BIANCHI & PARTNER

What is Requirements Engineering?

From Bernhard Bianchi

Aim of this article

With this article, I would like to introduce in a compact way in our definition of requirements engineering.

Summary

What does a computer do? It automatically processes the described business expertise for a specific area of interest. In the simple case, this expertise includes only a certain structuring of data (e.g. database applications). All other activities are left to the user. In the more extensive case, however, the expertise also includes the functionality (algorithms) necessary to provide certain services along with the data.

At the beginning of each project, a formal specification of this expertise is required in both cases, in order to then automate this knowledge as far as desired, whilst taking into account the technical limitations of a particular machine.

The formalized business knowledge (the domain model) forms the conceptual frame of reference without which no effective software, hardware or organizational development is possible.

The development of this formalized knowledge is called requirements analysis. It represents the most important step in requirements engineering and describes the abstract business solution. The details of the abstract business solution are the most important requirements in the development of a socio-technical solution. The main result of the requirements analysis is a requirements specification. For the description of the functionality we recommend an object-oriented model, or at most algebraic specifications.

The design of the socio-technical solution, which takes into account the technical, human and economic possibilities, we call system design. This design is derived from the abstract business model. The main result of the system design is an architectural specification. As a description language, we also recommend an object-oriented model here, or at most algebraic specifications. For the subareas with human organizational processes, business process models are also suitable.

The formalized knowledge is built up step by step starting from a first, minimal functional core. It is up to each project to decide whether it would like to implement every business step in a timely manner, or whether a broader basis of the formalized business solution is necessary before starting with implementation steps. This patience may be needed to reduce the risk of inappropriate architecture.

Requirements engineering leads to the specification of the conceptual frame of reference as well as to the specification of a socio-technical architecture.

Bianchi & Partner GmbH

Bianchi & Partner GmbH is an independent engineering company for system and software development. Since 1994, we have been training and accompanying engineers in introducing or improving requirements engineering. In the main, however, we do Requirements Engineering on assignment, depending on the situation with existing or our own teams. Basis and reference is our own requirements engineering method.

Bernhard Bianchi

Bernhard Bianchi successfully accompanies and implements projects with a focus on requirements engineering for more than 30 years. The company founder also works as a lecturer for requirements engineering at the University of Applied Sciences in Rapperswil and in various companies as a speaker, consultant and coach.

Definition of the term

After trying for many years to find a plausible definition of requirements engineering or even to design one ourselves, in 1998 we finally decided on our own – and since proven – definition.

Why a separate definition?

On the one hand, there was and is no universally accepted definition of what is meant by requirements engineering.

On the other hand, there were good definitions early on, for example those of Ross and Schoman: „Requirements definition must encompass everything necessary to lay the groundwork for subsequent stages in system development. Within the total process, which consists largely of steps in solution to a problem, only once is the problem itself stated and the solution justified - in requirements definition. Requirements definition is a careful assessment of the needs that a system is to fulfill. It must say why a system is needed, […]. It must say what system features will serve and satisfy this context. And it must say how the system is to be constructed." (Source: Douglas T. Ross, Kenneth E. Schoman: Structured Analysis for Requirements Definition. IEEE Transactions on Software Engineering, Vol. SE-3, No. 1, 1977).

However, we were always disturbed either by a too narrow definition of the system (for example, a restriction to IT systems) or by the lack of a solution design, without which no statements can be made regarding the feasibility or cost-effectiveness. We could not find a definition that satisfactorily covered both aspects.

Finally, we have decided on the following definition:

Requirements engineering is the development, formal specification and testing of essential requirements for an artificial, interactive system (requirements analysis)
and the design of a feasible and economical solution (system design)

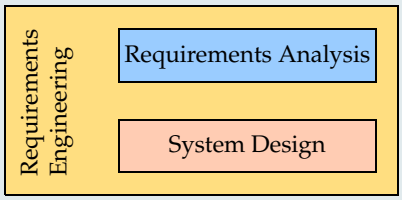

Figure 1

This definition covers all activities, starting from an idea or strategy up to a complete specification of a future product increment, including a binding cost and time schedule for its realization. Put simply, requirements engineering is the development of an idea into a concrete plan.

Possible products are all conceivable artificial, interactive systems in question.

A system is not just an IT system. Rather, it is a product in the holistic sense that can handle a business case or an industrial process in full. These systems include people who do work, machines and software. Examples of such systems are machines such as air traffic control systems, production lines or assembly machines and, of course, all service systems such as trading companies, banks or insurance companies.

