HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering

Using decision tools in deciding system product requirements: literature review and a behaviourally motivated lightweight tool

Jarmo Hurri

The thesis belongs to the special subject of software engineering
Supervisor professor Reijo Sulonen

Espoo, Finland 3.9.2000

HELSINKI UNIVERSITY OF TECHNOLOGY ABSTRACT OF THE LICENCIATE THESIS

Author and name of the thesis:
Jarmo Hurri
Using decision tools in deciding system product requirements: literature review and a behaviourally motivated lightweight tool

Date: 3.9.2000 **Number of pages:** ii + 75

Department: Department of Computer Science and Engineering
Professorship: Tik-76 Computer Science

Supervisor: Professor Reijo Sulonen

When system product requirements are decided, the objective is to set targets for properties of upcoming product releases. This problem has two important perspectives: it is both an investment problem and a goal setting problem for product development projects. From the investment point of view the objective is to select requirements that produce an optimal profit. From the point of view of goal setting, the objective is to set challenges for product development projects. The projects must strive to find efficient technical solutions to these challenges.

In decision making literature one can find a large set of more or less formal decision tools. The primary goals of these tools are the structuring of decision problems and assessment of impact of decisions. In requirements engineering literature one can find reports of applications of these tools. However, this literature is sparse: typically the objective of a research effort has been the application of one tool in a case study.

The most important contribution of this work is a synthesis of decision tools available for deciding product requirements. The tools reported in the literature are presented in a larger context, and their characteristics are analysed from several viewpoints.

In the analysis the existing set of tools has also been compared with the behavioural understanding of organisational decision making. In this analysis it has been observed, that theoretically advanced, mathematical methods seem to differ from typical organisational decision making with respect to several important dimensions. These formal methods have not been very successful among practitioners. On the other hand, those methods that have been successful in practical work seem to support better the natural ways in which organisations decide.

The second contribution of this work is a very simple behaviourally motivated lightweight decision tool. The objective of the development of this tool has been to support the natural decision making approach of product development projects. The objective of the tool is to make sure that the desirability, feasibility and risks of product requirements are assessed when requirements are decided.

Keywords: product development, requirements engineering, decision making, formal methods for decision making

TEKNILLINEN KORKEAKOULU **LISENSIAATTITUTKIELMAN TIIVISTELMÄ**

Tekijä ja työn nimi:

Jarmo Hurri
Using decision tools in deciding system product requirements: literature review and a behaviourally motivated lightweight tool
(Päätöstyökalujen käyttäminen systeemituotevaatimusten päätösongelmassa: kirjallisuuskatsaus ja käyttäytymistieteellinen kevyt päätöstyökalu)

Päivämäärä: 3.9.2000 **Sivumäärä:** ii + 75

Osasto: Tietotekniikan osasto
Professuuri: Tik-76 Tietojenkäsittelyoppi

Valvoja: Professori Reijo Sulonen

Systeemituotevaatimusten päätösongelmassa pyritään asettamaan tavoitteet tulevien tuoteversioiden ominaisuuksille. Ongelmalla on kaksi tärkeää luonnetta: se on sekä investointiongelma että tavoitteenasetantaongelma. Investointinäkökulmasta tavoitteena on valita sellaiset ominaisuudet, jotka tuottavat mahdollisimman hyvän voiton. Tavoitteiden asettamisen näkökulmasta pyrkimyksenä on valita sellaiset ominaisuudet, joihin tuotekehitysprojektin tulisi pyrkiä löytämään tehokas tekninen ratkaisu.

Päätöksentekokirjallisuudessa on esitetty koko joukko erilaisia enemmän tai vähemmän formaaleja päätöksentekotyökaluja. Näiden työkalujen pääasiallisia käyttötarkoituksia ovat päätösongelmien strukturointi sekä päätösten vaikutusten arviointi. Vaatimustenhallintakirjallisuudesta löytyy joukko raportteja näiden menetelmien soveltamisesta tuotevaatimusten päätösongelmaan. Kirjallisuudessa työkalujen käsittely on kuitenkin hajanaista — tyypillisesti yksittäisessä tutkimuksessa kokeillaan yhden menetelmän toimivuutta jossain sovellusprojektissa.

Tämän työn tärkein kontribuutio on synteesin tekeminen tarjolla olevista systeemituotevaatimusten päätöksentekotyökaluista. Kirjallisuudessa raportoidut työkalut esitetään laajemmassa kontekstissa, ja niiden ominaispiirteitä analysoidaan työssä useista eri näkökulmista.

Analyysissä on myös verrattu olemassa olevia menetelmiä käyttäytymistieteelliseen käsitykseen organisaatioiden päätöksentekotavoista. Tässä analyysissä on havaittu, että teoreettisesti kehittyneet, matemaattiset menetelmät näyttävät poikkeavan useilla tärkeillä akseleilla siitä lähestymistavasta, jota organisaatioissa tavallisesti käytetään. Nämä formaalit työkalut eivät ole menestyneet erityisen hyvin käytännön työssä. Toisaalta taas käytännön työssä hyvin menestyneet mutta teoreettisesti heikommat päätöksentekotyökalut tuntuvat tukevan paremmin organisaatioiden luonnollista päätöksentekotapaa.

Tämän työn toinen kontribuutio on erittäin yksinkertainen ja kevyt käyttäytymistieteellinen päätöksentekotyökalu. Työkalun kehittämisen tavoitteena on ollut tukea tuotekehitysprojektien luonnollista päätöksentekotapaa. Työkalun tavoitteena on varmistaa, että tuoteominaisuuksien haluttavuus, toteutettavuus sekä riskit kartoitetaan tuotepäätöstä tehtäessä.

Avainsanat: tuotekehitys, vaatimustenhallinta, päätöksenteko, päätöksenteon formaalit menetelmät

Preface

Requirements engineering is studied at Helsinki University of Technology in the QURE (Quality through Requirements) research project. This project is supported by TEKES (National Technology Agency) and a number of Finnish industrial partners. In autumn 1998 I had the pleasure to join the QURE project in its early stages in order to make my own contribution.

I would like to thank professor Reijo Sulonen for providing me with an opportunity to work in the project and prepare this thesis, and for his valuable comments that improved this end result. I would also like to thank professor Jyrki Kontio for his assistance in many critical situations during the project. My special thanks go to the QURE project group (Marjo Kauppinen, Sari Kujala, and Virve Leino) for their persistence and for creating a lively forum for the discussion of issues pertinent to the theme including, but not in any way limited to, the following: requirements engineering, product development, requirements engineering processes, user studies, traceability, research methods, project management, group work, motivation, and an endless list of issues concerning life outside the project. I would also like to thank Jenni Taavitsainen for providing the absolutely necessary distractions.

In Helsinki, 3.9.2000

Jarmo Hurri

Contents

CHAPTER 1

Introduction

The decision-making process has provided a fruitful framework, perhaps one of the most fruitful, within which to view the operation of private and public organizations. Both in describing what goes on in organizations and in making prescriptions for the improvement of their functioning, some of the most critical processes are the ways in which they identify problems, formulate them, generate alternative strategies for dealing with them, and select and implement those strategies.

— Herbert Simon in introduction to [**91**]

1. Is this a relevant research problem?

1.1. The problem: use of decision tools to facilitate deciding system product requirements. The problem addressed in this work is the *use of decision tools to facilitate deciding system product requirements*. Before the relevance of the problem can be considered we first have to describe shortly what it is that we are talking about.

In this work *system products* are products that contain electronics or software, consist of a number of interconnected components, and do not include human work processes in primary functions.[1] Examples of such products are software programs, mobile telephones, and modern elevators. Examples of products that by this definition are not system products include furniture, television programs, and banking services that are not automated.

Product requirements are objectives which have been set for future releases of a product, and which are relevant to stakeholders outside the developing organisation. For example, for a laptop computer such stakeholders might include users, legislators and computer service companies. Examples of requirements in this case might include the need to have an integrated mouse, the need to fulfil certain standards concerning electromagnetic interference, or the need to include a diagnostic program. Examples of objectives that are not product requirements in this case could include the use of a certain signalling mechanism between the integrated mouse and the main module, or the intention to write the diagnostic program in C++. If these objectives do not concern external stakeholders, they are not product requirements. Sometimes it is difficult to draw the line, since we may not know which properties of the product are relevant to the stakeholders, or become relevant in the future.[2]

Product requirement decision problems are problems in which we have to choose whether or not to include a certain requirement into future releases of the product, or have to choose goal values for measurable properties of the product. For example,

[1] This is not a rigorous formal definition, and finding one would not be very easy.

[2] As we can see, the definition of requirement depends on how we define the set of stakeholders. If we include potential new employees in the group of stakeholders, then programming language may well become a requirement.

we may have to decide whether or not to include the integrated mouse in the laptop, or decide the size of the display of the laptop.

Decision tools are techniques that are used to facilitate decision making by analysing the structure and impact of decisions. Examples of such tools include objective trees and networks, which are used to study relationships between objectives, and linear weighted methods, which are mathematical models that try to relate individual decision criteria to the overall goodness of a solution.

Now that we have defined the problem, we can consider whether the problem is relevant enough to deserve attention from the research community. Because this is applied research, we have to ask whether there is a need for product requirement decision making tools among practitioners. This question is answered in two steps by considering the importance of

- product requirement decision problem
- decision tools for the problem.

1.2. The importance of product requirement decision problems. Usually in product development projects more potential product requirements emerge than can or should be implemented in the next release of the product. All suggested requirements are not equally important. Some of them produce more revenue than others, some are more expensive to implement or manufacture, some are more critical for deciding the product architecture etc. Also it is easy to set economically infeasible goals for those "quality requirements" that will inevitably be properties of the product (such as expected lifetime).

Such considerations are central when the identity of the product develops. The most crucial decisions determine what is often called the *product concept* [**81**]. In addition to the major characteristics that give rise to the product concept a product can contain hundreds of other characteristics that give additional functionality, quality, and appeal to the product.

It is possible to let the set of product requirements evolve continuously during a development project, and define the exact behaviour of the product as the project ends: what you get is what came out. However, there are a number of reasons for deciding the set of requirements in a more controlled manner, especially in large development projects with tight schedules.

- Requirements are investment decisions that may require careful examination.
- External factors, such as need to close a sales deal before starting the development, may force us to decide system behaviour before starting the actual development work.
- Requirements are a set of goals for later stages of product development. If the goals are erroneous, a great deal of work can be wasted.
- In large development projects work has to be split. It may be easier to split the work if we have concrete goals. This is especially important in concurrent engineering, where different tasks of product development are run in parallel.
- If work is split, some mechanism should be established to form a coherent whole.
- Concrete goals are the basis for estimating schedule and costs correctly [**8**].

If we want to control the evolution of product requirements the set of requirements must be fixed or constrained somehow at some point of development. This leads to decision problems. For those requirements that either are or are not present in the product, such as the integrated mouse in the laptop, the following questions demand an answer: which requirements to include, which to discard, which are undecidable at this point of development? For those decisions that have a real-valued answer, such as the size of the display, it is a question of the level of the requirement: what is the final value, what is the range of acceptable values at this point of development?

Product development projects face such problems constantly. They can either address them ad hoc, or try to use some common language or framework with which and within which the problems can be analysed and discussed. This is where decision tools step in.

1.3. The importance of decision tools. The golden age of decision sciences was started by the work of von Neumann and Morgenstern, who developed a theory for making consistent choices under uncertainty [**88**]. This work resulted in the development of *decision analysis*, which is a fairly sophisticated mathematical machinery that can be used to analyse decisions. However, it can be stated with reasonable certainty that this tool is not used widely in deciding product requirements. For many such decisions the machinery is too heavy, since the analysis typically takes a lot of time and usually requires the assistance of a trained analyst [**69**]. But even when the product concept is selected — this is the largest of such decision problems — the academics suggest using simple table methods [**81**].

One could of course conjecture that there is not enough knowledge about these tools in the new product development community. However, decision analysis has been available in university textbooks at least since the 60's. In Finland every graduating MBA is nowadays exposed to decision analysis. Still, its use in deciding product characteristics is non-existent.[3] The cause of this discrepancy between theory and practice is an open issue. It may be that the way uncertainty is modelled in decision analysis does not fit the problem, or that the needed data is unavailable or too uncertain, or the amount of work required by the analysis is too large. But these are just examples of available hypotheses.

Decision analysis is of course not the only decision tool that could be applied to the problem. As in many other research areas, the linear simplifications form an important practical set of methods. For example the analytic hierarchy process (AHP) [**67**] is based on linearity. It has been around since the 70's, and has been applied to the problem of deciding product requirements [**39**]. No extensive empirical evaluation about the applicability of the tool has been done. To our knowledge the tool has not spread widely among practitioners.

This argumentation has its weak points. The use of such evolutionary justification may miss both dynamic and "random" components that can affect the way a technology spreads. It takes time for a technology to spread among practitioners. Also seemingly "random" factors such as the availability and goodness of teaching material, the number of motivated spokesmen, the suitability of first test cases etc. can affect the way a technology is taken into use. Our assumption here is that decision analysis and other rigorous mathematical methods discussed in this work have been available long enough, and that there have been enough enthusiastic people and good test cases for a good start. Also, as we shall see later on in this thesis, we can identify various attributes of these tools which do not seem to follow the way people actually make decisions.

The previous discussion might depress a researcher looking for a solution, but luckily there is a success story. Quality function deployment (QFD) is a "set of

[3]Another point of view can be borrowed from general R&D research. In his paper about the organisation of R&D Holmstrom discusses the use — or rather the lack of use — of the net present value as an investment criterion [**35**]:

> Those who think the problem is in management practice rather than the theory should be reminded of the speed with which financial markets have adopted modern asset pricing theories. Understanding net present value analysis is trivial compared with learning option pricing and valuation of other derivative securities.

communication and planning routines" [**32**], a "system to assure that customer needs drive the production process" [**74**], that is used in practice to facilitate decision making. QFD was developed in shipyard industry in Japan, and it has spread to other industries and continents. Some reports of the effects of the use of QFD are quite impressive. For example, Sullivan reports that Toyota reduced the start-up costs of automobile development by 61% when they deployed QFD [**74**]. QFD is probably the most famous quality tool at the moment.

QFD seems to enjoy limited success in systems development. Haag and his colleagues report in a survey that 6 out of 37 of the major software companies contacted in the survey used QFD in software development [**29**]. The most important areas of application were operating system software and embedded software. The survey indicated that the most knowledgeable QFD users in these companies saw QFD having a number of advantages when compared with traditional system development methods: better communication with users, management and technical personnel, better conformance to user requirements, less errors in developed systems, reduced programming time, and more consistent and complete documentation. However, except for the objective of producing good documentation, the differences were not dramatic.[4] In addition, all companies that were using QFD had adopted the larger Total Quality Management (TQM) paradigm in other operations before they adopted QFD in software development.[5] So Haag and his colleagues actually suggest that TQM is a necessary prerequisite for successful use of QFD in software development.

Looking at the direct reports about the use of QFD in system development (the article by Haag et al. is indirect information) it seems that the application of QFD to system development is somewhat problematic. System and software requirements are often binary, not real-valued or measurable. Also finding technical objectives for non-tangible software systems can be difficult [**10**]. One problem is that in (radically) new product development first versions of the product can be more experimental, and the future user community may be unknown [**39**]. It may not be feasible to find a set of goals that describe the goodness of the product — the way users see and value the product is found later when the product concept matures. This problem is also related to the problem of finding an appropriate level of abstraction to express the goals [**39, 10**]. The end result of these problems is the return to ad hoc methods or a search for new methods. For example, in Karlsson's case at Ericsson the development group abandoned QFD and tried to develop a new method. In our work with industrial partners we have had discussions which revealed that QFD had been evaluated, but it had been rejected after problems such as those stated above were discovered. All in all our understanding is that although QFD has been evaluated in a number of organisations, it has not been adopted in large scale outside the set of companies that follow the TQM paradigm in other operations as well.

So, to summarise, it seems that currently there is no widespread tool to facilitate deciding system product requirements. I suggest that this is not caused by the lack of knowledge among practitioners, but by the inadequacy of available tools. Although QFD does not seem to be widely applicable in system development, its success story provides a source of inspiration.

[4] On a five-point Likert scale (1: result not being achieved, 5: result being achieved very well) differences in mean for documentation was 1.2, and for other objectives mentioned here between 0.4 and 0.7.

[5] Total Quality Management consists of three parts: a planning methodology called Hoshin planning, QFD, and statistical process control [**29**].

2. The contribution of this work

The primary intended audience of this text is the requirements engineering research community, especially those researchers who are interested in the applicability of decision tools in deciding system product requirements. This work contributes to the requirements engineering research field in two ways.

- An overall picture is built to clarify the connections between different decision making tools and to describe how these tools can be applied in product requirement decisions. I also identify central attributes of the tools and use these to compare the tools.
- A new simple tool is proposed. The development of the tool has been inspired by the behavioural approach to decision making.

First of all, it is my firm opinion that the field of requirements decision making needs synthesis and clarification. In this section I will make a very rough outline of what has been done before. The literature written on the subject is described more thoroughly in subsequent chapters.

The most prolific researchers in this area have been Karlsson and Ryan along with a couple of colleagues. The most thorough description of their work is the licenciate thesis by Karlsson [**39**]. Most of the articles describe the same body of work [**40, 42, 43**]. The work is continued in [**41**]. The work of Karlsson and Ryan and their colleagues is based on three tools: QFD, AHP and cost-effectiveness analysis. Their contribution has been an important step in introducing some available practical decision tools into the requirements engineering community.[6]

Several other researchers have made an effort to bring decision tools into the field. They introduce risk management [**27**], decision structuring tools [**26**] and decision analysis [**13, 92**]. Some QFD-like tools are also available in general product development books, such as [**81**]. However, current research is missing an overall picture of the concepts and tools. The literature is sparse, and there is no synthesis that would build an overall picture of what is going on.

The most important contribution of this work is to build this overall picture. Actually even though the decision making research area is much more established than requirements engineering, finding an overall picture in that field was quite a challenge. The article by Schoemaker and Russo [**69**] and standard textbooks [**18, 83, 47, 88**] were a reasonable starting point. In addition to building an overall picture of decision making tools, an overview of product development is also provided, and the two worlds are bound together.

The second contribution of this work is the development of a tool for analysing and deciding product requirements. I have identified central attributes of decision tools, and assessed existing tools along these dimensions. Some reasons as to why the use of tools is so limited is suggested by the comparison of the existing set of tools against current understanding about the ways in which decisions are actually made in organisations. The goal of the development of the new tool has been a lightweight tool that would support the natural way of thinking. Currently available methods are sophisticated, elegant, theoretically correct et cetera, but they are not being used.

So I have made an attempt to improve the situation, and the approach has been tested in two cases. I hope that the method is powerful enough to provide insights for the decision maker, but simple enough so that the benefits outweigh the the

[6] The term *requirements engineering* is used in this work because it is an established concept in the field. Personally I think that requirements engineering as it is currently understood is pretty far from an *engineering* profession. An equivalent engineering profession would be the study of human mating behaviour in order to find out what chemical compounds chemical engineers should produce for cosmetics.

costs of using the method. But the truth is that there is no thesis associated with the method which we would be defending. The tests we have done do not constitute a rigorous evaluation. There is a hypothesis, and future will show whether it has to be rejected.

3. Focus: structuring objectives and assessing impact

Basically there are two major themes in this work: system product development and decision making. These themes converge in the problem that is studied here: system product requirement decision making. Both research areas, system product development and decision making, are extremely large, and the work has to be focused. The focus taken in this work is that of decision tools.

Decision tools have two primary goals: structuring the problem, and assessing the impact of different alternatives. We can structure

- relationships between objectives (objective trees and networks)
- causal relationships between decisions, chance events, and outcomes (influence diagrams)
- dynamic relationships between decisions and chance events (decision trees).

We can assess the impact

- directly (scoring, different scales)
- via subcriteria with linear methods (AHP, conjoint analysis)
- via subcriteria with nonlinear methods (decision analysis).

Of course when the problem is analysed a secondary effect can be enhanced communication, re-examination of problem definition etc. But the primary use is in structuring and assessment of impact.

Since the topic of this thesis is the use of decision tools, this naturally focuses our attention to structuring the problem, and assessing the impacts. However, with respect to structuring methods we have narrowed our focus still a bit: this work focuses on structuring objectives and their relationships. If you compare this focus with the goals of structuring methods listed above, you will notice that the relationship between decisions and chance events has been discarded. By doing so I actually follow tradition in this area, and this is a sensible approach since we are performing a literature review. Most research in this area discards chance events, and there is no research that would apply methods like decision trees or influence diagrams. At first sight this seems somewhat controversial, because the world of product development is truly a world of uncertainties. But decision trees and influence diagrams seem to work best in situations where possible states of world can be identified, and their number is limited. It may well be that such an analysis is not feasible in product development. Using the classification of Courtney and his colleagues [**20**], uncertainties in product development include a range of possible futures, or true ambiguity, where forecasting may be downright impossible. But this is just one hypothesis. Anyhow, when a synthesis of the field is made structuring methods that concern chance events are discarded, since the research reviewed here does not apply them. But we will return to questions about uncertainty later in this work.

To understand the things that we are discarding by selecting this focus, let us look at a model of what happens in decision situations. Based on the work of several decision theorists Carroll and Johnson have identified a seven stage model of decision making [**16**]:

- recognition
- formulation
- alternative generation
- information search

- evaluation / choice
- action / feedback

Structuring and assessment of impact (that is, decision tools) are mostly applicable in formulation and evaluation / choice stages. So these are the stages we will be focusing on in this thesis. For example, this thesis discards the extremely important area of information search, which includes activities like user studies and competitor analyses.

4. Chapters of this thesis

The material in this thesis has been divided as follows:

The context: product development and decision making: The purpose of this chapter is to introduce the world in which the activities take place, product development, and the activity in question, decision making. I describe shortly what product development is about, and how researchers have tried to conceptualise this world. Different viewpoints to decision making are described; not all decision making research belongs to the school of rational thought.

The decision problem: In this chapter the product requirement decision problem is defined: what is the problem, and what is its intuitive solution.

Decision tool review: This chapter contains a description of the different decision making tools that can be used to solve the problem. We review structuring methods, such as objective trees and networks, and methods for analysing the impact of decisions, such as ordinal, linear and nonlinear impact analysis, and cost-effectiveness and cost-benefit analysis. Also important attributes of tools are identified, and the tools are compared along these attributes.

A behaviourally motivated lightweight decision tool: This chapter introduces SATIRE, which is a simple behaviourally motivated decision tool that can be used to facilitate deciding system product requirements.

Conclusions and future work: The last chapter draws some conclusions from this work and depicts directions for future research in the area.

CHAPTER 2

The context: product development and decision making

1. Product development

1.1. Product development and its complexity. In the business world research and development is the search for new profitable ideas. In modern companies R&D activities can have a number of purposes [**14**]:

- developing new products or improvements to current products, that is, *product innovations*
- developing new processes or improvements to current processes, that is, *process innovations*
- creating new technologies which will enable a number of product and process innovations
- developing core competencies.

This thesis focuses on the first activity: creating new products or product improvements. This activity is called *new product development*, or NPD for short. However, we must note that the purposes listed above are often related: for example Wheelwright and Clark suggest that projects aiming at breakthrough products should be given freedom to also change the core processes [**90**].

This section is a short introduction into the world of new product development. There are two closely related restrictions that affect the usefulness and breadth of this introduction.

