Supporting requirement analysis through requirement rationale capture and traceability

Weili Dai

Imperial College London
Mechanical Engineering Department

A thesis submitted for the degree of Doctor of Philosophy

Declaration of Originality

Except where otherwise stated, this thesis is the result of my own research. This research was conducted in the Design Engineering Group at Imperial College between July 2010 and July 2014. I certify that this thesis has not been submitted in whole or in parts as consideration for any other degree or qualification at this or any other institute of learning.

Weili Dai

19th July 2014

Abstract

Manufacturers of complex engineering systems are increasingly recognising the importance of identifying, understanding and satisfying stakeholders' needs in order to produce high-quality products. The analysis of these needs into a formal requirement specification is a time consuming and complex process for which little support is offered to design engineers. This can result in requirements being poorly documented and with little or no traceability to their origins.

This dissertation reports an investigation to understand the process of requirement analysis and develop computational support for this important phase of the engineering design process. The key argument of this research is that the existing practice of requirement analysis can be improved by providing better support for requirement rationale capture and enabling greater requirement traceability.

The research consisted of three main phases. In the first phase, literature related to the requirement analysis was reviewed and led to the creation of a requirement analysis model. In the second phase, the practices of a global engineering organisation were investigated using document analysis as well as interviews with and shadowing of company engineers. The research found that requirement analysis lacks support for requirement rationale capture and traceability. On the basis of this result, a workflow for requirement analysis was proposed. The workflow involves the use of the Decision Rationale editor tool to capture requirement rationale and enable requirement traceability. In the third phase, four studies were undertaken to validate the workflow. These studies investigated: 1) application of the workflow to requirements generated through reverse-engineering a low-complexity consumer product; 2) requirements extracted from documents produced by a graduate engineering team during a twelve-week project; 3) the requirement analysis process undertaken by two graduate engineering teams during twelve-week projects; and 4) requirements for a new aircraft engine development programme. The studies showed that the proposed workflow is feasible, practical, and scalable when applied to engineering projects. Requirement rationales were classified into categories, namely product design and use, pre-existing rationale, and project management. In order to fully support requirement traceability, it was found that it is important to make traceable four types of requirement transformations: newly introduced, copied, updated, and deleted requirements.

The research demonstrated that the proposed workflow is a successful proof-of-concept and can lead to improved quality of requirement documentation and requirement traceability.

Acknowledgments

I would like to thank my principal supervisor Dr Marco Aurisicchio for his guidance, his inspiration, and the great deal of time he has dedicated to support my research. I also would like to express my gratitude to Professor Peter Childs for the many advice and opportunities he offered me.

I would like to thank my family for their unconditioned support and encouragement.

I am grateful to my colleagues and friends at Imperial College. They have made every day at the college unforgettable.

I would like to acknowledge the financial and project support of Rolls-Royce and EPSRC.

Table of Contents

List of abbreviations

DRed	Decision Rationale editor
IBIS	Issue-Based Information System
KAOS	Knowledge Acquisition in autOmated Specification
QFD	Quality Function Deployment
OMG	Object Management Group
SEURAT	Software Engineering Using RATionale
SysML	System Modelling Language
XWB	eXtra Wide Body

Chapter 1 Introduction

Chapter outline

1.1 Research motivation

This research project investigates the topic of requirement analysis with the aim to propose and evaluate a workflow to support engineers. The research has been conducted in collaboration with Rolls-Royce, particularly working with power systems design engineers and systems engineers. Requirement analysis is the process spanning from the elicitation of stakeholders' needs to the specification of product (or system) requirements. Research to improve the requirement analysis process is significant because it addresses many emerging challenges commonly faced in the development and manufacture of large-scale and complex engineering systems.

Challenge 1: Project management complexity and costs

Management of projects to develop large-scale engineering systems, such as designing gas turbines for aerospace applications and nuclear power plants, is a non-trivial task and the risk of cost overrun is very high. It is widely acknowledged that one of the commonest causes of cost overrun is requirement errors, i.e. what is built is different to what stakeholders had wanted. In addition, it is known that the cost to correct misinterpreted requirements is exponentially more than the cost to get it right at the beginning of development (Glass, 2003). Boehm et al. (2001) claims, for example, that the ratio of fixing requirement errors early in the development process versus fixing them late can be as high as 1:100. Similar views are shared by Dowlatshahi (1992) and Miles and Swift (1998). Therefore, being able to conduct requirement analysis effectively directly impacts the delivery of design solutions within budget.

Challenge 2: Evolutionary products and redesign

Large-scale engineering systems are evolutionary in nature. Design is based on incremental improvements rather than attempting a complete new design, as the re-use of a proven design solution reduces costs and risks. In order to re-use past designs, it is important to be able to understand the design requirements. Therefore, being able to document easily-interpretable requirements is a key factor influencing the effectiveness of design re-use.

Challenge 3: Global product development

A geo-distributed workplace is now very common and collaboration using new communication technologies is becoming the norm. For example, in Rolls-Royce there are 40,000 employees of which 45% are outside the UK (Glazier, 2011). Requirement elicitation and analysis is a collaborative decision-making activity (Christel and Kang, 1992). However, in a geo-distributed workplace traditional ways of communicating and analysing requirements such as brainstorming new requirements, structuring requirements, or debating the necessity of requirements, may no longer be adequate. Face to face requirement communication is increasingly being substituted by digital communication and information sharing, such as email exchanges or video-conferencing. One innovation which embraces this transition to digital communication is issue tracking systems such as JIRA[1], primarily used in the software industry to record discussions. The benefit offered by this platform is its standardised procedure to present and resolve an issue. Similarly, this research project provides a workflow, consisting of software support and methods to conduct requirement analysis; the workflow provides an approach to support engineering teams to contribute and share requirements in multiple geographic-locations and at different times.

Challenge 4: Increased outsourcing
According to Corbett (2010), we have entered the era of outsourcing, i.e. delegating non-core business functions to specialist suppliers instead of relying on in-house workforce. Corbett argues that the outsourcing strategy brings the benefit of market competition. Modern aircraft programmes, such as the Airbus A350 XWB (eXtra Wide Body) programme (AIRBUS, 2012), feature around 100 major work packages that are subcontracted to a large variety of suppliers. Specialist suppliers like these must strive for high-quality and low-cost in order to survive. Outsourcing is considered the natural extension to previous efficiency improvement movements (Sharp and McDermott, 2008), most notably process specialisation that shifted industry from craft production to mass production, and business specialisation that saw businesses changing from a monolithic entity to specialised functional units. When outsourcing is used, being able to produce a set of clearly defined requirements is especially important. The requirement list will serve as a contract that is signed off between directly interfacing partners (Browne and Zhang, 1999). Therefore, only a set of high-quality concise requirements can ensure that the supplier designs the right solutions.

Challenge 5: Increasingly transient workforce
Projects such as gas turbines can span over a period of many years, and it is unrealistic to expect everyone involved in the project to be present from project conception to completion. As a

[1] www.atlassian.com/software/jira

result, there is a need for any valuable project knowledge to be made readily transferrable. The requirements list is one such type of knowledge. In order to be transferrable, a requirements list should be easy to interpret including understanding the meaning of each requirement and how the requirements were stated in their current form from an initial set of vague stakeholders' needs. This research project proposes a workflow to capture *requirement rationale* and enable *requirement traceability* so that requirements are made easier to interpret and traceable.

1.2 Research aim and objectives

This project was carried out with the overall aim to investigate how to improve requirement analysis and documentation by supporting computer-based capture of requirement rationale and enabling improved requirement traceability. Requirement rationale is defined as information to support or oppose a requirement as well as detailing alternative requirements considered in the design process. Requirement traceability is defined as the ability to navigate forward and backward requirement models by means of capturing information to trace the evolution of requirements. Requirement rationale and traceability information are considered as types of requirement metadata. Requirement metadata is defined as any supplementary information enriching requirements.

In order to address the research aim the following objectives were established.

Objective A: Develop an understanding of requirements, requirements metadata, the requirement analysis process, and methods and tools to support requirement analysis.
Objective B: Develop an understanding of requirement analysis in existing engineering practice.
Objective C: Develop a workflow to support the capture of requirement rationale and to enable improved requirement traceability whilst integrating with existing practices.
Objective D: Validate the workflow in a range of contexts of increasing complexity.
The aim and objectives of this research project are based on research gaps identified during the literature review, and the needs of the collaborating company.

1.3 Research approach

In this project, empirical research was undertaken to investigate the aim and objectives stated in the previous Section. In order to fulfil the research aim and objectives three main research phases were undertaken, see Table 1. The aim and objectives have been fulfilled by means of a literature review and five case studies.

Table 1 Project information

Research phase	Objective	Study	Data collection method	Outcome	Chapter
Phase 1	A	Literature review	Publication review	• Reviewed definitions regarding to requirement, requirement lifecycle, requirement metadata, and requirement rationale; • Defined requirement analysis in the domain of design engineering, systems engineering, and requirements engineering; • Synthesised a model of requirement analysis; • Evaluated support for requirement analysis.	2
Phase 2	B C	Study 1	Interviews, document analysis, and observations with shadowing	• Constructed a view of the current requirement analysis practice at the collaborating company; • Identified areas of improvements in requirement analysis support; • Proposed a workflow for performing requirement analysis.	4
Phase 3	D	Study 2	Reverse-engineering and analysis to structure requirements	• Validated the feasibility of the proposed workflow.	5
		Study 3	Document analysis and interview	• Validated the feasibility of the proposed workflow; • Defined a model to describe requirement evolution.	6
		Study 4	Action research, document analysis, and interview	• Validated the feasibility of the proposed workflow; • Validated the practicality of the proposed workflow.	7
		Study 5	Document analysis, interviews, and questionnaire	• Validated the feasibility of the proposed workflow; • Validated the practicality of the proposed workflow; • Validated the scalability of the proposed workflow.	8

Table 1 shows the mapping between the research phases, the objectives, and the studies undertaken to fulfil them. For each case study Table 1 shows also the data collection method employed, the main outcome, and the Chapter presenting it.

The research was divided into three phases: 1) Knowledge foundation; 2) Research grounding and proposal of a requirement analysis workflow; 3) Implementation and evaluation of the workflow. The phases are now introduced showing the research undertaken in each of them.

Phase 1 – Knowledge foundation

The goal of this phase was to review literature on the requirement analysis process and related concepts. This consisted of five steps. In the first step, an understanding of requirements is first explored from a low-level. This includes surveying literature on definitions of requirement, requirement types, and requirement metadata. In the second step, an understanding of the requirement process is explored from a high-level. This consisted of surveying literature on engineering processes that include requirement analysis as a sub-process. The engineering processes investigated include the design engineering process, systems engineering process, and requirement engineering process. In the third step, the knowledge gained from the previous literature explorations was consolidated and a model of requirement analysis was synthesised. In the fourth step, existing methods and tools support available for requirement analysis were evaluated. In the fifth step, a research gap was identified based on the previous literature survey and evaluations of existing requirement analysis support.

Phase 2 – Research grounding in industrial practice and proposal of a requirement analysis workflow

The goal of this phase was to propose a workflow for requirement analysis that would incorporate improvements identified in this phase and in Phase 1. Improvements were identified by evaluating the current requirement analysis practice at the collaborating company, combined with the evaluation of various types of requirement analysis support in Phase 1. Information about the current practice was obtained through interviews, document analysis, and shadowing existing engineers working at the collaborating company. The current practice was analysed in terms of the methods and tools used, and the process that define how and when the methods and tools should be used.

Phase 3 – Implementation and evaluation of the workflow
The third phase is implementing the proposed workflow and evaluating how well the workflow performs when used in different scenarios. The proposed workflow has been applied to several projects. These projects are selected so that there is a wide range of variation in project properties, for example, in terms of method of research intervention (i.e. active or passive), data origin, and project size. These controlled parameters help to validate the proposed workflow in terms of feasibility, practicality, and scalability. Chapter 3 elaborates on the methodologies of validation.

1.4 Contributions

This dissertation has several contributions. First, a model of the requirement analysis process was formulated through consolidation of relevant literature. Second, a workflow to conduct requirement analysis was proposed based on the model of requirement analysis. The workflow integrates and extends current practices by using a novel approach to capture requirement rationale and enabling requirement traceability. Third, the data collected in this research enabled the characterisation of the requirement analysis practices of a global company. Fourth, new knowledge was developed about requirement rationales. This knowledge includes the common categories of requirement rationale, and the circumstances in which requirement rationale would likely conceive from. Fifth, new knowledge is gained about requirement evolution and how best to support requirement traceability. Sixth, this research has developed an understanding of the requirement analysis process for physical systems, whilst the majority of similar researches are aimed at software systems development.

1.5 Thesis outline

Chapter 1 introduces the research motivation, aim, objectives, approach, and contributions.

Chapter 2 places requirement analysis into context. The Chapter discusses requirement analysis on a micro-level, by presenting research about requirements, and on a macro-level, by discussing processes that subsume requirement analysis. In addition, a Section is dedicated to discussing methods and tools available to support requirement analysis.

Chapter 3 discusses the research framework used for this research project and how the research was validated.

Chapter 4 describes a case study to identify practices used in industry to support requirement analysis. The practices, combined with the methods and tools discussed in Chapter 2, are evaluated to formulate a set of criteria for effective requirement analysis support. A workflow is then designed and proposed to meet those criteria.

Chapter 5 presents a case study to determine the feasibility of the proposed workflow using data generated from reverse-engineering a hair-dryer.

Chapter 6 presents a case study to determine the feasibility of the proposed workflow using industrial data about a portable material sampling machine.

Chapter 7 presents a case study to determine the feasibility and practicality of the proposed workflow by working with graduate engineers from the collaborating company.

Chapter 8 presents a case study to determine the feasibility of the proposed workflow when it is scaled. The requirement analysis process for a whole-engine project was studied.

Chapter 9 summarises the main results showing how they answer the research objectives, presents the research contributions, discusses the research results and draws the overall conclusions.

Chapter 2 Literature review

Chapter outline

Requirement analysis is a key part of any engineering project. Yet, there is little consensus in the literature as to what exactly is the process of requirement analysis. This Chapter of the dissertation aims to review literature related to requirement analysis and to consolidate existing understanding.

This Chapter consists of five main Sections that investigate the process of requirement analysis from different angles. First, from a micro-perspective, requirement types, the requirement lifecycle, and requirement metadata are explored. Second, from a macro-perspective, different process models that subsume the requirement analysis process are analysed. Third, from a design knowledge-perspective, a model of the requirement analysis process is synthesised based on other process models in literature. Fourth, from the perspective of practical implementation, methods and tools available to support requirement analysis are evaluated. Fifth, the definitions and importance of requirement rationale and requirement traceability are explained.

2.1 Requirements in engineering processes

This Section provides a micro-perspective on requirement analysis and defines terms that shall be referred to in the rest of the dissertation.

In requirement elicitation and analysis, a requirement is considered the most basic unit of analysis – a piece of natural language statement describing what the system to be designed should be or should do (Alexander and Stevens, 2002). The system could be either a product or a process (Hull et al., 2010). In addition, the requirement statement should have properties that improve the quality of requirements (Génova et al., 2013). These properties include, amongst others, unambiguity and understandability, completeness, traceability, and atomicity (Hull et al., 2010; Génova et al., 2013; IEEE STD 1220-1998, 1998; Buede, 2009; Grady, 1993; Davis, 2005). The typical lifecycle of a requirement is described next.

2.1.1 Requirement lifecycle

A requirement would typically experience three stages of evolution during its lifetime (Hull et al., 2010). First, a requirement begins its life as a *stakeholder requirement*. Second, through an initial process of categorisation, the stakeholder requirement evolves into one or more *system requirement(s)* (Buede, 2009). Third, the system requirements enter a phase of decompositions that breaks down the system requirement into more detailed *subsystem-level* and *component-*

level requirements. Figure 1 shows this process on a diagram, part of the system engineering model of project development which will be elaborated later in this Chapter.

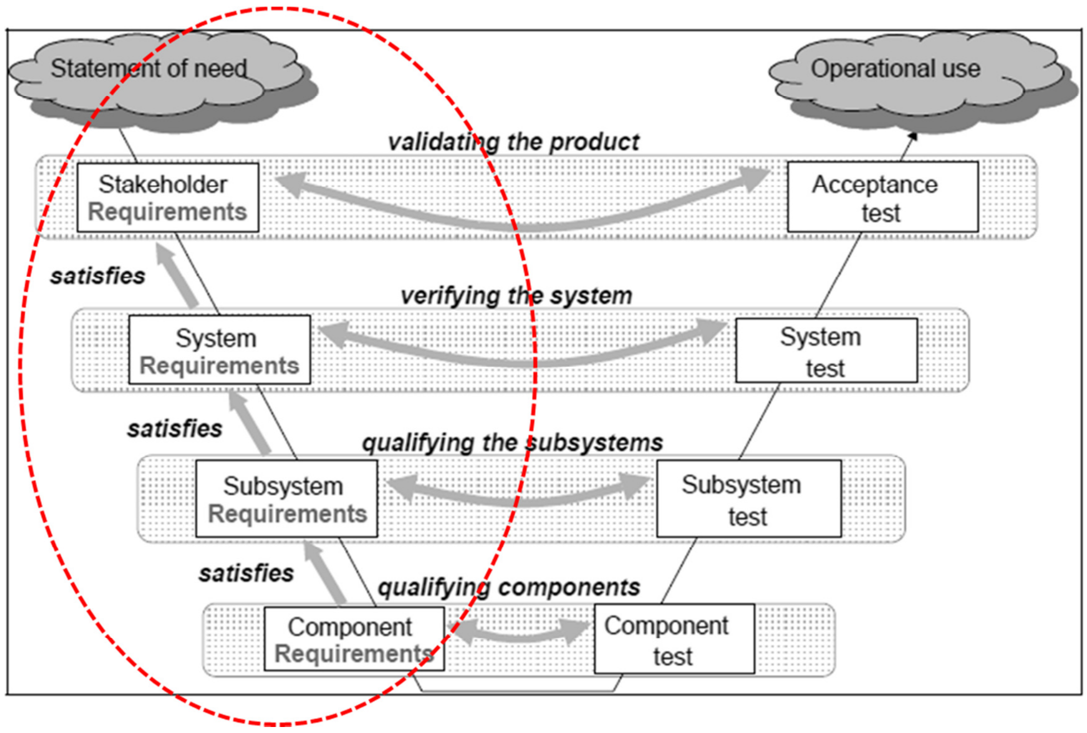


Figure 1 Requirement lifecycle (Dick and Chard, 2004)

Stakeholder requirements

Stakeholder requirements, often known as the Voice of the Customer (Akao, 2004), define what the stakeholders want the solution system to be and do at high level (Hull et al., 2010), normally focusing on the boundary of the system (Buede, 2009). By nature stakeholder requirements are general, ambiguous, and un-measurable (Burge, 2007).

System requirements

System requirements, often known as the Voice of the Engineer (Akao, 2004), define what the solution system will do to meet the stakeholder requirements (Hull et al., 2010). System requirements are a translation (or derivation) of the stakeholder requirements into engineering terminology (Buede, 2009). System requirements also define *how much* using measurable requirement satisfaction criteria. System requirements are specific, precise, and measurable (Burge, 2007).

The process of evolving stakeholder requirements into system requirements is a process of categorisation. System requirements can be either functional – describing what the system should do, or non-functional – describing what the system should be or have, i.e. its qualities (Young, 2003; Burge et al., 2008), see Table 2. Hull et al. (2010) further divide the two types of requirements for a product or system such that the functional requirement can be either operational or functional; and a non-functional requirement can be either a design characteristic (such as portability, reliability, and maintainability) or a constraint (such as limit on size, life-expectancy, and frequency of maintenance of a system) (Hull et al., 2010), see Table 2. The Holistic Requirement Model (HRM) (Burge, 2006) in Figure 2 consists of five requirement types and can be considered as an extension to the four requirement types defined by Hull et al. (2010), see Table 2. In addition, the HRM establishes causal relationships between different types of requirements. In this model, operational requirements define the major purpose of a system. Functional requirements specify what the system has to do in order to achieve the operational requirement. Non-functional system requirements define characteristics that apply to the whole or a significant proportion of the system. Non-functional implementation requirements define how a system is to be built in terms of specific technology. Non-functional performance requirements define how well a function has to perform. Table 3 shows an example for each of the five requirement types in the HRM.

Table 2 Requirement categories by different authors

General consensus (Young, 2003; Burge et al., 2008)	Hull et al (2010)	HRM (Burge, 2006)
Functional	Operational	Operational
	Functional	Functional
Non-functional	Design characteristic	Non-functional systems
	Constraint	Non-functional implementation
		Non-functional performance

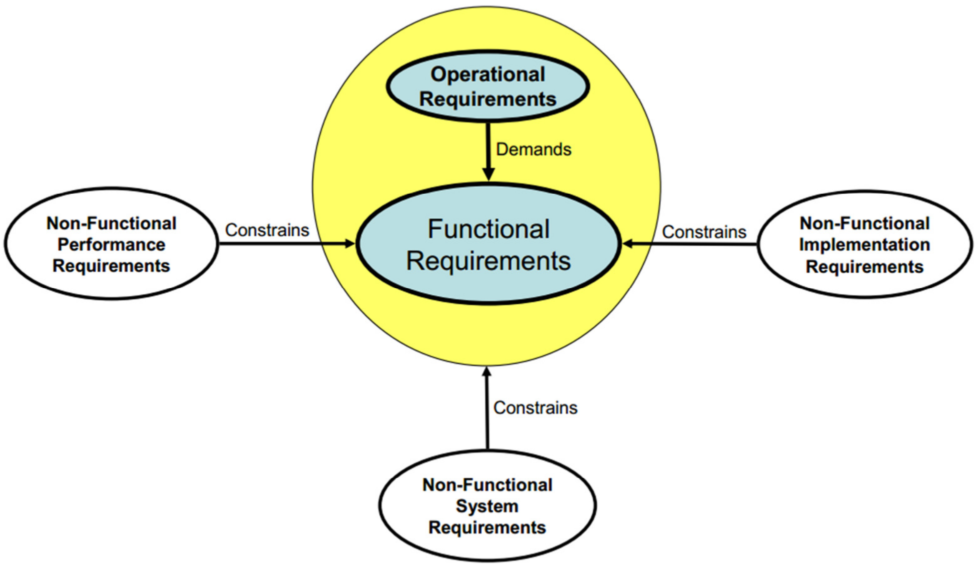

Operational Requirements
Demands
Non-Functional Performance Requirements
Constrains
Functional Requirements
Constrains
Non-Functional Implementation Requirements
Constrains
Non-Functional System Requirements

Figure 2 The Holistic Requirements model (Burge, 2006)

Table 3 Examples of requirements categorised in the Holistic Requirement Model (HRM)

Requirement type in HRM	Requirement statement
Operational requirement	Toast bread
Functional	Provide heat to bread
Non-functional systems	Safe to use
Non-functional implementation	Must use UK domestic 13 amp plug
Non-functional performance	Provide heat with a tolerance of 5 degree Celsius

Requirement decomposition

Once requirements are correctly categorised, an iterative process of decomposition can begin on functional requirements. The iterative process continues until functional requirements are describing atomic components. Burge (2006) argues that the HRM of requirement categorisation can be consistently applied to all engineering projects to facilitate requirement decomposition. Once requirements are categorised, requirement categories on the system level would be mapped to certain categories on the sub-system level, see Figure 3. For example, functional requirements on the system level map to operational requirements on the sub-system level.

