should not be there or some requirement or driver is missing.

8 Example Motorway Management

Figure 10 shows an example of the requirements analysis of a motorway management system. The keydrivers of a motorway management owner are:

- Safety
- Effective Flow
- Smooth Operation
- Environment

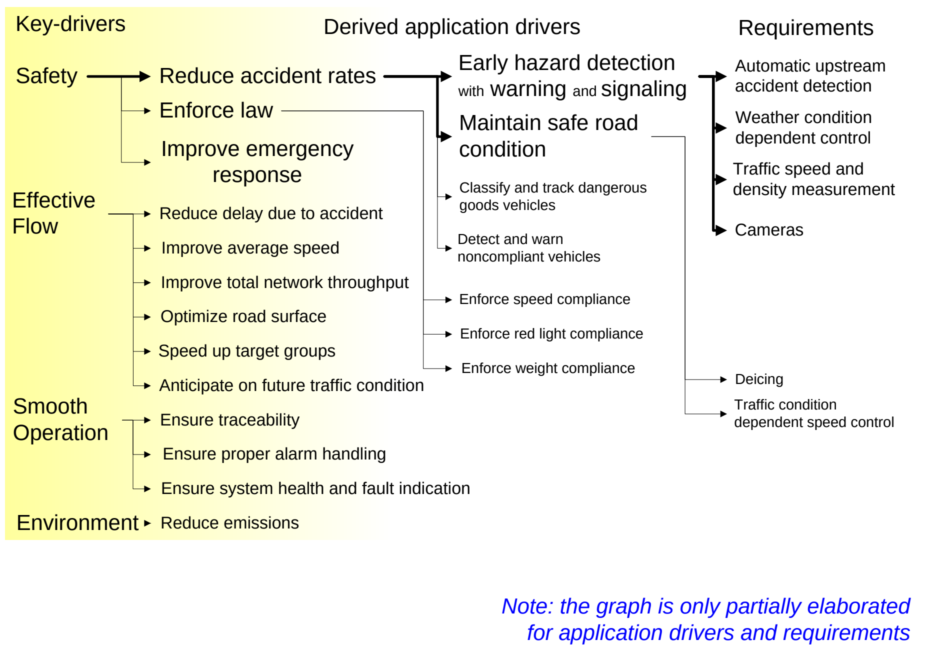


Figure 10: The key drivers, derived application drivers and requirements of a Motorway Management System

To realize these key drivers the owner applies a number of application processes. This leads to the derived application drivers. For instance to realize safety it is important to prevent accidents and to have immediate response by emergency departments in case of accidents.

9 Requirements Value and Selection

The collection of customer and operational needs is often larger than can be realized in the first release of a product. A selection step is required to generate a product specification with the customer and operational needs as input. Figure 11 shows the selection process as black box with its inputs and outputs.

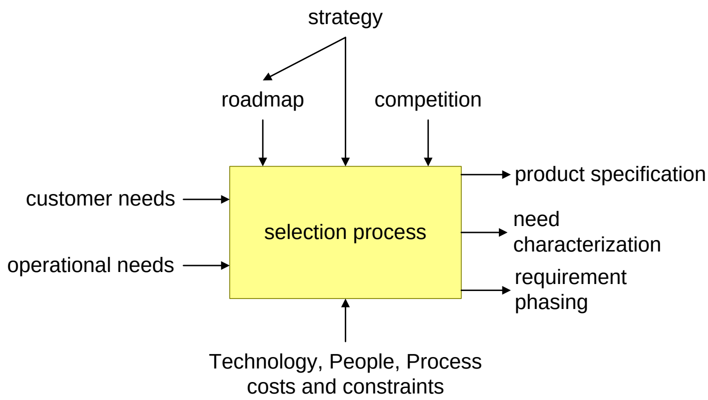


Figure 11: The selection process produces a product specification and a phasing and characterization of requirements to prevent repetition of discussion

The selection process is primarily controlled by the strategy of the company. The strategy determines market, geography, timing and investments. The roadmap, based on the strategy, is giving context to the selection process for a individual products. The reality of the competitive market is the last influencing factor on the selection.