- The world of new product development is inherently extremely complex. At least currently there is no model that would explain all the complexities of this world.
- There exists an enormous amount of literature on new product development. Only a tiny fraction of what is available can be covered. This short introduction is based mostly on the article by Brown and Eisenhardt [**11**], and books by Ulrich and Eppinger [**81**], Twiss [**80**] and Burgelman, Maidique and Wheelwright [**14**].

To begin, we might consider for a while the reasons for the complexity and difficulty of product development. Some reasons are given by Ulrich and Eppinger [**81**, page 6].

trade-offs: Product cost, timetable, features et cetera are typically all related: if you change one, at least one of others will change as well.

dynamics: The development is done in a dynamic environment, with changing demand, competition, technologies etc. This effect is pronounced in NPD since it is typically based on new opportunities, whether they are brought by new technology or changes in the society.

details: In addition to making the correct large scale decisions it is often also critical to make the right small scale decisions. For example, requirement decisions spread in many different places: design, manufacturing, all over the world with shipped products etc. Errors that seem relatively small can in fact be costly in the long run.

time pressure: Market entry is important. Risks must be taken to be among the first ones to enter.

We might add a couple more reasons to this list.

uncertainty: Uncertainty is inherent in the development of something new. The number of unknowns can be overwhelming: design and production problems, users, competitors, legislation etc.

creativity: Creativity is also inherent in new product development. Creativity in organisations seems to be a tricky business, requiring appropriate challenge, freedom, enough resources, a motivated and cooperative cross-functional work-group, supervisory encouragement, and top-level organisational support [**1**].

global markets and competition: Both the standardisation of products to global markets [**89**] and increased global competition require extensive analysis and vision.

cross-functional tasks: The successful development of new products typically requires cross-functional communication between marketing, sales, manufacturing and R&D [**80**, pages 8–12].

Although the world of NPD is inherently complex and even chaotic, and researchers and practitioners recognise this, the conceptualisation of NPD is seen as a very important task [**11, 31**]. The next section discusses various efforts to conceptualise product development.

1.2. Conceptualisations of successful product development.

1.2.1. *Rational, communication and problem solving paradigms.* Basically any development activities that result in the birth of a new product can be called new product development. Therefore conceptualisations of NPD activities have focused on describing models of *successful* NPD. In their excellent review article Brown and Eisenhardt have studied this vast research area, and have identified three different research streams that conceptualise successful NPD activities in different ways [**11**]:

- product development as a *rational plan*
- product development as a *communication web*
- product development as *disciplined problem solving.*

Table 1 shows some central concepts from these streams.

The conceptual view that sees development as a communication web is basically a focused viewpoint that can coexist with the other views, although it does add interesting flavours such as the role of the project manager as a negotiator for project resources. So let us focus on the other two views for a while.

- The rational plan emphasises superior product (cost, quality, uniqueness, fit with competencies), attractive market and rational organisation of NPD activities. Top management support is required.
- The disciplined problem solving paradigm emphasises harnessing the problem solving power of product development personnel to achieve a coherent product vision.[1] The personal influence of top management and a heavyweight project manager is crucial in setting a demanding strategic goal developing the product vision and subtly influencing its implementation, while the product team has great autonomy in organising itself and deciding on day-to-day activities [**75**].

[1]The "product vision" can be something as high-level and abstract as the "rugby player in a business suit" used by the Honda Accord development team in the beginning of the 1990s [**17**].

	rational plan	communication web	disciplined problem solving
key idea	success via superior product, attractive market, rational organisation	success via internal and external communication	success via problem solving with discipline
theory	mostly atheoretical	information and resource dependence	information including problem solving
research methods	bivariate analysis; single informant; many independent variables	deductive and inductive; multivariate; multiple informants	progression from inductive to deductive; multiple informants; single industry, global studies
product	product advantage — cost, quality, uniqueness, fit with core competencies	—	product integrity — product vision that fits with customers and firm
market attributes	size, growth, competition	—	—
senior management	support	—	subtle control
project team	cross-functional, skilled	—	cross-functional
communication	high cross-functional	high internal, high external — various types and means	high internal
organisation of work	planning and "effective execution"	—	overlapped phases, testing, iterations, and planning
project leader	—	politician and small group manager	heavyweight leader
customers	early involvement	—	—
suppliers	early involvement	—	high involvement
performance (dependent variable)	financial success (profits, sales, market share)	perceptual success (team and management ratings)	operational success (speed, productivity)

Table 1. Main properties of three conceptually different NPD research streams [**11**]

Both paradigms emphasise cross-functional teams and involvement of suppliers.

What does the existence of these two paradigms tell us? The most important points are probably that, on one hand, there is no consensus on how products should be developed successfully, but on the other, at least currently researchers seem to agree on some crucial ingredients. Not surprisingly, whether you call it a coherent product vision or a superior differentiated product, the product seems to be the most important thing. However, as could also be expected, no algorithm exists for inventing that product. Cross-functional teams seem to be important. But opinions on how to manage these teams disagree: one view emphasises planning and coordinating efficient execution, while the other emphasises "subtly" harnessing the problem solving power of team members to work towards the product vision. In addition, one view emphasises planning and execution, while the other emphasises iterativeness.

To bring the discussion closer to the subject of this thesis, consider how the goodness of a requirement could be determined in these two paradigms.

- In the rational plan paradigm a requirement is good if the product gets closer to being a superior product, targets a large and growing market, and is a basis for the efficient execution of subsequent development stages.
- In the disciplined problem solving paradigm a requirement is good if it fits the product vision and can be implemented by harnessing the problem solving power of the team.

These are two somewhat different views about the goodness of a requirement. The disciplined problem solving paradigm, which relies on the subjective product vision[2] of the heavyweight project manager, and relies heavily on the problem solving capabilities the development team, is probably harder to model and conceptualise. As a consequence, the rational plan paradigm is probably better suited to the use of decision tools.

In the next section we will take a closer look at some of the activities that take place in product development. This more detailed conceptualisation is probably better suited to the rational plan paradigm, but similar activities take place in all product development activities.

1.2.2. *Product development processes.* The more detailed conceptual model that is used to study the rational plan paradigm is from the book by Ulrich and Eppinger [**81**]. Their book presents a phase model of new product development projects, and a set of structured methodologies for completing new product development activities. The viewpoint is that of a *development process and project.*[3]

Ulrich and Eppinger divide the new product development process into five different phases:[4]

1. concept development
2. system-level design
3. detail design
4. testing and refinement
5. production ramp-up

Table 2 summarises the activities of each of these phases. One can immediately see the enormous amount of different activities that take place in new product development.

Now let us take a look at how requirements fit into this process. Especially in the rational plan paradigm of product development the requirements of a product have two roles.

---

[2]See footnote 1 on page 9.

[3]For additional information from the viewpoint of *development organisation* see the book by Twiss [**80**], and from the viewpoint of *company-wide strategic technological considerations* see the book by Burgelman et al. [**14**].

[4]In literature the number of phases and their names differ. For example, in [**31**] Hart and her colleagues use six phases: idea generation, concept development, build business case, product development, market testing, market launch.

	concept development	system-level design	detail design	testing and refinement	production ramp-up
marketing	define market segments, identify lead users, identify competitive products	develop plan for product options and extended product family	develop marketing plan	develop promotion and launch materials, facilitate field testing	place early production with key customers
design	investigate feasibility of product concepts, develop industrial design concepts, build and test experimental prototypes	generate alternative product architectures, define major subsystems and interfaces, refine industrial design	define part geometry, choose materials, assign tolerances, complete industrial design control documentation	do reliability testing & life testing & performance testing, obtain regulatory approvals, implement design changes	evaluate early production output
manufacturing	estimate manufacturing cost, assess production feasibility	identify suppliers for key components, perform make-buy analysis, define final assembly scheme	define piece-part production processes, design tooling, define quality assurance processes, begin procurement of long-lead tooling	facilitate supplier ramp-up, refine fabrication and assembly processes, train work force, refine quality assurance processes	begin operation of entire production system
other	finance: facilitate economic analysis, legal: investigate patent issues	finance: facilitate make-buy analysis, service: identify service issues		sales: develop sales plan	

TABLE 2. A generic product development process [**81**]

- They define the behaviour, look and feel of the product to the stakeholders outside the development organisation. These stakeholders include users and customers.
- They define the plan for subsequent stages of development, that is, requirements are used to guide development activities such as architecture design and implementation.

We could say that requirements are both an external and internal plan. Looking at these two roles it is easy to understand why requirements are so central in the rational plan paradigm. For example, when Cooper listed his "eight actionable critical success factors" [**19**], the number one factor was "solid up-front homework — to define the product and justify the project".[5]

Remembering this twofold role of requirements and looking at Table 2, we can see that when decisions about requirements are made during the first stages of development, an enormous number of different viewpoints have to be considered. Requirements are relevant for marketing, development and manufacturing aspects. Their role is more crucial during the first planning phases, concept development, system-level design and detail design, but they are also needed during the testing phase to evaluate the implementation. Because the requirements are both an external and internal plan, when requirements are decided, we should consider their impacts to users, customers, design, marketing, manufacturing and testing. Furthermore, if the view of development activities is non-iterative, requirements may have an even bigger role as *the* plan.

We will return to these issues in Chapter 5 when we discuss assessing the goodness of requirements. But now we will move on to the second theme of this work: decision making. The name "rational plan" that has been given to the paradigm that emphasises planning and effective execution implicitly associates rationality with careful planning and assessment before actions are taken. Rationality pops up quite naturally in decision making, since typically the objective of decision tools and other prescriptive decision research is to help the decision maker or organisation behave more rationally. But rationality may take many different forms, and people and organisations may prefer other forms of rationality than the planning approach.

2. Decision making and rationality

2.1. Descriptive, normative and prescriptive decision research. There are basically three different types of decision research, which illustrate three different approaches to decision making [**6**].

- The *descriptive* view studies how and why people actually make decisions. Descriptive studies have much in common with psychology and sociology.
- *Normative* research studies how "idealised, rational, super-intelligent people should think and should act". Normative studies are theoretical and mathematical, and they have a common theoretical background with economics.
- Those with a *prescriptive* point of view are interested in what can actually be done to make better decisions. The research starts from descriptive results and often takes its goals from normative results, but has to relax the idealised assumptions of normative studies. This is very much like consulting work.

This research is prescriptive research, which typically has to borrow results from both descriptive and normative schools to understand where people are now and to know where they would like to be heading.

[5]It is interesting to notice that "seek differentiated, superior products" was number three in Cooper's list, with "build in the voice of the customer" taking place number two.

The objective of this section is not to go through descriptive and normative research areas in detail. Several aspects of normative research are covered below. For introductions into descriptive research see [**38, 34**]. Here we will discuss the role of *rationality*, which is a general theme that ties all of the three approaches together. When prescriptive decision research tries to help people decide "better", the meaning of "better" is often explicitly or implicitly closely related to rationality. The underlying assumption is that current behaviour is not rational, and prescriptions should help the decision maker to become more rational, to become more like the decision maker in the normative approach. But what is rationality?

2.2. Rationalities.

> *In the face of complexity, uncertainty, conflicting objectives, and multiple decision makers, the typical response, at least of Western man, is to attempt to be rational. We seek a rational framework to help us think through our decisions. ... Rationality is a concept that we intuitively look for; it is, however, more difficult to define than might appear at first sight.*
>
> — Stephen Watson and Dennis Buede in [**88**]

Theoretically most business people would probably agree if we said that product requirement decision problems should be studied as *microeconomic* problems. Microeconomics studies the allocation of resources in the world of consumers and firms (see, for example, [**71**] for a good exposition of the basics, and [**51**] for more advanced microeconomics). The wellbeing of firms is measured with profits, and the wellbeing of consumers is measured with utility, which is a function of the different goods they have.

Rationality can be defined fairly neatly in a microeconomic world: if a decision problem is defined and consequences of possible actions are known, rational firms choose actions that yield the highest profits. Similarly rational consumers choose actions that yield the highest utility. This is what might be called "classical domain rationality". However, in the real world people may for example not know what their utilities are or may enjoy a certain decision making process more than the end result.

Alternative rationalities suggest that individuals, collectives or systems are behaving rationally, but that rational behaviour can be limited, focused or biased, or they challenge the whole meaning of the word "rational" [**52**]. Probably the best known example of alternative rationalities is *bounded rationality*, for which Simon received the Nobel price in economics. Basically bounded rationality focuses on the effects that the limits of knowledge and information processing capabilities of humans have on decision making and "rationality" expressed by humans [**72**].

Some of the most central themes in bounded rationality are the replacement of abstract high level goals with more concrete *subgoals* and the tendency to *satisfice* instead of optimising.

- "When the goals of an organization cannot be connected operationally with actions, then decisions will be judged against subordinate goals that can be connected." [**72**, page 353]
- Satisficing means that instead of looking for an optimal solution, the decision maker has a (possibly dynamic) threshold called *aspiration level* which he or she wants to exceed. When the threshold is exceeded the decision maker terminates the search and selects the alternative.

From our point of view both phenomena are very important. First of all, some available decision tools try to connect requirements to added income and cost, or to

rationality	properties	examples of consequences
calculated rationalities		
bounded	difficulties in considering all alternatives and all information	concrete subgoals, satisficing, step utility functions, simple search rules, uncertainty avoidance
contextual	influence of actors and other social and cognitive structures in choice situation	interrelationships between factors depend on their timing (simultaneity)
game	actors with self-interest act in relation to each other	coalition formation, sequential goals
process	decisions depend on decision process	outcomes are secondary, pleasure of organising decision making primary
systemic rationalities		
adaptive	experiential learning	behaviour approaches perfect rational behaviour, knowledge of reasons implicit (in individuals or collectives)
selective	amplification of positive factors through survival and growth	evolutionary learning
posterior	action produces experiences that are organised into an evaluation	action precedes intentions and goals

Table 3. Alternative rationalities [**52**]

some abstract criterion called "importance". If these connections can not be made operational, the decision rule may be replaced with another, either implicitly or explicitly. A decision maker could prefer the use of subgoals, even if they result in suboptimisation. Second, none of the current decision tools support the notion of an aspiration level. Decision makers intuitive description of an aspiration level may be something completely different than the notions offered by a decision tool.

In addition to the concept of bounded rationality there are a number of other conceptualisations of rationality. Together these go under the heading of alternative rationalities. Table 3 lists the properties and some consequences of alternative rational approaches [**52**]. *Calculated rationalities* are based on the assumption that decisions are based on explicit calculations or processes that are a result of rational behaviour. In a decision situation a decision maker is still trying to maximise utility, but the computational capabilities can be limited, and utility may include social and psychological aspects that are difficult to measure, and not just the goods. Most of the calculated rationalities listed in Table 3 are conceptualisations of common sense. The idea that decision making is affected by social factors is very familiar, for example, for anyone who has been involved in face to face sales situations.

Systemic rationalities challenge the way the word "rational" is defined in the context of complex systems. Properties of the system may result in behaviour that would be traditionally defined as "rational" although there is no explicit calculation of consequences included in the system. For example, would you say that evolution is rational?

Alternative rationalities are important for the conceptualisation of organisational and individual decision making. Above we have already discussed the importance of bounded rationality. Other calculated rationalities can help in the conceptualisation of various human behaviour. For example, in a survey a person can see that the use of a decision tool has improved decision making or made it more rational simply because accepted quantified procedures relieves him or her from certain personal responsibilities (which may cause controversial, chaotic feelings). Or decision makers may want to avoid explicit consideration of uncertainty, since otherwise they might have to admit that what is in fact known is not very much. Systemic rationalities, on the other hand, can help us classify different organisational models of behaviour. Using the classification of Table 3 one can for example see that iterative product development employs several systemic rationalities. It emphasises learning from experience, and iterative development can use action to evaluate alternatives that are difficult to understand a priori.

As a second example of different conceptualisations of decision making let us consider three different ways of looking at *organisational* decision making as described by March [**53**]:

- rational
- appropriateness
- decisions as artifacts.

In the rational approach decisions are results of intendedly rational choice, so this is "classical domain rationality". In the appropriateness approach decisions are seen as results of organisational rules and practices which determine what is appropriate. In decision situations persons define what is appropriate by asking themselves questions like:

- How do I define what kind of a situation this is?
- What kind of a person am I?
- What is appropriate for a person like me in a situation like this?

Therefore, "appropriate" decision making can be seen as "an implicit agreement to act appropriately in return for being treated appropriately" [**53**, page 105]. This is a special form of contextual rationality.

But March argues that the central role of decisions in both the "rational" and "appropriate" approaches is based on the concepts of hierarchies and causal ordering. Decisions made on a higher layer of hierarchy control the lower layers. Decisions are made in order to achieve goals as consequences. In the "artifact" approach organisational decisions are seen as having a smaller role.

- Decisions are results of complex interaction in networks instead of hierarchies. Therefore decision makers and their decisions do not have such a great effect on other decision makers and their decisions.
- Temporal ordering plays a bigger role than causal ordering. For example, the "garbage-can" model is a decision making model in which different choice opportunities, problems, solutions and decision makers are available at different times — the model states that the way these entities are connected with each other depends very much on how close they are in time. Partly this is a result of the fact that the attention of decision makers is a scarce resource.
- Decision processes are concerned with many other things than just decisions. They can be used for clarification, discovery, interpretation, appraisal or blame, development of social relationships, shows of power, enjoyment of participation and social interaction etc.

To conclude, in this section we have seen that rationality, which is the basis of the normative and prescriptive decision approaches, may not be very easy to define. When considering the requirement decision problem we are dealing with humans and complex systems (organisations), the rational behaviour of which can be limited or biased, or can challenge the definition of what it is to be rational. This is very important to understand, because it can effect all phases of prescriptive research:

assessment of current decision behaviour, determination of target behaviour, development of processes and tools, and the assessment of improvements. For example, selective rationality in organisations is a high level rationality that can be pretty difficult to identify if one considers only the decision behaviour of individuals in some NPD project.

The last “artifact” view challenges us to consider whether decisions — and rationality — are even the central thing of study. Perhaps it is much more important for example to consider the timing of problems and decision makers in the spirit of the “garbage-can” model. In the next chapter I will first justify why the requirement decision problem deserves special attention, and then proceed to study the nature of the problem further.

CHAPTER 3

The decision problem

1. The significance of the problem

We have already discussed the importance of the requirement decision from several viewpoints above.

- Selecting a requirement to be implemented product is an investment problem. Some investments are not worth making (see page 2).
- The rational plan approach of product development emphasises the role of requirements as a description of both what should be done and what should be achieved (see page 9).

Considering what March wrote about decisions as artifacts (see page 14), is this decision problem central enough to deserve our attention? It is my understanding that this is the case. In especially the rational plan paradigm of product development these decisions represent high-level objectives that affect other decisions (the decisions are hierarchical), and at least try to precede them in both causal and temporal order. As was discussed above, in the rational plan paradigm requirements are an important focal point for development projects (see page 9). When the set of requirements (or product definition) is developed, the objective is to try to capture all relevant information, and to transform this into a description that serves as a goal for further stages. The objective is to make the plan before the technical design (temporal order), and also to use the plan to drive technical design (causal order).

However, the appropriateness of this causal "what before how" planning paradigm is an eternal source of conversations and arguments. Some typical arguments for and against the paradigm are as follows.

for: If you look too much at the technical implementation, you will focus on a certain solution instead of focusing on what has to be done. You will lose a lot of creativity and the connection to external stakeholders.

against: If you truly forget the "how" you will never touch reality. In reality you can never forget the "how", which is good, since otherwise you would set goals that are totally beyond the reach of realistic implementations.

Of course in the real world development work is a mixture of both. We have the goals and requirements, and we try to be creative in finding solutions. On the other hand, when projects are started we have feasibility studies and cost estimates, and we try to think whether our requirements can be implemented within time limits and budget constraints. But the discussion will certainly go on. We will return to this problem in Chapter 5 when the decision tool developed in this work is discussed.

But even if the view of requirements driving the technical design would be wrong, it is still true that requirements drive the actual implementation. That is, even if requirements would be evaluated on the basis of existing technological solutions, we would still have to select the requirements that would actually be implemented.

To summarise, the following arguments speak for the central role of requirement decision problem.

- The requirements are used to describe what will be done.
- The requirements are used to describe what should be achieved.

I believe that these make the requirement decision problem a central decision problem that requires careful examination.

While we study the decision problem we have to be aware that deciding the set of requirements too early or too rigidly may be counterproductive. Knowledge of both the usage of the product and the way the product is realized technically develop during the project. Experiential learning and evolution take place. This is pronounced in iterative product development.

Assuming that it is desirable to control the evolution of product requirements, the set of requirements must be fixed or constrained somehow at some point of development. This leads to decision problems. For those requirements that either are or are not present in the product, such as the integrated mouse in the laptop, the following questions demand an answer: which requirements to include, which to discard, which are undecidable at this point of development? For those decisions that have a real-valued answer, such as the size of the display, it is a question of the level of the requirement: what is the final value, what is the range of acceptable values at this point of development? That is, the decision situation is somewhat different for two different types of requirements.

2. Binary and adjustable requirements

For our considerations we have to distinguish between two different types of requirements. Borrowing from the software community, we start the classification from *functional* requirements and *non-functional* requirements [**77**].[1]

- Functional requirements are associated with the (idealised) input/output relationship of a system. With a certain input the system performs a certain function. For example, for a word processor a functional requirement could state that the user should be able to have a maximum of 32 multiple open documents simultaneously. The input consists of the actions of the user and the associated files, the output is the open document windows and a possible visual or auditory message given to the user if the upper limit is reached.
- Non-functional requirements describe the limitations and side effects of one or more functions. For example, in the multiple open document case there might be reliability problems with some windowing systems, so the function would not always produce the idealised result. In addition, performing the function would take some amount of time and memory.

The distinction between the two groups of requirements is not exact enough so that we could divide a set of system requirements into two non-overlapping subsets. To continue the example above, is the upper limit of 32 open documents a description of the function or a limitation? In addition, it would of course be possible to describe the function of the system so that execution time is included in the description (for example by using temporal logic) — then time constraints would become part of the functional requirement.

In this work we use a modified version of this classification. The motivation for using the modified version is as follows. First, we note that requirements which are traditionally considered to be non-functional contain an "adjustable set of acceptable values" which are real numbers. Most time and memory constraints, reliability constraints etc. can be given more slack or trimmed down. On the other hand, if we for example have a non-adjustable time constraint — such as "response from action X must come before response from action Y", then such a requirement is

[1] Another used naming convention is behavioural and non-behavioural requirements.

needed to make a certain function possible, so that the requirement is in fact a functional one.

Second, each requirement which is traditionally considered to be a functional requirement contains a logical statement which has a true / false nature and can not be adjusted. For example, the requirement of having a maximum of 32 simultaneously open documents contains an underlying need of having multiple open documents. The limit of 32 can be adjusted, but the primary need for multiple open documents can not.[2]

Motivated by the previous discussion we define in this work two sets of requirements: *binary* and *adjustable*. We associate the class of binary requirements with the class of functional requirements, and the class of adjustable requirements with the class of non-functional requirements. Note that this classification does not target presentation issues but underlying semantics. It is of course possible to describe an adjustable requirements with a binary expression. An example is the requirement of having a maximum of 32 open documents given above.

Typically these two types of requirements are treated differently. Consider the case where we want to make a product definition that reflects the "importance" of the different requirements.

- For binary requirements you have to decide whether or not to include the requirement in the product. The decision is clearly binary, like the nature of the requirement, yes or no.
- For a adjustable requirement it is a question of the level of the requirement. Assuming that the implicit binary requirement has been decided (for example, that we will have multiple open documents), the "importance" is inherently tied with the associated measure (the maximum number of open documents).

In the next sections we will see what these problems are and what it means to decide them.