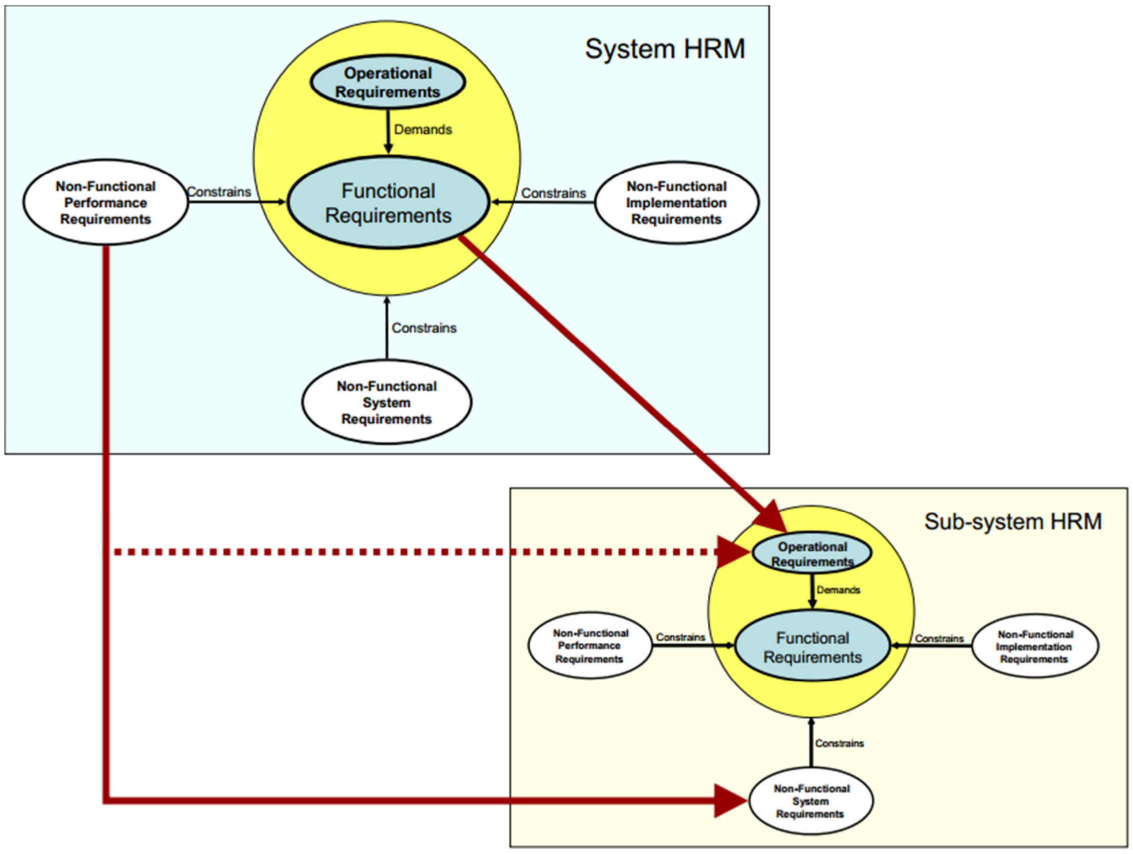

System HRM
Operational Requirements
Demands
Functional Requirements
Non-Functional Performance Requirements
Constrains
Constrains
Non-Functional Implementation Requirements
Constrains
Non-Functional System Requirements
Sub-system HRM
Operational Requirements
Demands
Functional Requirements
Non-Functional Performance Requirements
Constrains
Constrains
Non-Functional Implementation Requirements
Constrains
Non-Functional System Requirements

Figure 3 Decomposition in the Holistic Requirements model (Burge, 2006)

2.1.2 Requirement metadata

In the previous Section, a requirement was regarded as a textual statement. In addition to the statement, a requirement typically has metadata – this is supporting material such as, scenarios, diagrams, targets, and other values (Alexander and Stevens, 2002). A well-known template for capturing these metadata is the Volere index card (Robertson and Robertson, 1999). The Volere index card is a generic template for a requirement that specifies what supporting information should be captured along with the requirement, see Table 4. The template captures requirement metadata such as priority, originator, rationale, requirement type, and history of changes.

Table 4 Requirement metadata recommended by the Volere index card template

Type of metadata	Description
Description	A statement of the intention of the requirement
Use case	Use cases that need the requirement
Priority	A rating of the customer value
Fit criterion	A measurement of the requirement such that it is possible to test if the solution matches the original requirement
Originator	The person who raised the requirement
Rationale	A justification of the requirement
Requirement id	A unique id
Requirement type	The requirement type can be user defined classes, e.g. functional and non-functional
Customer satisfaction/dissatisfaction	The degree of stakeholder happiness if the requirement is successfully implemented The measure of stakeholder unhappiness if the requirement is not part of the final product.
Conflict	Other requirements that cannot be implemented if the requirement in consideration is
Supporting materials	Pointers to documents that illustrate and explain the requirement
History of changes	Changes, and deletions made to the requirement

Alexander and Beus-dukic (2009) proposed a similar model of requirements metadata, see Figure 4. They argue that there is no simple one-solution-fits-all template for requirements; instead, requirements are made of a set of commonly occurring elements that together define what is wanted by stakeholders. It is noteworthy that the metadata elements in Figure 4 have relationships between them.

Many attributes overlap between the Volere template and Alexander and Beus-duikic's model, and Table 5 shows the two compared side-by-side. It is worth noting that in both models, the requirement type is seen as a type of metadata. But in Alexander and Beus-duikic's model, the types are explicitly defined as function, quality or constraint. These three types have an approximate equivalence to the HRM mentioned in Section 2.1.1 where function refers to functional or operational requirement, quality refers to non-functional implementation requirements, and constraint refers to non-functional systems or non-functional performance.

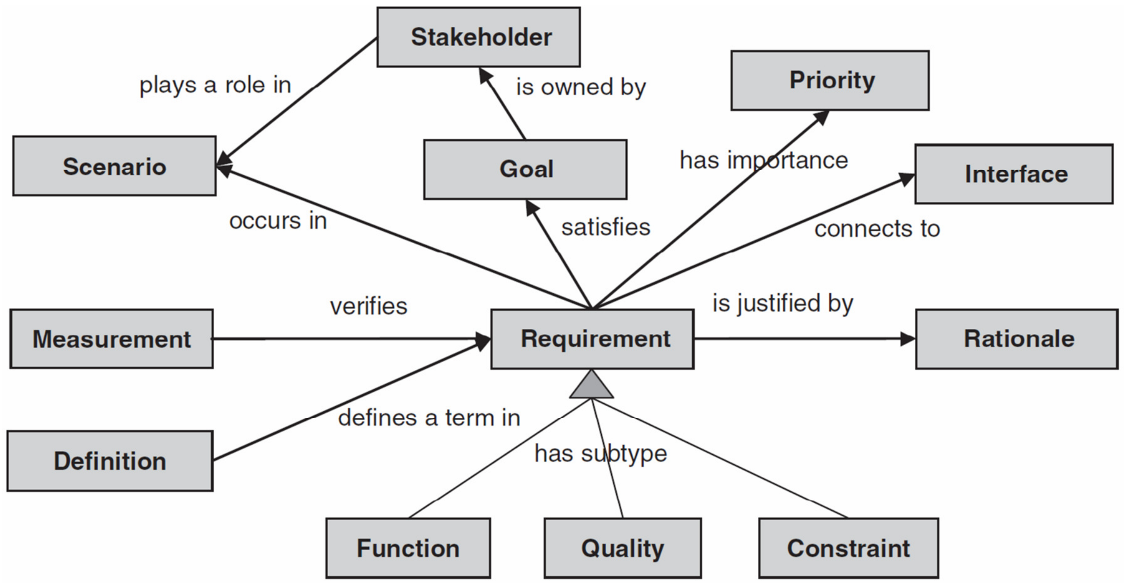


Figure 4 Diagram illustrating the contextual information of a requirement (Alexander and Beus-dukic, 2009)

Table 5 Comparison of two of the most popular requirement metadata models

Volere template	Alexander and Beus-dukic's model
Description	Requirement
Use case	Scenario
Priority	Priority
Fit criterion	Measurement
Originator	Stakeholder
Requirement type	Function, quality, or constraint
Rationale	Rationale
Requirement id	
Customer satisfaction/dissatisfaction	
Conflict	
Supporting materials	
History of changes	
	Goal
	Interface
	Definition

From Table 5 it can be seen that there is a large proportion of overlapping requirement metadata attributes between the two models. These are description, use case, priority, fit criterion, originator, requirement type, and rationale. The overlap in these two highly recognised models in the literature implies that these attributes can be considered as the fundamental types of requirement metadata. A comparison of the non-overlapping attributes indicates that the Volere template is more concerned with implementation details than the Alexander and Beus-duikic's model, for example requirement id, customer satisfaction/dissatisfaction, and history of changes.

Out of all the attributes described in Table 5 requirement rationale is the most generic and flexible requirement metadata attribute. This means that many types of requirements metadata can be expressed as a rationale. For example, the attribute *priority* can be substituted by a requirement rationale. A requirement rationale can justify importance and necessity of a requirement (Alexander and Beus-Dukic, 2009) because requirement rationale captures the underlying intent behind the requirement (Burge et al., 2008). The attribute, *the originator*, can also be substituted by requirement rationale. Dick argues that "the rationale would be a logical place to capture the source of the requirement" (Dick, 2005). The attribute, *use case*, on the Volere template and, *scenario*, on Alexander and Beus-duikic's model can both be described in the justification rationale of a requirement. For example, in OMG (2008) requirements for the design of a Hybrid Sports Utility Vehicle were elicited. One of the requirements is an implementation requirement which states that a 2-wheel drive-system must be used. The rationale for this requirement is a scenario describing the justification that a 2-wheel drive-system is the only way to obtain acceptable fuel economy, even though it compromises off-road capability. This trait of being generic makes requirement rationale an invaluable metadata to capture during requirement elicitation and analysis.

2.2 Engineering processes and requirement analysis

This Section provides a macro-perspective on requirement analysis; it explores well-known engineering process models that incorporate requirement analysis as a sub-process. These process models are introduced here in order to convey the contexts in which requirement analysis occurs. There are three streams of research which proposes process models that are supersets of the requirement analysis process: design engineering, systems engineering, and requirement engineering. Design engineering and system engineering are two different approaches to develop an engineering solution. Requirement engineering is a process specific to the handling of requirements in a project.

2.2.1 Design engineering

In the domain of design engineering, four well-known (Maffin, 1996; Wynn and Clarkson, 2005) process models of the design engineering process were proposed by French (1999) , Pugh (1991), Pahl and Beitz (1996), and Hales (2004). All of these models can be generalised to contain the stages of requirement specification and design.

The model by French, shown in Figure 5 is a stage-based model based on design practice observed in industry. It consists of four main *stages*, namely analysis of problem, conceptual design, embodiment of schemes, and detailing; and four types of *data*, namely need, statement of problem, conceptual design, and working drawings. First, a market *need* is observed, which is clarified in the *analysis of problem* stage. Second, the *analysis of problem* stage also elaborates the need into a detailed requirement specification, or *statement of problem*. The first two stages describe the requirement analysis process. It consists of activities that transform ambiguous market need into unambiguous requirement specification. Third, the *statement of problem* is used for the development of multiple concepts in the *conceptual design* stage. Fourth, selected concepts are represented as a set of physical principles for solving the problem, or *selected schemes*. These schemes are then solidified in the *embodiment of schemes* stage, and further optimised in *detailing* stage, before detailed work drawings are produced. The stages should be followed from top to bottom, but at every stage, it is possible to make use of newly acquired knowledge as feedback to enhance the data that had been captured before.

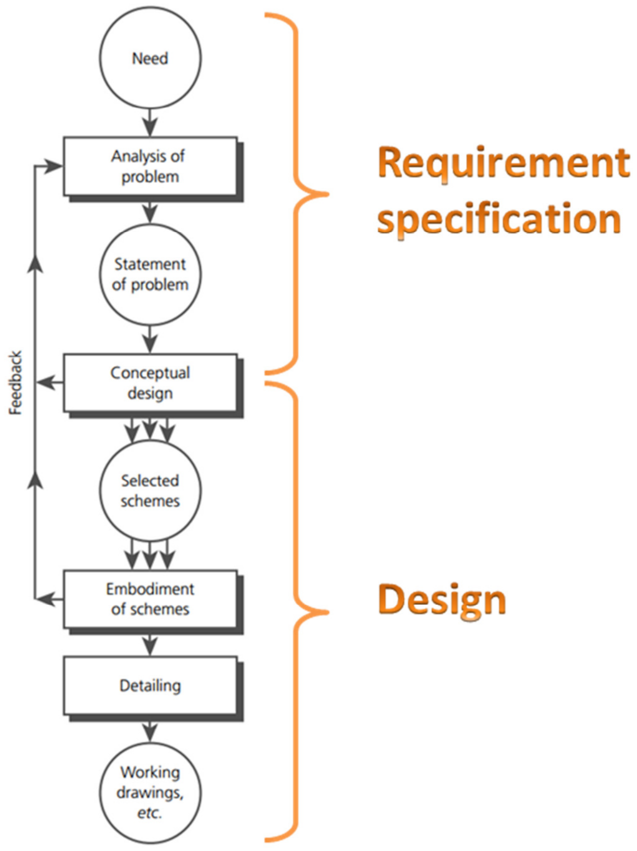


Figure 5 Model of engineering design by French (1999)

The model by Pugh, shown in Figure 6, extends French's model by including stages relating to manufacture, marketing and sales. But Pugh's model does not identify embodiment design as a stage between conceptual and detail design. Pugh's reasoning for this is that embodiment issues are often addressed very early in the design process, and embodiment is therefore a subset of conceptual design. Pugh also makes a distinction between original design (designing from the ground up) and adaptive design (building a variant of an existing solution). Similar to French's model, Pugh's model also views requirement analysis as a process between the identification of market needs and the definition of a requirement specification. However, Pugh additionally define the adaptive design scenario. In such circumstances, requirements would be constrained to existing product platforms and legislative requirements.

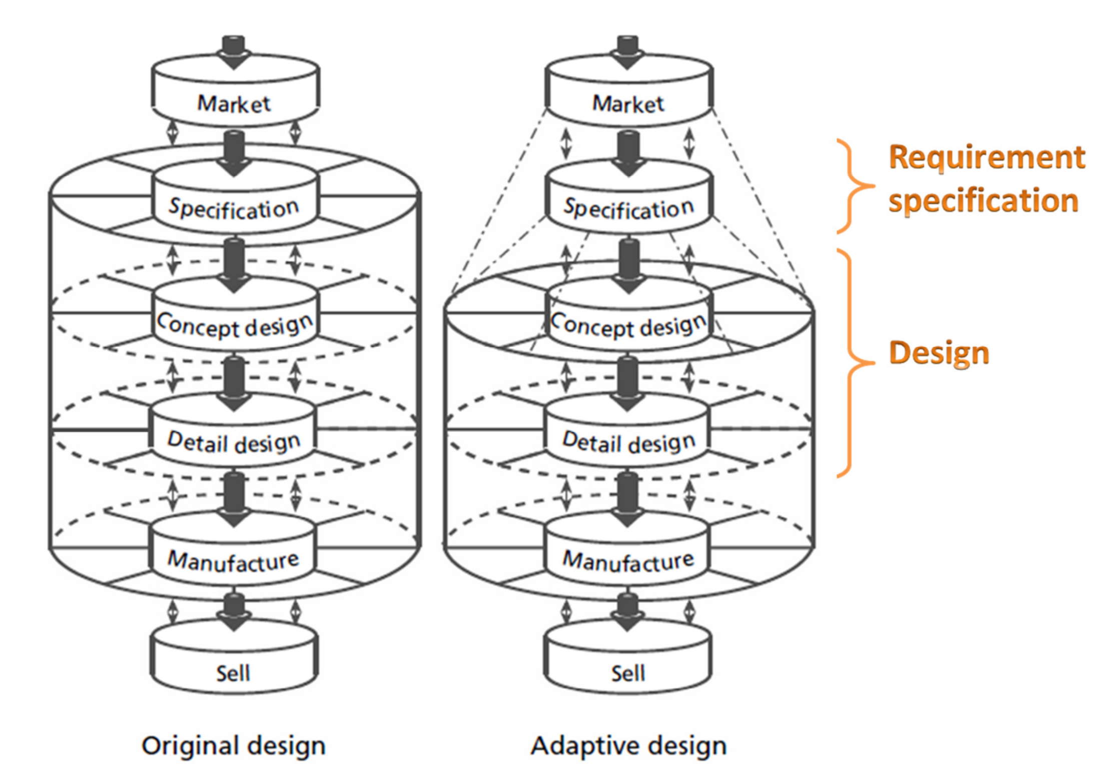


Figure 6 Model of engineering design by Pugh (1991)

The model by Pahl and Beitz, shown in Figure 7, is perhaps the most well-known model in mechanical design. It also consists of four main stages, namely clarification of the task, conceptual design, embodiment design, and detailed design. It differs from the previous two in that it extends the main stages to include intermediate stages. It contains four prescribed stages of working steps which are guidelines for design. Compared to French's model, Pahl and Beitz's model groups clarification and specification into one stage, whilst elaborates on the conceptual design stage by splitting it into two stages, i.e. conceptual design and embodiment design. Pahl and Beitz' model considers requirement analysis as identifying of the task and clarifying the task to evolve ambiguous requirements into an unambiguous requirement list.

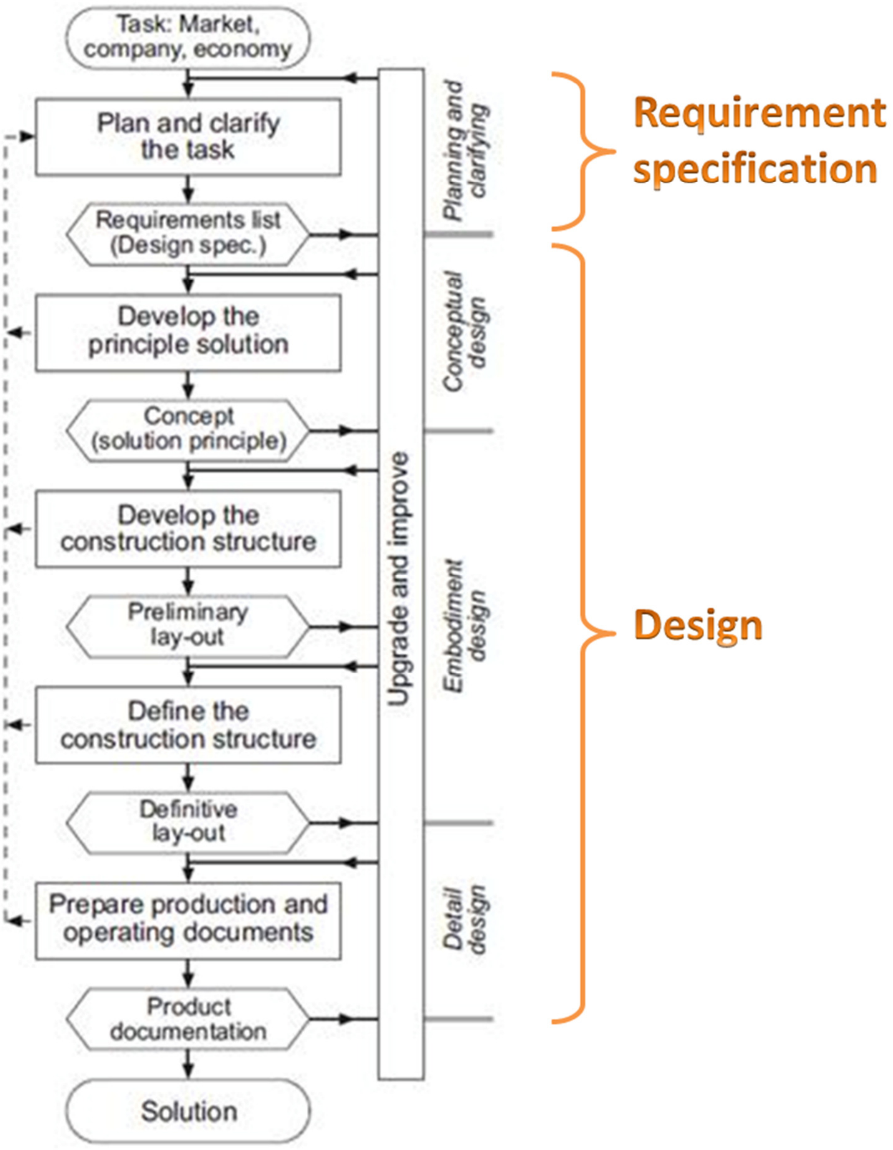

Task: Market, company, economy
Plan and clarify the task
Requirements list (Design spec.)
Develop the principle solution
Concept (solution principle)
Develop the construction structure
Preliminary lay-out
Define the construction structure
Definitive lay-out
Prepare production and operating documents
Product documentation
Solution
Upgrade and improve
Planning and clarifying
Conceptual design
Embodiment design
Detail design
Requirement specification
Design

Figure 7 Model of engineering design by Pahl and Beitz (1996)

The model by Hales, shown in Figure 8, integrates contextual influences into the design process, such as market, company, and management. However, at the core of this model it still retains many familiar components to the models presented previously. Hales' model, in terms of the design process, consists of four stages, namely task clarification, conceptual design, embodiment design, and detail design, and four types of data, namely specification, concept, layout, and manufacturing information.