Figure 2

Requirements engineering is not part of software engineering, but forms an interdisciplinary superstructure over the organization, hardware and software development.

We consider the business process modeling, which is widespread in the service sector, only as a possible means to specify the organization part of the second step in requirements engineering, the system design.

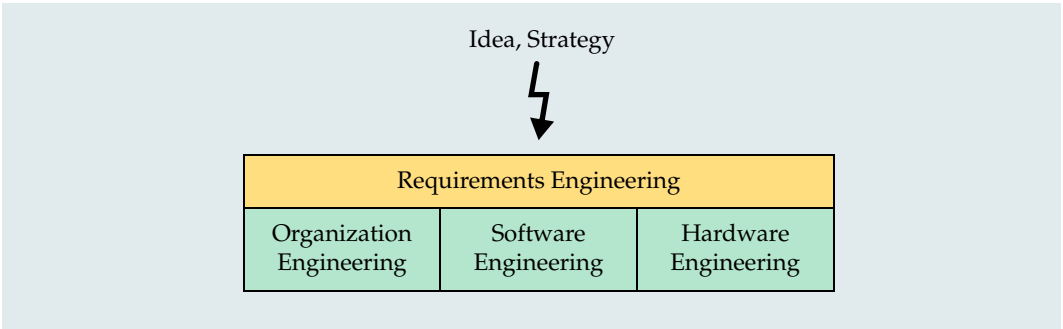

Figure 3

Requirements engineering is thus the first activity after the creation of an idea, whether we want to develop a software, seek an organizational optimization or change a socio-technical system.

A socio-technical system provides services through the collaboration of humans and machines as task carriers.

In the first step of requirements engineering, in requirements analysis (see Figure 4), the essential requirements are formalized. Essential requirements are those requirements that have to be fulfilled regardless of the choice of the future form of implementation.

Requirement analysis results in a complete, consistent, measurable, and interpretation-free specification of all functional and non-functional properties of the next increment of a system, regardless of whether these properties are met by humans, machines, or software.

Such a specification forms the stable basis for all future development opportunities. The contents of the deeper abstraction levels, e.g. the technology to be used or the organizational structure and processes of a company change at much shorter intervals than the underlying essence of the corresponding business or machine.

The essence is the reflection of business knowledge.

Without detailed (measurable) knowledge of the essence of an area of investigation, any factual discussion of the content, scope, cost and timing of a future solution is impossible. The participating discussion partners will have different ideas about the topic of discussion until a specification provides the necessary clarity.

The functional relationships are to be described in such detail that the reactions to all planned events can be predicted.

In the second step, the system design (see Figure 4), one or more technical or socio-technical implementations are designed. In the choice of options, among other things, the costs, the feasibility or socio-ethical criteria are considered. This specification is driven until the final criterion of requirements engineering is reached.

The completion criterion is a complete specification of the next increment of a product, including a binding cost and time schedule for its realization.

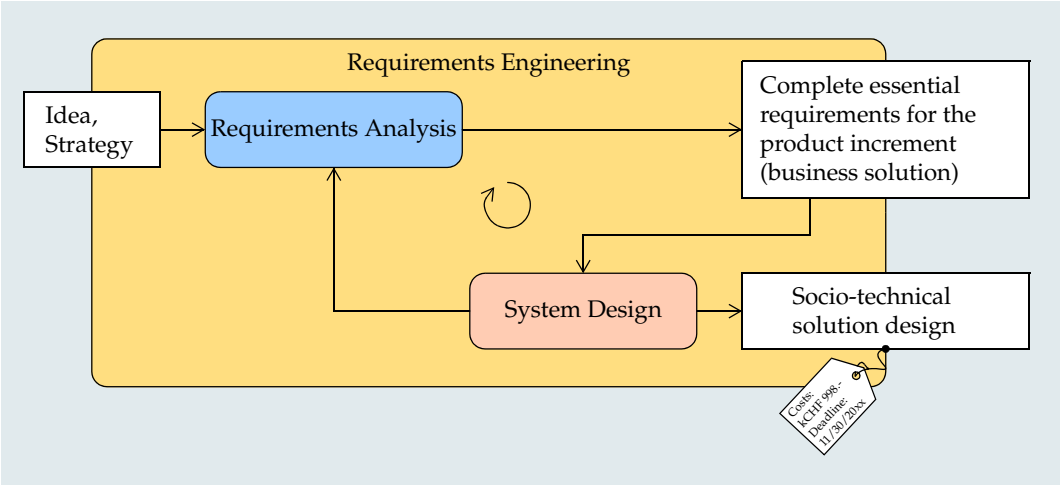

Figure 4

Effort

Creating a reliable plan involves considerable effort.