3. Deciding binary requirements

Like was already mentioned above, for binary requirements a straightforward way to control the rate of evolution is to say that some requirements will be included in the next release and some will not. That is, we can have two different selection classes, one class for those requirements that go into the release and another for those that do not. The problem of assigning binary requirements to selection classes is called the *requirements selection problem*.

In practice two classes with such strict definitions can be too strict an approach, and some room can be left for both indeterminacy and change.

- First, concerning indeterminacy, instead of using two selection classes — is included or is not included — we can use three classes: is included, is not included, and is not yet known. At the decision point it may well be that we identify requirements for which decisions can be made later, but not at this time. We could also try to solve this problem by not classifying such requirements at all. However, it can be important to identify such requirements explicitly, so that they are not mixed with new requirements that have not been classified yet.
- Second, concerning change, the definitions of the three classes can include procedures with which the decisions can be changed.

[2]Theoretically we could think that the requirement is there to limit the number of possible open documents, but then the underlying logical requirement is to have an upper limit, which is again binary.

selection class	definition
mandatory	Requirement is implemented in the next release of the product. Deleting or changing the requirement or its priority requires the acceptance of project steering group.
optional	Requirement is a candidate for the next release of the product. The final status of the requirement has to be decided before decision point 3. Changing the priority of the requirement to "mandatory" requires the acceptance of project steering group.
future	Requirement is a candidate for future releases of the system, but not the next release. Deleting the requirement requires the acceptance of project manager.

Table 1. An example of a set of selection classes used in selecting product requirements. Decision point 3 refers to a stage gate in a product development process, such as the decision point in which the requirements specification is frozen.

priority class	definition
mandatory	the customer cannot do without the requirement
important	the customer strongly wants the requirement
wish	the customer would like to have the requirement

Table 2. An example of priority classes [**9**]

Table 1 shows an example set of selection classes. Note that here each selection class has an operational definition, that is, the definition ensures that during the project each "mandatory" requirement is implemented, each "optional" requirement is decided to be either must or future, and all "future" requirements are left to future releases unless an *explicit* decision is made not to do so. The set of selection classes can be more complicated if requirements are selected for multiple consecutive releases at once. For example, you may have a number of subclasses for each upcoming release.

Requirements selection is the final operational problem that has to be solved when binary requirements are decided. However, there can be a number of analyses that lead to this decision. For example, in the literature one can find an analysis called *requirements prioritisation*, where binary requirements are given labels that describe the importance of a requirement from a certain point of view. Table 2 shows an example of priority classes [**9**]. Priority classes were originally developed for the development of tailored systems, where the mandatory need of a customer is practically mandatory for the project as well. However, in general product development this is seldom the case. For example, recent mobile phone inventions which make the mobile phone cheap, simple and disposable imply that in a mobile phone the "mandatory" user requirements are that the device has to be a phone and mobile. Selecting other requirement is a matter of tradeoff between production cost, price and demand.

4. Deciding adjustable requirements

Contrary to deciding binary requirements, there are no selection classes for adjustable requirements. The "selection" of a adjustable requirements means the assignment of its value, or a range of values. Typically the result of the assignment is a lower or upper limit, such as 32 for the number of open documents in our example above. In this thesis this problem is called the *requirement value assignment problem*.

However, when requirements are analysed the concept of importance of a adjustable requirement does have an intuitive appeal. For example, it is reasonable to ask how important the maximum number of open documents is for a user (although in a more specific way that relates the importance to a change in the value). Similarly different values of the requirement have different importances for the developing firm. If there are no interactions between requirements, then theoretically we would like to have an answer that shows how much our profit will increase for each value of the requirement.

Theoretically the determination of the utility of a adjustable requirement can be broken down into a cost-benefit analysis. Figure 1 shows one example of such an analysis. Figure 1(a) shows the benefit (revenue) curve of the requirement, and Figure 1(b) shows the cost curve of the requirement. Figure 1(c) is the overall utility (profit) curve, calculated by subtracting costs from benefits.

In practice finding out these curves is difficult or nearly impossible, and we have to make some assumptions that simplify the situation. For example, we might do one of the following.

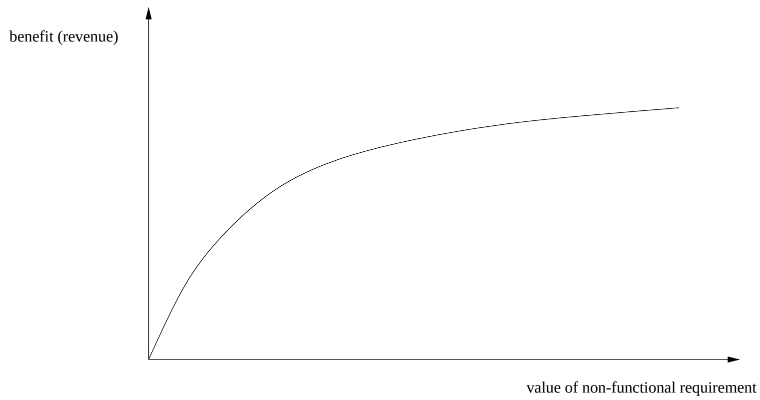
- We might assume that a function is a step function, like that in Figure 2(a), and try to find out where the steps are and how high they are.
- We might assume that the function is a straight line, like that in Figure 2(b), and try to find out its steepness.
- We might assume the existence of one or more *proxy* variables, which reflect the utility, benefit or cost of the requirement, but do not directly measure it. Such variables might for example be reliability or something more abstract like "importance". This is the subgoal approach (see Chapter 2, page 14).
- We might assess some specific points on the utility, benefit or cost line or proxy variable for certain values of the adjustable requirement with *direct preference measurements* [**47**, page 61]. An example of this is shown in Figure 2(c).
- We might create some ordered, qualitative labels such as "very important", "important", "not so important", and ask the user to assign the adjustable requirement to one of these.

The number of ways we can try to solve the situation is unlimited, and we will return to this topic again and again in this thesis.

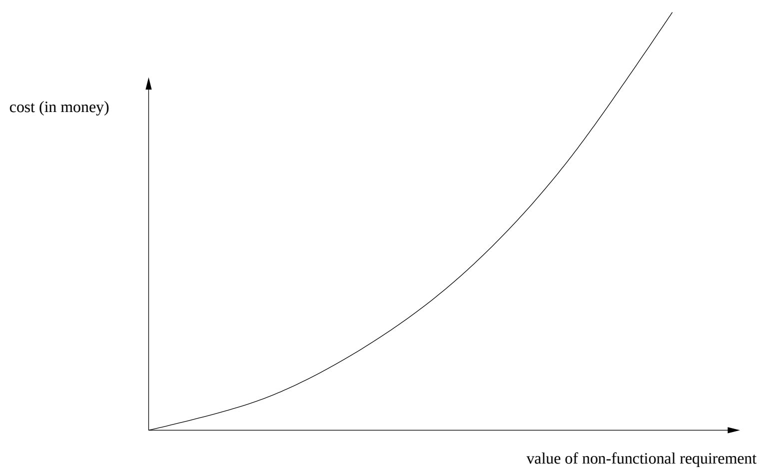
In general the concept of prioritisation is not an appropriate term for the analysis of adjustable requirements. As can be seen from the examples above, only in some cases is it possible to prioritise adjustable requirements, that is, order them from a certain viewpoint. In the approach with qualitative labels this is easy, for utility functions it's generally not possible.

There is another important view to the problem of specifying adjustable requirements. Some case studies suggest [**86, 13**] that overconstraining the specification of "technical" adjustable requirements at early stages of development may be a problem instead of being a strength. It is impossible to define what is meant by the term "technical", but loosely speaking it is used to denote the design space of engineers. In other words, you have to leave enough freedom to engineers to specify the details.[3] If you specify them too early or too rigidly, you will constrain the design space too much. This will cause both design problems in fitting the details

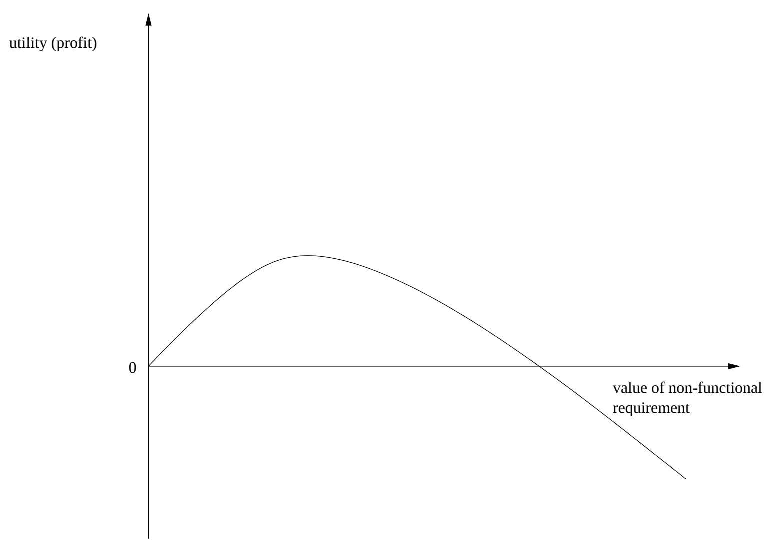
[3] Actually Ward and his colleagues even go further to say that you have to *stop* engineers from making the decisions, that is, delay the decisions consciously [**86**].



(a) The benefit (revenue) function of a adjustable requirement.

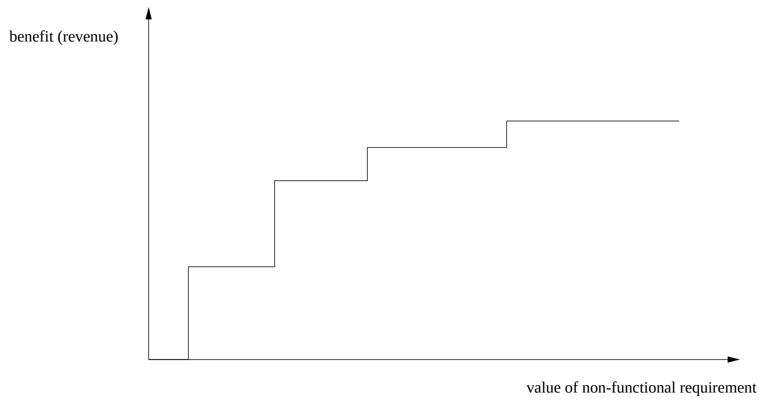


(b) The cost function of a adjustable requirement.

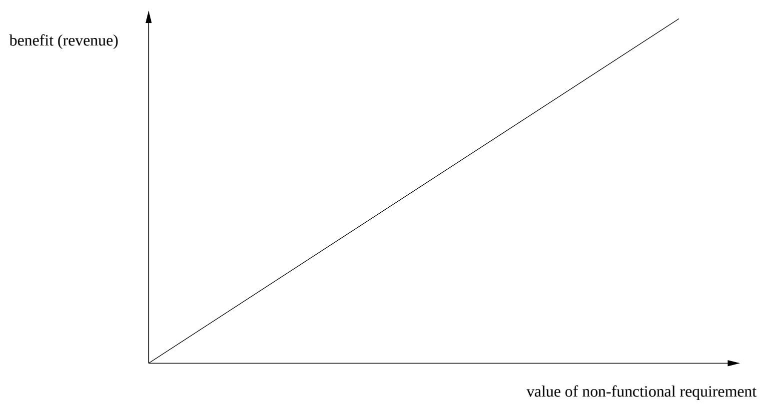


(c) The utility (profit) function of a adjustable requirement.

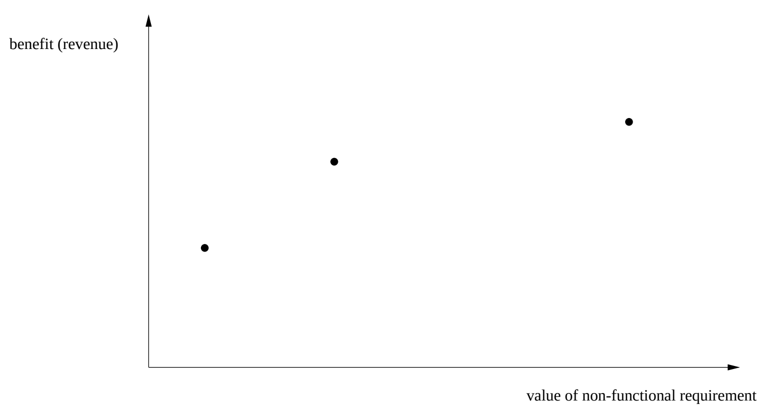
Figure 1. Cost-benefit analysis of a adjustable requirement.



(a) Approximation with a step function.



(b) Approximation with a linear function.



(c) Direct benefit measurements.

FIGURE 2. Examples of simplified approximations of a benefit function of a adjustable requirement

to work with other details [**86**], and problems in achieving higher level goals since low-level implementation has been constrained [**13**].

As can be seen from these discussions, the problem of deciding adjustable requirements is more versatile than the domain of binary requirements. But both problems are difficult to analyse. In this chapter we have shortly described the nature of decisions that we have to make finally, and some theoretical considerations about how individual requirements should be decided. But in the real world we do not have enough information to decide as theory would require, and in addition there are nasty interdependencies between the decisions. We will see this very clearly when we try to describe more formally the structure of decision problems and impact of decisions with decision tools. These tools are the subject of the next chapter.

CHAPTER 4

Decision tool review

1. Attacking the problem with tools — an overview

1.1. The need for systematic analysis. Although it is difficult to measure which decision problems are easy and which are difficult, it is my opinion that deciding the requirements of a product is a difficult one. There are a number of reasons for this.

The decision problem can be significant: The consequences of the decision can be far reaching and involve many stakeholders. A system requirement has a long and complex life-cycle, starting from design, and continuing its life in the environments of both the developing organisation and the user.

The problem can be complex: The number of requirements can be quite high, so decision makers have to make a large number of decisions. The influence of the requirement other stages of product lifecycle makes the task a difficult trade-off problem. Interactions between requirements can be complex.

The uncertainties can be high: There are high uncertainties involved because of lack of information (for example concerning competitors) and long life-cycle.

There are usually multiple decision makers: Even inside R&D different people have different objectives. The situation is even more complicated when the objectives of customers and end users are considered.

A number of remedies can help the situation: enhancing communication between stakeholders, assigning highly competent people to make the decisions, establishing standard stage gates or milestones with reviews etc.

One possible way to try to improve the situation is the use of decision making tools, that is, something systematic that could be used to facilitate decision making. The most important tools reported to date are described in this thesis. These tools can be used to structure and assess the decision problem, and provide a common language with which different stakeholders can express their views and preferences. Before we dig into the jungle of methods, we will take a look at the whole repertoire to get an overview.

1.2. Decision approaches and tools.

1.2.1. *Relevant research areas.* A number of different areas of science are relevant sources of information about relevant decision tools. The most important areas of science related to the problem are *requirements engineering* research, *technology assessment* research, and *decision sciences.*[1]

[1] One theoretically relevant research area is of course microeconomics. However, it is my opinion that microeconomic analyses are out of the scope of this applied research since their applicability to real-life situations at least at the moment is somewhat limited. Some related topics in microeconomics are modelling horizontal (trading one characteristic for another) and vertical (more of some or all characteristics) product differentiation by characteristics or representative consumer models [**87**] and the equivalence of different models [**2**], existence of equilibria in oligopolistic markets with differentiated products [**15**], and estimating product differentiation models from market share data [**7**].

- Requirements engineering research is the area of science closest to the subject of this thesis. It is a cross-disciplinary research area that studies the real-world goals for functions of systems and constraints placed on systems [**93**].[2]
- Technology assessment research is concerned with the assessment of the impacts of technology choices, both in public in private sectors [**62**]. When selecting product requirements we are actually selecting technologies, and when studying for example the costs and benefits of certain requirements we are assessing the impact of selecting certain technologies. Requirements selection and value assignment are forms of technology assessment.
- Decision sciences study the act of decision making [**18**]. In requirements selection and value assignment we have a set of alternatives to choose from, and a set of objectives we are striving for; it is clearly a decision problem.

In this section we will shortly describe decision sciences — because it is the foundation on which the structure of this thesis is built — and technology assessment — because it will not be covered in detail below. Relevant requirements engineering literature will be reviewed in sections below, and in Section 7 a summary of this literature will be given.

1.2.2. *Decision sciences.* We will view the set of decision making approaches as a list of methods with increasing power and complexity. The list, which is shown in Table 1, is a modified version of that published in [**69**]. I have added *structuring the problem*, *ordinal impact analysis*, *cost-effectiveness analysis* and *cost-benefit analysis* to the list in their appropriate places. Methods that appear later in the list are typically more costly, accurate and complex, and they are used to solve bigger problems and are applied less often.

As can be seen from the first column of the table, the presentation of these tools in this chapter follows this list. To make this even more explicit, the order in which different approaches are presented in this chapter is as follows.

1. structuring the problem
2. ordinal impact analysis
3. cost-benefit and cost-effectiveness analysis
4. linear impact analysis
5. nonlinear impact analysis

The direction of the presentation is basically first from qualitative towards more quantitative,[3] and then towards realistic modelling.

1.2.3. *Technology assessment.* As was mentioned at the beginning of this thesis, one of the focus areas of this thesis is the assessment of impacts of requirement decisions. Decision making methods described above are independent of the decision domain, and do not identify any specific classes of impacts that should be studied. Technology assessment tries to identify the most important decision criteria for choosing technologies, and utilises a range of methods to study the most important impacts of technologies with respect to these criteria [**62**]. The form of study can be *holistic* or *reductionist*. In holistic approach the impact field is viewed as a whole, whereas in reductionist approach the set of impacts is divided into subclasses. One possible division is EPISTLE [**62**], which divides impacts into the following subsets:

- environmental
- psychological
- institutional and political

[2]In Zave's definition the "systems" are actually "software systems". However, I do not see any reason why the definition could not be generalised to systems containing hardware, electronics, and mechanics.

[3]Quantitative methods presume at least the availability of an order relation [**58**, page 188]. Therefore ordinal methods are quantitative methods.

approach (section)	description	tools
intuitive judgements	deciding directly by intuition	—
structuring the problem (2)	studying the objectives and the alternatives, what affects what, what is the sequence of things happening etc.	objective trees, means-ends objectives networks, influence diagrams, decision trees, brainstorming, laddering
rules and shortcuts	establishing triggers, rules of thumb etc.	—
ordinal impact analysis (3)	assessing the order of importance of alternatives via ordering criteria and consequences	impact tables, quality function deployment
cost-effectiveness analysis (4)	calculating two measures — one that integrates all costs and one that integrates all benefits — and using the ratio of these measures as a combined measure of goodness	—
cost-benefit analysis (4)	calculating the monetary value of alternatives by pricing out all non-monetary criteria	—
linear impact analysis (5)	developing linear (weighted additive) utility models and assessing or estimating weights and criteria values	importance weighting, analytic hierarchy process, conjoint analysis
nonlinear impact analysis (6)	developing nonlinear utility models and assessing or estimating functional forms	value analysis

TABLE 1. A list of decision making approaches and tools (adapted from [**69**]). Approaches that appear later in the list are typically more costly, accurate and complex, and they are used to solve bigger problems and are applied less often.

- social
- technological
- legal
- economic.

For business problems the economic and technological subclasses are probably the most important ones.

The range of methods to study the impacts is quite large, including methods for

- information search
- survey and marketing
- decision making and group decision making

class	methods
economic analysis	cost-benefit analysis, cost-effectiveness analysis, life-cycle cost assessment (LCA), return on investment (ROI), net present value (NPV), internal rate of return (IRR), break-even point analysis, payback period analysis, residual income, total savings, increasing returns analysis
systems analysis	technology system studies, system dynamics, simulation modelling and analysis, project management techniques, system optimisation techniques, technology portfolio analysis
technological forecasting	s-curve analysis, Delphi, analytic hierarchy process, q-sort, R&D researcher hazard rate analysis, trend extrapolation
information monitoring	technical/scientific literature reviews, patent searches, Internet
performance assessment	statistical analysis, Bayesian confidence profile analysis, surveys, questionnaires, trial use periods, beta testing, technology decomposition theory, s-curve analysis, ergonomics studies, ease-of-use studies, outcomes research
risk assessment	simulation modelling and analysis, probabilistic risk assessment, environment & health and safety studies, risk-based decision trees, litigation risk assessment
market analysis	fusion method, market push/pull analysis, surveys, questionnaires, s-curve analysis
externalities analysis	social impact analysis, political impact analysis, environmental impact analysis, ethical issues analysis, cultural impact analysis

TABLE 2. Some important technology assessment methods [**33**]

- analysing economic efficiency
- optimisation
- statistical analysis
- simulation
- project management
- forecasting
- testing

As can be reasoned from the range of impacts and methods available for the task, the total number of practical technology assessment methods is quite large. Happily for us, Henriksen has tried to identify those methods that are the most important ones for private sector and has made a classification of these [**33**]. The result is shown in Table 2.

It is out of the scope of this thesis to study all of these methods here, and actually analysing them in detail is not important. Most of the methods listed in Table 2 can be used to study the developed system from a certain viewpoint — this viewpoint often represents a subgoal of the development task. Therefore they are not decision making methods per se, but can be used to obtain information used in the analysis. From the techniques listed in Table 2 we will review only cost-benefit analysis, cost-effectiveness analysis and analytic hierarchy process in this work. But researchers and practitioners should know about the existence of the toolbox of Table 2. We have to be aware of the range of different viewpoints to selection of technology and tools that are available for the analysis of certain viewpoints. In a decision situation these kinds of analyses can be consulted to obtain additional information from specific viewpoints.

1.2.4. *Tools covered in this thesis.* This thesis describes the decision approaches listed in Table 1 with one exception: the use of rules and heuristics will not be considered, although their use in system development has been brought up quite strongly, especially in [**64**], where the heuristic way is presented as a necessary complement of the rational systems architecting methodology. This thesis studies decision tools, which mostly support the rational approach. We will start by describing methods that can be used to structure the decision problem.

2. Structuring methods: fundamentals of requirements engineering

2.1. Overview of structuring methods. What does it mean to have a structured decision problem? According to Keller and Ho a fully structured decision problem contains [**48**]

- the *alternatives*, that is, possible options available in the situation
- possible *states of nature* which can have an impact on the outcomes of options.
- the *criteria*, that is, measures or rules for evaluating the alternatives.

In this thesis we focus on the set of criteria, and not so much on alternatives or states of nature because of the following reasons.

In the requirement decision problem the alternatives are the different binary requirements and values of adjustable requirements. The discovery of requirements for the requirement decision problem is an important and large research area. Several methods such as user studies and competitive studies can be used to identify such alternatives.[4] In order to maintain a reasonable focus in this thesis we discard this extremely important area. In the requirement decision problem we assume that the alternatives are known, that is, we already have a set of binary and adjustable requirements.

The modelling of possible states of nature includes the modelling of probabilities of the different states, and is equivalent to modelling the uncertainty in the decision situation. In the requirement decision problem these possible states of nature can be very complex, such as different competitive situations or different paths of technological development. The most important practical methods for modelling uncertain decision situations are *decision trees* and *influence diagrams* [**48, 54**], and more recent *sequential decision diagrams* [**21**], which all in turn use probabilities and density functions to model the underlying uncertainty. However, these modelling methods do not seem to be applicable in our problem. As was already discussed in Chapter 1 (see page 6), these methods seem to work best in situations where we have a discrete, well-identified set of possible futures [**20**]. Requirement decision problems are seldom like this. As a result, none of the currently available uncertainty modelling methods seem to match the problem. Even when experienced decision analysts have applied decision analysis to the problem [**46, 13**], they have applied nonlinear impact analysis but not influence diagrams or decision trees (which typically belong to their core method set). To summarise, decision trees or influence diagrams will not be covered in this review, and this is a logical choice since in the requirements engineering field there is no material on them to be reviewed. Interested readers can consult the citations above for further information.