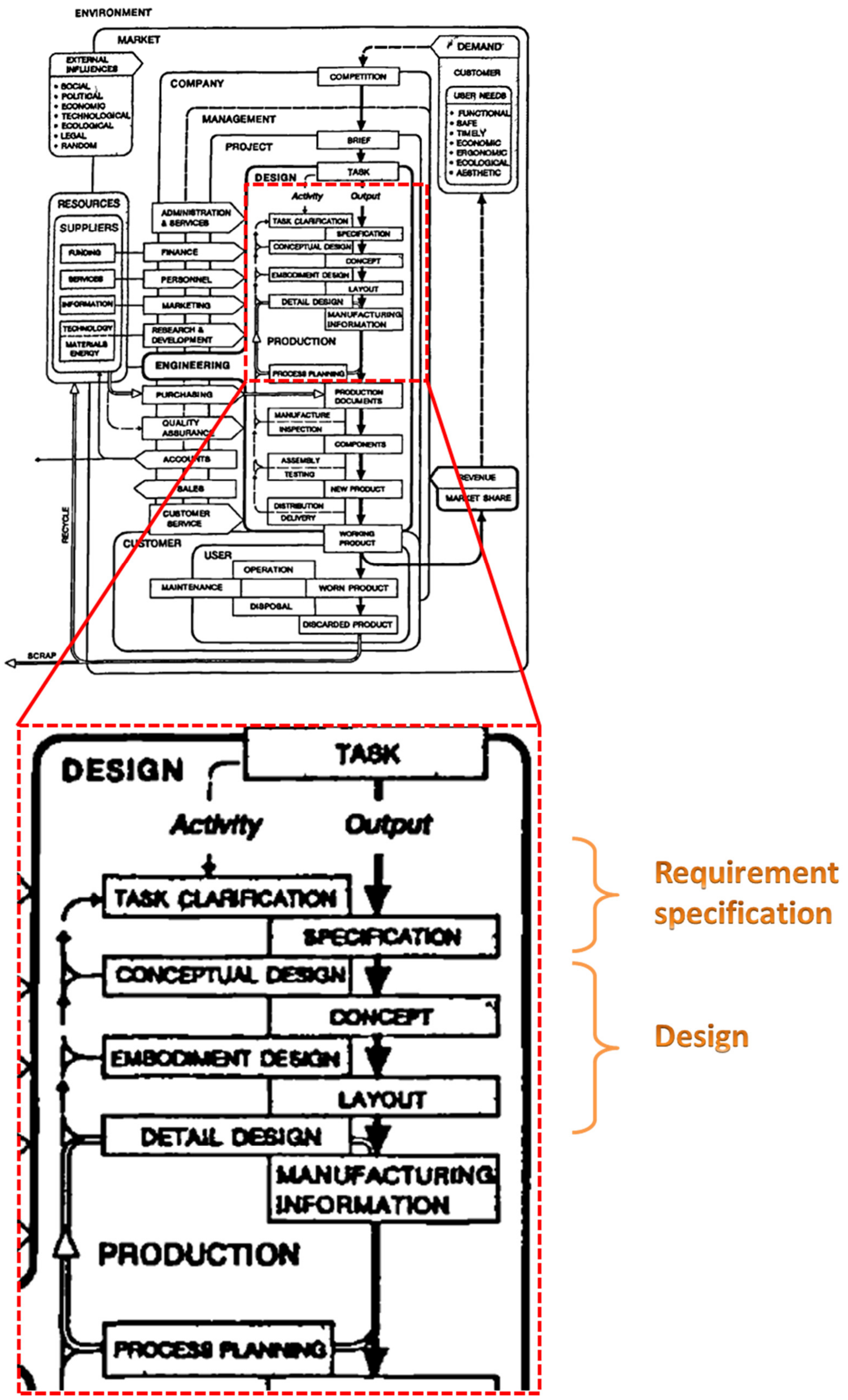


Figure 8 Model of engineering design by Hales (2004)

All four models of the engineering design process described above follow the pattern of first establishing requirements, then starting the process of design. In all four models, requirements are clarified and specified. Clarification is the act of ensuring that all requirements are captured in a way that can be clearly interpreted. It includes stating the problem (Bieniawski, 1993) and defining the stakeholders, i.e. "who needs to be involved in the decision-making as well as who will be affected by the problem's formulation and eventual solution" (Christel and Kang, 1992). Specification includes the phase of capturing the voice of the customer and translating it to system requirements. During this phase stakeholder requirements are redefined into more specific requirement types, in order to facilitate the construction of function structures (Pahl and Beitz, 1996) and basic architectures of the solution.

2.2.2 Systems engineering

Unlike in design engineering where there are many process models to guide engineering projects, in systems engineering there is just one model. The systems engineering model is universally accepted and known as the Vee model because it represents the shape of the English letter "V". One of the most widely referenced Vee model is by Forsberg and Mooz (1992), shown in Figure 9.

The Vee model is a graphical representation of the lifecycle of a system. On the left of the "V" there are requirements, and on the right there are matching solutions to the requirements. Note that the term "user requirements" in Figure 9 is equivalent to "stakeholder requirements" defined in the previous Section. Requirements and solutions are layered according to requirement scope with the requirements covering multiple systems at the top and component-specific requirements at the bottom. At every layer, requirements can be validated or verified. Validation checks whether system requirements are correctly interpreted from user requirements, verification checks if the solution satisfy requirements (Hull et al., 2010).

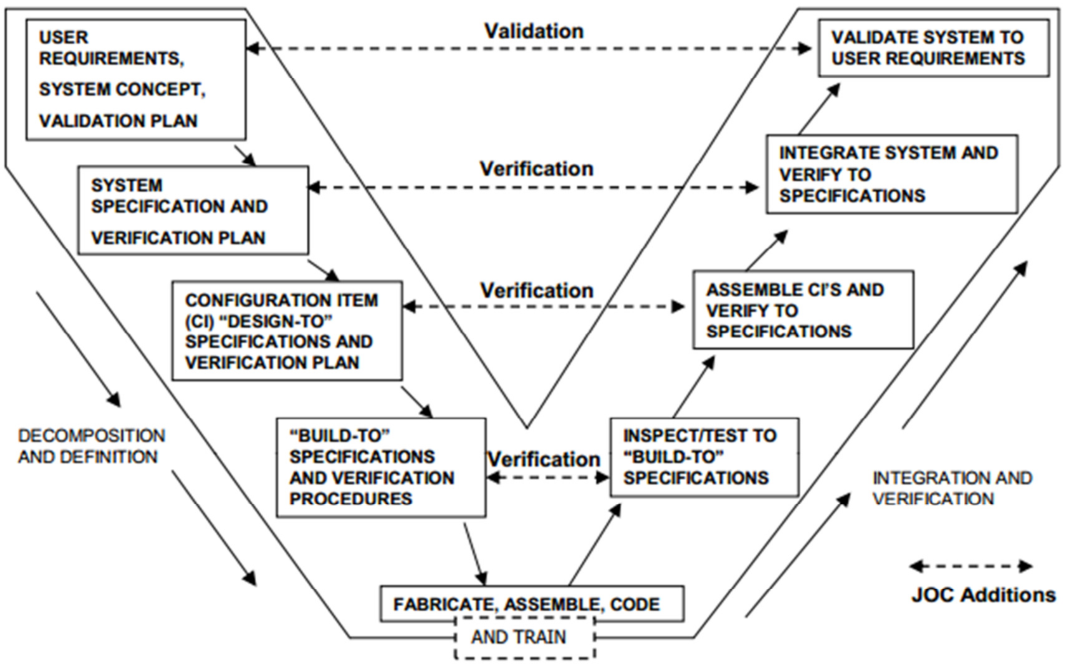

USER REQUIREMENTS, SYSTEM CONCEPT, VALIDATION PLAN
Validation
VALIDATE SYSTEM TO USER REQUIREMENTS
SYSTEM SPECIFICATION AND VERIFICATION PLAN
Verification
INTEGRATE SYSTEM AND VERIFY TO SPECIFICATIONS
CONFIGURATION ITEM (CI) "DESIGN-TO" SPECIFICATIONS AND VERIFICATION PLAN
Verification
ASSEMBLE CI'S AND VERIFY TO SPECIFICATIONS
DECOMPOSITION AND DEFINITION
"BUILD-TO" SPECIFICATIONS AND VERIFICATION PROCEDURES
Verification
INSPECT/TEST TO "BUILD-TO" SPECIFICATIONS
INTEGRATION AND VERIFICATION
FABRICATE, ASSEMBLE, CODE
AND TRAIN
JOC Additions

Figure 9 Vee model by Forsberg and Mooz (1992)

Compared to design engineering models, the Vee model describes the requirement analysis process in greater detail. In addition to the stages of clarification, namely clarifying and validating user requirements; and specification, namely specifying system requirements, the Vee model has the step of decomposition. Decomposition is the process of making requirement more specific at sub-system level (sometimes referred to as flow-down). This process of requirement decomposition has two effects (Sage, 1992): 1) requirements are described more precisely, often with numerical measurements; 2) requirements map to only a subset of components, instead of applying to the whole system.

2.2.3 Requirement engineering

This Section examines process models specific to activities which transform requirements. The previous two Sections discussed processes that describe all the activities taking place in the development of a system; all the processes agree that requirements begin as stakeholder requirements which are then transformed into system requirements. In his research on requirement engineering Gotel and Finkelstein (1994) identified two phases termed Pre-Requirement Specification (Pre-RS) and Post-Requirement Specification (Post-RS).

Pre-Requirement Specification (RS)

The Pre-RS phase generalises activities that are carried out before a requirement specification (previously referred to as systems specification) is formalised. Pre-RS is also known as requirement development (Brace and Cheutet, 2011; Wiegers, 2000). Two of the most cited models that elaborates in detail requirement development as a stage-based process are Kotonya and Sommerville (1998) and Robertson and Robertson (1999). A third model by Pohl (1993), also one of the most cited requirement engineering models, takes a different perspective by viewing requirement development as a continuous process with three goals. These models are now reviewed in detail.

Kotonya and Sommerville (1998) defines the requirement development process as requirement elicitation, analysis, and negotiation, see Figure 10. It is viewed as an iterative process such that new requirements are elicited after previous requirements have been analysed and negotiated. Elicitation consists of obtaining requirements from stakeholders. Requirement analysis consists of a series of checks, including checking for necessity, completeness, consistency, and feasibility. Negotiation is an activity that is tightly coupled with analysis. It consists of discussing those requirements uncovered during checking that are found to be unnecessary, conflicting, or infeasible. Figure 11 shows the coupling of requirement analysis and negotiation. It explains the causal relationship of *necessity checking* with *requirement discussion*, *consistency and completeness checking* with *requirements prioritisation*, and *feasibility checking* with *requirements agreement*. *Necessity checking* parses requirements for the ones that do not contribute to defining the problem to be addressed by the system; unnecessary requirements are discussed amongst stakeholders to allow them to justify whether these requirements are needed. *Consistency and completeness checking* involves cross-checking requirements to ensure that no contradictory requirements exists; and completeness means that no constraints which are needed have been missed out. Once conflicting requirements are identified, they are prioritised with justification to define critical requirements. *Feasibility checking* involves ensuring that the requirements are feasible in the context of the budget and schedule. Infeasible requirements identified are discussed amongst stakeholders so that they can agree on a compromising solution.

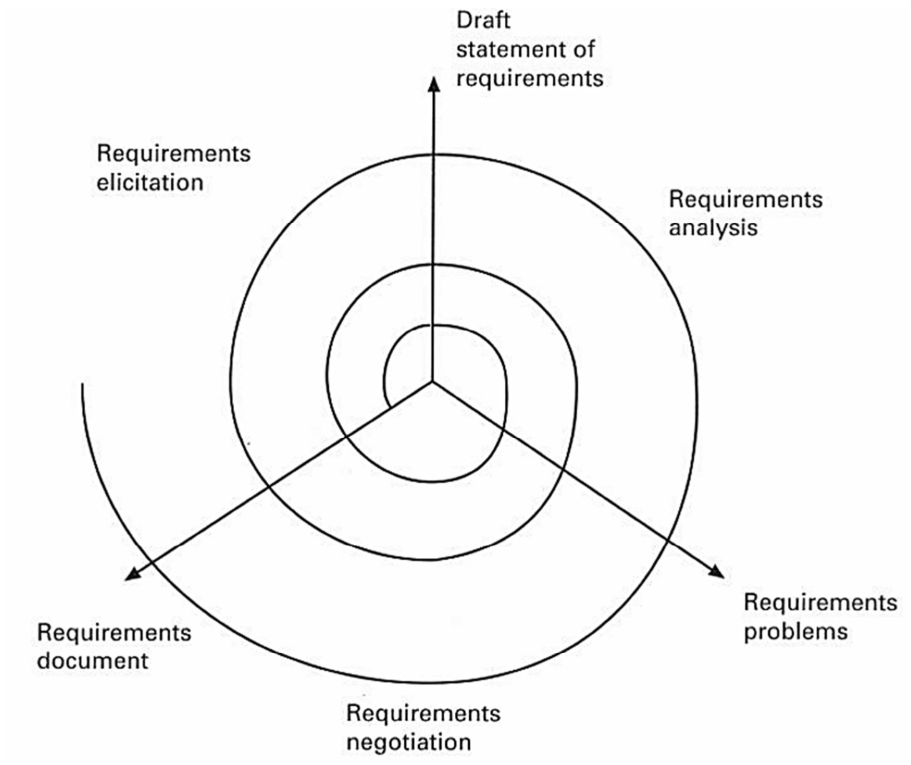


Figure 10 Model of the requirement development process by Kotonya and Sommerville (1998)

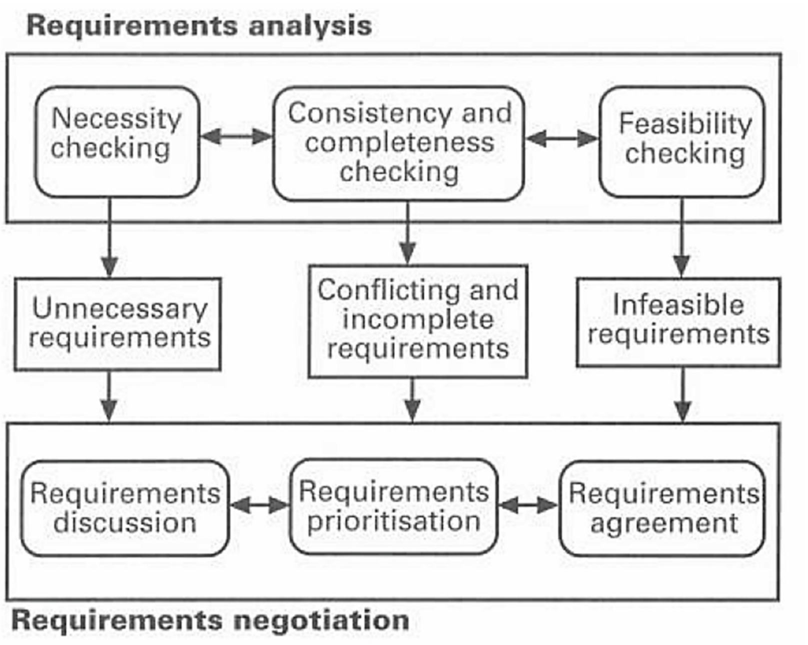


Figure 11 The relationship between requirement analysis and negotiation in Kotonya and Sommerville's model of requirement development (Kotonya and Sommerville, 1998)

Robertson and Robertson's model, shown in Figure 12, describe a superset of pre-requirement specification or the requirement development process. In Figure 12, the activities belonging to requirement development is labelled 1, 2, and 3. Robertson and Robertson's model is very similar to Kotonya and Sommerville's model. In Robertson and Robertson's model, "trawl for knowledge" is used to represent requirement elicitation. "Writing the requirements" is not a separate activity, but it is a parallel activity to trawling and quality gateway checks. It is an activity for capturing requirements and refining them. Finally, "Quality gateway" has the same effect of requirement analysis and negotiation on Kotonya and Sommerville's model. The quality gateway is a device for preventing incorrect requirements from becoming part of the specification. At the quality gateway, requirements are checked for completeness, consistency, necessity, and feasibility. And those requirements that do not pass the checks are discussed and resolved amongst stakeholders.

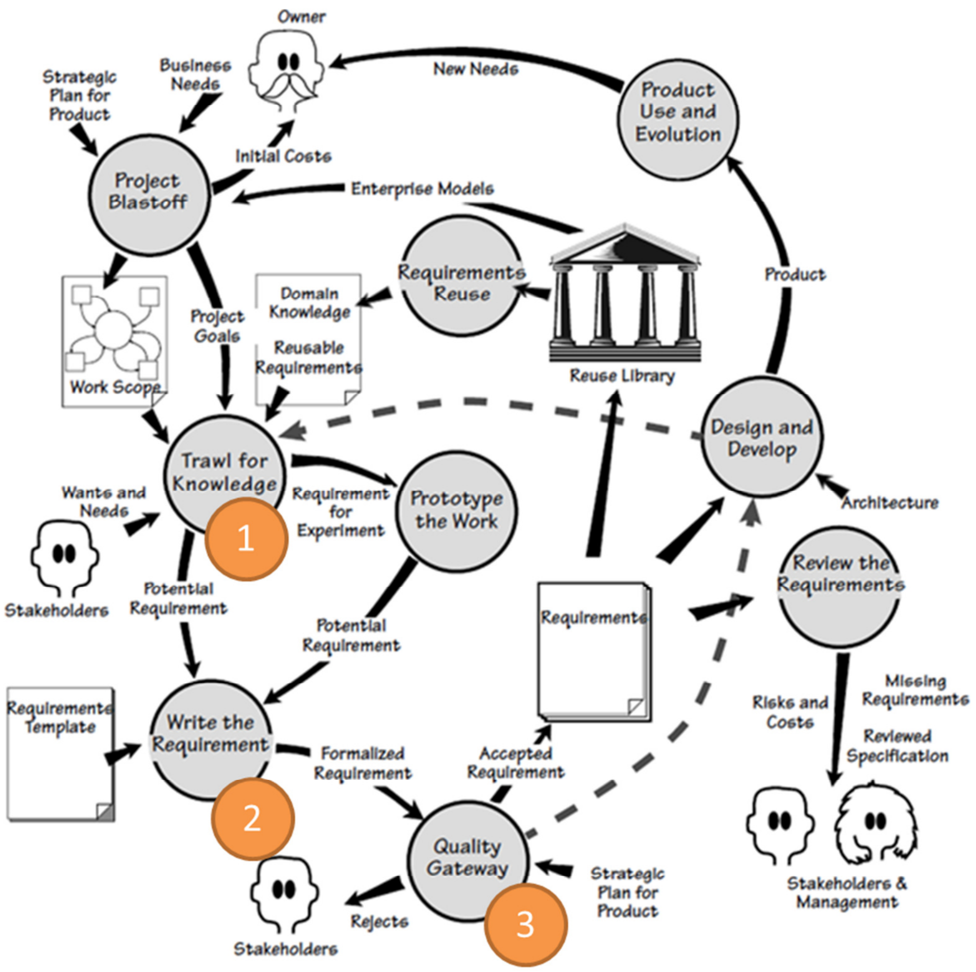


Figure 12 Model of the Pre-RS process by Robertson and Robertson (1999)

Pohl's (1993) research differs from the two previous models as it places emphasis on the transformation of requirements, rather than validation. Nevertheless, his work is no less significant in the field of requirement engineering. One of Pohl's most influential publications is an investigation (Pohl, 1993) that summarises contemporary requirement engineering research in order to deduce a pattern. This pattern was presented in the form of a requirement engineering framework. From this framework many research branches were spawn, such as (Ramesh and Jarke, 1999; Cleland-Huang, 2005; Nguyen and Swatman, 2003). The framework is known as the *Three Dimensions of Requirements Engineering*. A key concept in this framework is that of requirement evolution. The term *requirements evolution* is used in the rest of this thesis to describe the situation when an existing requirement is updated and refined. In Pohl's framework, requirements evolve from: 1) opaque (i.e. vague and fuzzy) requirements at the beginning to a complete (i.e. comprehensive) system specification; 2) ambiguous informal requirements into unambiguous formal requirements; and 3) potentially contradictory and stakeholder-biased requirements to a commonly agreed set of requirements. The framework is shown diagrammatically in Figure 13.

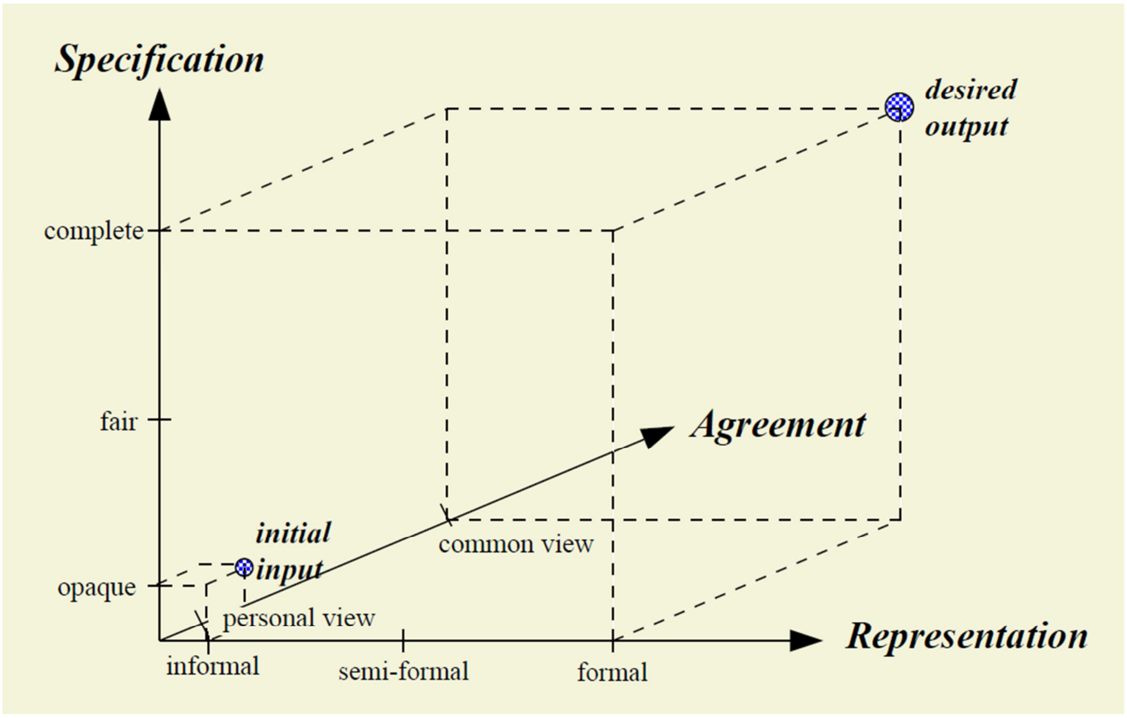


Figure 13 Diagrammatical view of the three dimensions of the Requirement Engineering framework

Post-Requirement Specification (RS)

Post-RS refers to activities occurring after requirement specification; it is the domain of requirements management. Paetsch (2003) views requirements management as a separate activity to requirement elicitation and analysis. Wiegers (2000) further expands on the notion of requirements management as including all activities concerned with *change control*, *version control*, *requirements traceability maintenance*, and r*equirements status tracking* (Wiegers, 1999b). In other words, requirement management is seen as consisting of all the activities related to maintaining a set of requirements and to ensure that requirement metadata is up-to-date.