When trying to determine the effort share of requirements engineering in the overall project effort, many decision-makers fall into the realization trap. It is believed that the realization is the decisive part of a product development. And this might even be true as long as the products are passive systems. For example a house, a table or a sack of wheat.

However, as soon as the process knowledge for the provision of a service is transferred in whole or in part to a machine, the specification of this knowledge becomes a great challenge. If an activity can be executed by a machine (e.g. an IT system), then requirements engineering takes more and more space compared to the realization effort.

That is for a simple reason: The realization of a software project requires comparatively little effort. On the other hand, the description of the requirements is extremely complex because complex processes are to be mastered and specified. The actual work thus happens in the requirements analysis and in the system design. In the following figure, the areas within the product development correspond to the expected cost distribution. The share for requirements engineering amounts to at least 35% to 50% of the total development effort.

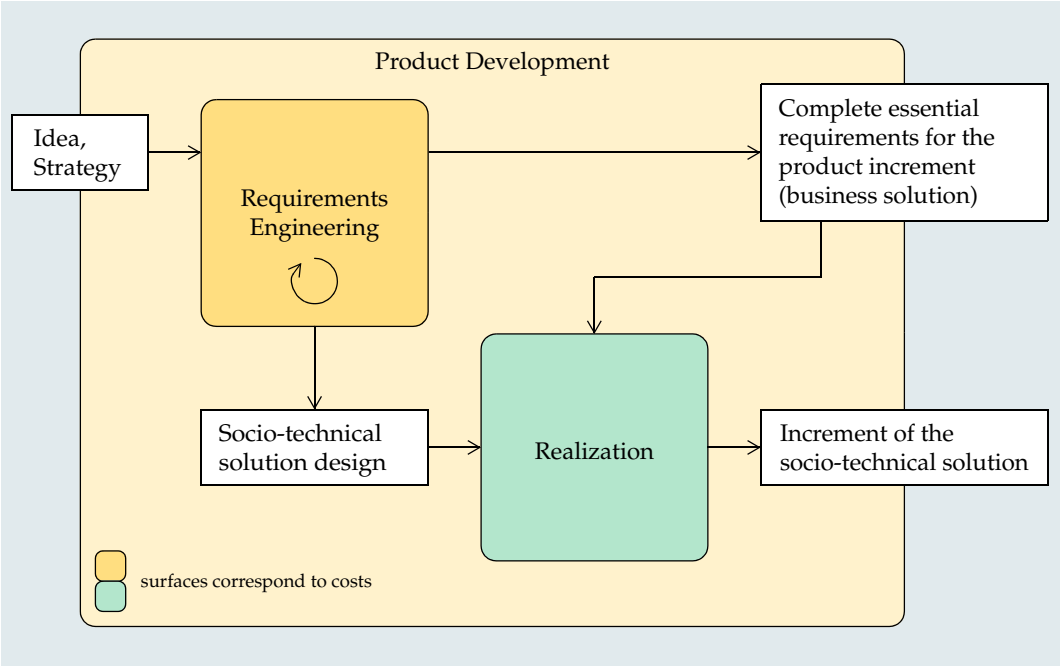

Figure 5

However, the expectations of decision-makers are often not oriented to the conditions of a computer science project. An accepted cost share of only 5% to 10% for requirements engineering is unfortunately still widespread in practice.

However, missing or wrong essential requirements cause a high correction effort. This effort increases dramatically, the longer these errors are not eliminated. Missing or incorrect requirements can still be solved inexpensively during the requirements analysis. But if these problems are first eliminated in the programming, then the troubleshooting is already 10 times more expensive. If the same issues are discovered and resolved at the time of acceptance, the troubleshooting will be at least 50 times more expensive (Source: Barry W. Boehm: Software Engineering Economics, Prentice-Hall, 1981).

For example, if we adopt ten fatal errors in a 100-page specification – the practice goes far beyond that – and we further assume that these errors could be remedied at the expense of one person-day in the requirements analysis, then the correction in the requirement analysis means an effort of 10 person-days. If the correction does not work, then they will be corrected in the programming – provided that they are discovered – at a cost of 100 person-days (5 person-months). If the errors remain undetected until acceptance, an effort of 500 person days (over two person years) is necessary to correct them. We do not want to address the correction costs during operation here.