[4]See [**59**] for research on general identification tactics (use of pre-existing ideas, tailoring practices of others, searching solutions, and creating or designing new solutions) and their effects on several qualities (such as adoption) of resulting decisions.

Although these important uncertainty modelling tools will not be covered, we will return to questions concerning uncertainty below in Section 6.3 and in Chapter 5.

So what is left for us at this point is the set of criteria. Luckily this is a very interesting area, since in objectives and criteria and their relationships is truly the heart of requirements engineering.

2.2. Structuring objectives and criteria: fundamentals of requirements engineering. There is no one universally accepted definition for the term *objective*. However, the concept of objective is pervasive in decision research, and the following are examples of informal descriptions of what an objective is.

- "An objective is a statement of something that one desires to achieve." [**44**]
- "An objective generally indicates the 'direction' in which we should strive to do better." [**47**]
- "Objective is a specific thing that you want to achieve." [**18**]

A *criterion* is a measure with which you can evaluate your alternatives: how does the alternative help you in achieving your objective.[5] As can be understood from the previous text, criteria are also objectives, so the concepts objective and criterion can refer to the same statement. However, not all objectives are criteria, since criteria should be measurable.

There are two important subproblems that have to be solved before we have a structured set of objectives and criteria: generating objectives and criteria, and structuring them by studying the relationships between them.

According to Keller and Ho [**48**] the procedures for generating criteria can be classified to general creativity methods, those that elicit new criteria from states of nature, from alternatives, or from criteria.

state-based procedures: In these the possible states of nature are gone through, and in each state the outcomes of currently available options are considered.

alternative-based procedures: For each alternative it is considered how the suitability of the alternative could be evaluated.

criterion-based procedures: These include examination of known criteria, and hierarchical representation of criteria considered in detail below.

general creativity methods: General creativity methods include fluent and flexible thinking, idea generating checklists, brainstorming and metaphorical thinking [**18**, pages 203–208]. These methods can be used to enhance creative thinking.

Generation of criteria is a very important task in requirements engineering, because it is closely related to modelling the world of the customer, the competitive environment etc. For example, in the state-based procedure one could think about the different tasks in which a user will apply the product, or consider different competitive situations. But in order to understand why generation of criteria is so closely related to, well, about anything else in requirements engineering, let us have a look at the relationships between criteria.

The most important relationships between objectives are *abstraction* and *causal* relationships. Abstraction relationships are represented with a *fundamental objectives hierarchy* [**44**]. Figure 1 shows one hypothetical example. To move down in the hierarchy, specialise by asking "what do you mean by that?", and to move up, generalise by asking "of what more general objective is this an aspect?" [**18**].

[5]Another word that is often used to denote the same concept is *attribute*. However, I will use the word criterion in this work, because attribute is a common word that can easily get mixed in different contexts.

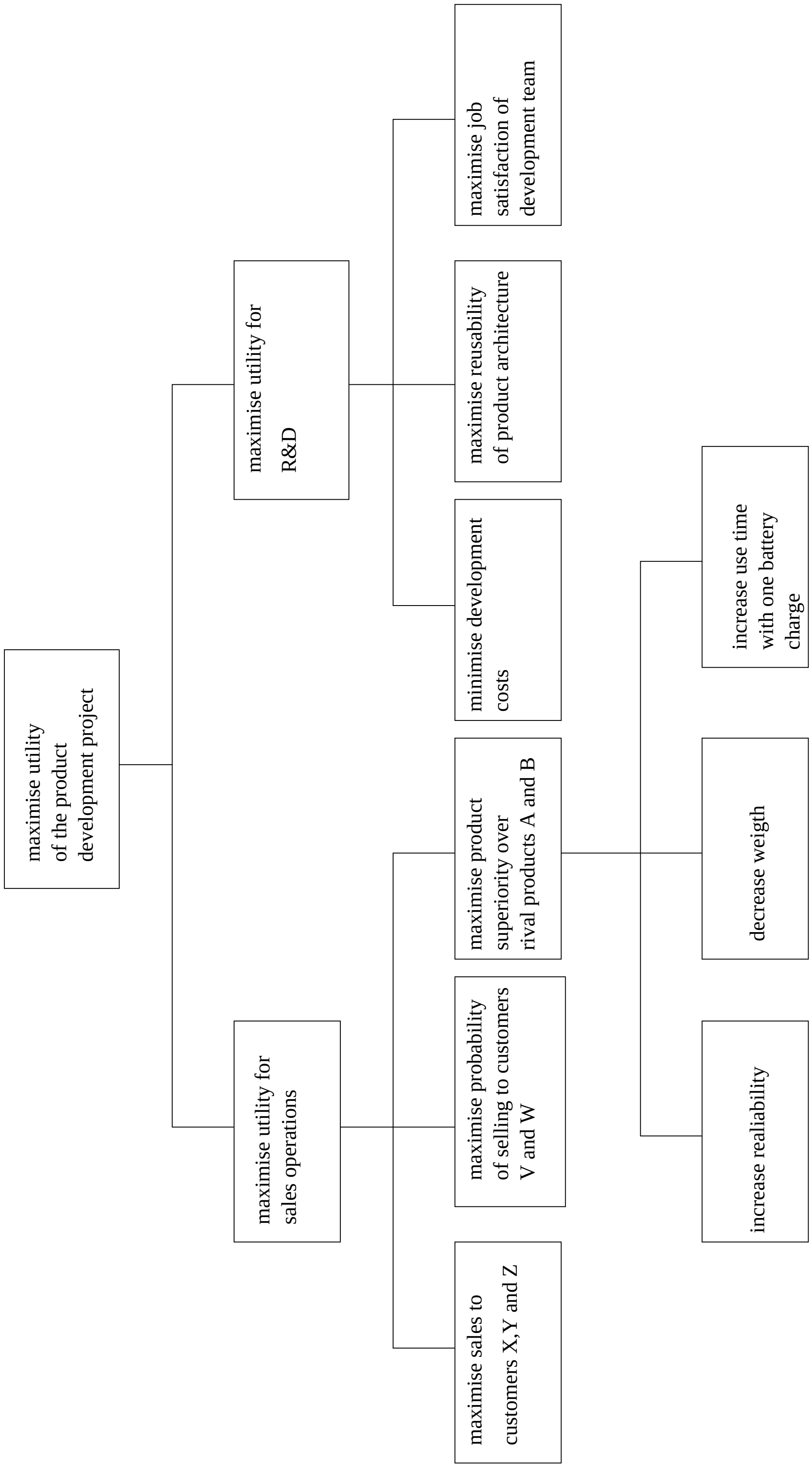


FIGURE 1. A (partial) fundamental objective tree for a hypothetical product development project. The lowest-level objectives should be measurable.

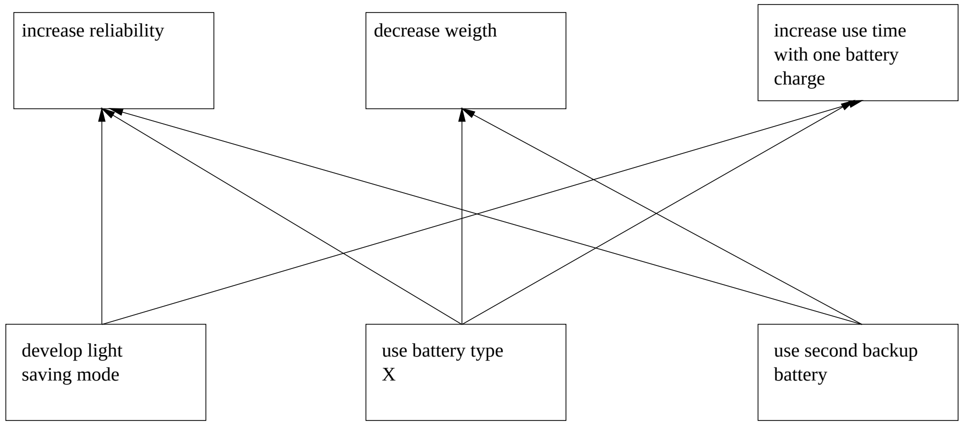


Figure 2. A (partial) means-ends objectives network for a product development project. Note that it contains some measurable lowest-level fundamental objectives from the hierarchy of Figure 1.

Causal relationships are represented with a *means-ends objectives network* [**44**]. Figure 2 shows one hypothetical example. To move from the tail of an arrow to the head in the network, examine the reason by asking "why is this important?" [**18**], or "what does this cause?", and to move from the head to the tail, examine the possible mean by asking "how could you achieve this?" [**18**], or "what causes this?". Note that a means-ends objectives network can contain bidirectional edges and cycles.

The fundamental objectives hierarchy defines the decision domain. It defines what we want to achieve, what kind of bounds will be set for the problem and how success will be measured. The lowest-level objectives of a fundamental objectives hierarchy should be measurable. The means-ends objectives network, on the other hand, examines possibilities for achieving the lowest-level fundamental objectives (which should be measurable). So in some sense it is a way for structuring alternatives, the alternatives being other subobjectives or criteria which have a causal relationship with the fundamental objectives. If the fundamental objectives hierarchy is changed, the problem is redefined. If we change the means-ends objectives network, we change the way we attack the problem.

Note that in this example all the elements of the network are not requirements: the user is probably not interested in the type of battery. But we might very well end up with a situation such as that described in Figure 2, because in the real world causal relationships are both powerful and complicated.

The representation of relationships between objectives is a conceptualisation of *the* central idea of requirements engineering. This is basically what requirements engineering is all about. It is about working with different levels of objectives and their relationships. Remembering the definition given by Zave, requirements engineering "studies the real-world goals for functions of systems and constraints placed on systems" [**93**]. Real-world goals are objectives, as are functions and constraints of systems also (they are objectives of subsequent stages of development projects). Remember that there are many levels of real-world goals. For example, if the objective of a system is to facilitate in performing a certain user task faster, this speedup is done to achieve something else (such as increase the overall throughput).

The causal (means-ends) conceptualisation of requirements engineering makes it easier to understand both the grief and rejoicing in requirements engineering. First of all, it is important to understand that we are talking about causality, and that the nature of causality can be pretty tricky to establish. For example, we are very lucky indeed if we know how the system will change the time it takes to perform a task. Second, the chain of causal objectives is practically infinite, so it is pretty difficult to state one reason (or requirement) for what is being done. Why are your eyes examining this text? Third, near the top things get quite religious. What is the ultimate reason for you to examine this text? The world of causal relationships is a hazy and complex world — this makes requirements engineering quite difficult. But there is a great potential in examining reasons for what is being done. In the words of Polish author Stanislaw Lec [**49**]: “Think before you think.”

The abstraction relationship can be used to explain another important phenomenon in requirements engineering. If objectives are too abstract, they cannot be made operational. If measures are too concrete, we may lose sight of higher levels of abstraction that could imply other concrete measures. For example, borrowing the classical example, if we just want to make a system “easier to use”, it is pretty difficult to understand how the objective could be tested. On the other hand, if we just want to “decrease the time it takes for a user to understand a message given by the system”, we may forget that this is just one aspect of ease of use. More general and more operational objectives can be found for example by using the questions introduced above: “what do you mean by that?”, and “of what more general objective is this an aspect?” The reader is encouraged to study Figure 1 with the help of these questions.

Currently in requirements engineering these two classes of relationships are mixed. To my knowledge none of the current requirements engineering methods explicitly recognise these two relationships. Although the fundamental objectives tree and means-ends objectives network are not unique, and there is a clear connection between the two, in my opinion the explicit recognition of these relationships could be a fruitful way to clarify the field. The “science of causal and abstraction relationships of objectives” is an inherent part of requirements engineering.

To bring the discussion back to earth, the use of several different levels of objectives is a natural thing in decision sciences and requirements engineering. Decision goals and requirements are often seen as goals for further development, and naturally if we broaden our decision context, then in a means-ends objectives network there are some goals for which the current goals are “only means”. In decision sciences this viewpoint is emphasised in *value-focused thinking* [**44, 45**], which stresses the role of values — “all that you care about in a decision situation” — because of their superiority as ultimate measures of goodness of decisions. In requirements engineering the use of objectives and goals in discovering and analysing requirements has become a significant research area in the 1990s. Anton and Potts describe a non-formal goal-based requirements analysis method in [**3, 4**]. Lee and Xue have developed a goal-driven use case analysis method [**50**]. Mylopoulos and his colleagues analyse requirements from the point of view of satisficing *softgoals*, for which there is no clear-cut criterion as to whether they have been satisfied [**57**]. They also refer to the work of Herbert Simon when selecting the purpose of satisficing, as I have done in this thesis. Van Lamsweerde and Letier have developed a very interesting conceptual model (see for example [**82**]), which combines goals, assumptions, agents, objects and operations, and uses temporal logic for the description of goals. This model seems to be one of the most promising conceptual models for representing applications [**56**].

3. Towards quantitative analysis: ordinal impact analysis

3.1. Impact tables.

	light saving mode	battery type X	backup battery
+reliability	yes	yes	yes
-weight	no	yes	yes
+use time	yes	yes	no

(a) A connection table showing the connections in means-ends objectives network. Entry (i, j) (ith row, jth column) indicates whether an arrow goes from j to i.

	light saving mode	battery type X	backup battery
+reliability	-	+	+
-weight		-	-
+use time	+	+	

(b) An impact table showing the "nature" of the causal relationship in means-ends objectives network. Entry (i, j) (ith row, jth column) indicates whether j causes a desired (+) or undesired (-) change in i.

	light saving mode	battery type X	backup battery
+reliability	-	+	++
-weight		-	-
+use time	++	++	

(c) An impact table showing the "nature" and ordinal magnitude of the causal relationship in means-ends objectives network. Entry (i, j) (ith row, jth column) indicates whether j causes a strong desired (++), weak desired (+), weak undesired (-) or strong undesired (- -) change in i.

Table 3. A connection table and two impact tables derived from the means-ends objectives network of Figure 2. Notation (+/-)criterion shows whether the criterion is to be increased or decreased.

3.1.1. *Adding direction and magnitude into networks.* Traditional means-ends objectives networks such as those in Figure 2 are used to show that there exists a causal connection between objectives or objectives and alternatives, but it does not say anything about the direction or magnitude of the *impact* of this connection. That is, it does not show whether the nodes support each other or are in conflict with each other, or say anything about how much the nodes affect each other.

It is of course straightforward to associate a sign with each connection to denote direction and an additional label or number to denote magnitude. Usually networks are then converted to a set of matrices or *impact tables*, each table showing the connections between two successive layers. This is done for two reasons:

- In the tabular format it is easy to sum up rows or columns to obtain overall descriptions of the connections of a node.
- Tables are easy to manipulate with standard tools (spreadsheets).

Table 3 shows a hypothetical *connection table*, which shows the connections of a means-ends objectives network in tabular format, and two different impact tables that have been derived from the means-ends objectives network of Figure 2.

scale	invariance
nominal	1-1 transformations
ordinal	monotonically increasing transformations
difference	$y = x + b$
ratio	positive linear transformations $y = ax$, $a > 0$
interval	positive affine transformations $y = ax + b$, $a > 0$

TABLE 4. Different measurement scales [**67**]

3.1.2. *The difficulty of assessing direction and magnitude of impacts.* When the assessment of the direction and magnitude of the impacts is included into the analysis, both the potential payoff and the difficulty of the analysis increase. Moving from qualitative analysis towards quantitative analysis makes it possible to perform more thorough analyses. However, assessing these quantities can be quite difficult.

First of all, the direction of the impact may not be constant. As an example, consider the size of a medical pill: on one hand, we want it to be small so that it can be swallowed easily, on the other, if it is too small it may be difficult to handle.

A much bigger problem than the direction is the magnitude of the impact. Good measures are difficult to develop and take into operational use. Consider the means-ends objectives network in Figure 2. How do we know how much the new light saving mode will affect product reliability?

The difficulties of measurement haunt all areas of science, so in that sense difficulties like this can be expected when an area of science develops. The most important measurement problem in decision sciences is probably the measurement of utility — even when we have a well-known measure like weight of a product the real value of having a specific weight is usually not directly proportional to the actual weight. Such problems come up when we want to convert decision criteria into fewer ones, for example, when we want to convert the weight of the product into increase in sales so that we can compare development costs with added revenue. We will return to this topic several times below.

3.1.3. *Towards optimisation: comparisons and the role of scales.* When good objective measures are difficult to develop or take into use, it is common to use subjective measures to assess the impacts. However, subjective measures are often only very approximate measures, so usually it is not realistic to try to get exact measures from decision makers. Actually, you often have to settle for comparative statements indicating that one impact is bigger than another, or statements that tell something about the magnitude of importance, like "A is much more important than B".

Measurement scales are central tools when expressing comparisons. Table 4 lists the different measurement scales that can be used [**67**]. Each scale is illustrated by the class of transformations that preserve the information in the scale.

Two scales are especially central to our analysis: *ordinal* and *ratio* scales. An ordinal scale means that you can put things into order, that is, you have labels such as "very important" "important" and "not so important" with which you can order the impacts. Ratio scale means that if you have two points on the scale you can say that one is r times more important than the other.

3.2. Quality function deployment (QFD). The objective of this section is to provide an overview of a method called quality function deployment (QFD). QFD is introduced at this point in the text because in its most orthodox form it is based on the concepts defined above. To be more exact, orthodox QFD is based on

- prioritisation from customer perspective

- structuring objectives
- ordinal impact analysis.

More modern versions of the tool can include linear impact analysis (see Section 5, page 43).

Quality function deployment, or House of Quality, is a set of planning and communication routines for [**32**]

- using the customers' desires and tastes as a primary driving force in product development and manufacturing
- resolving several kinds of tradeoffs — between different user needs, between engineering design choices, between product part characteristics etc.
- facilitating communication between different functional units in an organisation.

Secondary effects of the use of QFD can for example include identification of new market segments [**36**].

QFD tries to transmit the voice of the customer by the means of stages and impact analysis.

- New product development tasks are divided into stages, where each stage tries to achieve the objectives of a higher stage — the highest stage is the satisfaction of customers
- Impacts are analysed between key objectives of each stage.

The term "house" in the name House of Quality is inspired by the shape of artifacts that are the result of using QFD. If QFD is used to link the "voice of customer" all the way down to the details of the manufacturing process, the result is not just a single House of Quality, but a set of linked houses (Figure 3).

The division of development tasks into different stages is one form of a means-ends objectives network (Figure 4). The objectives of successive layers are means for achieving objectives of higher layers. Customer criteria are the the driving force since they are the fundamental objectives that populate the topmost layer.

The routine repertoire of QFD consists of the following methods of planning / analysis:

- determination of customer criteria and design and manufacturing choices at different stages (such as engineering characteristics)
- prioritisation of customer criteria
- competitor analysis
- impact analysis
- target setting

The following subsection explain each of these steps shortly.

3.2.1. *Determination of customer criteria and design and manufacturing characteristics.* Building the House of Quality starts by finding out what the customer criteria are. Typically the customer criteria are the lowest level objectives of a means-ends objectives network of the customer. The network can be built using domain knowledge, or product development teams may study the responses of users to existing versions of the product, or perform interviews about product concepts.

The selection of design and manufacturing characteristics is the domain of R&D personnel and engineers. Along the ceiling of each house engineering personnel should be able to find characteristics variables that capture critical factors of the stage. These should of course have an identifiable causal connection with the customer requirements. As can be expected, one of the most difficult aspects of this task is to represent the characteristics at an appropriate level of abstraction [**39, 10**]. The characteristics should of course be measurable, so that targets can be set and improvement measured (see also below Section 3.2.5).

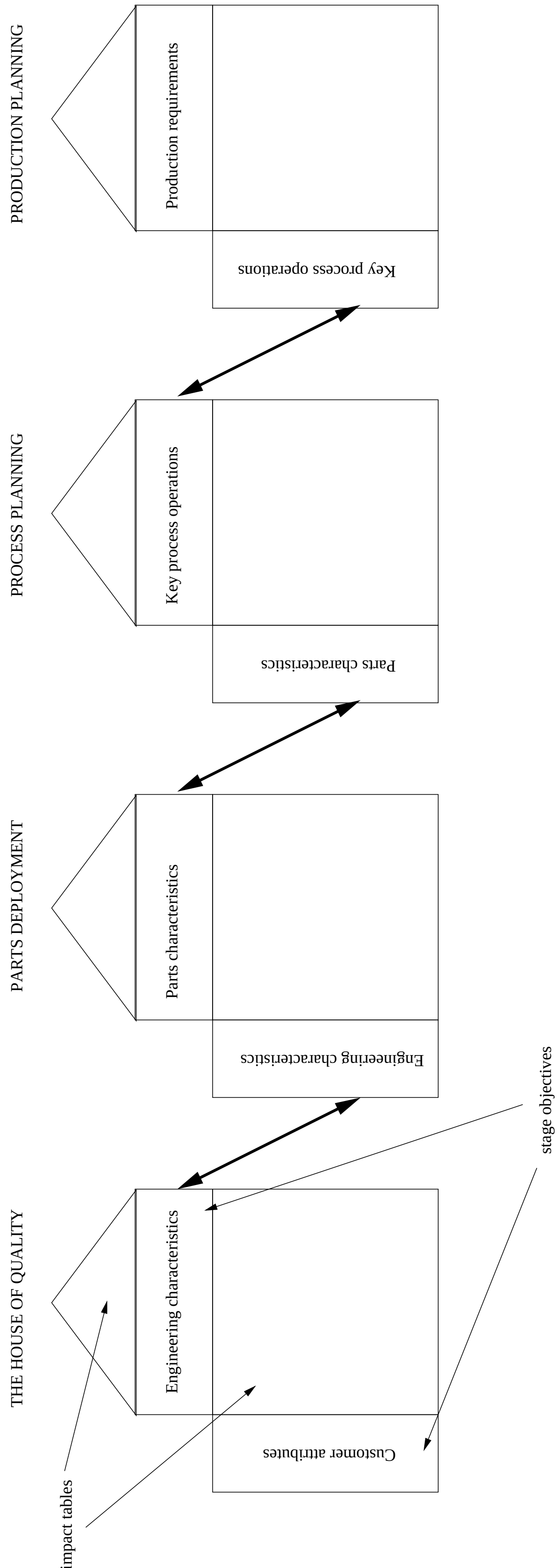


FIGURE 3. The flow of "the voice of the customer" through the houses of QFD [32]

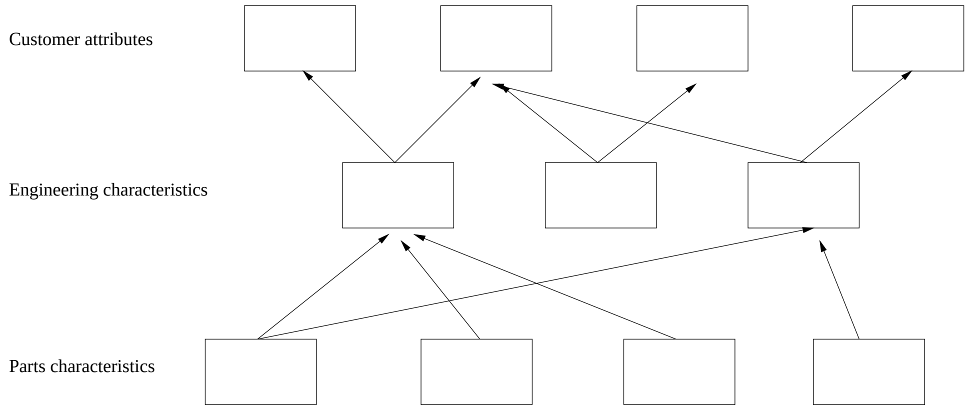


FIGURE 4. Illustration of the means-ends objectives structure of QFD (the first three stages). Intralayer links (roofs of houses) are not displayed here.