2.3 A process model of requirement analysis

The previous Section reviewed various engineering processes that are supersets of the requirement analysis process. However, there exists limited number of literatures to define solely the requirement analysis process. This Section consolidates a definition of the requirement analysis process from publications in literature. The consolidation involved comparing models of the three engineering processes as well as further literature that justify the definition. Three elements were identified that fully describe the requirement analysis process: 1) checking – performing checks on requirements; 2) structuring – structuring requirements; and 3) evolution – evolving requirements as a result of the checks and structuring. The relationships between these three elements and the three engineering processes are shown in Table 6.

Table 6 The presence of checking, structuring, and evolution in engineering processes

	Design engineering	Systems engineering	Requirement engineering
Checking		✓	✓
Structuring		✓	
Evolution	✓	✓	✓

In the next Sections, the three main elements of the requirement analysis process are described in detail.

2.3.1 Checking

Requirement checking, also referred to as requirement validation (Rzepka, 1989), is a process to discover problems (Rzepka, 1989) and answer the question "Have we got the right requirements?" (Kotonya and Sommerville, 1998). Requirement checking is found in both the systems engineering process and requirement engineering process models. In the systems engineering Vee-model, requirement validation is performed at every layer. The validation checks whether requirements are correctly interpreted from the layer above. In requirement engineering process models, a fundamental part of Kotonya and Sommerville's model is a series of checks, including checking for necessity, completeness, consistency, and feasibility; also, in Robertson and Robertson's model these checks are generalised as the "Quality Gateway".

Alexander and Stevens (2002) distinguish between checking a single requirement and checking requirements as a set. Individual requirements are checked for clarity, necessity, and feasibility. The need for clarity was highlighted in a study on requirements management in the automotive industry (Almefelt, 2003). The most significant problem found relates to the interpretation of requirement specifications, which are often unclear leading to misunderstandings. Andersson et al.'s (2003) explanation for the lack of clarity typically found in requirements points to requirements not providing adequate information about the context and underlying intent and rationale. Properties of clarity include singularity, completeness, context, comprehensibility, concision, precision, tolerance, correctness, non-ambiguity, and verifiability (INCOSE, 2011). A set of requirements is checked for completeness (DAU, 1991; Kotonya and Sommerville, 1998), uniqueness (Heumesser et al. 2004), redundancy (Olivier, 2009) and freedom from conflicts (ISO-15288, 2008). The need to check for these properties is summarised by Kotonya and Sommerville (1998). They argue that stakeholders should gather to start a formal meeting with the aim of reviewing requirements. Requirements should be reviewed to find those that do not appear to be implementable with the technology available, generate significant conflicts, miss information, or are simply badly expressed. ISO-15288 (2008) – a requirements standard for systems engineering – confirms that after analysis the requirements should be achievable and conflict free (ISO-15288, 2008). Kotonya and Sommerville (1998) recommends to resolve any issues arisen during checking by discussion and agreement amongst stakeholders. Similarly, Pohl (1993) suggested that the way to make stakeholders "end up on a common agreement on the final specification" is through communication, conversation, coordination, collaboration, and better decision support. Defining a set of requirements that is agreed by all stakeholders is a complex task and a *wicked problem* (Conklin, 2005b). With such problems agreement can be reached through "an exchange of arguments among stakeholders in which they bring their personal expertise and perspective to the resolution of design issues" (Kunz and Rittel, 1970).

2.3.2 Structuring

In addition to checking, Hull et al. (2010) believe that requirement analysis is about defining a structure within the set of requirements. Structuring includes: 1) differentiating requirements according to their types; 2) defining a hierarchy of requirements; 3) decomposing requirements with reference to the hierarchy; and 4) allocating decomposed requirements to the hierarchy. In the systems engineering Vee-model, structuring is applied in the form of requirement decomposition. Decomposition is the process of making requirements more specific at sub-system level.

Andersson et al. (2003) explicitly specify the need for hierarchical levels so that satisfaction of requirements can be visualised on an enterprise level as well as product level. Stoller (1988) also recognises the need for a requirement specification to be "partitioned according to levels including sponsor, project, system, subsystem, assembly, etc". Buede (2009) states that "just as there is a hierarchy associated with the physical components of the system, there needs to be a hierarchy of requirements". He believes that the hierarchy should have mission requirements at the first level – the most abstract point; stakeholder requirements at the second level; and detailed engineering requirement statements at the third level. A similar view is shared by Grady (1993), who advocates that decomposition is central to the process of requirement analysis that transforms customer needs into system requirements.

2.3.3 Evolution

The overall effect of requirement analysis is to transform requirements from informal *stakeholder requirements* to formal *system specifications* (ISO-15288, 2008; Gotel and Finkelstein, 1994; Ramesh and Jarke, 1999; Jarke and Informatik, 1992). Easterbrook and Nuseibeh describe it as "a process to recognise change through continued requirement elicitation, re-evaluation of risk, and evaluation of systems in their operational environment" (Easterbrook and Nuseibeh, 1995). In design engineering, all the process models imply the occurrence of requirements evolution. Figure 14 shows how each design engineering model describes the transformation from stakeholder requirements to systems requirements. In the systems engineering model, user requirements are evolved into system-level requirements, sub-system level requirements, and component-level requirements. Pohl's model in requirements engineering is characterised by requirement evolution in three dimensions: 1) evolution to a complete system specification; 2) evolution to unambiguous formal requirements; and 3) evolution to a commonly agreed set of requirements.

French	Need → Statement of problem
Pugh	Market → Specification
Pahl & Beitz	Task → Requirement list
Hales	Task → Specification

Figure 14 Requirement evolution in design engineering processes

Requirement evolution occurs as a result of the requirements being modified to accommodate corrections, environmental changes, or new objectives (Lamsweerde, 2000). Essentially it is a process of transformations (DAU, 1991). Such transformations can be thought of as moving requirements across a number of states in which needs become specifications by means of checking and agreement as well as by addition of new requirements, and revision and rejection of existing ones. Operations needed to transform requirements are those related to structuring as well as refinement. Refinement refers to changes made to requirements within the same level of abstraction. Capturing requirement evolution touches on the issue of requirement traceability, which is defined as "the ability to follow the life of a requirement, in both forwards and backwards direction, i.e. from its origins, through its development and specification, to its subsequent use" (Gotel and Finkelstein, 1994). To support traceability, references between requirements across separate views can be maintained as a way to navigate them. Overall, the evolution dimension seems to include aspects of checking and agreement, as well as aspects of structuring. As user needs are checked and agreed upon, the re-expressed needs become less personal and more formal in order to make them universally understandable. Also, as user needs are decomposed, they become less abstract moving from needs and desires towards implementation restrictions. Therefore, evolution can be interpreted as the recording of all the necessary information during the transition from informal to formal expressions of requirements.

Requirement evolution can be generalised into three types of transformations (Finkelstein, 1991): 1) ***one-to-one*** – (also known as requirement update or refine (Ramesh and Jarke, 1999)) is a predecessor-successor relationship described in the IEEE 610.12-1990 standard; 2) ***one-to-***

many – (also known as decomposition or allocation (Ramesh and Jarke, 1999)) is a master-subordinate relationship described in the IEEE 610.12-1990 standard; and 3) ***many-to-one*** – is a rare circumstance in which several requirements merge to become one requirement.

Requirement evolution can be distinguished as two types: ***within-document evolution*** and ***cross-document evolution***. Within document evolution is when a requirement evolution takes place within the same document. Enabling traceability in this scenario is often known as version control. Cross document evolution is when a requirement is transferred from one document to another. During the course of the transfer, the same requirement at the source and destination may be described differently. Enabling traceability in this scenario is often known as linking.

One of the biggest challenges in managing large, complex systems is due to the way in which requirements are constantly evolving and changing (Ramesh and Jarke, 1999). In focus group workshops conducted by Ramesh and Jarke (1999) they noted that in order to accurately reflect this volatility a requirement traceability scheme should be able to help document and understand the evolution of requirements. Ramesh (1999) concludes that requirements evolve as a result of either stakeholders changing their needs or new needs obtained in operation or testing.

The three elements checking, structuring, and evolution that make up requirement analysis are illustrated graphically in the model in Figure 15. Requirement analysis is the process of requirement evolution from stakeholder requirements to system requirements. Requirement evolution is a result of refining requirements through checking and structuring.

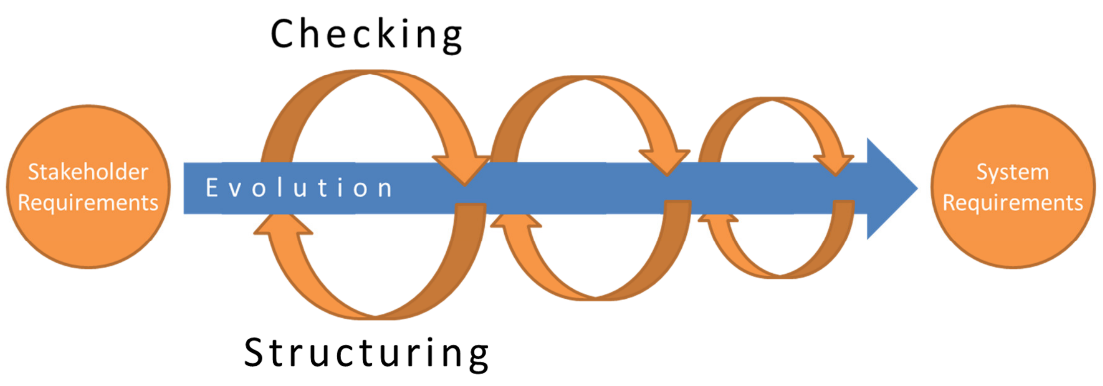


Figure 15 Graphical representation of the synthesised requirement analysis process model

2.4 Requirement analysis support tools

This Section provides an implementation-perspective of requirement analysis; it defines the boundaries of current support tools for requirement analysis to understand how this research can extend them. The term “support tool” shall be used hereafter to refer to a package that consists of one (or a combination of) method of analysis, format of requirement capture, and support software. It is not practical to present all support tools in existence, therefore the tools selected for presentation are either (or both) they are the most widely used tools or because they appear in the course of this research investigation. The tools are considered to support requirement analysis if they support any of the three aspects outlined in Section 2.3, i.e. requirement checking, structuring, or recording evolutions. Checking support is the ability of a tool to make it easy for requirement engineers to check individual requirements for clarity, necessity, and feasibility; or to check a set of requirements for completeness, uniqueness, redundancy and freedom from conflicts. Structuring support is the ability of a tool to show requirement categorisation, or to show requirements in a hierarchy. Recording evolution support is the ability of a tool to show the process of requirement evolution, i.e. recording a history of changes between requirement versions.

The tools that support requirement analysis can be categorised by structure into three types: hierarchy-based, diagram-based, and table-based. The hierarchy-based structure is characterised by a set of requirements organised in a tree consisting of parent-children relationships. The diagram-based format is characterised by requirements organised in a network structure consisting of nodes and arcs. The table-based format is characterised by a set of requirements organised in a tabular format.

2.4.1 Hierarchy-based tools

Affinity diagram (Tague, 2005) is a hierarchy-based tool that support checking and structuring aspects of the requirement analysis process. It is a widely-used tool that is often used as a precursor to more formal recording of requirements, particularly before constructing a Quality Function Deployment table (CIRI, 2011; Crow, 2011). Affinity diagramming facilitates re-arrangement of requirement snippets so that requirements can be grouped, checked, and refined. The grouping helps requirement engineers to surface the deep structure in stakeholder requirements (Mazur, 1993). Affinity diagramming tool is a method-based tool that can work on any software supporting text rearrangement. Figure 16 shows an affinity diagram of requirements for the design of a washing machine. Note that the requirements are grouped by themes such as environmental impact, performance, looks, and ease of use.

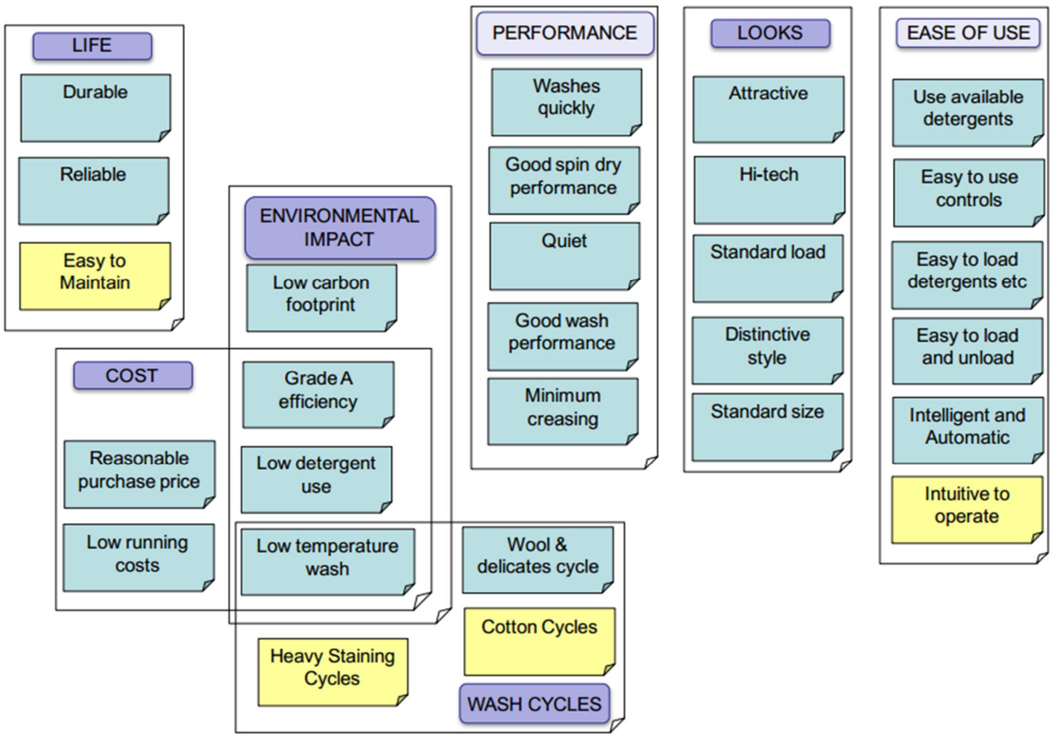


Figure 16 Affinity diagram to organise requirements for a washing machine (Burge, 2011a)

Viewpoint Analysis (VPA) (Burge, 2011a) is a tool that consists of a method and a format. It is facilitates requirement structuring by representing a set of requirements as a hierarchy. As well as supporting structuring, VPA also facilitates checking for requirement completeness. VPA is a relatively new tool therefore literature on its use is limited. However its use has been observed in several projects documented in this dissertation. Creating a VPA chart has three steps. First, requirements are classified into the types external functional, internal functional, and non-functional. Second, the functional requirements are structured as a tree. The tree has a top-level consisting of operational requirements and all subsequent levels contain functional requirements. Third, non-functional requirements (appearing as ‘bubbles’) are attached to corresponding functional requirements on the tree. A VPA chart can be created using any software capable of showing a hierarchical tree of requirements. The recommended tool is Microsoft PowerPoint or Visio. VPA can be combined with affinity diagramming. Affinity diagram can be used for grouping requirements, and VPA used for representing the requirements as a hierarchical tree. Figure 17 shows a set of requirements for a washing machine in the structure of a VPA chart. The requirements are developed from Figure 16.

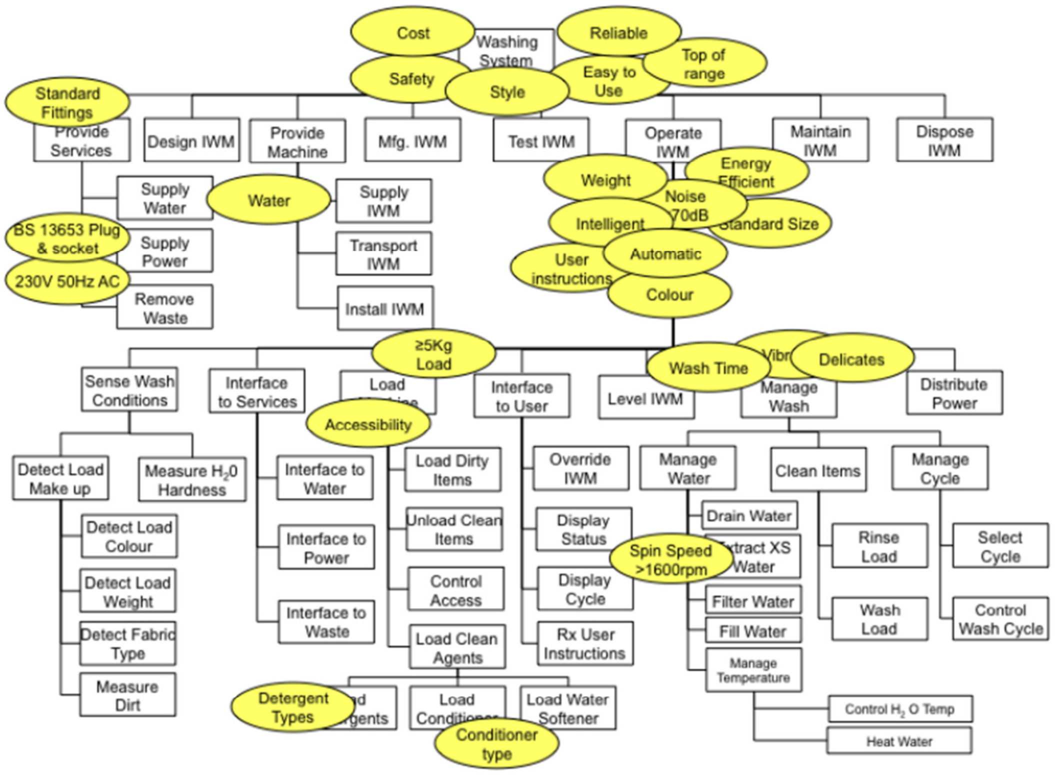
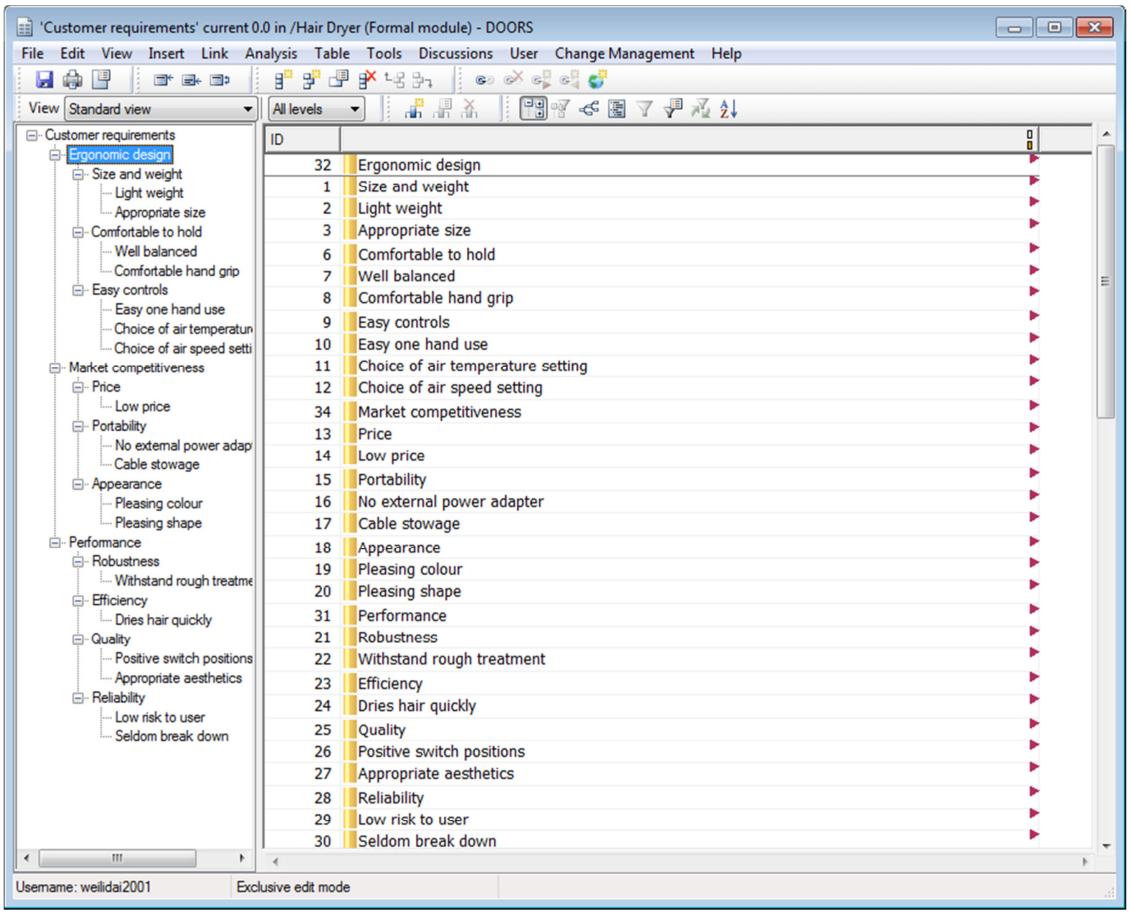


Figure 17 Viewpoint analysis for a washing machine (Burge, 2011b)

IBM DOORS (IBM, 2011) is a tool that consists of a software program. It is used to provide version control and traceability to a set of requirements (Wiegers, 1999a). Therefore DOORS supports recording requirement evolutions. IBM claims DOORS has had a dominant market share of requirement engineering software tools for over a decade (IBM, 2009). For example, it has been used at Airbus in various projects since 2003 (Kossmann et al. 2009). A screenshot of requirements captured in DOORS is shown in Figure 18. The screenshot is a default view consisting of a vertical split-screen. On the left there is a tree-view of requirements. And on the right there is list of requirements which is the requirement tree on the left flattened. DOORS is usually used after stakeholder needs have been elicited and analysed into systems specifications. DOORS is designed to support multi-user access. It implements a client-server model which allows requirement engineers to input and update requirements from their client machines by connecting to a centralised DOORS database. DOORS is also designed to scale for large requirement sets (typically in the magnitude of thousands). It stores requirements in a database instead of a file and has sophisticated management logic. One feature of DOORS that is relevant to this research is its ability to automatically track changes to requirements. This is in effect a means to automatically creating requirement traceability. However, tracking changes is only limited to when the requirement performs a one-to-one evolution, i.e. an update. If a requirement is decomposed into several requirements, extra operations will be required to make the decomposition traceable.