If the effort for requirements engineering illustrated in Figure 5 is not pursued, the realization costs increase exponentially as a result of the inevitable errors.

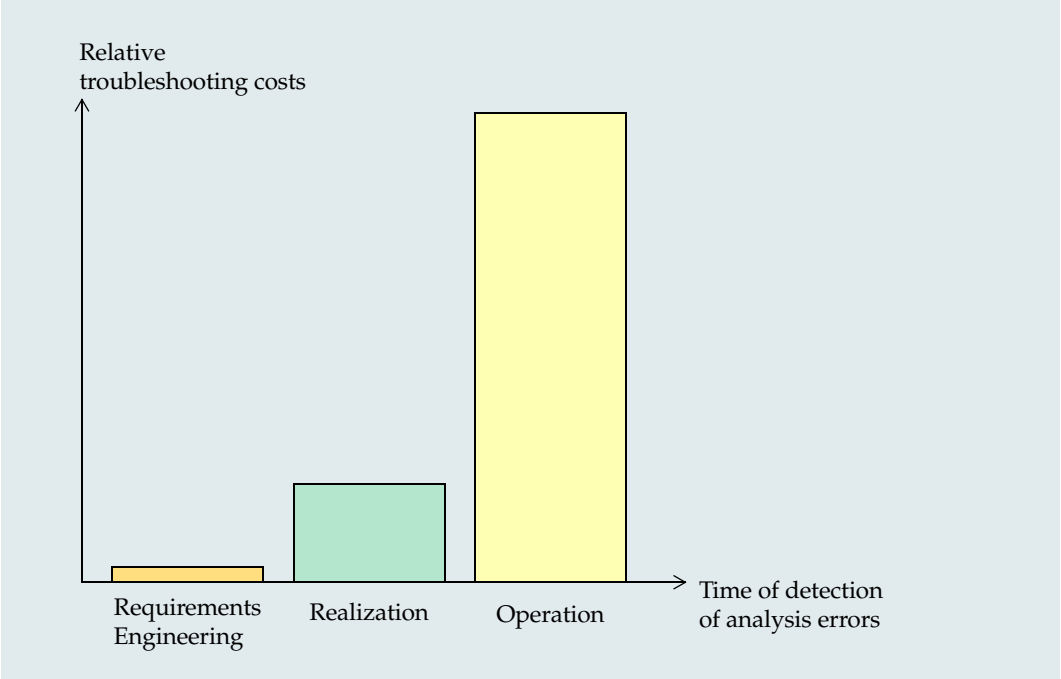

Figure 6

The budget can not be met. The decision-makers now have three choices: cancel the project, limit the functionality or extend the budget. At the time of this (mostly late) realization, large investments had already been made. The previous economic calculations are no longer correct.

Well-founded statements about the costs or the duration of a solution development can only be made after a complete requirements engineering of the next increment of a product.

With a complete and precise step-by-step analysis of the essential requirements, the entire project effort can be significantly reduced.

Clear targets

Most decision-makers say they have very clear ideas about the goals and characteristics of a business, and therefore about the technical or socio-technical systems to be implemented. On closer inspection, these clear ideas turn out, in the best case, to be what they really are – cloudy, superficial ideas.

Figure 7

On the basis of these ideas, a binding cost ceiling for their realization as well as an introduction time are then determined. With extensive but superficial essays and presentations, these sizes are underpinned – unfortunately never really comprehensible. Only in the context of a realization project the original idea will then be analyzed for the first time. However, according to our experience, the resulting process models or specifications still remain so superficial that each involved person develops a different idea about the future product. With a lot of effort, the original idea was illuminated from all sides, but it is still stuck at the stage of an idea.

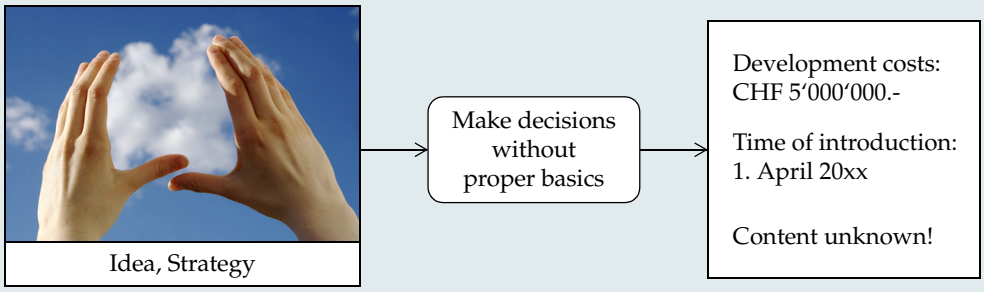

Figure 8

The information on the costs and the time of introduction are thus fictitious and correspond to a reflection of their own expectations – unfortunately without any factual substantiation.