3.2.2. *Prioritisation of customer criteria.* Customers seldom view all criteria as equally important. Therefore the method includes a phase in which customer criteria are prioritised. Prioritisation can be done by using a direct ordinal scale [**32**] (such as numbers 1–5), or different scoring schemes like dividing 100 points among criteria [**28**], or for example, using the analytic hierarchy process [**26**] (see Section 5, page 43). The article by Hauser and Griffin [**28**] describes an excellent study of the first phases that lead to the identification and prioritisation of customer criteria.

3.2.3. *Competitor analysis.* Expert opinions, scientific studies, marketing studies et cetera can be used to study the position of current product versions with respect to customer criteria. This is crucial in order for product development teams to decide where they seek competitive advantage. A method called *perceptive mapping* is often used for this purpose. In it different versions of the product are simply positioned on one or more competitive (ordinal or cardinal) scales.

3.2.4. *Impact analysis.* The interior of the house is filled with impact factors that describe causal relationships between the objectives (left hand side of the house) and engineering characteristics (the ceiling). The purpose of these factors is to "carry the voice of the customer through the houses" so that it influences the choices made at different stages.

Characteristics, such as engineering characteristics, may well have interdependencies that should be considered during product design. A second impact table is used to describe the relationships between characteristics. This cross-impact table constitutes the roof of the house.

The impact factors may again be of various kinds. Usually QFD employs a direct ordinal scale (such as numbers 1–5). However, more sophisticated methods like the analytic hierarchy process (see Section 5, page 43) can be used on the whole means-ends objectives network to find the impact factors [**26, 85**].

3.2.5. *Target setting.* The most concrete product of the method is the set of targets for design and manufacturing characteristics. Based on previous steps, the product development team tries to set a measurable target for each characteristic.

3.2.6. *Appraisal and criticism of QFD.* Case studies and the breadth of application areas suggest that QFD is indeed useful for product development. QFD was born in Japanese ship building industry in 1972, and has since then been applied to a wide variety of product development tasks [**32**]. Some reports of the effects of the use of QFD are quite impressive. For example, Sullivan reports that Toyota reduced the start-up costs of automobile development by 61% when they deployed QFD [**74**]. QFD is probably the most famous quality tool at the moment.

QFD seems to enjoy limited success in systems development. Haag and his colleagues report in a survey that 6 out of 37 of the major software companies they contacted used QFD in software development [**29**].[6] The most important areas of application were operating system software and embedded software. The survey indicated that the most knowledgeable QFD users in these companies saw QFD having a number of advantages when compared with traditional methods: better communication with users, management and technical personnel, better conformance to user requirements, less errors in developed systems, reduced programming time, and more consistent and complete documentation. However, except for the objective of producing good documentation, the differences were not dramatic.[7] In addition, all companies that were using QFD had adopted the larger Total Quality Management (TQM) paradigm in other operations before they adopted QFD in software development.[8] So Haag and his colleagues actually suggest that TQM is a necessary prerequisite for successful use of QFD in software development.

Looking at the direct reports about the use of QFD in system development it seems that the application of QFD to modern system and software development has proved to be somewhat problematic. The following problems were reported by Karlsson [**39**] and Brandt [**10**].

analysing binary requirements: QFD was born and developed in application areas where adjustable requirements are a natural way to think about the wishes and needs of users. The main functionality of cars, sawing machines, boats et cetera has been evolving for decades, and some stabilisation of the interface between man and machine has taken place in these areas. Since the functionality has converged, it has become fairly clear in what kinds of tasks the user wants to apply the product. When the tasks become known and routine, it is easier to measure the success of the product in the tasks. These measures of success are very natural candidates for adjustable requirements. In my opinion there is a difference between the rate of birth of new man-machine interfaces in conventional product development and general system product development, especially modern systems that include software. One distinctive feature of the computing era is the spreading of automated information processing to new application domains, resulting in the birth of new information processing models and human-machine interfaces.

When there are no natural adjustable requirements, finding them and analysing causal connections between them and the binary requirements can be quite difficult. This was reported by Karlsson [**39**, pages 73–74], who had noticed that converting binary requirements to adjustable ones was cumbersome. In their approach binary requirements were handled separately. The problem can be even a bit worse in software development: finding technical objectives needed in later stages of QFD can be difficult for non-tangible software systems [**10**].

developing radically new products: As was already discussed above, when a product concept has matured, it may be easier to find measures with which the suitability of the product can be assessed. The situation is different with

[6] For some general discussion about the benefits of QFD in software development see [**78**].

[7] On a five-point Likert scale (1: result not being achieved, 5: result being achieved very well) differences in mean for documentation was 1.2, and for other objectives mentioned here between 0.4 and 0.7.

[8] Total Quality Management consists of three parts: a planning methodology called Hoshin planning, QFD, and statistical process control [**29**].

radically new products. It is difficult to collect information from users about product concepts that do not exists. Situation may be even much worse than this suggests: with new products we may not know who the customers is,[9] or because of confidentiality problems it may be impossible to "hear the voice of the customer" that should be "captured" [**39**, pages 72–73].[10]

expressing requirements at an appropriate level of abstraction: A major issue in the use of QFD is the selection of the number of user criteria and characteristics, that is, the size of the "house". Although some practitioners report that they have had as many as 130 customer criteria [**32**], it seems that other practitioners recommend having something like 30 criteria [**39**, page 83]. In any case, systems with more than 130 requirements are not unusual.

If the number of customer criteria becomes too large for the use of QFD, limiting this number has an impact on the level of abstraction used to describe the requirements [**39**, pages 76–78 and 83]. That is, we may have to hide the causal links from which we already have knowledge. [**39, 10**].

The end result of these problems is the return to ad hoc methods or a search for new methods. For example, in Karlsson's case at Ericsson the development group abandoned QFD and tried to develop a new method. In our work with industrial partners we have had discussions which revealed that QFD had been evaluated, but it had been rejected after problems such as those stated above were discovered. All in all our understanding is that although QFD has been evaluated in a number of organisations, it has not been adopted in large scale outside the set of companies that follow the TQM paradigm in other operations as well.

4. Cost-effectiveness and cost-benefit analysis

Cost-effectiveness and cost-benefit analysis are conceptually fairly simple methods, which can be used by themselves or in combination with other, more complex methods. In cost-effectiveness analysis all costs are converted to one measure — usually money — and all benefits are converted to a second measure. In cost-benefit analysis [**68**] the measures used for costs and benefits are the same — money — and costs are subtracted from benefits to see which action(s) should be taken. Originally cost-benefit analysis was developed as a method to assess the social desirability of public projects.

These methods can be used in deciding requirements as follows.

cost-benefit analysis: Compute the net profit for each requirement, that is, determine the additional revenue resulting from the implementation of a requirement, subtract from it the cost of designing and implementing the requirement. Choose those requirements that yield the highest (non-negative) profits.

cost-effectiveness analysis: Karlsson and Ryan [**39, 66, 43**] have used the cost-effectiveness approach, where all costs are converted to one measure, and all benefits are converted to a second measure, but these two measures are not converted to a single one. Instead of comparing the costs vs. benefits by subtracting benefits from costs, the problem is analysed as a two dimensional optimisation problem with a cost-benefit plot.

[9] Karlsson's title for section explaining this phenomenon is very much to the point: "Will the real customers please stand up?" [**39**, page 72]

[10] Dearden and Howard, however, did develop a method targeted specifically for innovative product development that uses QFD in one of its stages [**24**]. Unfortunately their paper does not discuss the availability or goodness of impact data.

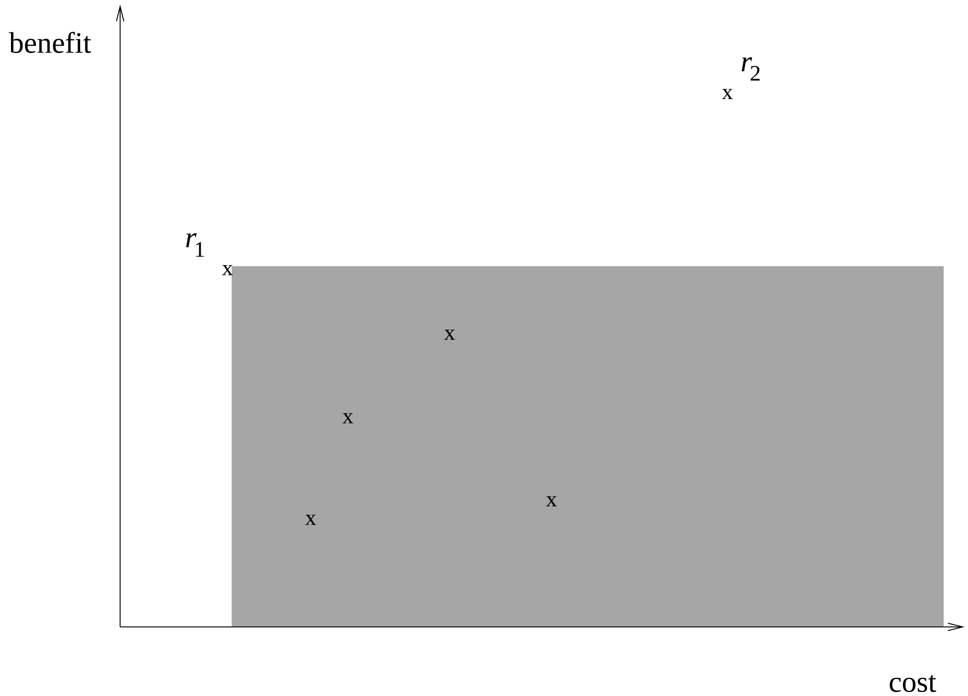


FIGURE 5. A plot showing the costs and benefits of a set of requirements. Each cross corresponds to a requirement, and r_1 and r_2 are labels for two requirements. Requirement r_1 dominates the shaded region.

Figure 5 shows one example of such a plot. In the figure each cross corresponds to a requirement of the system. Each requirement has an associated cost and benefit. Generally those requirements that have high benefits and low costs are good. Some requirements are clearly better than others. For example requirement r_1 *dominates* all requirements in the shaded region, since r_1 has higher benefit and lower cost. However, requirement r_2 does not dominate any requirements and is not dominated by any requirements. Therefore from this plot it is difficult to state the relationship between the importance of r_2 and other requirements. The cost and benefit of each requirement can be estimated for example by using the analytic hierarchy process (see Section 5, page 43).

As is evident, cost-benefit and cost-effectiveness analysis are fairly simple methods once the costs and benefits have been determined. But as can be expected, the estimation of costs and benefits is the difficult part. Such estimations typically require *pricing out*. Pricing out means the determination of price for non-monetary criteria [**47**, page 125]. Whereas pricing out the well being of humans and other creatures of nature is a controversial subject in modern society, pricing out all proxy variables has a very natural appeal in business environment. However, converting all criteria to monetary units can be fairly difficult.

So if you can price out your criteria, please do so. But unfortunately this is seldom the case because of the uncertainties and subjective assessments involved. In the next sections we will see some methods which aim at converting all criteria to common numerical values, although not necessarily to money. These methods

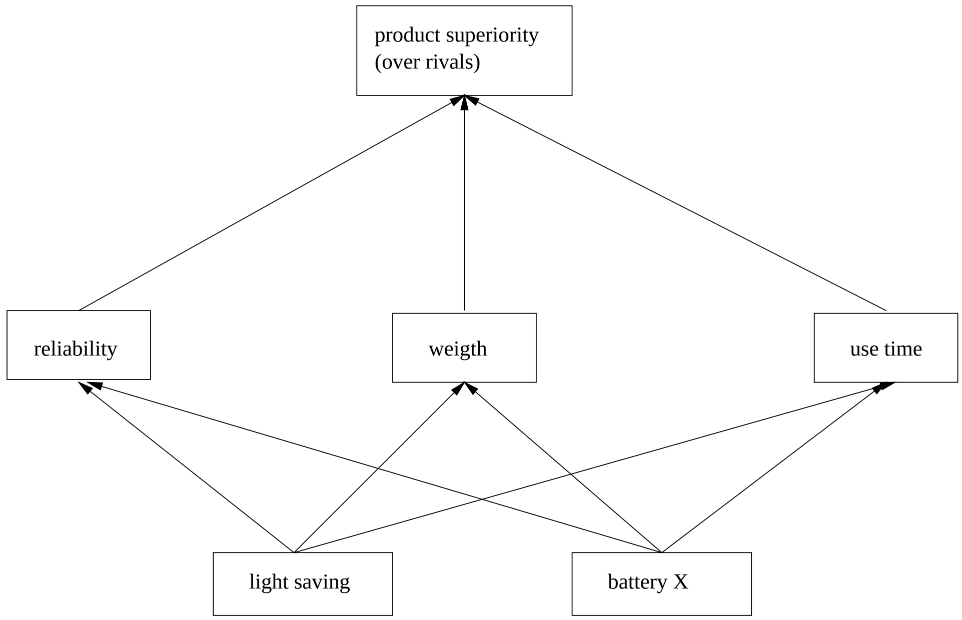


FIGURE 6. First step: constructing a means-ends objectives network. X and Y are two different battery alternatives.

try to relate individual criteria to the "priced-out" goodness of a solution with the help of mathematical models. The simplest of such models are linear.

5. Linear impact analysis

5.1. Analytic hierarchy process. We will illustrate the concepts of linear impact analysis by examining a method called the analytic hierarchy process (AHP). AHP is a quite well known decision tool, and it has been applied to the problem of deciding requirements.

Analytic hierarchy process is a decision support method for quantitative ranking of a set of alternatives [**67**]. The most important ideas behind AHP are

- structuring the problem as a hierarchical means-ends objectives network
- pairwise quantitative comparison of network nodes that are at the same layer of the network
- elicitation of verbal statements of importance, and replacement of these statements with quantitative ones
- determination of the weights and their consistency from the pairwise comparisons using the methods of linear algebra.

We will first demonstrate the use of AHP with an example, and then take a look at its elements and some criticism of the method.

EXAMPLE 5.1. In this example we will use the fundamental objective and means-ends objectives network shown in Figures 1 (page 32) and 2 (page 33). Now assume that we want to analyse whether it would be best to develop a light saving mode or to use battery type X if we want to outperform the competition. So we want to analyse which one of these properties makes our product best in terms of superiority over rival products.

The means-ends objectives network of the problem is illustrated in Figure 6. The network consists of three different layers:

1. the root objective
2. layer containing the subobjectives
3. layer containing the alternatives

qualitative importance of A over B	quantitative ratio A/B
equal importance	1
weak importance of A over B	3
essential or strong importance	5
very strong or demonstrated importance	7
absolute importance	9

TABLE 5. Converting qualitative pairwise comparisons into quantitative ratios in AHP [**67**]

	reliability	weight	use time
reliability	1	7	$\frac{1}{3}$
weight	$\frac{1}{7}$	1	$\frac{1}{9}$
use time	3	9	1

TABLE 6. Second step: comparing the nodes of the second layer (subobjectives) with each other with respect to the highest layer objective ("maximise product superiority"). Value 7 in cell $(1, 2)$ of the matrix indicates that reliability is "demonstrably or very strongly more important" than weight. Value $\frac{1}{7}$ in cell $(2, 1)$ is just a restatement of the same relationship.

	light saving	battery X
light saving	1	$\frac{1}{3}$
battery X	3	1

(a) comparison with respect to objective reliability

	light saving	battery X
light saving	1	9
battery X	$\frac{1}{9}$	1

(b) comparison with respect to objective weight

	light saving	battery X
light saving	1	3
battery X	$\frac{1}{3}$	1

(c) comparison with respect to objective use time

TABLE 7. Second step (continues): comparing the nodes of the third layer (alternatives) with each other with respect to second layer objectives.

Note that the objective is to measure the goodness with respect to product superiority over rival product. There is no natural scale for this objective, so one will be constructed in terms of the subobjectives.

After the means-ends objectives network has been constructed, the nodes are compared pairwisely with other nodes at the same layer with respect to each higher-layer node. In this example, we first compare the subobjectives (layer 2) with respect to importance to "maximise superiority over rival products" (layer 1). Table 5 shows the mapping that is typically used to convert qualitative comparisons into quantitative ones. Table 6 shows the results of comparing the objectives.

The alternatives at layer 3 have to be compared with respect to all objectives at layer 2. Table 7 shows the results of these pairwise comparisons.

Next we determine the *absolute* importance of different layer nodes from their *relative* importance recorded in Tables 6 and 7. That is, we determine a single number of importance that can be associated with each node with respect to its higher layer node.

For those matrices containing only a single consistent comparison[11] (Tables 7(a)–7(c)) this operation is very easy. Consider Table 7(a). We are looking for two weights $w_{\mathrm{L,rel}}$ (describes the importance of the light saving solution with respect to reliability) and $w_{\mathrm{B,rel}}$ (describes the importance of the battery X solution with respect to reliability) that describe the importance of the two alternatives with respect to objective for which

$$\begin{aligned} \frac{w_{\mathrm{B,rel}}}{w_{\mathrm{L,rel}}} &= 3 \\ w_{\mathrm{L,rel}} + w_{\mathrm{B,rel}} &= 1, \quad \text{normalization of weights} \end{aligned}$$

The solution is $w_{\mathrm{L,rel}} = \frac{1}{4}$ and $w_{\mathrm{B,rel}} = \frac{3}{4}$.

For matrices containing more than one comparison (Table 6) determination of weights involves some linear algebra. The weights are the elements of the eigenvector associated with the largest eigenvalue of the comparison matrix (justification of the procedure is given [**67**]). For matrix in Table 6 the largest eigenvalue is $\lambda_{\max} \approx 3.08$ and the associated normalised eigenvector is $e_{\lambda_{\max}} \approx [0.29\ 0.05\ 0.66]^T$. The elements of this vector describe the importance of reliability, weight and use time, respectively.

To summarise, after the computation of intermediate weights we have the following weights:

$$\begin{aligned} w_{\mathrm{rel}} &= 0.29 \\ w_{\mathrm{weight}} &= 0.05 \\ w_{\mathrm{time}} &= 0.66 \\ w_{\mathrm{L,rel}} &= 0.25 \\ w_{\mathrm{B,rel}} &= 0.75 \\ w_{\mathrm{L,weight}} &= 0.89 \\ w_{\mathrm{B,weight}} &= 0.11 \\ w_{\mathrm{L,time}} &= 0.75 \\ w_{\mathrm{B,time}} &= 0.25 \end{aligned}$$

These intermediate weights are used to determine the final absolute importance of different alternatives. The lowest-layer comparisons are propagated to the highest objective layer, where the alternatives can then be compared in terms of a single combined measure. Figure 7 shows the overall means-ends objectives network, and weights associated with each interlayer connection. We can compute the importance of an alternative by starting from the lowest layer of the tree, and propagating each weight upwards by multiplying it with the importance of the weight at the higher layer. So the overall "goodness" of the light saving mode is

$$w_{\mathrm{L}} = w_{\mathrm{rel}} w_{\mathrm{L,rel}} + w_{\mathrm{weight}} w_{\mathrm{L,weight}} + w_{\mathrm{time}} w_{\mathrm{L,time}} \approx 0.68. \tag{1}$$

Correspondingly $w_{\mathrm{B}} \approx 0.32$. This result suggests that from the point of view of outperforming competition the light saving mode is better.

5.2. Elements of AHP and some criticism.

5.2.1. *What does 0.68 measure?* In the last example the three low-level criteria were converted to one common measure, which hopefully describes product superiority over rivals. In practise, the criteria were "priced out", although not in terms of money. Such operations take place when we want to assess the goodness of a solution in terms of non-leaf nodes of a fundamental objectives network (see

[11] The method actually allows having *inconsistencies* in the matrices, for example, in Table 7(a) we might have a value of 5 in cell $(2,1)$ even when we have value $\frac{1}{3}$ in cell $(1,2)$. These cases are handled with the eigenvector approach illustrated below.

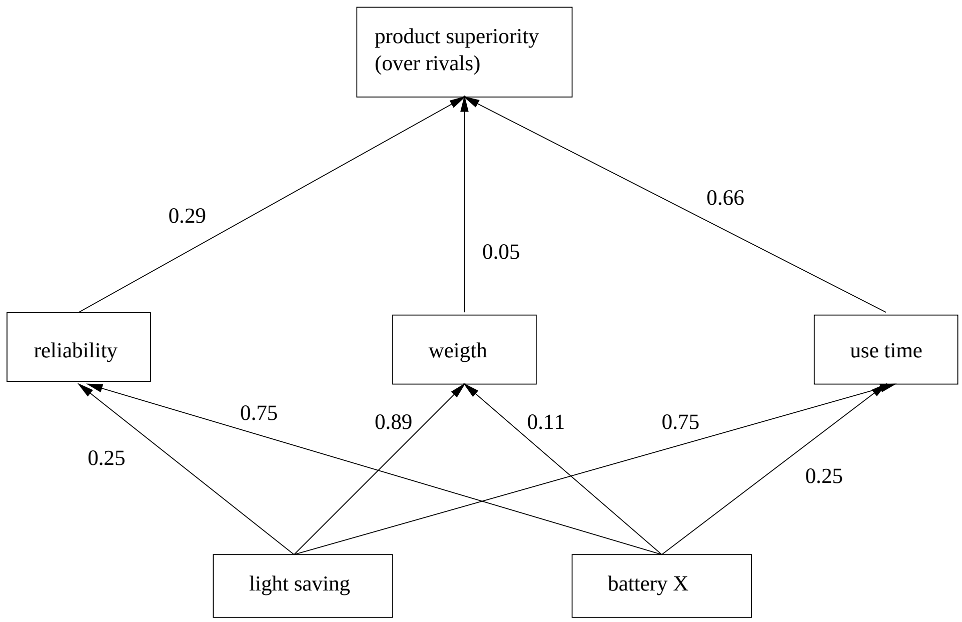


Figure 7. The means-ends objectives network with intermediate weights.

Figure 1). This is because non-leaf nodes are abstract and not measurable. We have to create new measures of goodness for these nodes.

There is one superior measure that is both abstract and operational: money. The beauty of the price system lies in the adaptive conversion of utility to a common unit. Therefore, if we have to use a common unit, money is a natural choice — at least for most business problems. But sometimes it is not feasible to price out all criteria, and we still have to make decisions. Then one alternative is to resort to new abstract measures, as was done above.

5.2.2. *Linear utility function.* AHP employs the concept of linear utility. Basically this means that the computation of the final "goodness" can be expressed as a weighted sum, such as that in equation (1). More generally, let x denote the values of the lowest-level criteria for an alternative. Then the final measure of goodness $u(x)$ is given by $u(x) = w^T x$, where vector w can be computed from the network of weights.