'Customer requirements' current 0.0 in /Hair Dryer (Formal module) - DOORS

- Customer requirements
  - Ergonomic design
    - Size and weight
      - Light weight
      - Appropriate size
    - Comfortable to hold
      - Well balanced
      - Comfortable hand grip
    - Easy controls
      - Easy one hand use
      - Choice of air temperatur
      - Choice of air speed setti
  - Market competitiveness
    - Price
      - Low price
    - Portability
      - No external power adap
      - Cable stowage
    - Appearance
      - Pleasing colour
      - Pleasing shape
  - Performance
    - Robustness
      - Withstand rough treatme
    - Efficiency
      - Dries hair quickly
    - Quality
      - Positive switch positions
      - Appropriate aesthetics
    - Reliability
      - Low risk to user
      - Seldom break down

ID	
32	Ergonomic design
1	Size and weight
2	Light weight
3	Appropriate size
6	Comfortable to hold
7	Well balanced
8	Comfortable hand grip
9	Easy controls
10	Easy one hand use
11	Choice of air temperature setting
12	Choice of air speed setting
34	Market competitiveness
13	Price
14	Low price
15	Portability
16	No external power adapter
17	Cable stowage
18	Appearance
19	Pleasing colour
20	Pleasing shape
31	Performance
21	Robustness
22	Withstand rough treatment
23	Efficiency
24	Dries hair quickly
25	Quality
26	Positive switch positions
27	Appropriate aesthetics
28	Reliability
29	Low risk to user
30	Seldom break down

Username: weilidai2001 Exclusive edit mode

Figure 18 Requirements for a hair dryer captured in IBM DOORS

The Issue-Based Information System (IBIS) (Kunz and Rittel, 1970) is a tool consisting of a method and a format of four notations. It can be used to support requirement checking. This tool is not yet a widely-used system for requirement analysis. One example of using IBIS for requirement analysis is by Selvin et al. (2001) who showed that it is feasible to use the IBIS-based Compendium tool to record stakeholder discussions on requirements. IBIS promotes the capture of requirement rationales. The action of defining requirement rationale has the effect of checking individual requirements for clarity, necessity, and feasibility. Andersson et al. (2003) argue that rationale explains why stakeholders "have made certain decisions and what information they took into account when making them". In (Dai and Aurisicchio, 2012) an approach was introduced to map non-functional requirements with a tree structure using the IBIS notation, which put emphasis on the capture of design rationale to justify requirements. Rooksby et al. (2006) proposed an approach that uses a combination of cognitive mapping and IBIS to explicitly map and record conflicting requirements; the research concluded that their approach is better than no pre-planned approach, which is often the case. The IBIS method uses a limited set of notations to represent questions, ideas, pros and cons, as shown in Figure 19. Common software tools that can be used to support the IBIS notation are designVUE (2014), Compendium (Selvin et al. 2001), gIBIS (Buckingham-Shum et al. 2006), and DRed (Bracewell et al., 2009a).

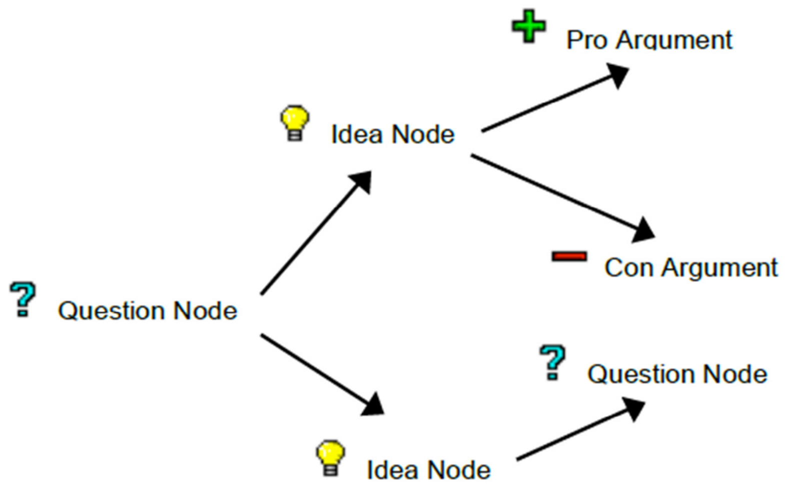


Figure 19 IBIS notation illustrated using the gIBIS tool (Conklin, 2005a)

2.4.2 Diagram-based tools

The Function Flow Diagram (FFD) is a diagram-based tool consisting of a distinctive format. It is used in requirement analysis to support requirement checking. FFD can be used to check individual functional requirements for necessity and feasibility, and to check a set of requirements for completeness (Robertson and Robertson, 1999). Hull et al. suggest that the use of modelling tools “introduces a degree of formality into the way systems are defined” and provides validation for the decisions contained in system requirements (Hull et al., 2010). The vocabulary of FFD is limited to only three types: function, terminator, and flow, see Figure 20. It portrays the system to be designed in terms of its component functions and the logical interdependencies or "flows" between the functions (Burge, 2011c). FFD can be created using most diagramming software that support creation of blocks and connectors, such as Microsoft Visio or PowerPoint.

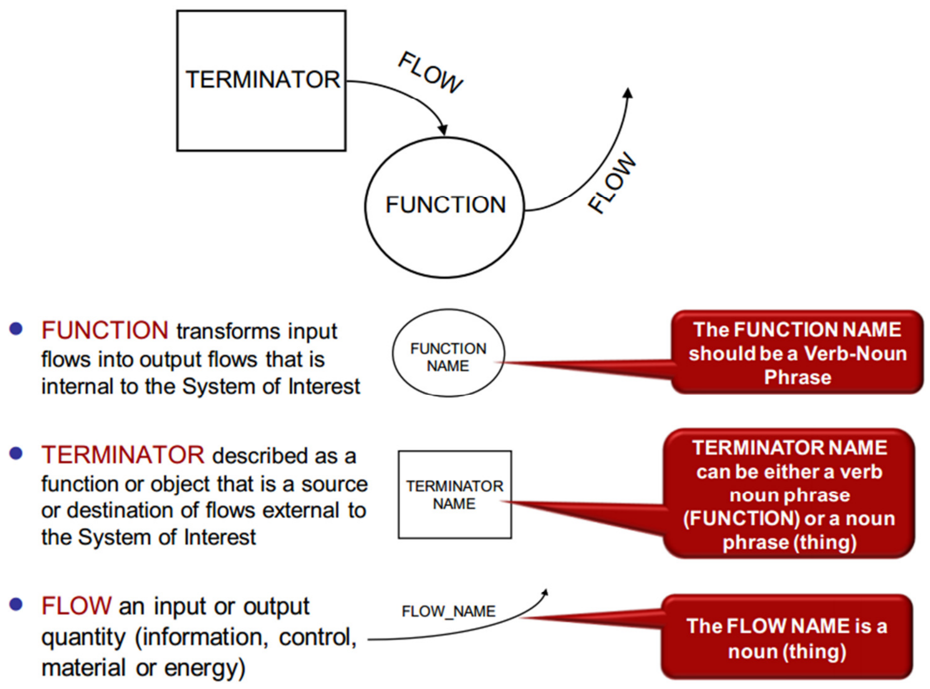

TERMINATOR
FLOW
FUNCTION
FLOW
FUNCTION transforms input flows into output flows that is internal to the System of Interest
FUNCTION NAME
The FUNCTION NAME should be a Verb-Noun Phrase
TERMINATOR described as a function or object that is a source or destination of flows external to the System of Interest
TERMINATOR NAME
TERMINATOR NAME can be either a verb noun phrase (FUNCTION) or a noun phrase (thing)
FLOW an input or output quantity (information, control, material or energy)
FLOW_NAME
The FLOW NAME is a noun (thing)

Figure 20 Basic building blocks of a Function Flow Diagram (FFD) (Burge, 2011c)

Systems Modelling Language (SysML) (SysML.org, 2006) is a diagram-based modelling language. SysML indirectly facilitates requirement checking when a system to be designed is modelled. SysML is relatively new, having been just introduced in 2006 and is still being finalised in 2012 (OMG, 2012). However, it has received a lot of attention and support from industry (Friedenthal et al., 2009). SysML is a UML derived graphical modelling language used in systems engineering. It contains nine diagram types that can model structure, behaviour, and requirements of a system. SysML was introduced as a universal representation for requirements and design so that design solutions can be traced back to requirements. Requirements are usually linked to design using allocation tables (Hause and Francis, 2008). However, the use of SysML in engineering projects is far less common than the use of UML in software projects (Hause and Francis, 2008). When SysML is used to model requirements, these are captured as blocks. A requirement block has a text-based requirement, an id, and any user defined properties such as verification (Friedenthal et al., 2011). Requirements blocks can relate to other requirements through relationships including DeriveReqt, Satisfy, Verify, Refine, Trace, and Copy (Hove et al, 2008; Friedenthal et al., 2008). Requirement blocks can also be justified using rationale blocks. The justification can be attached to a requirement block, or to a link block that describes a relationship between two nodes. Rationale has a supportive role in SysML as commented in (INCOSE 2012). An example of SysML requirement diagram is shown in Figure 21.

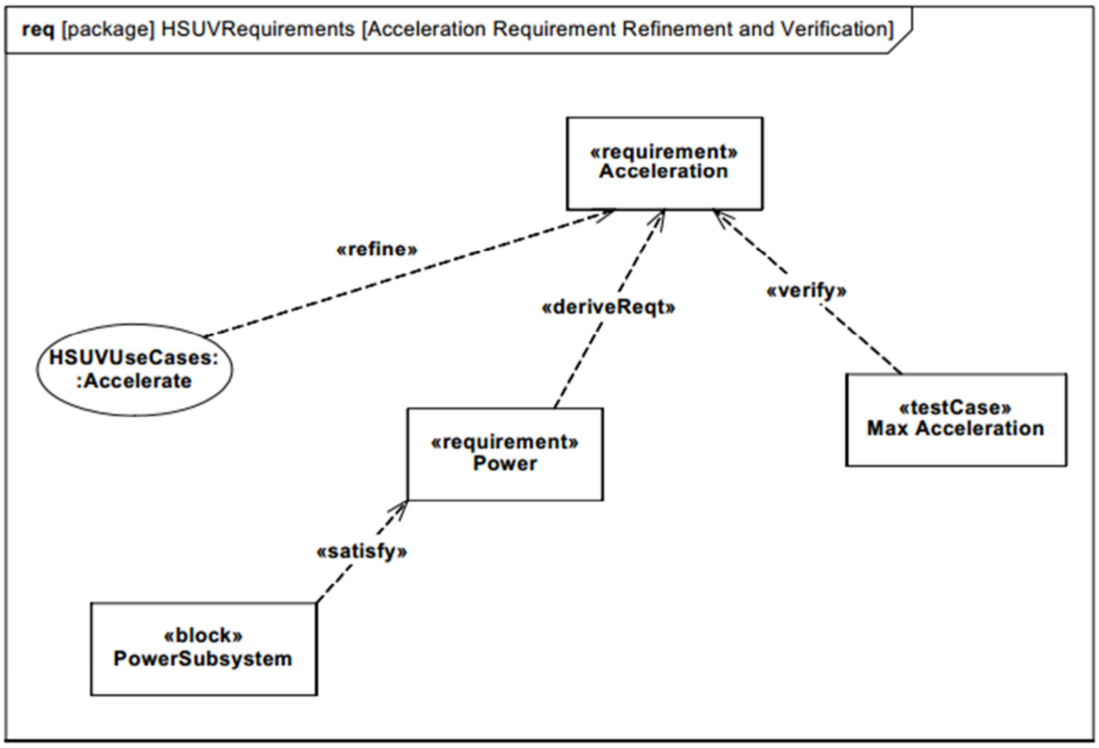
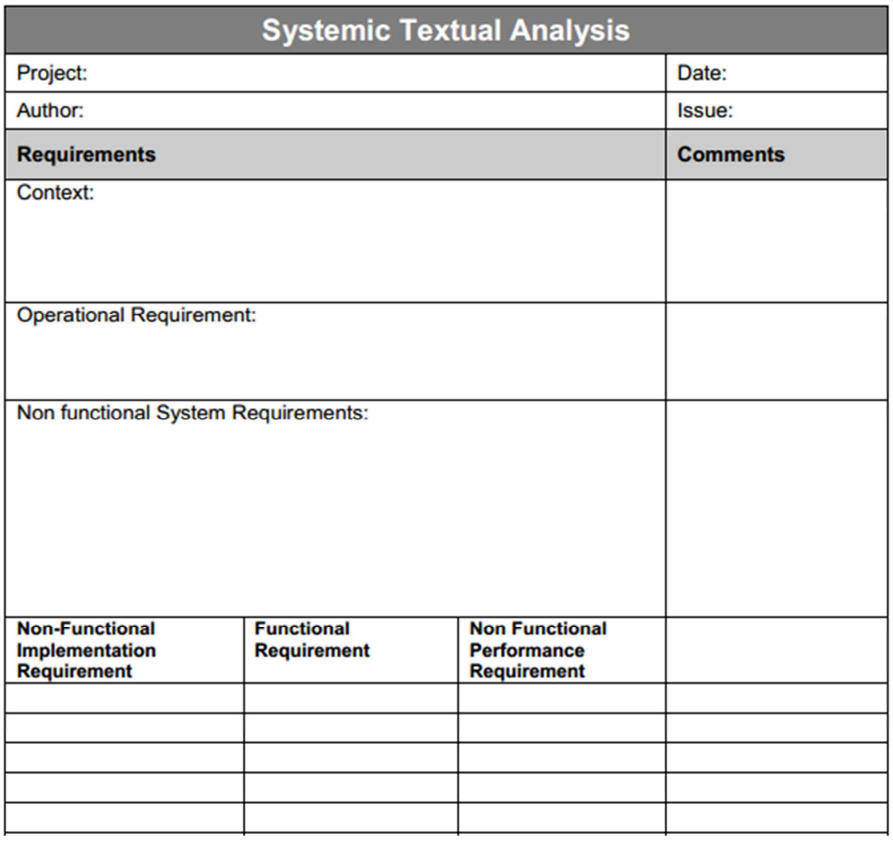

req [package] HSUVRequirements [Acceleration Requirement Refinement and Verification]
«requirement» Acceleration
«refine»
«deriveReqt»
«verify»
HSUVUseCases: :Accelerate
«requirement» Power
«testCase» Max Acceleration
«satisfy»
«block» PowerSubsystem

Figure 21 An example of a SysML requirement diagram (Friedenthal et al., 2009)

2.4.3 Table-based tools

Systemic Textual Analysis (STA) (Burge, 2004) is a table-based tool that consists of a method and a table format. STA facilitates requirement checking and structuring. STA facilitates checking for missing requirements in a set through visual inspection of matching functional requirements to non-functional requirements. It also supports structuring by facilitating visual interpretation of requirement hierarchy relationships, and relationships between functional and non-functional requirements. STA is a tool that has been observed in use in several investigations in this thesis. It is used to allocate requirements to the five types defined in the Holistic Requirements Model (Burge, 2006). The process of categorising a raw set of requirements into correct types helps check requirements for conflicts and consistency. Also STA facilitates crosschecking of requirements to discover missing requirements. An example of STA is shown in Figure 22. Creating a STA table has three steps. First, existing requirements are given a type based on the five HRM types, namely operational requirement, non-functional system requirement, non-functional implementation requirement, functional requirement, non-functional performance requirement. Second, requirements are allocated to different areas of the STA table corresponding to their type. Third, a process of checking for missing requirements begins by obeying a pattern that for every non-functional implementation requirement there could be one or more functional requirement and non-functional performance requirement. STA can be created in any software that supports table creation and manipulation such as Microsoft Word and Excel.



Systemic Textual Analysis			
Project:			Date:
Author:			Issue:
Requirements			**Comments**
Context:			
Operational Requirement:			
Non functional System Requirements:			
Non-Functional Implementation Requirement	**Functional Requirement**	**Non Functional Performance Requirement**	

Figure 22 Systemic Textual Analysis table (Burge, 2004)

Quality Function Deployment (QFD) (Crow, 2011) is a table-based tool that has a unique format. QFD is a well-known tool in industry (Koski et al, 2003; Herzwurm and Schockert, 2006; Mazur, 1993) and its use has been observed in investigations in this dissertation. QFD supports requirement checking and structuring. The effect of QFD is similar to STA in that it checks a requirement set for conflicts and missing requirements (Robertson and Robertson, 1999), and structures requirements by allocating requirements according to types. Figure 23 shows a QFD table that organises requirements by type according to the Holistic Requirements Model (Burge, 2006). A QFD can be created using general software that supports table creation such as Microsoft Excel. There are also specialised software tools for QFD such as Qualica Planning Suite[2].

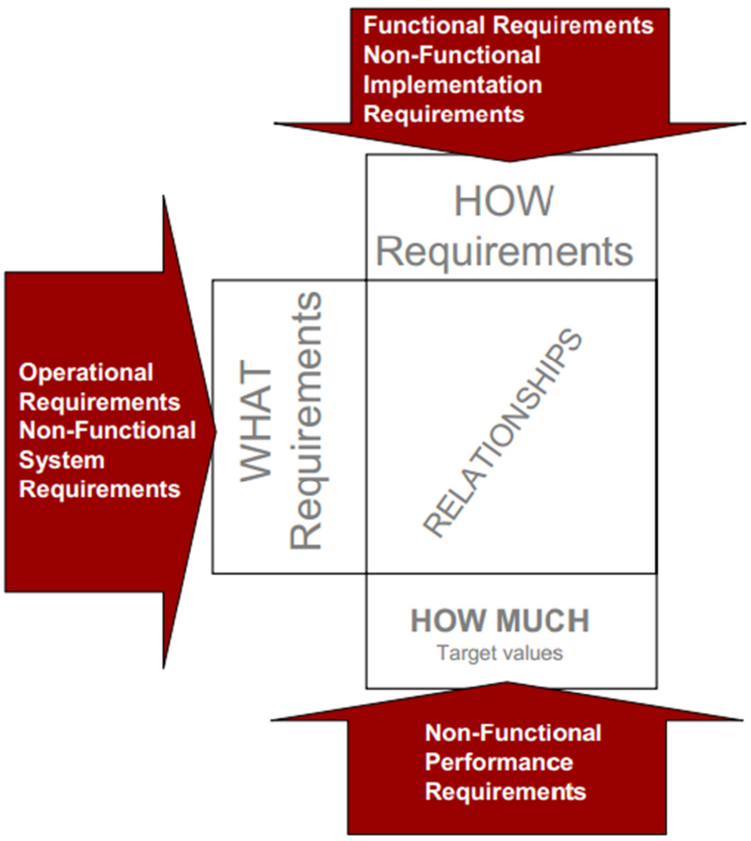


Figure 23 Layout of a typical QFD table (20]

2.5 Requirement rationale and traceability

The previous Section introduced some of the most popular support tools for requirement analysis. This Section addresses a research gap left by the tools, i.e. loss of contextual knowledge. An issue possessed by most of the tools is that they capture only a small proportion

[2] http://www.qualica.de/qps_qfd.html

of the knowledge available at the time of requirement analysis; typically only the results of decisions are captured. A consequence of this is that the intentions of the stakeholders are only partially transferred into requirements, reducing the interpretability of requirements. Tools such as affinity diagram, viewpoint analysis, systemic textual analysis, and QFD are transient tools. They are transient because they are designed to process, not to store, requirements. They take as an input a set of requirements, allow requirement engineers to apply their knowledge to check and structure the requirements, and output a set of evolved requirements. These tools only capture requirement statements, not additional requirement metadata such as requirement rationale. Also, most of these tools do not enable requirement traceability. Therefore requirements lose their contextual information once they are processed by these tools. Contextual information refers to how the requirement was arrived at and why was it necessary to modify (or keep) the requirement.

Some tools such as DOORS and SysML may initially appear to have addressed the issue of lack of requirement rationale and requirement traceability. These tools are designed to record requirement evolutions, by enabling i.e. requirement traceability. For example, the DOORS software has requirement version control built-in, and the SysML can indicate requirement evolutions through its <<refine>> and <<deriveRqt>> notations. They also support requirement rationale capture. DOORS allows requirements objects to store any type of metadata such as rationale. SysML has rationale nodes to store requirement rationales. However, these tools have been designed to *store* requirements and their metadata, not to process them. They are inadequate also because they only accept *validated* system requirements.

This research project addresses the issue of contextual knowledge loss by improving the support of requirement rationale capture and enabling requirement traceability. Requirement rationale can be either a statement focused on a single requirement or a statement about several requirements (Andersson et al., 2003). In the literature, requirement rationale is already considered as an important element that should be captured during requirement analysis (Liang et al, 2010; Rooksby et al. 2006; Selvin et al, 2001; Mead, 2008). Requirement traceability is defined as the ability to locate the requirement at any time within its lifecycle, both backwards and forwards (Gotel and Finkelstein, 1994). Therefore, requirement traceability is closely linked to requirement evolution which is defined as the transformations that requirements are subject to as a result of analysis.

The need to support requirement rationale and requirement traceability is important for at least two reasons: team collaboration and requirement quality. Requirement rationale and requirement traceability have an especially profound effect on team collaboration in large projects where the requirement list is contributed to by multiple people (Finkelstein and Fuks, 1989). Traceable requirements allow engineers new to a project to trace back to follow the reasoning (captured by rationale) that has taken place; therefore a team can easily continue the analysis on a set of requirements that another team has created. The scenario of accumulating requirements collaboratively is especially evident in geographically-spread workplaces. For example, the aerospace engineering company Airbus relies on co-operation with 2000 suppliers in 20 countries[3]. In such environments, creating requirement traceability allows teams to integrate their individually created requirement documents (Finkelstein, 1991). Requirement rationale and requirement traceability also affects the quality of requirement documentation in three ways. First, the presence of requirement rationale and traceability adds confidence to the requirement statement (Ramesh and Edwards, 1993). It shows that a requirement has an origin, and is evidence that a requirement has been considered before and validates that a requirement is necessary. Second, by imposing the condition that every requirement is traceable is a way to ensure that no existing requirements are overlooked and no new unverified requirements are introduced (Ramesh and Edwards, 1993). This is especially useful to ensure that stakeholder requirements are directly mapped to design specifications without loss. Third, requirement rationale and traceability can increase the interpretability of requirements (Jarke, 1998). Engineers are more likely to understand a requirement if they can see where the requirement originates from and the intentions behind the requirement. Therefore, requirements are less likely to become scrambled in translation (Kotonya and Sommerville, 1998) as requirements evolve.

2.6 Discussion

This Chapter reviewed the literature relevant to this research project. The main findings are discussed below.