The selection will often be constrained by technology, people, and process. The decisions in the selection require facts or estimates of these constraints.

During the selection a lot of insight is obtained in needs, the value of needs, and the urgency. We recommend to consolidate these insights, for example by documenting the characterization of needs. The timing insights can be documented in a phased plan for requirements.

The amount of needs can be so large that it is beneficial to quickly filter out the"obvious" requirements. For some needs it is immediately obvious that they have to be fulfilled anyway, while other needs can be delayed without any problem. Figure 12 shows a number of qualitative characterizations of needs, visualized in a two-dimensional matrix. For every quadrant in the matrix a conclusion is given, a need must be included in the specification, a need has to be discarded or the need must be discussed further.

This simple qualitative approach can, for instance, be done with the following criteria:

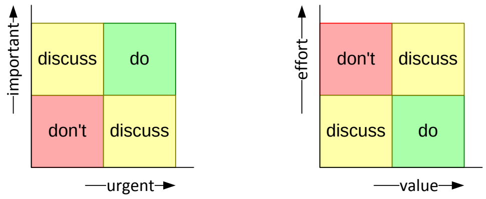


Figure 12: Simple methods for a first selection

- importance versus urgency
- customer value versus effort

In the final selection step a more detailed analysis step is preferable, because this improves the understanding of the needs and results in a less changes during the development.

A possible way to do this more detailed analysis is to "quantify" the characteristics for every requirement for the most business relevant aspects, see for examples Figure 13.

- Value for the customer
- (dis)satisfaction level for the customer
- Selling value (How much is the customer willing to pay?)
- Level of differentiation w.r.t. the competition
- Impact on the market share
- Impact on the profit margin

Use relative scale, e.g. 1..5 1=low value, 5 -high value
Ask several knowledgeable people to score
Discussion provides insight (don't fall in spreadsheet trap)

Figure 13: Quantifiable Aspects for Requirements Selection

These quantifications can be given for the immediate future, but also for the somewhat remote future. In that way insight is obtained in the trend, while this information is also very useful for a discussion on the timing of the different requirements. In [1] a much more elaborated method for requirement evaluation and selection is described.

The output of the requirement characterization and the proposed phasing can be used as input for the next update cycle of the roadmap.

10 Acknowledgements

The platform project within Philips Projects provided a clear analysis of amongst others a motorway management process. Wil Hoogenstraaten contributed significantly in the creation of this requirements analysis and sharpened the model from key driver towards features and functions.

Stimulating discussions with Henk Obbink and Jürgen Müller helped to shape this article.

Shakil Ahmed added the regulations viewpoint.

References

[1] Jean-Marc DeBaud and Klaus Schmid. A systematic approach to derive the scope of software product lines. In *21st international Conference on Software Engineering: Preparing for the Software Century*, pages 34–47. ICSE, 1999.

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

History

Version: 1.5, date: June 28, 2010 changed by: Gerrit Muller
- textual adaptations
- more consistent use of the terms requirements, product specification and needs

Version: 1.4, date: May 21, 2010 changed by: Gerrit Muller
- replaced some lists by figures
- renamed operational requirements in life cycle requirements

Version: 1.3, date: August 28, 2008 changed by: Gerrit Muller
- added system as a black box

Version: 1.2, date: April 20, 2005 changed by: Gerrit Muller
- added regulations viewpoint

Version: 1.1, date: February 17, 2005 changed by: Gerrit Muller
- updated key driver figures
- improved readability of some figures

Version: 1.0, date: May 31 2002 changed by: Gerrit Muller
- added abstract
- increased readability of some figures
- modified stakeholder figure
- updated keydriver figure
- modified Motorway management example

Version: 0.2, date: June 22 2001 changed by: Gerrit Muller
- annotated the figure with the reference architecture
- added a list of recommendations with respect to the definition of keydrivers

Version: 0.1, date: February 16 2001 changed by: Gerrit Muller
- new layout

Version: 0, date: October 25 1999 changed by: Gerrit Muller
- Created, no changelog yet