The fact that AHP can be reduced to a linear utility model does in fact give us an opportunity to analyse it further. Since AHP is only one of the methods described here, it is not sensible for us to dwell too much in its details. However, as one example of the consequences, in Appendix A it is shown that assuming we have two alternatives for which the initial pairwise comparison ratio for the first low level objective is r, if this ratio is changed to $r + \delta r$, then the difference of utilities of alternatives changes by $\frac{2\delta r w_1}{(r+\delta r+1)(r+1)}$. Since δr is fixed in AHP (change of 2 from one level to another), from the equation one can verify that for small values of r changes between the qualitative levels of AHP have a larger impact on the difference between utilities than for large values of r. This implies that utilities resulting from the use of AHP are more sensitive when moving between lower levels of scale than between higher levels. Mathematically this is understandable, since adding or subtracting constants in linear models produce greater changes in output at smaller input values, since the proportional change is greater with smaller inputs. However, the qualitative pairwise comparison scale used by AHP does not directly support this idea. It is difficult to say whether a change from "weak" to "strong" importance is larger than a change from "demonstrated" to "absolute" importance (see Table 5).[12]

To illustrate another important property of linear decision rules let us assume for a while that the lowest-level assessments (the x) are correct. Assume that such a w exists that $u(x) = w^T x$ is a good description of the goodness of x. Also assume that the variables are relevant, normalised, and uncorrelated or oriented so that they are positively correlated. Then for example assigning equal or even random values with the same sign to each element of w can be a pretty good approach [**69, 84**]. This is because the variance explained by w is relatively insensitive to the values of w [**84**]. Interestingly enough, these considerations were not originally prompted by theoretical examinations, but by experimental comparisons of human decision making and decision making with linear models [**23**], which suggested that linear models with equal weights are in many tasks superior to human decision makers. On the other hand, when studying the college admission problem Schoemaker and Waid found that equal weights performed somewhat worse than other methods, but it did not really matter which weight determination scheme was used [**70**].[13]

So the problem here is probably not the selection of weights in the upper part of the tree, but the original assumption of a linear relationship, the selection of relevant variables, and determination of the values of these variables. Expressing knowledge as real-valued variables in the uncertain world of new product development can be quite a challenge.

5.2.3. *Pairwise comparisons.* The pairwise comparisons of AHP seem to have their pros and cons.

- Pairwise comparisons provide a way of checking the consistency of the comparisons since they overdetermine the preferences [**67, 88**]. We know that we should have $\frac{w_i}{w_j} = \frac{w_i}{w_k}\frac{w_k}{w_j}$ for the weights in the model. Since the decision maker is asked to supply all of these ratios $\frac{w_i}{w_j}$, $\frac{w_i}{w_k}$ and $\frac{w_k}{w_j}$, the consistency of the assessments can be checked. AHP provides a *consistency index* and *consistency ratio* for this purpose [**67**].
- Providing all the pairwise comparisons requires a lot of work when the number of requirements grows [**66, 43**]. For N requirements there are $\frac{N(N-1)}{2}$ comparisons — if $N = 100$, the number of comparisons is 4950. The interdependencies of the weights can be used to reduce the number of comparisons, but then we lose the check of consistency and automate some procedures which in fact might require a lot of thinking effort from the decision maker. More sophisticated methods have been introduced and used [**66, 43**] to alleviate the problem, but when the problem is alleviated, some power of consistency checks is given up.
- Pairwise comparisons do not take into account interdependencies between requirements [**66, 43**]. An alternative A may well depend on another alternative B. However, when the utility of A changes, it is not automatically reflected as an increase in the utility of B.

[12]This problem is analogous to that used by Karlsson in [**39**, pages 81–83] to motivate the use of AHP with its ratio scales instead of ordinal ranking scales. Unfortunately, the problem will not disappear if we use the qualitative assessments of AHP, since deep down this is again based on an ordinal scale (numbers 1, 3, 5, 7, 9 used in comparisons). The problem *will* disappear if we see AHP for what it really is, that is, if we see the pairwise comparisons as coefficients in a linear model. Unfortunately other problems may then turn up, for example, regarding the ease of use of the tool.

[13]The set of schemes was multiple regression, AHP, tradeoff analysis, point allocation, and equal weighting.

5.2.4. *Elicitation of verbal statements of importance and their replacement with quantitative ones.* One of the most interesting aspects of AHP is the elicitation of pairwise ratios as qualitative statements, and the replacement of these statements by (predefined) quantitative counterparts (see Table 5). This property of AHP seems to be highly contradictory. Although Saaty argues quite strongly for the use of weights 1, 3, 5, 7 and 9 [**67**], some other researchers argue against such violent forcing of assessments. As Watson and Buede state [**88**]

> Although it may be easier to state that one criterion is weakly more important than another than to state that a given unit of one scale is three times as significant as a unit on the other scale, AHP makes the untested assumption that, given the first, we may infer the second. The problem of making difficult judgements is taken away from the decision-maker and replaced by standard assignments.

As has been shown above (see Section 5.2.2, page 46), this problem does not only apply to the individual comparisons, but to the assessment as whole: associating qualitative statements with quantitative ones may obscure the true meaning (in terms of the value of the utility function) of qualitative comparisons from the decision maker.

Example 5.1 can be used to study some of these effects. Remember that use time was “weakly more important” than reliability (Table 6), and that from the point of view of use time light saving “is weakly more important” than battery X (Table 7(c)). However, the “score” light saving gets from the viewpoint of use time is $w_{\mathrm{time}} w_{\mathrm{L,time}} \approx 0.50$. Use time is really the factor that counts, although the importances were only “weak”.

5.3. Conclusions. Karlsson and his colleagues used AHP in their work by comparing requirements from two different viewpoints — cost and benefit — and using a cost-effectiveness analysis to assess the results. So they had two comparison matrices, one for costs and one for benefits, and each requirement was compared with each other in both matrices. Concerning the evaluation of the method to the problem, Karlsson writes that [**39**, page 138]

> The people who have used the contribution-based method subjectively found the ideas interesting and appealing. As with all novel approaches, the concepts need to be further improved.

So it would seem that the method is somewhat applicable to the problem at hand. Also it seems that in the set of linear weighting methods AHP is relatively easy to use [**70**]. However, as we have discussed above, the results, although they are nicely quantitative, can be misleading. The use of a linear utility model and quantitative values for verbal statements may very well hide the things that are really taking place. In the next section we will see a method that avoids some of these pitfalls, but probably introduces new ones — especially concerning the work needed to complete an analysis.

6. Nonlinear impact analysis

6.1. Decision analysis. Decision analysis, which could also be called the decision theoretic approach, is the most extensive, rigorous and mathematical of the decision methods reviewed here (probably also of those that are used in practice [**69**]).[14] A thorough exposition of the subject is not feasible here. The interested reader is referred to [**18**], [**88**] or [**47**] (ordered by increasing mathematical difficulty) for excellent presentations of the subject.

[14] The development of the theory is closely connected to the theory of games and microeconomics. A nice review of the development of the field until 1987 can be found in [**88**].

We will illustrate the method with an example from the development of military systems following [**13**]. Other examples in literature include [**46**], which describes a case in which a planned system is evaluated against competing ones by modelling customer preferences, and [**92**] which employs both fuzzy logic and decision analysis.

EXAMPLE 6.1. The problem used as a case has been inspired by the real-world problem described in [**13**], which described the planning phase of a mobile, helicopter transportable weapons system. The overall objective of the development task was to develop a system which can provide assault fire support and anti-armour capability. This example is a simplified version of the original case.

The decision analytic part of the development work starts — as with most other decision methods — with the construction of the objective network. Figure 8 shows a partial objective tree constructed in the example.

From the objectives tree constructed in step one it is possible to extract a set of adjustable requirements for the system — military personnel call these "operational effectiveness parameters". The second step consists of developing *utility curves* for these requirements. The idea behind utility curves is to assess what the real utility of having a certain value of a adjustable requirement is. That is, what is the real utility of having a system which weighs 13 tons compared with that having a system which weighs 15 tons.

Formally the utility curve (Figure 9(a)) $u(x)$ is a function that associates a value between 0 and 1 with each feasible value of the adjustable requirement. If $u(x_1) = 0$, then x_1 has the lowest utility of all feasible values of x. Conversely, if If $u(x_2) = 1$, then x_2 has the highest possible utility of all feasible values of x. Typically for a requirement either more is better or less is better, that is, utility functions are usually (monotonically) increasing or decreasing. There exists a number of methods for the assessment of utility curves [**88, 47**], such as indifference and bisection methods.

The utility function in Figure 9(a) is monotonically decreasing. Notice also that if the weight of the system is decreased by one ton, it has a greater impact on utility near the upper limit than the lower limit. Similar utility functions to that depicted in Figure 9(a) are assessed for other requirements as well. Figure 9(b) shows a utility curve for speed.

The utility curve tells us the importance of different values of the requirement with respect to other values of the same requirement. However, since utility curves are normalised (between 0 and 1), they can not be used directly as measures utility with respect to other requirements. The approach used by Buede applies weighting to solve this problem. Each adjustable requirement is given a corresponding weight, and the overall utility of a design is evaluated by adding up the weighted utilities of requirements. The weights can be assessed for example by using AHP, or they can be assessed directly by giving points to each criterion. In our example, assume that we have only the two requirements introduced above: weight and speed. The decision-maker assesses that their weights are $w_\mathrm{w} = \frac{1}{3}$ and $w_\mathrm{s} = \frac{2}{3}$.

Assume that we have two different designs A and B for the system. We also have two adjustable requirements weight and speed introduced above, denoted by variables x_w and x_s, respectively. The utility curves $u_\mathrm{w}(x)$ and $u_\mathrm{s}(x)$ and the weights w_w and w_s of the requirements are as given above. The values of the variables for designs are as follows: for design A, $x_{\mathrm{w},A} = 13$, $x_{\mathrm{s},A} = 60$, for design B, $x_{\mathrm{w},B} = 15$, $x_{\mathrm{s},B} = 80$. The final utility of a design is computed by

$$u\left(x_\mathrm{w}, x_\mathrm{s}\right) = w_\mathrm{w} u_\mathrm{w}\left(x_\mathrm{w}\right) + w_\mathrm{s} u_\mathrm{s}\left(x_\mathrm{s}\right). \tag{2}$$

Substituting the values for A and B we get

$$\begin{aligned} u\left(x_{\mathrm{w},A}, x_{\mathrm{s},A}\right) &\approx 0.62 \\ u\left(x_{\mathrm{w},B}, x_{\mathrm{s},B}\right) &\approx 0.67. \end{aligned}$$

By this evaluation design B delivers more utility, which suggests that it should be selected.

6.2. Comparison between decision analysis and AHP. If we compare the previous example to the one which illustrated AHP (see Example 5.1), we see that the examples are much alike. We construct the objective network, and for each alternative we assign a measure of importance between 0 and 1 with respect to all

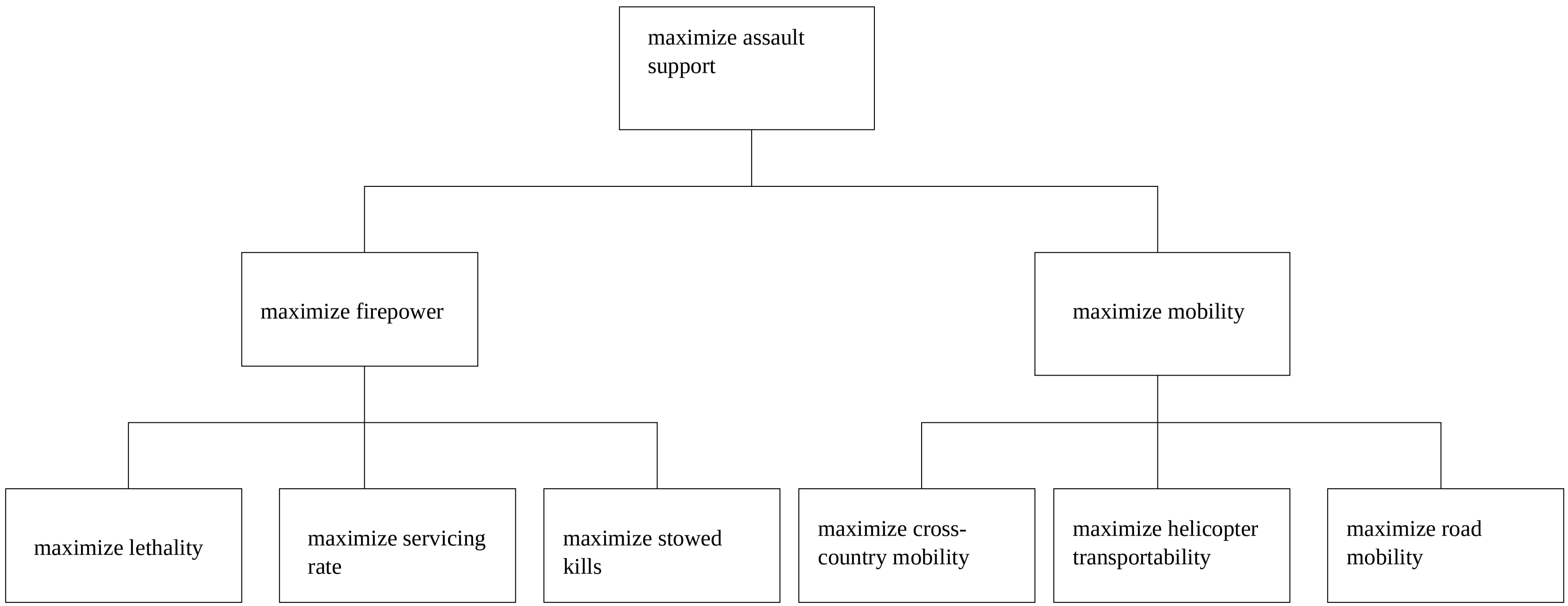
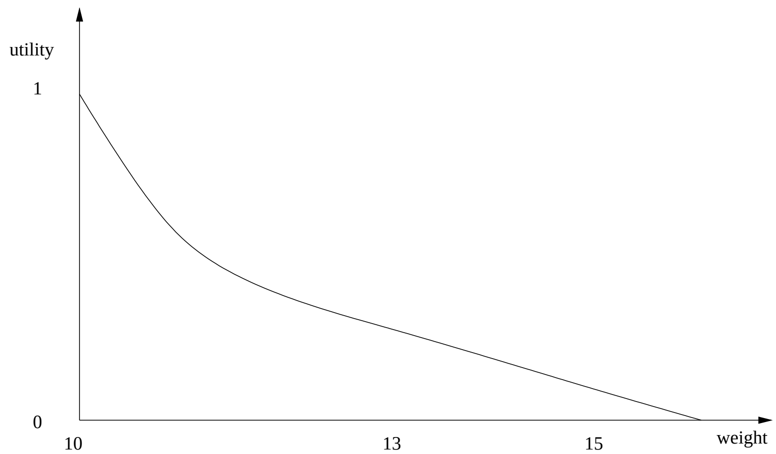
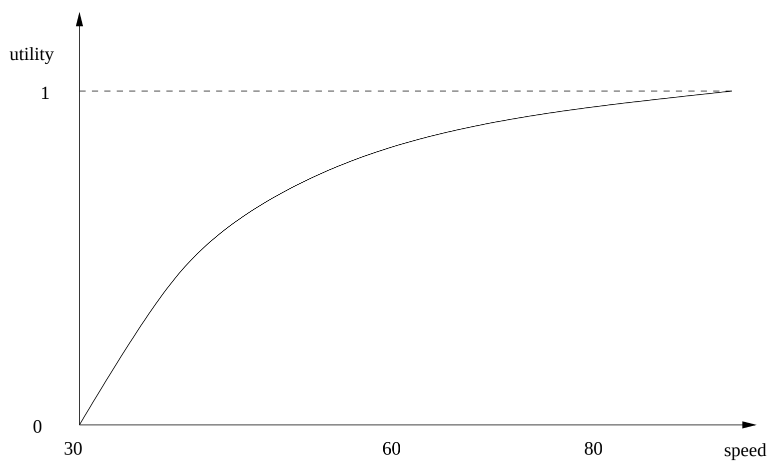


FIGURE 8. First step of decision analytic approach: constructing an objective network. The real objective network used in the case described in [**13**] is much larger.



(a) utility of different equipment weights



(b) utility of different equipment speeds

FIGURE 9. Example utility curves

lowest level objectives. We compute the final evaluation as a weighted sum. As mentioned above, the weights of the decision analytic example can even be assessed by using AHP.

The only relevant difference between the examples is the use of utility curves vs. the use of pairwise comparisons.

- Imagine how AHP would be used in the military example. The goodness of designs A and B would be assessed with respect to the low level objectives weight and speed by pairwise comparisons. Because the exact values of the variables are known (for example, $x_{s,A} = 60$, $x_{s,B} = 80$) one would in fact assess how important a speed of 60 is compared with that of 80. Put in another way, one would assess the utility of speed 60 vs. speed 80. This corresponds to assessing two points of the utility curve. In the decision analysis example we assessed the normalised utility of all feasible values of a variable, whereas using AHP we would only assess the utility of those values of the variable which come up in the alternatives.

- Imagine how the decision analytic method of the military example would be used in the AHP example. The difference here is that in the AHP example (see Example 5.1) we only have binary requirements. Therefore, a variable is either present or it is not present, which means that utility curves would be pretty simple: 1 if the property is present and 0 if it is not. In such a case the use of the decision analytic method reduces to the determination of weight, which can be done, for example, using AHP. Therefore, the decision analysis brings no added value when considering adjustable requirements.
- Imagine that in the military example we would have a third objective of selling the developed system to a certain customer. How would we handle this requirement with the two methods? AHP offers a straightforward method for doing this: the decision maker can compare the different alternatives in the usual pairwise way. However, in the decision analytic method we would have to device some way of measuring the value of this variable. This could be done in several ways. We could try to find out which of our other objectives are the most relevant to this customer, and build up a measure from our current objectives. We could also assess the probability of selling to this customer, and use the probability as a variable.

The discussion above illustrates the following points:

- The decision analytic method is at its best when handling adjustable requirements. It does not really produce added value when binary requirements are being decided.
- For adjustable requirements the decision analytic method assesses the complete utility curve, whereas AHP assesses only the values of single points. The “decision analytic movement” could say that their method *reveals the preference structure* better than AHP. Figure 9(a) does tell us quite a few things about the valuation of weight. Of course it may also take more work to obtain this whole plot.
- AHP incorporates quite naturally subjective assessments when we have no natural metrics with which to measure the performance of our system. In the decision analytic method we have to build variables for handling such objectives or assess alternatives directly (then the usefulness of utility curves is lost). It is again a relevant question whether building such variables would reveal the structure of the problem and preferences better than direct subjective assessment.

6.3. Interdependencies and uncertainty. The example discussed here does not utilise all the machinery of decision analysis. In principle decision analysis can also handle

- interdependencies between utility curves
- uncertainty.

Although it has not been explicitly stated above, both AHP and the previous example assume that the utility of an alternative with respect to one criterion is independent of the utility with respect to another criterion. As an extreme example of the dependency phenomenon consider striving for seeds, water and light when your objective is to plant as many trees as possible — it is impossible to assess the utility of having a number of seeds without the knowledge of other resources. Similarly, it could be possible in the previous example to have tasks where the combination of weight and speed is valuable, such as crossing icy rivers (where combination of low weight and high speed is good), or breaking through obstacles (where combination of high weight and high speed is good). Then the utility curve of weight might be different for different speeds, or the weight coefficient associated with variable weight could depend on speed. Similarly in AHP, it could be that the weights of the network could depend on the value of other criteria.

It is possible to handle interdependencies between different criteria in decision analysis, but then the nice additive form of the utility functions is lost (see equation (2)) [**47**]. This complicates things considerably.

The explicit consideration of uncertainty in decision making is a central contribution of decision analysis [15], whereas none of the other methods discussed here try to tackle this problem.[16] As has been mentioned several times before, new product development involves a great deal of uncertainty. There is uncertainty about the external behaviour of the product, technical implementation, users, competitors etc. Typical "amount" of uncertainty in new product development projects is much larger than for example in production tasks. Capturing the relevant information and building models based on which estimates can be computed can be impossible when building a new product. On the other hand, in day to day operation of a paper machine good estimates about the operation of the plant can be computed. (Of course problems involving nasty uncertainties come up in long-term strategic planning in paper industry.)

Uncertainty could come up in many forms in the examples we have considered. We might be uncertain about whether we can design and build a product with the given specification at all. We might have uncertainty about the actual value of adjustable requirements we can achieve. But, most importantly, strong arguments can be made for considering uncertainty about the utility of developing a product that fulfils the requirements. Utility typically results from the sales and usage of the system. Both of these are often long-term activities in turbulent environments. For sales this is fairly easy to see: we want to sell the product with good profit for a long time, and our competitors want to beat us every day. The military system considered above is a good example of the uncertainties in usage. Who knows in what kind of crisis situations the system will be employed? What do we know about the other side, or weaponry of future that we may have to attack or defend against?

6.4. Conclusions. Nonlinear utility functions are a powerful way to analyse adjustable requirements. However, decision analysis may take quite a bit of work to accomplish. Furthermore, we may not have all the necessary information to assess utility curves. All in all, the method is probably too heavy for solving most of the problems that come up when requirements are decided. But it may well be very valuable when deciding the most important adjustable requirements.

7. Summary of relevant requirements engineering literature

Now that the reader has a basic understanding of the different decision tools, we will summarise what has been written about the requirement decision problem in requirements engineering literature. Table 8 associates existing literature with each of the different decision making approaches that are presented in this thesis.

The research efforts that produced the articles in Table 8 can be described as follows.

- Most of the articles by Karlsson and Ryan, namely [**40, 42, 43, 66**], and the licenciate thesis by Karlsson [**39**], are overlapping and refer to the same body of work. The tools that were used were an ordinal prioritisation method

[15]See footnote 14 on page 48.

[16]Greer and colleagues have developed a basic scheme in which cost-efficiency analysis is augmented with ordinal risk classification [**27**].

approach	literature
direct method	Karlsson [**39, 40**], Karlsson and Ryan [**66, 43**]
structuring the problem	Fung, Ren and Xie [**26**]
ordinal utility analysis	Karlsson [**39, 40**], Karlsson and Ryan [**42, 66, 43**], Karlsson, Olsson and Ryan [**41**], Greer, Bustard and Sunazuka [**27**], Fung, Ren and Xie [**26**]
cost-effectiveness analysis	Karlsson [**39, 40**], Karlsson and Ryan [**42, 66, 43**], Karlsson, Olsson and Ryan [**41**], Greer, Bustard and Sunazuka [**27**], Jung [**37**]
cost-benefit analysis	—
linear utility analysis	Karlsson [**39, 40**], Karlsson and Ryan [**42, 66, 43**], Karlsson, Olsson and Ryan [**41**], Fung, Ren and Xie [**26**]
nonlinear utility analysis	Keeney and Lilien [**46**], Buede [**13**], Yen and Tiao [**92**]

TABLE 8. A summary of current requirement decision literature

and another method which combines a linear utility model with a cost-effectiveness approach. The contribution is the development of the latter method and application of the two methods in two case projects at Ericsson. This work was continued in [**41**] with the development of a better prioritisation process and a tool, and the use of simplification rules to reduce the amount of work needed by the original method.

- Jung continues in [**37**] the development of the method of Karlsson and Ryan. He solves the best set of requirements from the results of the cost-benefit analysis by first adding an additional cost constraint and then using a two-stage integer optimisation algorithm. Note that this results in maximum benefit within given cost constraint.
- In [**27**] Greer, Bustard and Sunazuka describe a method which uses decision criteria for cost, benefit and risk, and several ordinal scales.
- Fung, Ren and Xie demonstrate in [**26**] the application of a decision structuring method and an impact analysis method which combines ordinal impact analysis and linear impact analysis in a case study of compact disc player design.
- In [**46**] Keeney and Lilien assess nonlinear utility functions in order to estimate how customers would react to planned system versions and competitive products.
- In [**13**] Buede studies the use of nonlinear utility functions in a case study in which the purpose is to develop a transportable weapons system.
- In [**92**] Yen and Tiao expand the nonlinear utility function approach by integrating fuzzy logic to facilitate the use of linguistic terms, reasoning and composition of more complex rules. They also demonstrate shortly the use of the method in a conference room scheduling system case study.