[3] According to http://www.airbus.com/company/

The review of requirements in engineering showed that a requirement is defined by more than just a statement of text. A requirement can be of different types. And it can have many metadata that decorate the requirement statement to convey context. Out of all the types of contextual metadata, requirement rationale is deemed the most important because many other types of requirement metadata can be presented as requirement rationale.

The literature review also showed that requirement analysis is viewed as a crucial part of design engineering, systems engineering, and requirement engineering process models. In design engineering, requirement analysis is viewed as the process of clarifying stakeholder requirements into unambiguous system requirements. In systems engineering, requirement analysis is considered as the process of transforming stakeholder requirements into system requirements through a series of decomposition. In requirements engineering, requirement analysis is considered as a combination of activities including checking, validation, and requirement evolution.

Based on the review of design engineering, systems engineering, and requirement engineering, a model of requirement analysis was synthesised. This model is defined as having three parts: 1) checking, 2) structuring, and 3) evolution. The model will be referenced in the subsequent Chapters of this dissertation as the formal process of requirement analysis.

Finally, the review of the support available for requirement analysis showed that current tools lack two kinds of support: 1) requirement rationale support and 2) requirement traceability support. Current tool support can be grouped into two groups. One group consists of transient use-once tools that are designed to evolve requirements through checking and structuring, but do not support the capture of the reasoning behind requirement evolutions. Another group consists of requirement repository tools that do support requirement rationale capture and traceability, but are not suitable for handling pre-validated requirements.

2.7 Conclusion

There are five main outcomes from this Chapter. First, the term "requirement" has been defined. A requirement is not only a statement of text, but can be decorated with contextual metadata to convey additional information. Second, an understanding of the requirement analysis process has been developed. Requirement analysis is a sub-process of higher abstraction-level engineering processes. Requirement analysis takes the roles of checking and structuring stakeholder requirements so that they evolve into system requirements. Third, a model has been synthesised to describe the requirement analysis process as checking, structuring, and requirement evolution. Fourth, a gap in current requirement analysis support has been discovered. In particular, the need to support requirement rationale capture and enabling requirement traceability in requirement tools was identified.

The main conclusion is that requirement analysis could benefit from further investigation into support for requirement rationale and requirement traceability. The investigation so far in this Chapter could provide the basis to develop improved tools to support requirement rationale capture and enabling requirement traceability during requirement analysis.

Chapter 3 Research approach

Chapter outline

This Chapter is divided into two Sections. Section 1 defines a research framework that is used to structure the entire project. Section 2 describes the application of the research framework to the thesis, and elaborates on the methodologies of data validation.

3.1 DRM framework

The Design Research Methodology (DRM) by Blessing and Chakrabarti (2002) is a framework to guide design research. In general, a research framework can save time compared to customising an alternative research methodology, and a framework is more likely to be already proven in terms of efficiency, effectiveness, and rigor.

Under the DRM a typical design research project would consist of investigating an existing work process with the aim of making improvements to it. This research project follows this pattern. Existing processes for requirement analysis in the collaborating company are examined, and improvements are proposed and validated.

The DRM framework consists of four iterative stages, see Figure 24. The four stages are detailed below:

1) ***Research Clarification*** is to establish the existing process and define success criteria of research.
2) ***Descriptive Study I*** is to gain in-depth understanding of the existing process through first-hand experience and analysis of evidence, and identify factors that influence the formulated criteria.
3) ***Prescriptive Study*** – define the desired process through an increased understanding of the existing process (established in Descriptive I). To reach the desired process, improvements are proposed.
4) ***Descriptive Study II*** is to evaluate the proposed improvement.

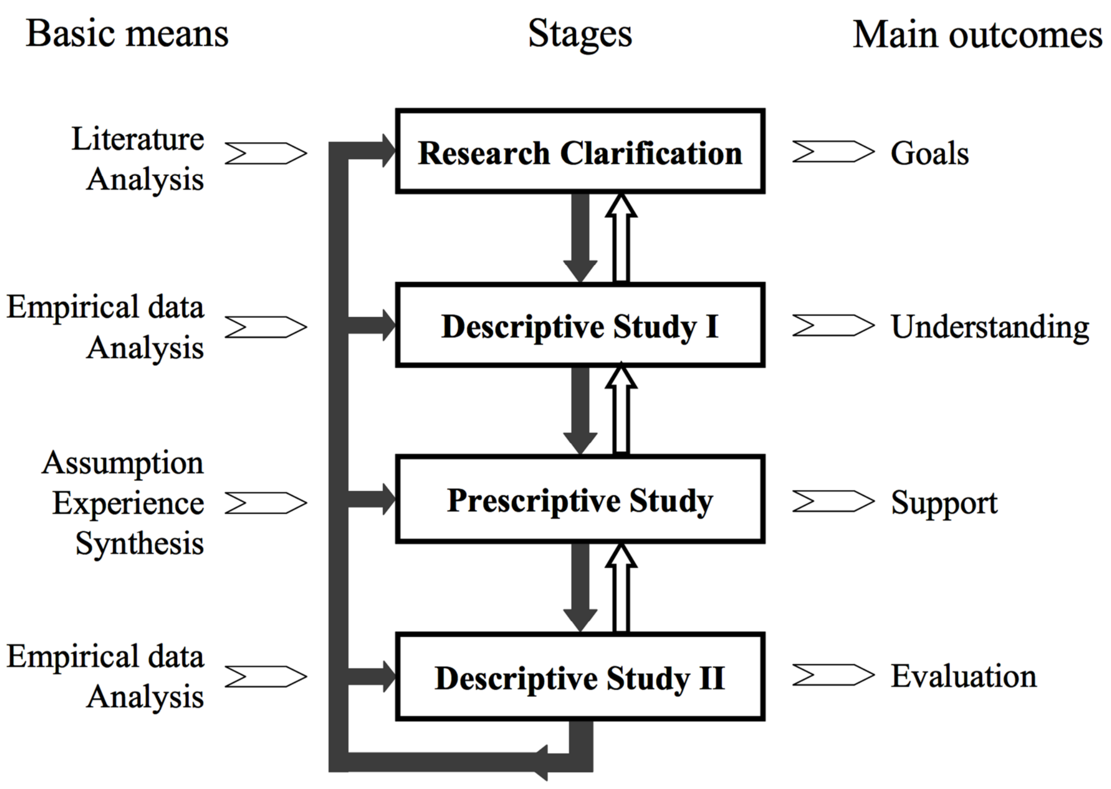


Figure 24 The DRM framework

3.2 The relation between the DRM framework and research studies

Figure 25 shows how the literature review (in Chapter 2) and the research studies (in Chapters 4, 5, 6, 7, and 8) undertaken in this project fit within the DRM framework. As can be seen the results of each study feed into the next one informing its development. In addition, the four stages of the DRM are not traversed linearly. Iterative traversal of the DRM is not uncommon. For example, Aurisicchio and Bracewell (2013) designed their research to iterate between the Prescriptive stage and the Descriptive II stage.

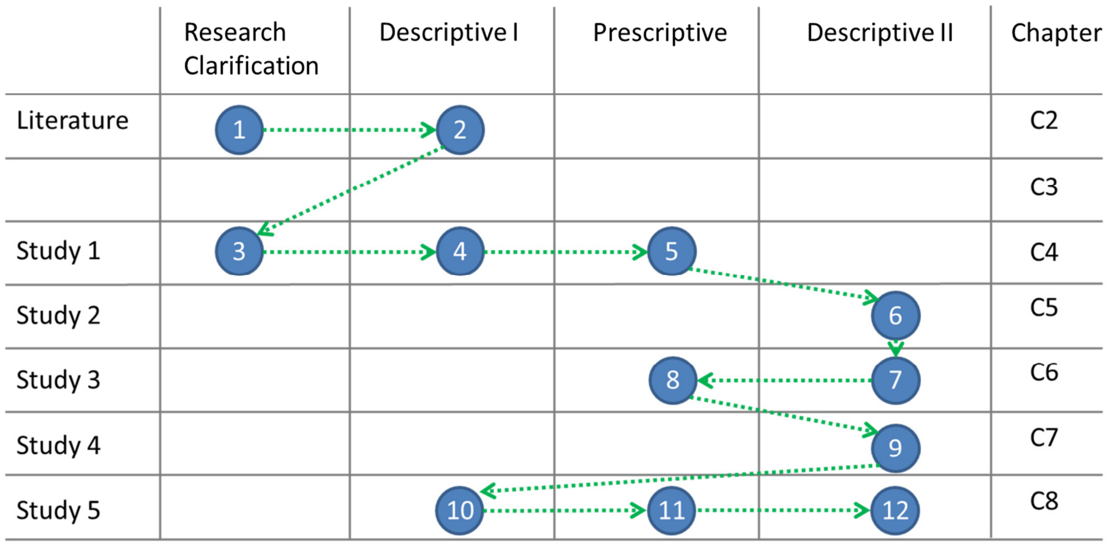


Figure 25 Relationship between the DRM model and thesis Chapters

The numerical labels in Figure 25 and their relations to the literature review and the research studies are described below:

The literature review was used to develop an understanding (Figure 25-2) of the requirement analysis process and identify a research gap that would become the success criteria (Figure 25-1). The research gap was identified as a lack of effective support for conducting requirement analysis. The success criteria were identified as to increase the quality of requirements through thorough requirement analysis and the specification of interpretable requirements (Figure 25-1). Two crucial factors identified as directly contributing to increased requirement quality are the availability of requirement rationale and the traceability of requirements.

Study 1 aimed at understanding current practice to support requirement analysis in the collaborating company (Figure 25-3, Figure 25-4). In this research project, current practice was reconstructed by shadowing engineers whilst they were analysing requirements, interviewing engineers after they had performed requirement analysis, and analysing requirement documents produced in engineering projects. A detailed description of data collection for Study 1 is documented in Section 4.1. On the basis of this work, a new requirement analysis workflow was proposed (Figure 25-5). The workflow proposes to use an IBIS-based tool to capture requirements and rationale as well as making traceable the captured requirements in all subsequent usages.

Studies 2, 3, 4, 5 validate the proposed workflow (Figure 25-6, Figure 25-7, Figure 25-9, Figure 25-12) using data sources that vary in terms of selected parameters; a detailed explanation of the validation process is given in Section 3.3. In addition to validation, Study 3 introduces an extension to the proposed workflow (Figure 25-8) that improves requirement traceability support. In Study 5, additional knowledge is gained about current requirement practice at the collaborating company (Figure 25-10), which has also led to modifications to the proposed workflow (Figure 25-11).

3.3 Research validation

The proposed improvement was validated in the subsequent studies. Three forms of research validation were sought, namely feasibility, practicality, and scalability. Feasibility was defined as a demonstration of the conceptual soundness of the workflow by testing it with an engineering design data set. Practicality was defined as a demonstration of the conceptual soundness and industrial relevance of the workflow by testing it with practicing engineers. A similar approach was used in (Aurisicchio and Bracewell, 2013), where the feasibility and practicality of a tool were tested using data from engineering projects in industry. Finally, scalability was defined as a demonstration of the conceptual soundness, industrial relevance and applicability to complex projects of the workflow by testing it on a large engineering programme in industry. Eres et al. (2014) conducted a similar research project on large engineering programmes in the aerospace domain to demonstrate scalability. In addition to these three forms of validation, the following parameters were identified to help design the studies: *environment* in which the study was produced, *expertise of data producer*, *method of data generation*, and *complexity*. The complexity parameter was further subdivided into *size of dataset*, *project duration*, project *team size*, and *solution complexity* in terms of number of components in a project.

The case studies and their controlled parameters are shown in Table 7. Only case studies 2 to 5 are listed in Table 7 because they are the studies that are used for the purpose of validation. Each case study was selectively chosen so that they cover a wide range of varying parameters. This would allow the proposed workflow to be validated under different contexts.

Table 7 Validation of the proposed workflow

Case studies / Controlled parameters	Study 2: Hair dryer	Study 3: Scoop sampling machine	Study 4: Cable routing and mould cleaning	Study 5: Whole engine
Type of validation	Feasibility	Feasibility	Feasibility Practicality	Feasibility Practicality Scalability
Environment	Academia	Industry	Industry	Industry
Expertise of data producer	Non-domain expertise	Low domain expertise	Low domain expertise	High domain expertise
Method of data generation	Reverse-engineering	Real-time / Reconstruction	Real-time	Real-time / Reconstruction
Complexity	Low	Medium	Medium	High
Size of dataset	Less than 50 requirements	Less than 100 requirements	Less than 100 requirements	Number of requirements in the order of thousands
Project duration	N/A	12 weeks	12 weeks	Around 10 years
Project team size	1	4	3	In the thousands
Solution complexity	Number of components less than 100	Number of components less than 100	Number of components less than 100	Number of components in the order of ten-thousands

Case study 2 validates the feasibility of the workflow. In this study a low-complexity engineering product (hair dryer) was reverse-engineered to extrapolate product requirements and requirement rationale. A detailed description of data collection for Study 2 is documented in Section 5.1. This case study validates the feasibility of both requirement rationale support and traceability support by the workflow. Requirement rationale capture was enabled by capturing requirements in an IBIS-format. Requirement traceability was enabled through the use of hyperlinking. Examples were shown to demonstrate the feasibility of creating traceability between requirements stored in different tool formats.

Case study 3 validates the feasibility of the proposed workflow. Data was obtained empirically from an engineering project. The project was undertaken by graduate engineers employed by the collaborating company. They worked on a low complexity engineering product (material sampling device). The data for this project was collected partly by using documents produced in real-time by the engineers, and partly through an interview. A detailed description of data collection for Study 3 is documented in Section 6.1. This case study shows the feasibility of requirement rationale capture and the potential to enable requirement traceability. Requirement rationales were obtained retrospectively; and it was shown that the workflow can process the obtained requirement and requirement rationale in order to generate a set of requirements that is richer in requirement rationale than the original set. This case study was also analysed to find opportunities to improve requirement traceability. The analysis let to a proposal to enable requirement traceability. It is worthy to note that the validation of practicality was not investigated in this case study as requirement analysis had already completed before the proposed workflow could be introduced.

Case study 4 validates both the feasibility and practicality of the proposed workflow. Data was obtained empirically from two engineering projects. The projects were both undertaken by graduate engineers employed by the collaborating company. They both worked on low complexity engineering solutions (cable routing and fan blade mould cleaning). The data for this project was captured in real-time by the engineers themselves. Practicality has been validated because the engineers have been given the proposed workflow to apply. A detailed description of data collection for Study 4 is documented in Section 7.1. This case study validates the feasibility and practicality of requirement rationale capture. The validation is demonstrated as engineering teams from both projects accepted and implemented requirement rationale capture according to the proposed workflow. This case study also showed high potential in enabling requirement traceability. One engineering team implemented traceability mechanisms used to enable requirement traceability, but not for requirements. The other engineering team was one step away from enabling traceability in their requirements as their requirements were aligned to satisfy all the pre-conditions needed to enable traceability.

Case study 5 validates the feasibility, practicality, and scalability of the proposed workflow. Data was obtained empirically from a whole-engine engineering project. The project is large-scale, and very complex. It spans several years and is developed by hundreds of engineers. Data collection for this project involved project document analysis, an interview to reconstruct the existing requirement analysis practice, a questionnaire to validate the improved workflow, and an interview to reconstruct requirement rationales. A detailed description of data collection for Study 5 is documented in Section 8.1. This case study validates the feasibility and practicality of the requirement rationale capture aspect of the proposed workflow. In this case study, requirement rationales were recreated through interviewing a project stakeholder, and were modelled in real-time according to the proposed workflow. This case study also validates the feasibility, practicality, and scalability of the requirement traceability aspect of the proposed workflow. It was demonstrated in this case study that the proposed workflow can be used to model the existing requirements in this large dataset to improve traceability in existing requirements.

Chapter 4 An investigation of requirement analysis practice in industry and proposal of a workflow

Chapter outline

This Chapter presents an investigation to understand current practices for requirement analysis in the collaborating company, identify areas of the practice for potential improvement, and propose a workflow that realises the potential for improvement.

4.1 Data collection and analysis

The data for this case study are based on: interviews, document analysis, and observations with shadowing. The interviewees were conducted with system engineers who oversee the implementation of best practices. The documents analysed include company best practices, presentation material used by system engineers to communicate company best practice, and past project documents. Shadowing involved observing engineers who apply the company best practices.

4.1.1 Interviews

A total of six interviews were conducted to understand requirement analysis practice. See Table 8 for the interview theme, questions discussed, interview duration, format of data capture, and participants. All the interviewees were systems engineers, but differed in project experience, speciality, and level of expertise, see Table 9. Interviews were intentionally arranged so that there is a wide diversity in the interviewees' area of expertise. The data from the interviews were analysed to:

- uncover reoccurring concepts – all the interviews have had their contents analysed for overlapping concepts; re-occurring concepts were treated as company best practice.
- uncover new sources of information – every interviewee was asked to recommend other people to contact, or suggest document resources, that would be useful for this investigation.
- endorse sources of information – the act of an interviewee making references to other sources of information was considered endorsing the source. Sources of information that have been referred to multiple times were considered more authoritative.

Table 8 An overview of interviews

Interview number	Theme	Questions discussed	Duration	Format of data capture	Interviewee
1	Company best practice in requirements engineering	What are the current best practices in the company? What tools are used for requirement analysis? How much training is given to engineers on requirement analysis?	1 hour	Notes	A
2	Company best practice on a large business project	What tools are used for requirement analysis? What is the workflow used for requirement analysis?	1 hour	Notes and presentation	B, C
3	A demonstration of DOORS – a company approved requirement tool	How does the DOORS fit within the company recommended workflow?	1 hour	Notes	B, C
4	Company best practice in requirements engineering	What are the current best practices in the company?	2 hours	Notes	D
5	Currently used practice on an engine project in development	What is the company recommended workflow for requirement analysis?	1 hour	Notes	D
6	Currently used practice on an engine project in conceptual design stage	What is the company recommended workflow for requirement analysis?	1 hour	Notes	E

Table 9 An overview of interviewees

	Job title	Experience	Gender
A	Chief of World Class Systems	Project managed large-scale projects for over 10 years and authored the requirement section of the company's engineering best practice guide	Male
B	Systems Engineer	Project managed several large-scale projects in nuclear power systems	Male
C	Senior Engineer/ Engineering Information Management Technical Lead	Project managed several large-scale projects in nuclear power systems	Female
D	Global Chief of Systems Engineering	Led systems engineers and defined process and tools capability into the whole engineering division in the company	Male
D	Project Systems Engineer/Whole Engine Design Engineer	Experienced in integrating diverse multi-functional systems on major engineering projects	Male
E	Project Systems Engineer	Led improvement in Systems Engineering capability within a department of approximately 150 people. Responsible for defining and executing the Systems Engineering strategy including requirements management	Male

4.1.2 Document analysis

Requirement analysis related documents from nine separate sources were collected, see Table 10. These documents describe either directly or indirectly the current practice at the collaborating company. The documents were analysed for clues that relate to the current company best practice in requirement analysis. These documents have been published either internally for company employees or externally by company representatives at conferences and other information exchange platforms. Document analysis was informed by the patterns emerging from the interviews and aimed at identifying re-occurring patterns such as multiple sources referring to the same requirement analysis technique.

Table 10 An overview of documents analysed

Title of document(s) examined (Format)	Purpose of document(s)
Generic Design Process (Booklet)	An internally published document that communicates company recommended tools to practicing engineers
Short course on Systems Engineering (Booklet of presentation slide printout)	A week-long seminar that teaches to practicing engineers processes and tools for systems engineering
Deployment of Requirements Management in [the collaborating company] (PowerPoint)	A presentation given at a systems engineering conference event to illustrate requirement management techniques used at the collaborating company
Implementing the Systems Engineering approach in [the collaborating company] (PowerPoint)	Presentations given at a systems engineering conference event to illustrate systems engineering practices being implemented at the collaborating company, including requirement workflows
Service Systems Engineering think piece input to kick-off for INCOSE UK (PowerPoint)	
Service Systems Engineering Working Group Wrap up (PowerPoint)	
Systems Engineering within the wider enterprise (PowerPoint)	
QFD An integrating Systems Engineering Tool (PowerPoint)	A presentation given at a systems engineering conference event to illustrate the use of QFD in the collaborating company
Documents from 12 separate projects (Electronic documents of various formats)	Past project documents relating to requirement analysis

4.1.3 Observations with shadowing

Practicing engineers were shadowed in order to gather first-hand knowledge about requirement analysis practice. The researcher observed graduate engineers who were carrying out 12-week long projects. Engineers from 3 projects were shadowed in parallel for a week. The week was chosen to coincide with the period when the engineers in each team were in the phase of eliciting and analysing requirements. During the shadowing, the researcher observed and took notes on the way engineers worked. In particular, the researcher was present when the engineers held requirement discussions, stakeholder interviews, and site visits.

Shadowing gave to the researcher the opportunity to observe how requirement analysis techniques are implemented in practice. Observations focused on identifying common practices, and identifying which references the shadowed engineers follow to perform requirement elicitation and analysis.

The following Section describes how the 3 project teams were shadowed:

Project team A
Project description: Design and build a test rig to emulate heating cycles of an aircraft engine fan blade.

Method of data collection: The researcher shadowed the team for four consecutive days, observed the workflow and tools that the team used to elicit and analyse requirements, and recorded observations in the form of notes. One event that was particularly interesting was when the project team met with their customer. The meeting was a session in which the project team discussed existing requirements. During the meeting, the customer checked existing requirements (such as requirement feasibility, importance, and scope) and gave justifications. New requirements were also elicited during the session when the customer was evaluating existing concepts. Figure 26 shows a snapshot taken during the meeting.

Figure 26 Project team A discussing proposed concepts with a customer to refine requirements and elicit requirement rationale

Project team B
Project description: Design and build a test rig consisting of a scaled-down turbine combustion chamber, so that a new technique to increase the efficiency of combustion can be tested.

Method of data collection: The researcher shadowed the team for four consecutive days, observed the workflow and the tools that the team used to elicit and analyse requirements, and recorded the observations in the form of notes. Figure 27 shows a snapshot of a tool that the project team used.