Conclusion

Our requirements engineering is the development, formal specification and testing of essential requirements for an artificial, interactive system (requirement analysis) and the design of a feasible and economical solution (system design).

Requirements engineering forms an interdisciplinary superstructure over organizational, hardware and software development.

With the essential requirements (the domain model) the necessary basis for a next product increment is worked out, so that a concrete solution specification with binding cost ceiling and binding introduction date can be designed.

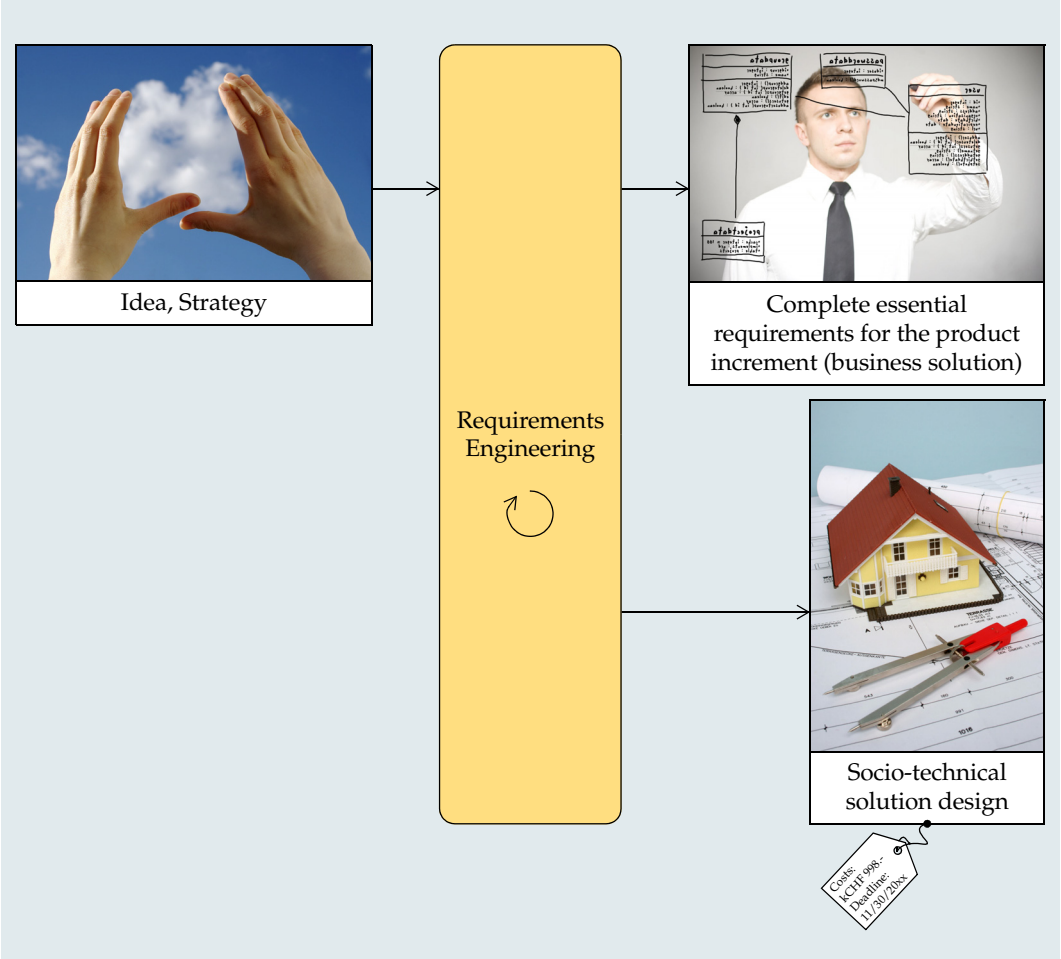

Figure 9

Imprint:

Bianchi & Partner GmbH
Sternenmatt 7
CH-6423 Seewen SZ

Phone +41 41 811 99 11
fachartikel@bianchi-partner.ch
www.bianchi-partner.ch

Version 4 of May 9, 2019