The classification of decision approaches used in Table 8 is not comprehensive: in selecting the classes some generality has been sacrificed to obtain simplicity. There are some elements in the research efforts described above that do not fit into our classification. Also there are some research efforts that fall almost completely outside the classification.

First of all, from those research efforts described above one work considers *uncertainty* (risk) explicitly: the work of Greer, Bustard and Sunazuka [**27**]. Uncertainty has been considered shortly above in Section 6.3, and will be discussed again in Chapter 5.

Second, requirement decision problems are often *group decision problems*[17], possibly also *negotiation*, *consensus* or *voting* problems. The paper by Park, Port and Boehm addresses these issues by developing a co-operative model for prioritisation and applying it in one case study [**61**]. Their paper also addresses a third important area of *decision support systems* by developing a system that supports co-operation in prioritisation. Group decision making and decision support systems are not handled in this thesis.

Now we have had a short review of the most important decision tools and associated literature. In the next chapter the review is summarised by comparing the tools with respect to some important dimensions.

8. Comparison of different decision tools

> *Human beings seem to believe that the theory of choice exaggerates the relative power of a choice based on two guesses compared with a choice that is itself a guess.*
>
> — James March [**53**, page 100]

8.1. Attributes of decision tools. Our comparison of the different decision tools is based on a simple model of requirement decision situations.

1. The tool has a certain objective, for example, selection or analysis of value.
2. The decision maker(s) start with a set of alternatives, that is, a set of binary and adjustable requirements.
3. The tool may suggest general decision criteria, or the decision maker(s) may have to generate the criteria.
4. The impact of the requirement on the criteria is assessed.
5. The uncertainty of the impacts may be assessed.
6. The tool is used to perform its objective.
7. The final decision is a selection class label for binary requirements, or a value assignment for adjustable requirements.

Going through each stage in the previous simple model I have identified the following attributes.

optimising or satisficing (objective): Does the tool aim at optimising or just satisficing? (Note that an optimising tool is also a satisficing one, since an optimal solution clearly satisfices.)

geared for adjustable or binary requirements, or both (objective): Is the tool geared for deciding binary or adjustable requirements? That is, which of the following problems does the tool target: selection of requirements, value assignment, or both?

criteria: What are the decision criteria in the tool? Are they fixed or are they developed separately for each decision problem?

conversion to common measure vs. multiple criteria (criteria): Are all criteria converted to one common measure, or is the result an examination across multiple criteria?

formalism for impact: What kind of formalism is used to denote the impact — ordinal, linear, nonlinear?

[17] In *group decision problems* a group of people is making the decisions.

	QFD	cost-benefit	AHP	decision analysis
Attributes related to problem domain				
geared for requirements	adjustable	binary	binary	both (when combined with weighting method)
consideration of uncertainty	no	no	no	yes
Attributes related to decision approach				
optimising or satisficing	satisficing (implicit, see text)	optimising	optimising	optimising
criteria	situation specific, grouped and generated by roles (user, engineer, manufacturing)	general cost and benefit	undefined	undefined
conversion to common measure or multiple criteria	multiple criteria	conversion	conversion	conversion
formalism for impact	ordinal	real number	linear	nonlinear

TABLE 9. Attributes of different decision tools.

consideration of uncertainty (assessment of impact): Can uncertainty be handled by the tool?

The tools are evaluated along these dimensions in Table 9. Most of the attributes are explained above in sections where the tools are described. However, labelling QFD as a satisficing method demands an explanation. First, QFD can be considered as an implicit satisficing tool, since it examines the effect of different design decisions against the user requirements. If from some user aspects the result is not good enough, the design can be considered to have failed.[18] So there is some implicit satisficing in QFD, but it remains to show that this satisficing is not optimising.

First, optimisation requires a notion of "betterness", that is, an order relation. In all optimising tools there is such a notion. In the cost-benefit approach it is clear: more money is better. In AHP the order relation is defined by the value of the linear impact analysis, and in decision analysis by the value of the nonlinear impact analysis. In QFD we also measure the goodness with at most two measures: the prioritisation of customer attributes, and possibly also the importance of engineering characteristics with a weighted linear model.

Second, optimisation means that we choose the alternative that is the best with respect to the order relation. In the tools that optimise this is the case. Or to be more exact, the tools suggest the best alternative, and we can of course discard the suggestion. In cost-benefit analysis the best alternative results in biggest profit, and in AHP and decision analysis the most utility. But when goals are set for

[18]Of course we have to remember that satisficing can also be a result of the change in aspiration levels (see Chapter 2, page 14).

	microeconomic	behavioural
optimising or satisficing	optimising	satisficing
conversion to common units or multiple criteria	conversion (pricing out)	multiple criteria
criteria	money	operational subgoals
formalism for impact	exact (nonlinear)	operational

Table 10. Comparison of theoretical and behavioural decision approaches. Operational subgoals are criteria which can be connected operationally with alternatives (see Chapter 2, page 14). Operational formalism of impact means any notion that can be used to express the impact with respect to the operational subgoals. For example, formalism can be yes/no impacts for questions such as "will customer X buy the product", and monetary for some impacts, such as "what will be the cost of materials of the product".

the adjustable requirements in QFD, the tool does not suggest any target values, and there is no comparison of the target values with any other with respect to the measures of goodness.

8.2. Comparison of the tools. As was mentioned at the beginning of this thesis, there are two major themes in this work: product development and decision making. In the same spirit two subclasses can be identified in the previous set of decision tool attributes. The first subclass concerns properties of the decision domain, product development, and contains two attributes: problem type(s) the tool is geared for, and uncertainty. The second set concerns properties of the decision approach: optimising or satisficing, given or situation specific criteria, conversion to common unit or multiple criteria, and formalism for impact. These subsets have been separated in Table 9. To get a deeper understanding of the capabilities of the tools I will next compare the tools against some yardstick with respect to both themes

In product development the yardstick is simple. We know that in product development we have both kinds of requirements — binary and adjustable — and high uncertainties. Looking at these attributes in Table 9, we can see that decision analysis is the only tool that supports both tasks and uncertainty. We also know that decision analysis is very heavy and laborious, and has not really been applied to the problem. This suggests that there may be some room for improvement along these two dimensions.

With respect to the decision approach the yardstick is a bit more complicated. First of all, let us take a look at the two most profound competing decision approaches — the microeconomic and behavioural approaches — from the perspective of these attributes. The two approaches are compared against each other in Table 10. We can see that the attributes capture some major differences between these approaches.

Let us now compare the tools reviewed in this work against the behavioural approach (Table 11). We observe that most methods differ quite a bit from the behavioural approach, and — what is most interesting — that in many respects QFD seems to be closer to the behavioural approach than other methods. Remember that QFD is the most successful one of the methods reviewed here. So maybe we have identified some critical attributes that really affect the usefulness of the tools.

The next chapter introduces a behaviourally motivated decision tool that has been developed in this work. The attributes studied in this section act as guidelines in the development of the method. The development of the tool is based on the

	behavioural	QFD	cost-benefit	AHP	decision analysis
optimising or satisficing	satisficing	satisficing	optimising	optimising	optimising
conversion to common unit or multiple criteria	multiple criteria	multiple criteria	conversion	conversion	conversion
criteria	operational subgoals	situation specific subgoals grouped by roles and stages (user, design engineer, manufacturing)	cost and benefit	undefined	undefined
formalism for impact	operational	ordinal	real number	linear	nonlinear

TABLE 11. Comparison of different decision tools against the behavioural decision approach. The grayed entries indicate attribute values that differ markedly from the behavioural approach.

assumption that tools that satisfy the attributes of the problem domain and follow the behavioural approach are more suitable for the task.

CHAPTER 5

A behaviourally motivated lightweight decision tool

First you guess. Don't laugh, this is the most important step. Then you compute the consequences. Compare the consequences to experience. If it disagrees with experience, the guess is wrong. In that simple statement is the key to science. It doesn't matter how beautiful your guess is or how smart you are or what your name is. If it disagrees with experience, it's wrong. That's all there is to it.

— Richard Feynman in a television program "The Best Mind Since Einstein"

1. Principles behind the tool

At the end of the previous chapter we examined some central attributes of decision tools from the perspective of problem domain (product development) and the behavioural decision approach. In this chapter we will discuss a lightweight tool called SATIRE (SATIsficing REquirements). The central goal of the development of SATIRE was to create a tool that would fulfil the needs of the problem domain and would be based on behavioural decision approach. More specifically, the goal was to develop a method that

- supports deciding both binary and adjustable requirements
- supports examination of uncertainty
- aims at satisficing
- uses a set of operational subgoals as decision criteria
- does not convert the subcriteria to a common unit
- uses an operational formalism for impact.

In addition, we want the method to be lightweight so that it can be used even when deciding a great number of requirements. The method must be capable of supporting both intuitive and extensive, rigorous analyses.

We start the description of SATIRE with the set of subgoals, because the selection of decision criteria lies at the heart of any decision tool. Instead of leaving the set of subgoals to be completely decided by the decision maker(s), SATIRE suggests a general set of subgoals. This set is based on knowledge of the system product development domain.

2. Decision criteria: measures for goodness of requirements

The objective of this section is to study decision criteria for requirement decision problems. We would like to have an answer to the question: what are the criteria of a good requirement? Can *general* criteria of goodness for requirements be determined? Furthermore, could this set of criteria be the set of operational subgoals that could be used as decision criteria?

From a microeconomic point of view a good requirement results in a good profit. In the real world, we have to use proxy criteria or subgoals, which are thought to have a causal relationship with the final objective. For example, market share is one typical proxy criterion for profit. How can we identify these criteria?

Above in Chapter 4 (page 31) criterion generation methods were divided into four subclasses:

- state-based procedures
- alternative-based procedures
- criterion-based procedures
- general creativity methods.

Alternative-based procedures and general creativity methods are quite situation-specific. Here we try to find general criteria of goodness — criterion-based and state-based procedures are the best for this purpose. Both of these will be used to approach our problem.

First, concerning criterion-based procedures, there exists tons of research on success factors of product development. The main objective of these studies is to relate a number of factors to the ultimate financial success of the product. This is an obvious source for decision criteria, even though the results are not very strong or surprising.

In their review article Brown and Eisenhardt have studied a wide range of product development literature [**11**]. One of their contributions is a model that integrates current findings about the success factors of new product development.

Figure 1 shows the causal network of success factors in the integrated model. Details about the different factors can be found in the article [**11**]. Most of the factors are outside the domain of requirement decision problems. Such factors include suppliers, development team and management. The remaining relevant factors are the following.

market: The requirements should target a large and growing market with little competition.

product concept effectiveness: The requirements should form a product that fulfils market needs and is fit with key competencies of the company.

customers: The requirements should fulfil needs that customers have expressed.

This set of criteria is basically what can be borrowed from this line of research. These criteria will be called *product advantages* to denote that these criteria advance product success in the product development game. Note that key competencies are also advantages from the point of view of competition.

Second, concerning state-based procedures, when a requirement is selected to be implemented into a product, it has to go successfully through a number of states in order to become reality. It has to fit product architecture, has to be implemented, tested, taken into use, possibly ported to other platforms etc. So we have a model of the life of a requirement, and we examine the goodness of a requirement by determining whether it will survive all states.

The simple lifecycle used here is based on system and software development models (see, for example, [**81, 73**]) and logical reasoning of the steps that a product must go through in its lifetime. In each step of the lifecycle we can assess the goodness of the requirement. The states and corresponding success criteria that I have identified by this state-based procedure are as follows.

design: The requirement can be fulfilled with the planned product architecture. The design can be verified / validated if necessary.

implementation: The necessary resources (skills, workforce) are available for the implementation. The project will stay in budget and timetable.

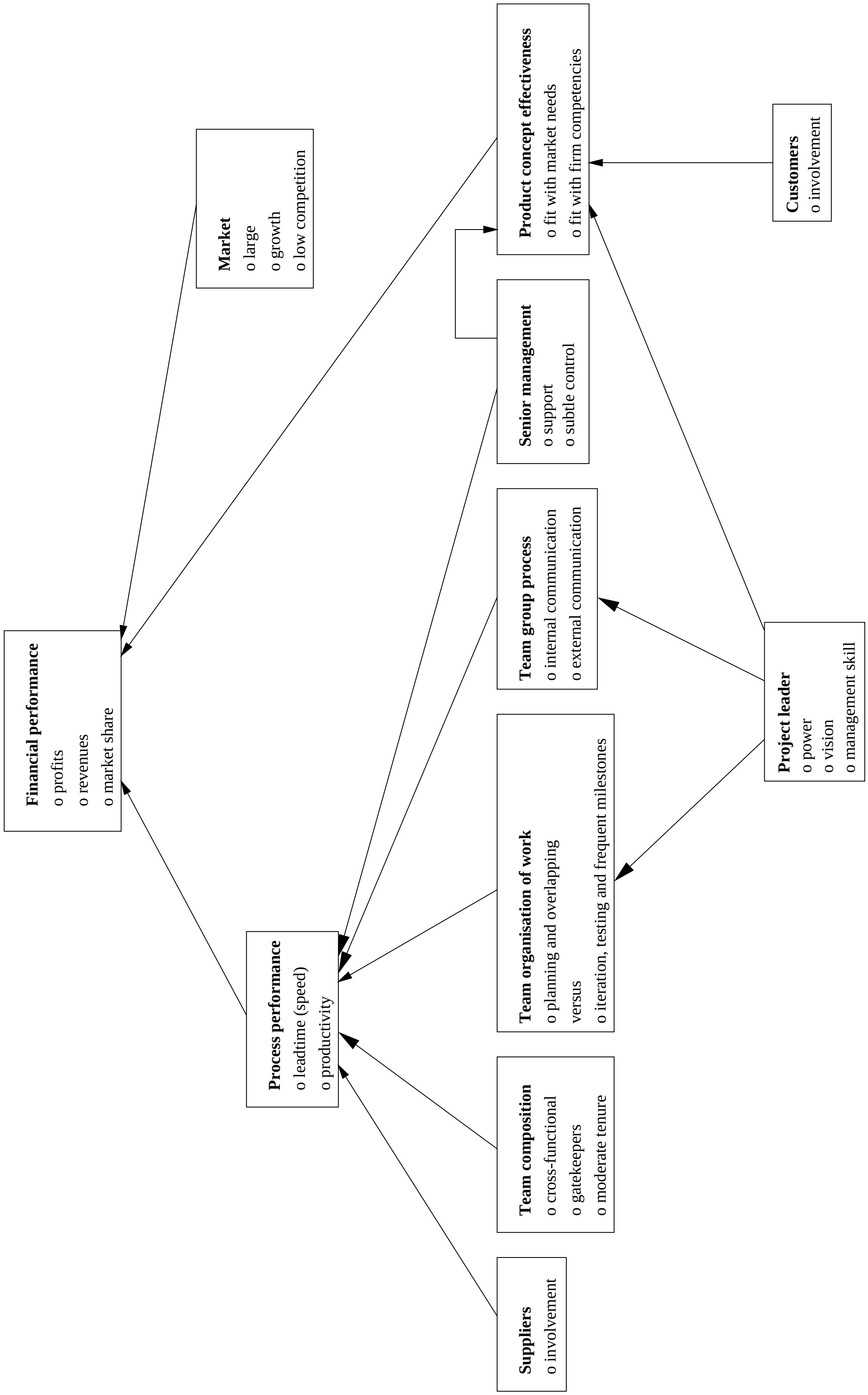


FIGURE 1. Some success factors of new product development [11]

manufacturing: Product cost will be reasonable. Manufacturing costs will remain reasonable. The production can take place without logistic or subcontracting problems.

testing/piloting: Piloting experiences can be used to assess the requirement. The end product can be tested against the requirement.

marketing: The requirement can be used as an marketing argument.

sales: The requirement helps selling the product. It responds to the needs of the users. It outperforms competition.

installation: The product that fulfils the requirement can be installed without problems.

use: The product that fulfils the requirement is usable, fulfils the standards of the domain, fulfils performance needs, is reliable, safe and secure and works with other systems if necessary.

maintenance: The product can be upgraded as planned, and the system can be ported to other possible platforms.

We will call this set of criteria *product feasibilities* to denote that these criteria assess the feasibility of successful implementation and use of the product. It should be noted that although this set of states resembles a waterfall model, it is not required that each requirement should pass each state at the same time, or that it should pass a state only once.

By default SATIRE studies the goodness of requirements from these two viewpoints, advantages and feasibilities. The following section introduces the SATIRE matrix, in which the analysis is performed.

3. SATIRE matrix

The analysis in SATIRE is performed in SATIRE matrix. Table 1 shows an example of such a matrix. The matrix is read as follows.

- Each row of the matrix corresponds to one requirement. For example, the first row of the matrix above corresponds to requirement labelled REQ 1.
- The first column **REQ** identifies the requirement.
- The second column **SEL** stores the final selection class of the requirement if one is given to the requirement (selection classes are mostly sensible for binary requirements, see page 21) such as “mandatory” or “future”, with mandatory and future of course defined appropriately (see Table 1, page 21). For example, REQ 1 is a “mandatory” (M) requirement.
- The rest of the columns are divided into two parts: the first part contains an analysis of the advantages of the requirements and the second part an analysis of the feasibility, as discussed in the previous section.
- For each requirement an advantage or feasibility slot is filled with a symbol and / or colour to indicate the goodness of the requirement from the point of view of that criterion. For example the + in the first advantage column of REQ 1 indicates that this requirement is needed to support the planned tasks of system users. On the other hand, the − in the competition column of REQ 1 indicates that our competitors are better than us with respect to this requirement.

For advantages the explanations of the symbols / colours are as follows:

(empty) / no colour: The requirement is neither an advantage or disadvantage. For example, requirement REQ 1 is neither following market trends nor being against them.

+: The requirement is considered sufficiently good to be an advantage. For example, requirement REQ 1 supports user tasks.

		ADVANTAGES			STAGE FEASIBILITIES					use						
REQ	**SEL**	supporting user tasks	meeting or outperforming competition	following market trends	design	implementation	manufacturing	testing	delivery	usability	performance	reliability	safety	security	interoperability	maintenance
REQ 1	M	+	−	?	+	+	?	?	+		−	+	?	+	?	+
REQ 2	O	?	−	?	+	−	+	+	+		?	−	?	+	+	?
REQ 3	F	+	+	+	−	−	?	?	?		?	−	−	−	?	−

TABLE 1. An example SATIRE matrix with three requirements

–: The requirement is considered sufficiently bad to be a disadvantage. For example, requirement REQ 1 loses to what the competitors have.

?: We don't have enough information to say whether the requirement is an advantage or not. For example, we don't yet know whether REQ 1 is following market trends or against them.

For stage feasibilities the explanations of the symbols / colours are as follows:

(empty) / no colour: This stage is irrelevant for the analysis of the requirement.For example, for the analysis of requirement REQ 1 the consideration of usability is not relevant. For example, it may be that this property does not have an interface with the user.

+: We have sufficient knowledge to assume that the requirement will pass the stage. For example, it is assessed that REQ 1 can be delivered with no major problems.

–: We have reason to believe that the requirement will not pass the stage without major problems. For example, there seems to be some problems in the use of REQ 1 with respect to performance.

?: We don't have enough information to say whether the requirement will pass the stage or not. For example, we don't yet know whether REQ 1 can be tested.

The advantages are used to analyse the motivation for including the requirement in the next release of the system (see page 61). The *minimum* recommended set of advantage viewpoints (used in the matrix above) is the following:

supporting user tasks: Is the requirement needed to support the planned tasks of the users?

meeting or outperforming competition: Is the requirement needed to meet or outperform a competitive product?

following market trends: Is the requirement needed to follow market trends?

The set of advantages may requires some tailoring in each company, as we have done in our case company. For example, if the company does "semi-tailored" product development, in which single customers are important enough (by themselves or as reference customers), the set of advantages may contain criteria like *sold* (if you have already committed to the requirement), or *needed by key customers*. Also, you can use a criteria like *fits key competencies*, assuming that you know what your key competencies are [**63**]. The requirement can also be *needed by internal customers* in some organisations.

The stage feasibilities are used to analyse whether a product that fulfils the requirement can pass the stages of new product development. The recommended set of stages is the following.

design: Can the requirement be fulfilled with product architecture?

implementation: Can the requirement be implemented with current resources, given timetable, and reasonable cost?

manufacturing: Can the product fulfilling the requirement be manufactured with reasonable cost and without major problems in logistics or subcontracting?

testing: Can the end product be tested against the requirement?

delivery: Can we assume that the requirement will not make the product unnecessarily difficult to install or take into use?

use: Is the product fulfilling the requirement easy to use? Can we assume that the operational costs and the operational lifetime of the product remain reasonable? Can we assume that the requirement does not cause usability, performance, reliability, safety, security or interoperability problems?

maintenance: Does the requirement support planned upgrade path? Can we assume that the requirement does not cause portability problems.

4. What are the steps in filling the SATIRE matrix

When you begin to analyse your requirements you have an empty matrix. The following sequence of steps can be used to carry through the analysis.

1. Decide which priority classes to use (e.g., mandatory, optional, wish, see above).
2. Change the columns if you want to.
3. Add all the requirements that you want to analyse as rows in the matrix.
4. Go through the requirements one at a time, preferably with project group. Leave empty entries with no symbol / colour when you think that the consideration of advantage or feasibility is not relevant. When the analysis is relevant, mark +, −, or ? as explained above.
5. For those entries that received the ?, you can
   - leave the entry and requirement as such
   - interview an expert or consult more thorough analyses such as user or marketing or competitor studies to find out the status of the entry
   - modify the requirement to clarify the situation.
6. For those entries that received the −, you can
   - leave the entry and requirement as such
   - try to modify the disadvantage or infeasibility so that it no longer is a problem - for example, you can recruit a new person that has the needed capabilities to implement the requirement
   - modify the requirement to solve the problem.
7. Assign a priority for the requirement, if the requirement is binary. If you are selecting requirements for just the next release of the product, all adjustable requirements should have the highest priority. The situation can be different if you are selecting requirements for several upcoming releases (see page 21).
8. When you have done the above for all requirements, consider the cumulative effect of all requirements, that is, even though in SATIRE matrix we have for example analysed the feasibility to implement each requirement, in the end it has to be assessed whether we can implement all MUST requirements.

5. What does SATIRE not do

SATIRE does not

- relieve decision maker(s) from the pressure of thinking hard
- guarantee the decision maker(s) that all relevant things have been considered: although it suggests a common, general set of advantages and stages, other relevant criteria do exist
- show the total (cumulative) effect of all requirements in SATIRE matrix: each requirement is considered separately — the decision maker(s) should consider the total (cumulative) effect after SATIRE has been applied
- tell the decision maker(s) what it means to be good enough to be counted as an advantage or feasible, or what it means to be bad enough to be counted as an disadvantage or unfeasible — this has to be decided by the decision maker(s)
- tell you when the advantages are worth taking a risk

- replace the need for more detailed analyses such as user, task and customer analysis or marketing studies, and designs such as architectural design, design for manufacturing etc. — SATIRE collects the results of such analyses in one place so that you can assess the overall advantages and feasibility
- show interdependencies between requirements.