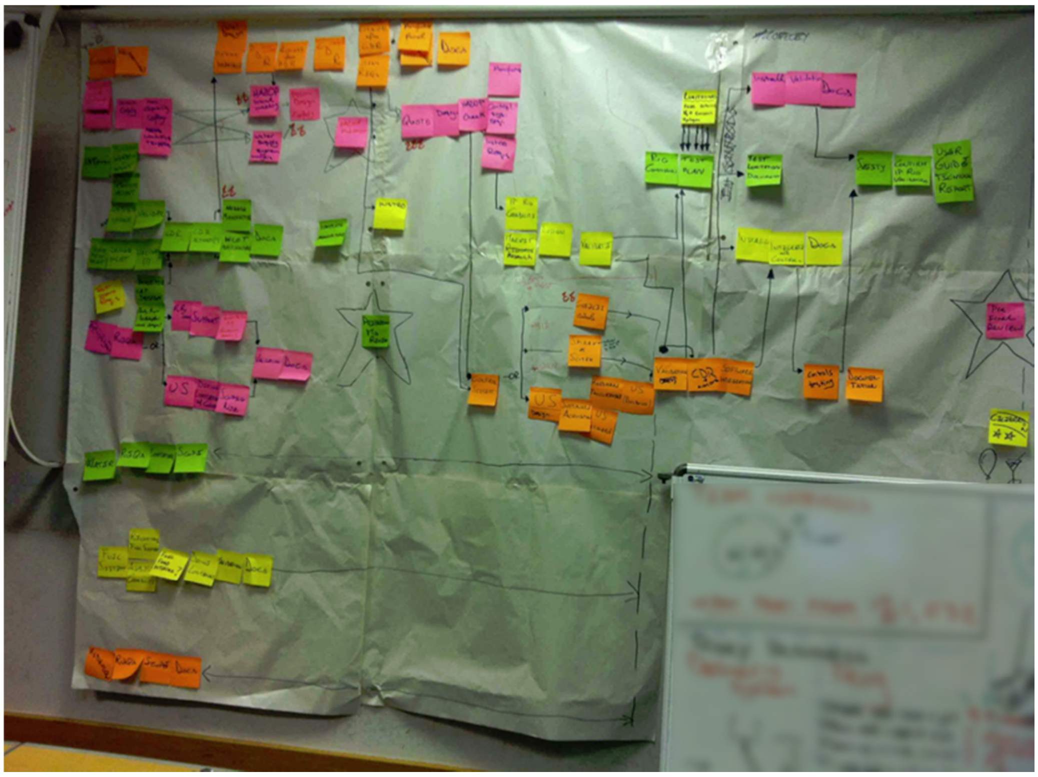


Figure 27 Project team B using Post-it notes on an activity diagram to calculate time and budget requirements

Project team C

Project description: Design and build a test rig that would produce a similar effect to reinforced sandpaper, to test the feasibility of a new technique to grind aircraft turbine fan blades whilst they are in-situ.

Method of data collection: The researcher shadowed the team for four consecutive days, observed the workflow and tools that the team used to elicit and analyse requirements, and recorded the observations in the form of notes. During the course of the shadowing, the researcher followed the project team on a site visit that helped the team elicit requirements, see Figure 28 and Figure 29.

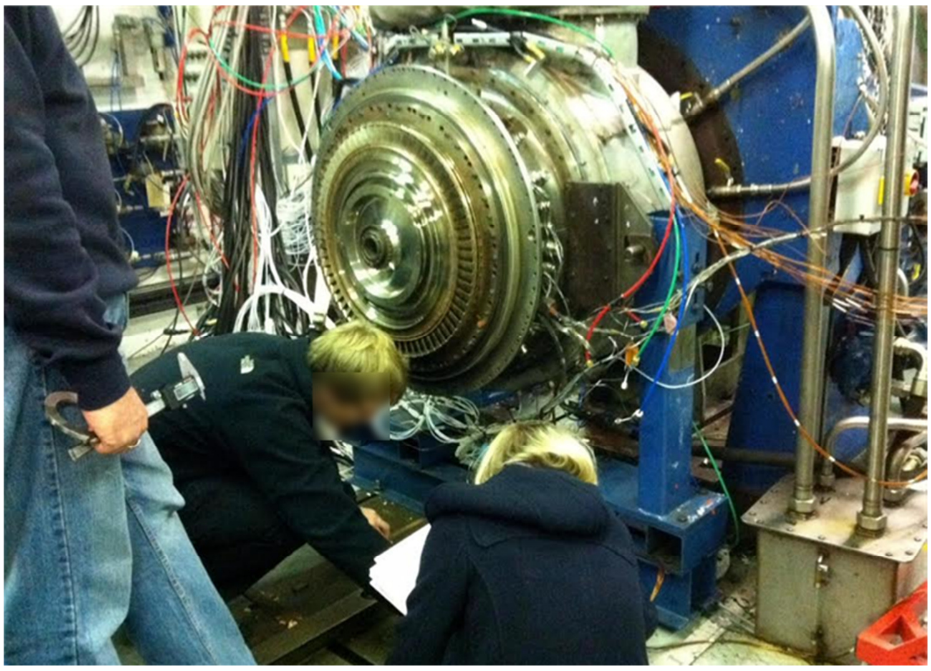


Figure 28 Project team C taking measurements of an existing rig to define interface requirements for their solution

Figure 29 Project team C consulting an expert to elicit requirements and understand rationale behind requirements

4.2 Results

The results are organised into two main Sections. The first focuses on processes for requirement analysis. The second focuses on tools. The processes refer to procedures that engineers follow. The tools refer to techniques which engineers can adopt as they follow the processes.

4.2.1 Processes

At the time of data collection, one process has been consistently referred to and that process is shown in Figure 30. The process consists of three steps: 1) capture requirements using software tools; 2) validate requirements; and 3) flow-down requirements to decompose high-level requirements. In *requirement capture*, requirements are assumed to be captured through various methods including interviews with airframe manufacturer and regulations authorities. *Requirement validation* involves checking the captured requirements so that conflicts are identified and traded. Validation can involve the use of methods such as Systemic Textual Analysis and Viewpoint Analysis as described in Section 2.4. *Flow-down* involves decomposing requirements. This process is further illustrated in Figure 31. It assumes that most requirements defined by the context of the project are high-level requirements. Hence, these are grouped at the environment-level. These requirements are aggregated into the Business Requirement Document (BRD). Once the BRD is formed, several iterations of flow-down occur to decompose high-level requirements into component requirements. The highest level is enterprise, followed by system, sub-system, and finally component. For each iteration there exists one definition document, and one or more evidence documents. A definition document captures design-decisions. Design decisions play a key role in requirement decomposition. Sometimes a requirement cannot be decomposed further unless a design decision is made. For example, when decomposing a requirement to provide thrust, requirements cannot be made about the thrust provider unless a design-decision is made to use a gas turbine as opposed to using an electric motor. A definition document can also include *evidence* of some form to justify the decision for the selection of the definition. For example, a gas turbine is able to produce the thrust needed to satisfy a top-level requirement, but an electric motor cannot. Evidence can be viewed as rationale; it is any contextual metadata that support design-decisions in the definition document. Evidence can include: design decomposition evidence; validation and verification methods evidence; validation and verification results summary; satisfaction evidence summary; test results or test reports; and allocation evidence.

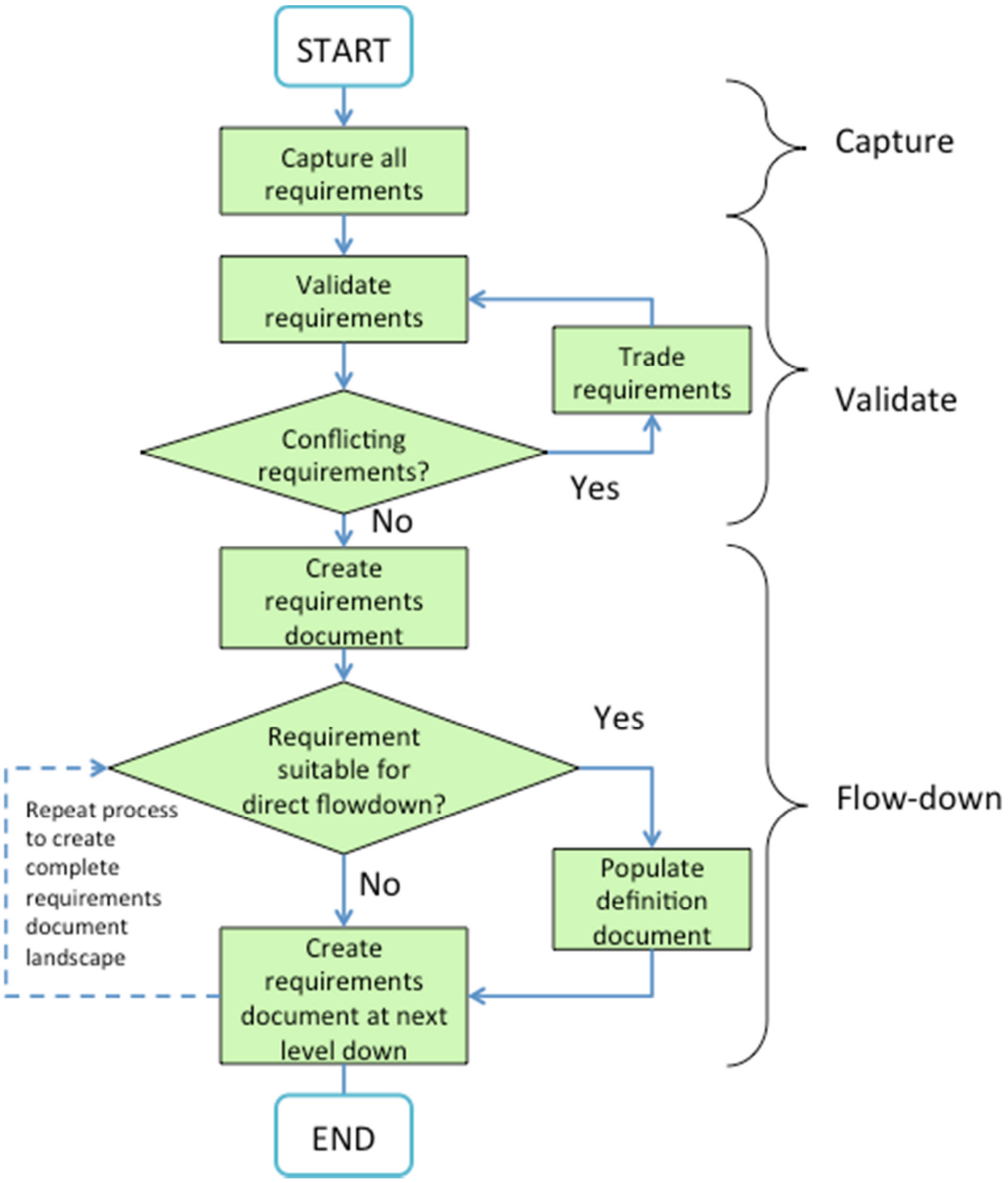

START
Capture all requirements
Validate requirements
Trade requirements
Conflicting requirements?
Yes
No
Create requirements document
Requirement suitable for direct flowdown?
Yes
No
Populate definition document
Create requirements document at next level down
Repeat process to create complete requirements document landscape
END
Capture
Validate
Flow-down

Figure 30 Formal process of requirement analysis

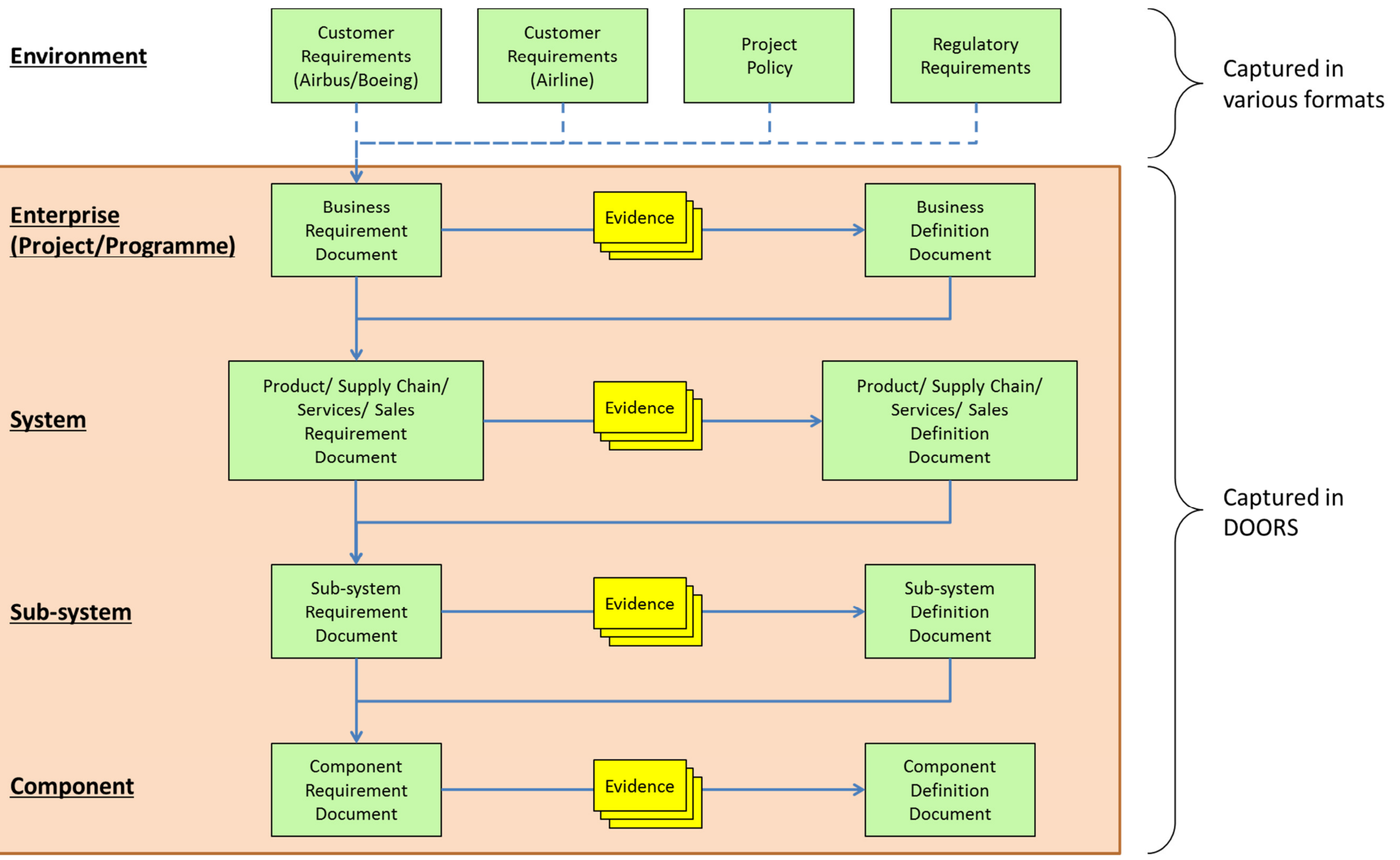

Environment
Customer Requirements (Airbus/Boeing)
Customer Requirements (Airline)
Project Policy
Regulatory Requirements
Captured in various formats
Enterprise (Project/Programme)
Business Requirement Document
Evidence
Business Definition Document
System
Product/ Supply Chain/ Services/ Sales Requirement Document
Evidence
Product/ Supply Chain/ Services/ Sales Definition Document
Sub-system
Sub-system Requirement Document
Evidence
Sub-system Definition Document
Component
Component Requirement Document
Evidence
Component Definition Document
Captured in DOORS

Figure 31 Formal process of requirements flow-down

4.2.2 Tools

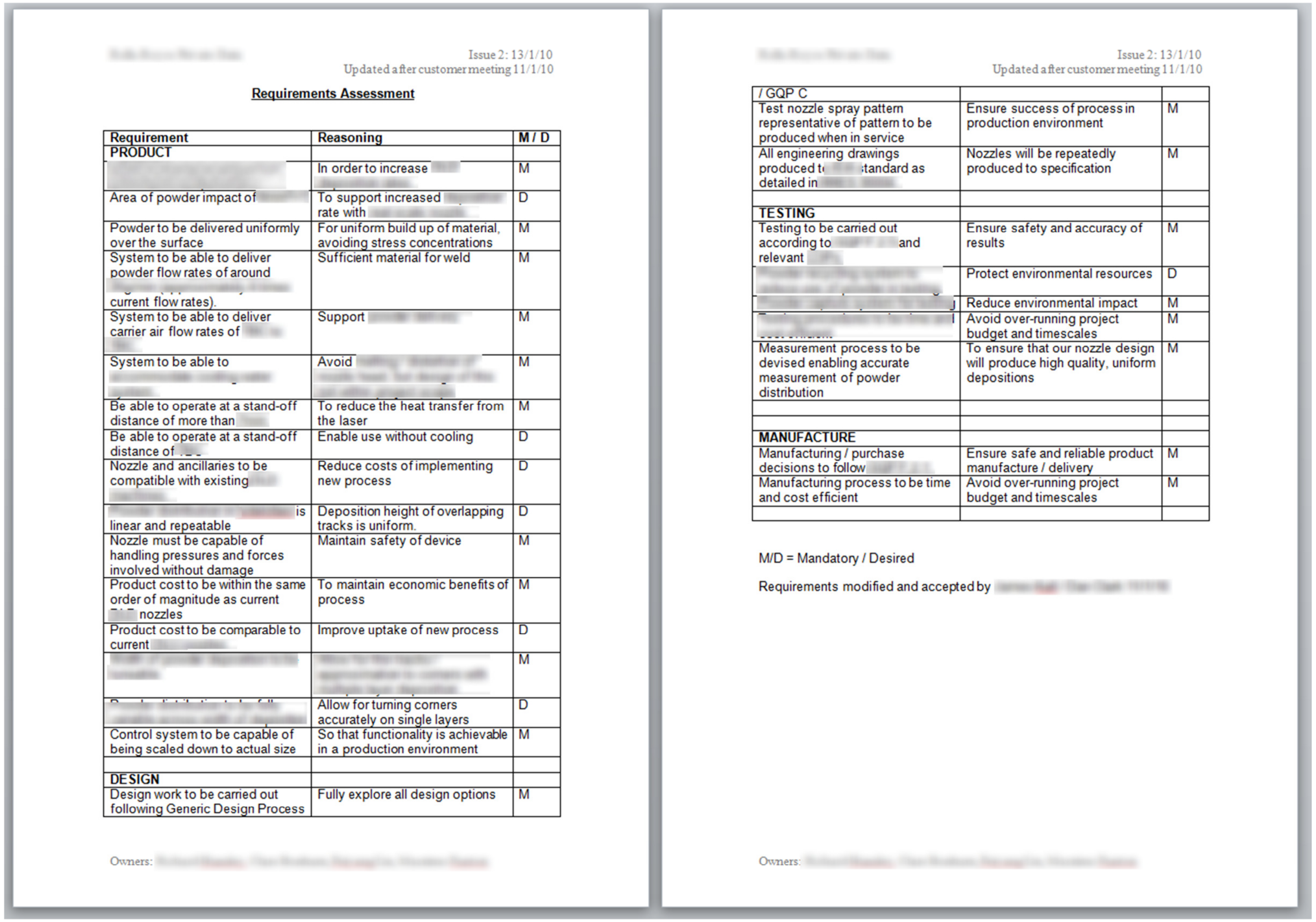
The requirement analysis process shown in Figure 30 also has associated with it a range of requirement tools. The process is divided into three stages: capture, validate, and flow-down. The tools are presented according to which of these stages they are used in. The tools are:

- Capture
  - Word processors and spreadsheets
- Validation
  - Functional Flow Diagram (FFD)
  - Viewpoint Analysis diagram (VPA)
  - Systemic Textual Analysis (STA)
- Validation and flow-down
  - Quality Function Deployment (QFD)
- Flow-down
  - IBM DOORS

With the exception of word processors and spreadsheets, the remaining tools are documented in the company recommended toolset for engineering design.

Capture

Word processors and spreadsheets are the most frequently used tools for requirements capture. They are preferred because most engineers are familiar enough with the interfaces of these tools; this means that requirements capture can keep the pace of requirement elicitation. Many variations of formatting have been observed for requirement capture using word processors and spreadsheets. Sometimes requirements are captured in tables, see Figure 32, whilst others in lists, see Figure 33 and Figure 34.