6. Analysis and evaluation of SATIRE

At the beginning of this chapter the goals that drove the development of SATIRE were stated:

- supports deciding both binary and adjustable requirements
- supports examination of uncertainty
- aims at satisficing
- uses a set of operational subgoals as decision criteria
- does not convert the subcriteria to a common unit
- uses an operational formalism for impact.

How does SATIRE achieve these goals?

binary and adjustable requirements: The analysis is independent of the type of the requirement.

uncertainty: In SATIRE the ? symbol expresses decision maker(s) uncertainty about the impact of the requirement.

satisficing: The + and − symbols are used to express whether the requirement satisfices the subgoal or not.

operational subgoals: These are of course the advantages and stage feasibilities of SATIRE.

avoiding conversion to common unit: The non-numeric subgoal-oriented view of SATIRE naturally avoids conversion to common unit.

operational formalism for impact: In SATIRE the decision maker(s) have a complete freedom in deciding how +, ? and − are defined.

Using the terms of requirements engineering, SATIRE resembles a viewpoint development approach [**22**], but with very little methodological support for comparing viewpoints or managing conflicts or inconsistencies. The devotion to the behavioural decision approach has kept the method very simple. In fact it is a good question why one would want to use tools to support the behavioural decision approach in the first place: is it not optimally supported by itself? But in the development of SATIRE I have wanted to make the behavioural decision making process a bit more systematic and transparent: the use of SATIRE should at least guarantee that some viewpoints are systematically checked, and that a record is left behind.

It would of course be very nice if SATIRE had been evaluated rigorously, but unfortunately this is not the case. We have used SATIRE in two test cases with one of our industrial partners. The participants in these sessions thought that the method was promising. Probably the strongest argument for the tool is that at the time of this writing the method is being implemented into the product development process of the company. But the real value of SATIRE in this company is revealed when several projects have used the development process. And this is just one company, so generalisation of these results requires evaluation in several companies.

So in reality the evaluation of the method is just beginning. The experimental evaluation of SATIRE is currently limited to comparison with the behavioural decision approach, and the two test cases mentioned above.

CHAPTER 6

Conclusions and future work

The first principle is that you must not fool yourself — and you are the easiest person to fool.

— Richard Feynman [**65**, page 212]

1. Some critical analysis of this work

As was mentioned at the beginning of this thesis, the contribution of this work is twofold.

- This work tries to build an overall picture of the different decision tools that are available for the requirement decision problem.
- This work introduces a new, behaviourally motivated lightweight decision tool for the problem.

Has this work really delivered what was promised, and what is the end result like?

First, concerning the overall picture, the picture depicted here is based on the role of decision tools as tools for structuring decisions and assessing impacts. The resulting examination of the properties of these tools is rather selective. We have only examined a few issues related to organisational decision making — such as subgoals and satisficing — and have not discussed many important ones like motivation, power, or different organisational structures [**12**], or the extremely important area of communication between R&D and marketing [**80**] and factors affecting it [**55**], such as credibility of information and communication channel used. We have also not discussed the psychological aspects. Research on biases under uncertainty [**79**], such as the tendency to underestimate the probability that something goes wrong in a long sequence of uncertain actions, or more general psychological traps [**30**], such as tendency to favour status quo, can make the final difference in a decision situation. And although the focus in this work lies heavily in assessing impact, the accuracy with which different effects can be estimated has not been covered. For example, effort estimation is of course an important part of the problem.

It would have been possible to select another picture frame. One possible choice would have been to analyse the problem from the point of view of different alternative evaluation tactics as they have been classified by Nutt [**60**]. He has divided evaluation tactics into analytical, bargaining, subjective and judgmental tactics with further subclassifications. Table 1 lists analytical and subjective evaluation tactics. A picture of the set of decision tools drawn inside this frame would have been very different, probably emphasising support for different kinds of information and multiple sources, and the treatment of this information. But had this frame been selected we would have been led very quickly outside the domain of decision tools into issues that are closer to the frame, that is, issues concerning flow of information, teamwork, data storages etc.

So the frame chosen here has been selected to keep the discussion close to the primary use of decision tools: structuring decisions and assessing impacts. The selected conceptualisation is, in my opinion, a good framework when discussing

tactics	distinctive features
analytical	
quantitative-data	choice based on conclusions drawn from manipulating data in records
quantitative-pilot	choice based on conclusions found by manipulating data extracted from a field test
quantitative-simulation	choice based on conclusions found by manipulating data drawn from a mockup
subjective	
subjective-data	choice based on making value judgements about the meaning of archival performance data in records, which created new performance conclusions
subjective-sponsor opinion	choice based on sponsor citing "facts" that support a particular alternative
subjective-expert opinion	choice based on "facts" that support a particular alternative provided by an expert
subjective-user opinion	choice based on views of the merits of the alternatives provided by users

TABLE 1. Analytical and subjective evaluation tactics [**60**]

decision tools, but it has also kept the discussions somewhat technical. If we for example would want to discuss information sources and organisational issues, the classification by Nutt would probably be a better way to look at the world. But the goodness of the review is finally judged by the readers: if they find that decision tools are an interesting area, and that the analysis and synthesis develops insight and builds coherence into the problem field, the result is a success, otherwise it is not. At least the review helped *me* see things more clearly.

Second, concerning the development of a lightweight behaviourally motivated decision tool, the tool that has been developed here is still very much a prototype. The tool has been derived from a comparison between current existing tools and the behavioural decision making approach — this analysis was the first evaluation of the tool. The focus in this work has also constrained the analysis of the tool. For example, I have written very little about SATIRE as a mechanism for communication or motivation. The second evaluation of the tool was the application of the tool in two test cases. These were not scientific, experimental evaluations, as can easily be guessed from the "breadth" with which I have covered the experiments. But practitioners found the tool good enough to deserve a place in the next version of their product development process. That's already *something*. But the method is very lightweight, and time will tell whether it creates enough added value, even if it is very easy to use.

2. Future challenges

Instead of making a long list of possible future challenges here I will restrict this discussion to two different context: what are, in my opinion, the most import future challenges

- with respect to this thesis
- for the research area of decision tools?

For this thesis the most crucial future challenge is of course the empirical evaluation of SATIRE. One evaluation is under way at the moment and, realistically, this will be the most important evaluation since it will provide the motivation or the demotivation for other practitioners.

For the decision tool research area the most important challenge, in my opinion, is to develop an integrated toolbox of decision tools in which the decision maker could use a number of different decision models. For some mathematical models, such as AHP and decision analysis, there already exists some tools that can take care of a number of methods. But the challenge is to cover a whole range of decision making approaches with one tool, including the behavioural approach, and to provide transitions from one tool to another as smoothly as possible. Such a tool, with software support, would provide both an important set of methods for the practitioners, and an important research platform for researchers.

APPENDIX A

The effect of changes in pairwise ratio in AHP

PROPOSITION 1. *Consider an AHP network with two alternatives in which the weights of each edge have been computed from pairwise comparisons. Let w be the composite linear utility mapping of the network, that is, $u(x) = w^T x$, and let (x_1, x_2) and (x_I, x_{II}) be two different pairs of alternatives, for which $x_{1,i} = x_{I,i}$ $x_{2,i} = x_{II,i}$ for $i \neq 1$, $\frac{x_{2,1}}{x_{1,1}} = r$, and $\frac{x_{II,1}}{x_{I,1}} = r + \delta r$. Then*

$$u(x_{II}) - u(x_I) - [u(x_2) - u(x_1)] = \frac{2\delta r w_1}{(r + \delta r + 1)(r + 1)}. \tag{3}$$

Proof: Since the weights are normalised $(x_{1,1} + x_{2,1} = 1)$ and $\frac{x_{2,1}}{x_{1,1}} = r$, we have $x_{1,1} = \frac{1}{r+1}$ and $x_{2,1} = \frac{r}{r+1}$. Similarly $x_{\mathrm{I},1} = \frac{1}{r+\delta r+1}$ and $x_{\mathrm{II},1} = \frac{r+\delta r}{r+\delta r+1}$. Let $N = n(L)$ be the dimension of the alternative vectors. Let $\sigma_1 = \sum_{k=2}^{N} w_k x_{1,k}$ and $\sigma_2 = \sum_{k=2}^{N} w_k x_{2,k}$. Then the utility functions of the alternatives are

$$\begin{aligned} u(x_1) &= w_1 \frac{1}{r+1} + \sigma_1 \\ u(x_2) &= w_1 \frac{r}{r+1} + \sigma_2 \\ u(x_{\mathrm{I}}) &= w_1 \frac{1}{r+\delta r+1} + \sigma_1 \\ u(x_{\mathrm{II}}) &= w_1 \frac{r+\delta r}{r+\delta r+1} + \sigma_2. \end{aligned}$$

From these

$$u(x_{\mathrm{II}}) - u(x_{\mathrm{I}}) - [u(x_2) - u(x_1)] = w_1 \left(\frac{r+\delta r}{r+\delta r+1} - \frac{1}{r+\delta r+1} - \frac{r}{r+1} + \frac{1}{r+1} \right).$$

Equation (3) follows directly by symbolic manipulation.

Bibliography

[1] Amabile, T. M. How to kill creativity. *Harvard Business Review* (Sept.–Oct. 1998), 77–87.

[2] Anderson, S. P., De Palma, A., and Thisse, J.-F. Demand for differentiated products, discrete choice models, and the characteristics approach. *Review of Economic Studies 56* (1989), 21–35.

[3] Antón, A. Goal-based requirements analysis. In *Proceedings of the IEEE International Conference on Requirements Engineering (ICRE)* (1996), pp. 136–144.

[4] Antón, A., and Potts, C. The use of goals to surface requirements for evolving systems. In *Proceedings of the IEEE International Conference on Software Engineering (ICSE)* (1998), pp. 157–166.

[5] Bell, D. E., Raiffa, H., and Tversky, A., Eds. *Decision making*. Cambridge University Press, 1988.

[6] Bell, D. E., Raiffa, H., and Tversky, A. Descriptive, normative and prescriptive interactions in decision making. In *Decision making* [**5**], ch. 1, pp. 9–30.

[7] Berry, S. T. Estimating discrete-choice models of product differentiation. *RAND Journal of Economics 25*, 2 (1994), 242–262.

[8] Boehm, B. W. *Software engineering economics*. Prentice Hall, 1981.

[9] Brackett, J. Software requirements. Tech. Rep. SEI Curriculum Module SEI-CM-19-1.2, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, USA, 1990.

[10] Brandt, S. Applying QFD in software engineering. Master's thesis, Tampere University of Technology, Tampere, Finland, 1996.

[11] Brown, S. L., and Eisenhardt, K. M. Product development: Past research, present findings, and future directions. *Academy of Management Review 20*, 2 (1995), 343–378.

[12] Buchanan, D., and Huczynski, A. *Organizational Behaviour: An Introductory Text*, 3rd ed. Prentice Hall, 1997.

[13] Buede, D. M. Integrating requirements development and decision analysis. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC)* (Oct. 1997), vol. 2, pp. 1574–1579.

[14] Burgelman, R. A., Maidique, M. A., and Wheelwright, S. C. *Strategic Management of Technology and Innovation*, 2nd ed. McGraw-Hill, Inc., 1996.

[15] Caplin, A., and Nalebuff, B. Aggregation and imperfect competition: On the existence of equilibrium. *Econometrica 59*, 1 (Jan. 1991), 25–59.

[16] Carroll, J. S., and Johnson, E. J. *Decision Research: A Field Guide*, vol. 22 of *Applied Social Research Method Series*. Sage Publications, 1990.

[17] Clark, K. B., and Fujimoto, T. The power of product integrity. *Harvard Business Review* (Nov.–Dec. 1990), 107–118.

[18] Clemen, R. T. *Making hard decisions: an introduction to decision analysis*, 2nd ed. Duxbury Press, 1996.

[19] Cooper, R. G. From experience: The invisible success factors in product innovation. *Journal of Product Innovation Management 16* (1999), 115–133.

[20] Courtney, H., Kirkland, J., and Viguerie, P. Strategy under uncertainty. *Harvard Business Review* (Nov.–Dec. 1997), 67–79.

[21] Covaliu, Z., and Oliver, R. M. Representation and solution of decision problems using sequential decision diagrams. *Management Science 41*, 12 (Dec. 1995), 1860–1881.

[22] Darke, P., and Shanks, G. Stakeholder viewpoints in requirements definition: A framework for understanding viewpoint development approaches. *Requirements Engineering 1* (1996), 88–105.

[23] Dawes, R. M., and Corrigan, B. Linear models in decision making. *Psychological Bulletin 81* (1974), 95–106.

[24] Dearden, A., and Howard, S. Capturing user requirements and priorities for innovative interactive systems. In *Proceedings of the Australasian Computer Human Interaction Conference* (1998), pp. 160–167.

[25] Dorf, R. C., Ed. *The Technology Management Handbook*. The Electrical Engineering Handbook Series. CRC Press, IEEE Press and Springer Verlag, 1999.

[26] Fung, R. Y., Ren, S., and Xie, J. The prioritisation of attributes in customer requirement management. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC)* (1996), vol. 2, pp. 953–958.

[27] Greer, D., Bustard, D., and Sunazuka, T. Prioritisation of system changes using cost-benefit and risk assessment. In *Proceedings of the IEEE International Symposium on Requirements Engineering (ISRE)* (1999), pp. 180–187.

[28] Griffin, A., and Hauser, J. R. The voice of the customer. *Marketing Science 12*, 1 (1992), 1–27.

[29] Haag, S. E., Raja, M., and Schkade, L. Quality function deployment usage in software development. *Communications of the ACM 39*, 1 (1996), 41–49.

[30] Hammond, J. S., Keeney, R. L., and Raiffa, H. The hidden traps in decision making. *Harvard Business Review* (Sept.–Oct. 1998), 47–58.

[31] Hart, S., Hultink, E. J., Tzokas, N., and Commandeur, H. R. How industrial companies steer their new product development processes. In *1998 PDMA Proceedings* (1998), pp. 137–154.

[32] Hauser, J. R., and Clausing, D. The house of quality. *Harvard Business Review* (May–June 1988), 63–73.

[33] Henriksen, A. D. P. A technology assessment primer for management of technology. *International Journal of Technology Management 13*, 5–6 (1997), 615–638.

[34] Hogarth, R. M. *Judgement and Choice*, 2nd ed. Wiley, 1987.

[35] Holmstrom, B. Agency costs and innovation. *Journal of Economic Behavior and Organization 12* (1989), 305–327.

[36] Hunter, M., and Van Landingham, R. Listening to the customer using QFD. *Quality Progress 27*, 4 (Apr. 1994), 55–59.

[37] Jung, H.-W. Optimizing value and cost in requirements analysis. *IEEE Software* (July–Aug. 1998), 74–78.

[38] Kahneman, D., Slovic, P., and Tversky, A., Eds. *Judgement under uncertainty: Heuristics and biases*. Cambridge University Press, 1982.

[39] Karlsson, J. *Towards a Strategy for Software Requirements Selection*. Linköping University, S-581 83 Linköping, Sweden, 1995. Licenciate thesis.

[40] Karlsson, J. Software requirements prioritizing. In *Proceedings of the IEEE International Conference on Requirements Engineering (ICRE)* (1996), pp. 110–116.

[41] Karlsson, J., Olsson, S., and Ryan, K. Improved practical support for large-scale requirements prioritising. *Requirements Engineering 2*, 1 (1997), 51–60.

[42] Karlsson, J., and Ryan, K. Supporting the selection of software requirements. In *Proceedings of the 8th International Workshop on Software Specification and Design (IWSSD)* (1996), pp. 146–149.

[43] Karlsson, J., and Ryan, K. A cost-value approach for prioritizing requirements. *IEEE Software* (Sept.–Oct. 1997), 67–74.

[44] Keeney, R. L. *Value-Focused Thinking*. Harvard University Press, 1992.

[45] Keeney, R. L. Creativity in decision making with value-focused thinking. *Sloan Management Review* (summer 1994).

[46] Keeney, R. L., and Lilien, G. L. New industrial product design and evaluation using multiattribute value analysis. *Journal of Product Innovation Management 4* (1987), 185–198.

[47] Keeney, R. L., and Raiffa, H. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge University Press, 1993.

[48] Keller, L., and Ho, J. Decision-problem structuring. In *Concise Encyclopedia of Information Processing in Systems & Organizations*, A. P. Sage, Ed. Pergamon Press, 1990, pp. 103–110.

[49] Lec, S. *Unkempt thoughts*. St Martins, New York, 1962. Original Polish work published in 1954. Text of the book is available on the world wide web at address `http://www.geocities.com/Athens/Forum/5344/lit/lecengl.html`.

[50] Lee, J., and Xue, N.-L. Analyzing user requirements by use cases: A goal-driven approach. *IEEE Software* (July–Aug. 1999), 92–101.

[51] Luenberger, D. G. *Microeconomic Theory*. McGraw-Hill, Inc., 1995.

[52] March, J. G. Bounded rationality, ambiguity, and the engineering of choice. In Bell et al. [**5**], ch. 2, pp. 33–57.

[53] March, J. G. How decisions happen in organizations. *Human-Computer Interaction 6* (1991), 95–117.

[54] Matzkevich, I., and Abramsson, B. Decision-analytic networks in artificial intelligence. *Management Science 41*, 1 (Jan. 1995), 1–22.

[55] Moenaert, R. K., and Souder, W. E. Context and antecedents of information utility at the R&D/marketing interface. *Management Science 42*, 11 (Nov. 1996), 1592–1610.

[56] Mylopoulos, J. Information modeling in the timeof the revolution. *Information Systems 23*, 3/4 (1998), 127–155.

[57] Mylopoulos, J., Chung, L., and Yu, E. From object-oriented to goal-oriented requirements analysis. *Communications of the ACM 42*, 1 (Jan. 1999), 31–37.

[58] Niiniluoto, I. *Johdatus tieteenfilosofiaan: käsitteen- ja teorianmuodostus.* Otava, 1980. In Finnish. Free translation of name in English: Introduction to philosophy of science: formation of concepts and theories.

[59] Nutt, P. C. The identification of solution ideas during organizational decision making. *Management Science 39*, 9 (Sept. 1993), 1071–1085.

[60] Nutt, P. C. How decision makers evaluate alternatives. *Management Science 44*, 8 (Aug. 1998), 1148–1166.

[61] Park, J.-W., Port, D., and Boehm, B. Supporting distributed collaborative prioritization. In *Proceedings of the sixth Asia Pacific Software Engineering Conference (APSEC)* (1999), pp. 560–563.

[62] Porter, A. R., Rossini, F. A., Carpenter, S. R., Roper, A., Larson, R. W., and Tiller, J. S. *A Guidebook for Technology Assessment and Impact Analysis*, vol. 4 of *North Holland series in system science and engineering.* North Holland, 1980.

[63] Prahalad, C. K., and Hamel, G. The core competence of the corporation. In *Strategic Management of Technology and Innovation* [**14**], pp. 64–76.

[64] Rechtin, E., and Maier, M. *The art of systems architecting.* Systems engineering series. CRC Press, 1997.

[65] Robbins, J., Ed. *The pleasure of finding things out: the best short works of Richard P. Feynman.* Penguing Press, 1999.

[66] Ryan, K., and Karlsson, J. Prioritizing software requirements in an industrial setting. In *Proceedings of the IEEE International Conference on Software Engineering (ICSE)* (1997), pp. 564–565.

[67] Saaty, T. L. *The Analytic Hierarchy Process.* McGraw-Hill, Inc., 1980.

[68] Samuelsson, W. Cost-benefit analysis. In Dorf [**25**], pp. **4**–35–**4**–40.

[69] Schoemaker, P. J., and Russo, J. E. A pyramid of decision approaches. *California Management Review* (fall 1993), 9–31.

[70] Schoemaker, P. J., and Waid, C. An experimental comparison of different approaches to determining weights in additive utility models. *Management Science 28*, 2 (Feb. 1982), 182–196.

[71] Schotter, A. R. *Microeconomics : a modern approach*, 2nd ed. Addison-Wesley Educational Publishers, 1997.

[72] Simon, H. A. Rational decision-making in business organizations. In *Nobel Lectures in Economic Sciences 1969–1980*, A. Lindbeck, Ed. World Scientific Publishing Co., 1060 Main Street, River Edge, NJ 07661, USA, 1992, pp. 343–371. Original Nobel Memorial Lecture given on 8 December 1977.

[73] Sommerville, I. *Software engineering*, 4th ed. International computer science series. Addison-Wesley, 1992.

[74] Sullivan, L. P. Quality function deployment. *Quality Progress* (June 1986), 39–50.

[75] Takeuchi, H., and Nonaka, I. The new product development game. *Harvard Business Review* (Jan.–Feb. 1986), 137–146.

[76] Thayer, R. H., and Dorfman, M., Eds. *Software requirements engineering*, 2nd ed. IEEE Computer Society, 10662 Los Vaqueros Circle, P.O. Box 3014, Los Alamitos, CA 90720-1314, USA, 1997.

[77] Thayer, R. H., and Thayer, M. C. Software requirements engineering glossary. In Thayer and Dorfman [**76**], pp. 489–528.

[78] Tran, T.-L., and Sherif, J. S. Quality function deployment (QFD): An effective technique for requirements acquisition and reuse. In *Proceedings of the second IEEE International Software Engineering Standards Symposium (ISESS)* (1995), pp. 191–200.

[79] Tversky, A., and Kahneman, D. Judgement under uncertainty: Heuristics and biases. *Science 185* (1974), 1124–1131. Reprinted in [**38**].

[80] Twiss, B. C. *Managing technological innovation*, fourth ed. Financial Times Pitman Publishing, 1992.

[81] Ulrich, K. T., and Eppinger, S. D. *Product Design and Development.* Management and Organization Series. McGraw-Hill, Inc., 1995.

[82] van Lamsweerde, A., and Letier, E. Integrating obstacles in goal-driven requirements engineering. In *Proceedings of the IEEE International Conference on Software Engineering (ICSE)* (1998), pp. 53–62.

[83] von Winterfeldt, D., and Edwards, W. *Decision Analysis and Behavioral Research.* Cambridge University Press, 1986.

[84] Wainer, H. Estimating coefficients in linear models: It don't make no nevermind. *Psychological Bulletin 83*, 2 (1976), 213–217.

[85] Wang, H., and Xie, M. A comparative study of the prioritization matrix method and the analytic hierarchy process technique in quality function deployment. *Total Quality Management 9*, 6 (Aug. 1998), 421–430.

[86] Ward, A., Liker, J. K., Cristiano, J. J., and Sobek II, D. K. The second Toyota paradox: How delaying decisions can make better cars faster. *Sloan Management Review 36*, 3 (1995), 43–61.

[87] Waterson, M. Models of product differentiation. *Bulletin of Economic Research 41*, 1 (1989), 1–27.

[88] Watson, S. R., and Buede, D. M. *Decision Synthesis.* Cambridge University Press, 1987.

[89] West, D. C. Global marketing. In Dorf [**25**], pp. **22**–1–**22**–5.

[90] Wheelwright, S. C., and Clark, K. B. Creating project plans to focus product development. *Harvard Business Review* (Mar.–Apr. 1992), 70–82.

[91] Witte, E., and Zimmermann, H.-J., Eds. *Empirical Research on Organizational Decision-Making.* North Holland, 1986.

[92] Yen, J., and Tiao, W. A. A systematic tradeoff analysis for conflicting imprecise requirements. In *Proceedings of the third IEEE International Symposium on Requirements Engineering (ISRE)* (1997), pp. 87–96.

[93] Zave, P. Classification of research efforts in requirements engineering. *ACM Computing Surveys 29*, 4 (1997), 315–321.