[redacted]

Issue 2: 13/1/10
Updated after customer meeting 11/1/10

Requirements Assessment

Requirement	**Reasoning**	**M / D**
PRODUCT		
[redacted]	In order to increase [redacted]	M
Area of powder impact of [redacted]	To support increased [redacted] rate with [redacted]	D
Powder to be delivered uniformly over the surface	For uniform build up of material, avoiding stress concentrations	M
System to be able to deliver powder flow rates of around [redacted] current flow rates).	Sufficient material for weld	M
System to be able to deliver carrier air flow rates of [redacted]	Support [redacted]	M
System to be able to [redacted]	Avoid [redacted]	M
Be able to operate at a stand-off distance of more than [redacted]	To reduce the heat transfer from the laser	M
Be able to operate at a stand-off distance of [redacted]	Enable use without cooling	D
Nozzle and ancillaries to be compatible with existing [redacted]	Reduce costs of implementing new process	D
[redacted] is linear and repeatable	Deposition height of overlapping tracks is uniform.	D
Nozzle must be capable of handling pressures and forces involved without damage	Maintain safety of device	M
Product cost to be within the same order of magnitude as current [redacted] nozzles	To maintain economic benefits of process	M
Product cost to be comparable to current [redacted]	Improve uptake of new process	D
[redacted]	[redacted]	M
[redacted]	Allow for turning corners accurately on single layers	D
Control system to be capable of being scaled down to actual size	So that functionality is achievable in a production environment	M
DESIGN		
Design work to be carried out following Generic Design Process / GQP C	Fully explore all design options	M
Test nozzle spray pattern representative of pattern to be produced when in service	Ensure success of process in production environment	M
All engineering drawings produced t[redacted]standard as detailed in [redacted]	Nozzles will be repeatedly produced to specification	M
TESTING		
Testing to be carried out according to [redacted] and relevant [redacted]	Ensure safety and accuracy of results	M
[redacted]	Protect environmental resources	D
[redacted]	Reduce environmental impact	M
[redacted]	Avoid over-running project budget and timescales	M
Measurement process to be devised enabling accurate measurement of powder distribution	To ensure that our nozzle design will produce high quality, uniform depositions	M
MANUFACTURE		
Manufacturing / purchase decisions to follow [redacted]	Ensure safe and reliable product manufacture / delivery	M
Manufacturing process to be time and cost efficient	Avoid over-running project budget and timescales	M

Owners: [redacted]

[redacted]

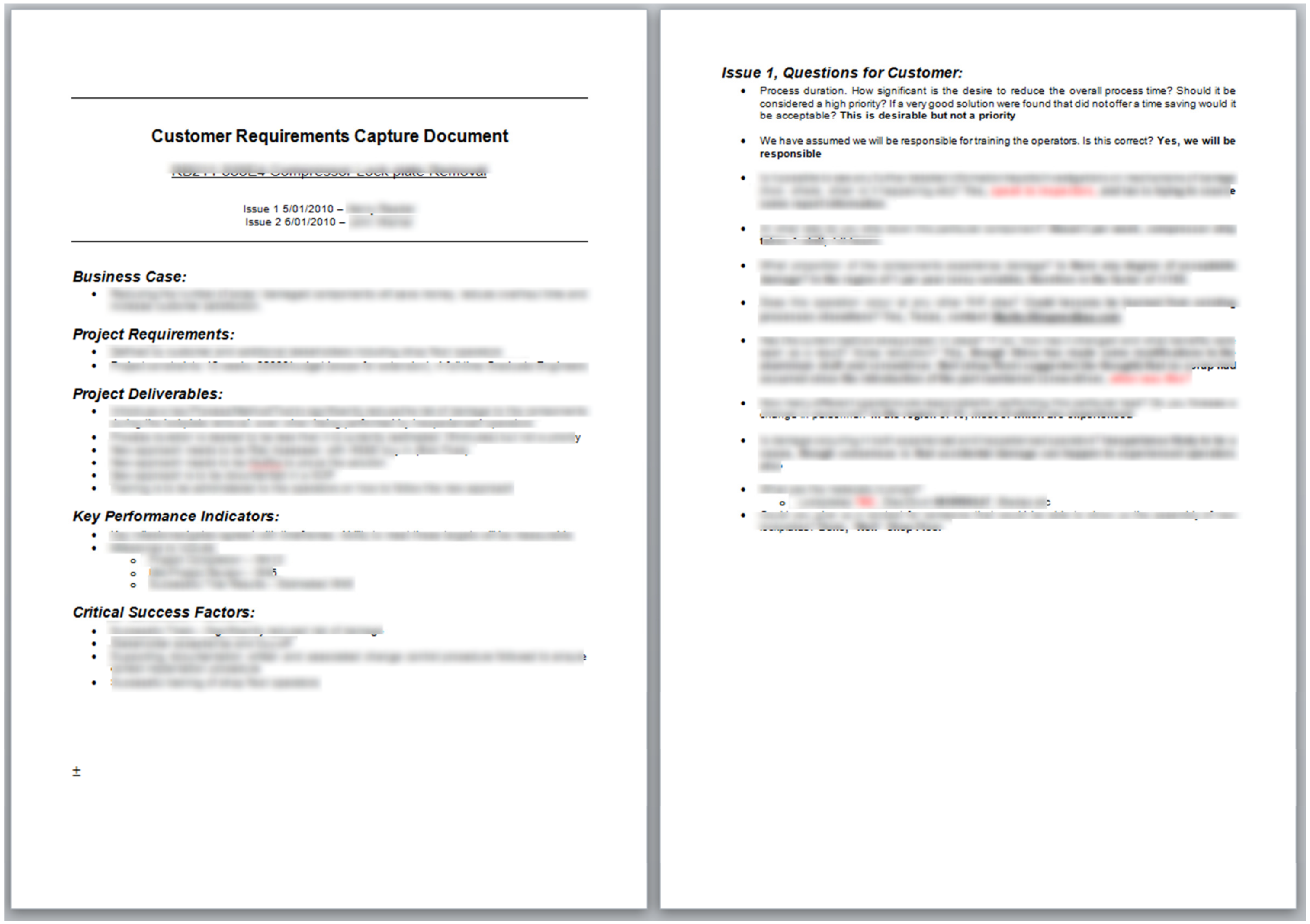
Issue 2: 13/1/10
Updated after customer meeting 11/1/10

M/D = Mandatory / Desired

Requirements modified and accepted by [redacted]

Owners: [redacted]

Figure 32 Requirements in a table (intentionally marked for confidentiality)



Customer Requirements Capture Document

[redacted]

Issue 1 5/01/2010 – [redacted]
Issue 2 6/01/2010 – [redacted]

Business Case:
- [redacted]

Project Requirements:
- [redacted]
- [redacted]

Project Deliverables:
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]

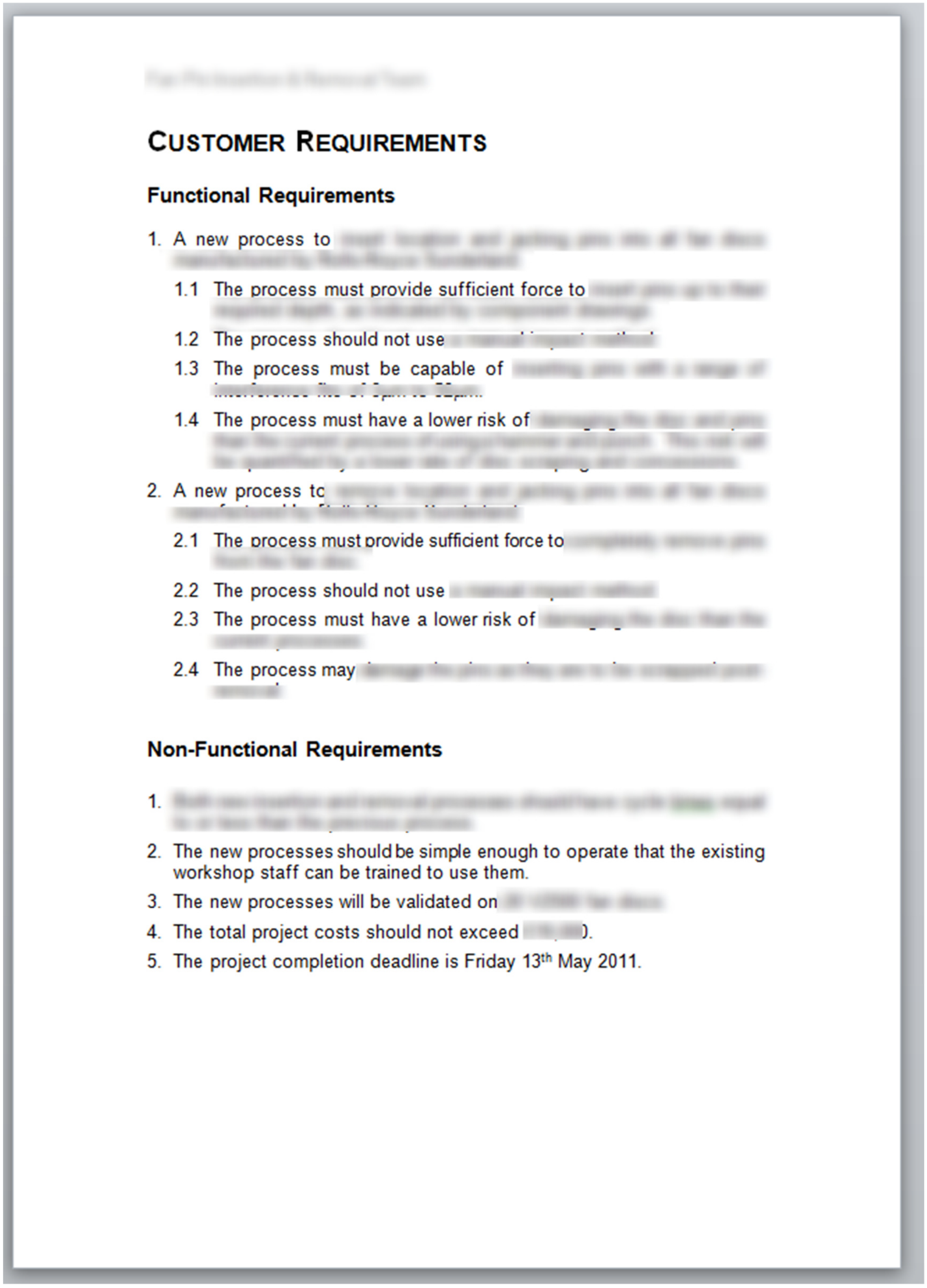
Key Performance Indicators:
- [redacted]
- [redacted]
  - [redacted]
  - [redacted]
  - [redacted]

Critical Success Factors:
- [redacted]
- [redacted]
- [redacted]
- [redacted]

±

Issue 1, Questions for Customer:
- Process duration. How significant is the desire to reduce the overall process time? Should it be considered a high priority? If a very good solution were found that did not offer a time saving would it be acceptable? **This is desirable but not a priority**
- We have assumed we will be responsible for training the operators. Is this correct? **Yes, we will be responsible**
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
  - [redacted]
- [redacted]

Figure 33 Requirements as a list (intentionally marked for confidentiality)



[redacted]

Customer Requirements

Functional Requirements

1. A new process to [redacted]
   1.1 The process must provide sufficient force to [redacted]
   1.2 The process should not use [redacted]
   1.3 The process must be capable of [redacted]
   1.4 The process must have a lower risk of [redacted]
2. A new process to [redacted]
   2.1 The process must provide sufficient force to [redacted]
   2.2 The process should not use [redacted]
   2.3 The process must have a lower risk of [redacted]
   2.4 The process may [redacted]

Non-Functional Requirements

1. [redacted]
2. The new processes should be simple enough to operate that the existing workshop staff can be trained to use them.
3. The new processes will be validated on [redacted]
4. The total project costs should not exceed [redacted]).
5. The project completion deadline is Friday 13th May 2011.

Figure 34 Requirements as an ordered list (intentionally marked for confidentiality)

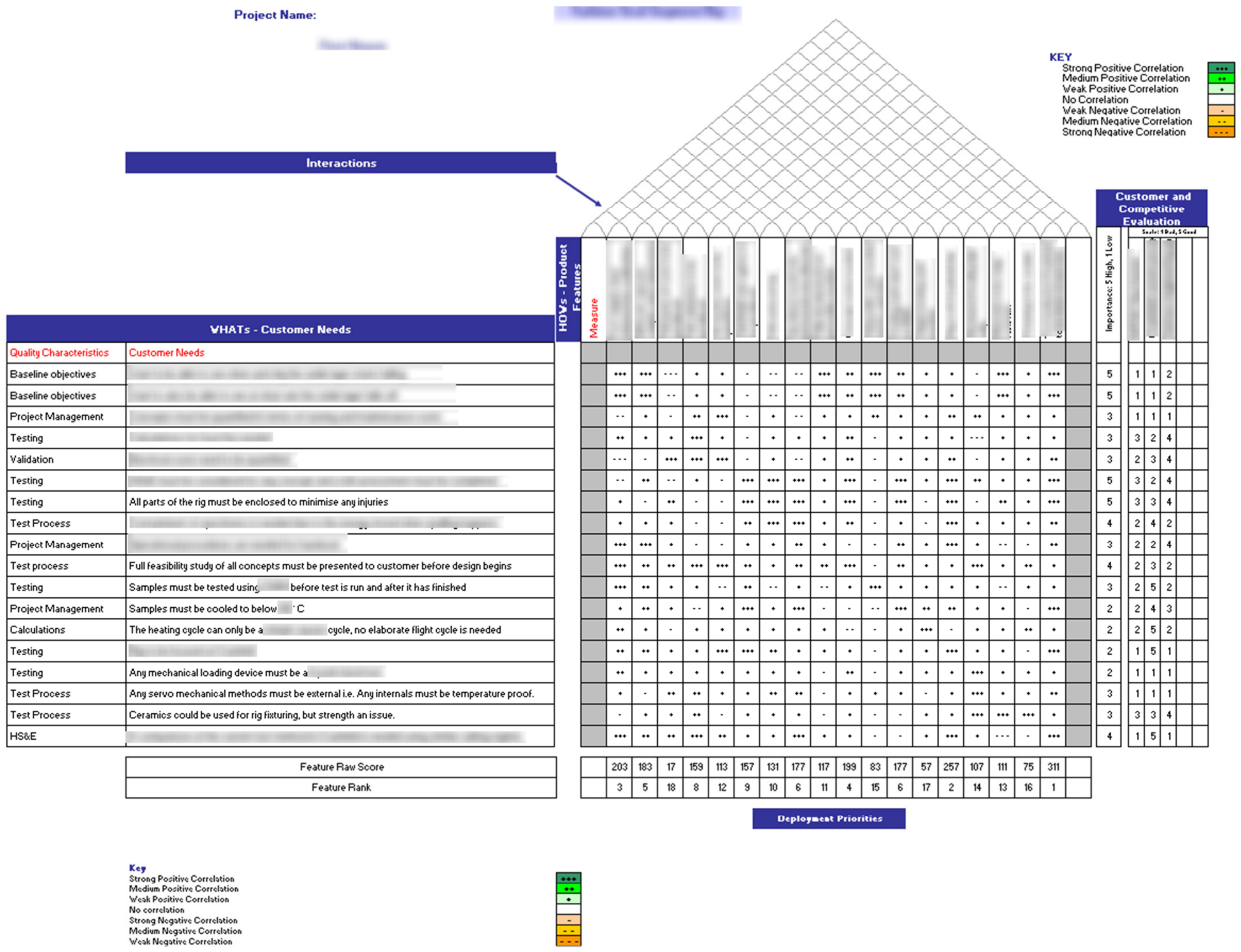
Validation
FFD, VPA, and STA were observed to be used to help validate captured requirements. The engineers who used these tools received training in a one-week taught course on systems engineering. The tools were used in the same way as instructed in the course material. These tools were introduced in Section 2.4.

Validation and flow-down
QFD has been observed to be used for both validating requirements as well as showing requirement flow-down. It is particularly favoured in projects which are original, as opposed to adaptive or evolutionary. In the collaborating company, there exists dedicated QFD software, known as Qualica, but it is not available on every computer. For this reason, most engineers use Microsoft Excel to create and edit QFD. There exists a company-defined QFD template in Excel to remove the need for engineers to building a House of Quality table from scratch, and an example of QFD that was built using this template is shown in Figure 35. QFD does not inherently support the capture of requirement rationale. In terms of traceability, QFD relies on requirements being consistently worded between QFD tables, e.g. between QFD1 and QFD2.

Flow-down
IBM DOORS was observed to be used for documenting requirement flow-down. It is particularly suited for large projects, and especially when the project has matured to a state that a requirement list becomes too large to be managed using a single document. Typically with large projects, a dedicated systems engineer is assigned to it. The systems engineer would define a format to capture requirements in DOORS.

Requirement flow-down can also be supported with a DOORS add-on called TraceLine (Pulham, 2008). It is, however, important to note that this add-on is not adopted by all engineering teams within the collaborating company. TraceLine is used to visualise flow-down of requirements, and it has views dedicated to display requirement trace links and requirement contextual metadata. These views help ensure that all requirement flow-downs are traceable and have rationale to justify how the children requirements satisfy the parents.

Requirements captured in DOORS follow a pre-defined format, shown in Figure 36. This format encourages rationale to be captured with requirements. It also reinforces the capture of other requirement metadata such as requirement id, a change flag, and hyperlinked references to related requirements.



Project Name: [redacted]

[redacted]

KEY

Strong Positive Correlation	•••
Medium Positive Correlation	••
Weak Positive Correlation	•
No Correlation	
Weak Negative Correlation	-
Medium Negative Correlation	- -
Strong Negative Correlation	- - -

Interactions

Customer and Competitive Evaluation

WHATs - Customer Needs

HOWs - Product Features

Quality Characteristics	Customer Needs	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	Importance: 5 High, 1 Low	[redacted]	[redacted]	[redacted]		
	Measure																								
Baseline objectives	[redacted]	•••	•••	- - -	•	•	-	- -	- -	•••	••	•••	••	•	•	-	•••	•	•••	5	1	1	2		
Baseline objectives	[redacted]	•••	•••	- -	•	•	-	- -	- -	•••	••	•••	••	•	•	-	•••	•	•••	5	1	1	2		
Project Management	[redacted]	- -	•	-	••	•••	-	•	- -	•	•	••	•	•	••	••	•	•	•	3	1	1	1		
Testing	[redacted]	••	•	•	•••	•	-	•	•	•	••	-	•	•	•	- - -	•	•	•	3	3	2	4		
Validation	[redacted]	- - -	-	•••	•••	•••	-	•	- -	•	••	-	•	•	••	-	•	•	••	3	2	3	4		
Testing	[redacted]	- -	••	- -	•	-	•••	•••	•••	•	•••	-	•••	•	•••	••	•	•	•••	5	3	2	4		
Testing	All parts of the rig must be enclosed to minimise any injuries	•	-	••	-	-	•••	•••	•••	•	•••	-	•••	-	•••	-	••	•	•••	5	3	3	4		
Test Process	[redacted]	•	•	•	-	-	••	•••	•••	•	••	-	•	-	•••	•	•	•	••	4	2	4	2		
Project Management	[redacted]	•••	•••	•	-	-	•	•	••	•	-	-	••	•	•••	•	- -	-	••	3	2	2	4		
Test process	Full feasibility study of all concepts must be presented to customer before design begins	•••	••	••	•••	•••	••	•	••	••	•••	-	••	•	•	•••	•	••	•	4	2	3	2		
Testing	Samples must be tested using [redacted] before test is run and after it has finished	•••	••	•	•	- -	••	- -	•	- -	•	•••	•	•	•	•	- -	•	•	3	2	5	2		
Project Management	Samples must be cooled to below [redacted] °C	•	••	•	- -	•	•••	•	•••	-	-	- -	•••	••	••	•	•	-	•••	2	2	4	3		
Calculations	The heating cycle can only be a [redacted] cycle, no elaborate flight cycle is needed	••	•	-	•	•	•	•	•	•	- -	-	•	•••	-	•	•	••	•	2	2	5	2		
Testing	[redacted]	••	••	•	•	•••	•••	••	•	•	•	-	•	•	•••	•	•	-	•••	2	1	5	1		
Testing	Any mechanical loading device must be a [redacted]	••	•	•	•	•	•	•	•	-	••	-	•	•	•	•••	•	•	•	2	1	1	1		
Test Process	Any servo mechanical methods must be external i.e. Any internals must be temperature proof.	•	-	••	••	•	•	••	••	-	•	•	•	-	•	•••	•	•	••	3	1	1	1		
Test Process	Ceramics could be used for rig fixturing, but strength an issue.	-	•	•	••	-	•	•	•	-	•	-	-	•	•	•••	•••	•••	•	3	3	3	4		
HS&E	[redacted]	•••	••	••	•••	••	•	•	•••	•	•	-	-	•	•••	•	- - -	-	•••	4	1	5	1		

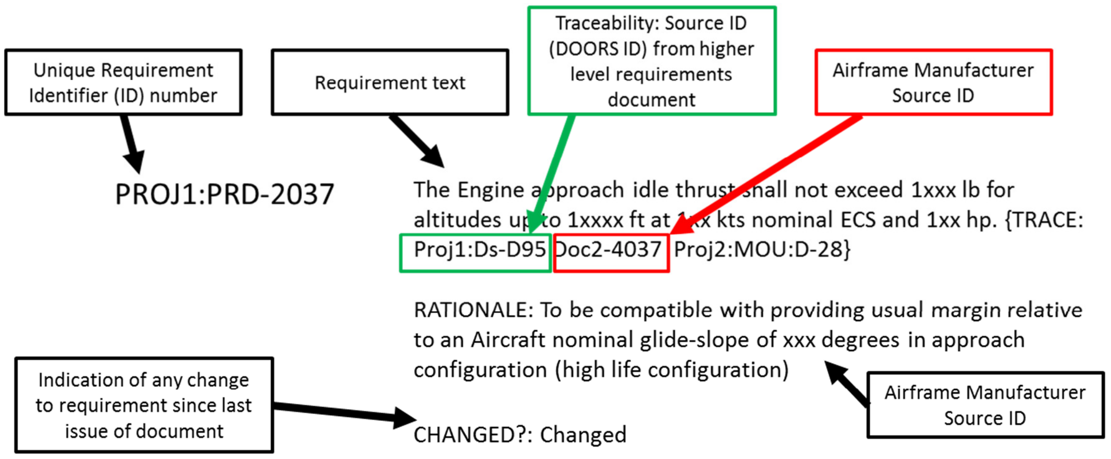
Feature Raw Score	203	183	17	159	113	157	131	177	117	199	83	177	57	257	107	111	75	311
Feature Rank	3	5	18	8	12	9	10	6	11	4	15	6	17	2	14	13	16	1

Deployment Priorities

Key

Strong Positive Correlation	•••
Medium Positive Correlation	••
Weak Positive Correlation	•
No correlation	
Strong Negative Correlation	-
Medium Negative Correlation	- -
Weak Negative Correlation	- - -

Figure 35 Requirement analysed on a QFD (intentionally marked for confidentiality)



Unique Requirement Identifier (ID) number
Requirement text
Traceability: Source ID (DOORS ID) from higher level requirements document
Airframe Manufacturer Source ID
PROJ1:PRD-2037
The Engine approach idle thrust shall not exceed 1xxx lb for altitudes up to 1xxxx ft at 1xx kts nominal ECS and 1xx hp. {TRACE: Proj1:Ds-D95 Doc2-4037 Proj2:MOU:D-28}
RATIONALE: To be compatible with providing usual margin relative to an Aircraft nominal glide-slope of xxx degrees in approach configuration (high life configuration)
Indication of any change to requirement since last issue of document
Airframe Manufacturer Source ID
CHANGED?: Changed

Figure 36 Template for capturing requirements in DOORS

All engineers who use DOORS in the collaborating company would have had at least a 4-hour intensive training. According to a systems engineer expert, some users within the company even reached a level where they are able to write scripts to automate certain processes in DOORS with ease. From the interviews with senior systems engineers it emerged that DOORS was selected over similar software because it has powerful change control and the ability to integrate with other tools (such as the ability to import and export into Microsoft Office documents).

4.3 Scope for improving existing practice

The analysis of existing practice has shown opportunities to improve requirement analysis in the collaborating company by focusing on requirement rationale and requirement traceability. The need and importance to have support for requirement rationale and traceability was discussed and justified in Section 2.5. These two research issues are linked to both the process and the tools. The opportunities presented in this Section are based both on the analysis of the documents and the interviews.

4.3.1 The process

The existing process of requirement analysis has scope for improvement in two ways. First, the need for capturing requirement rationale could be made more evident. For example, in the flow-diagram shown in Figure 30 in Section 4.2.1, there is no instruction as to when or how to capture requirement rationales. Second, requirement traceability could be enabled for all the stages of requirement analysis rather than being restricted to requirement flow-down. The workflow for requirement flow-down, shown in Figure 31 in Section 4.2.1, has been carefully crafted to enable requirement traceability across documents. However, there is no instruction as to how to enable traceability for requirements before they are flown-down.

The need for requirement rationale capture and rich traceability emerged also in the interviews. For example, a systems engineer responsible for overseeing a current large complex systems project acknowledged the need for “explicit traceability” and “collection of evidence around rich tracing”, i.e. evidence to cover the intention and fulfilment of a requirement. In particular, he indicated that the company was facing various challenges in modifying a complex system designed in the 70s without explicit rationale captured along with requirements.

4.3.2 The tools

Similar to the process, most of the tools currently used for requirement analysis also do not provide support for requirement rationale capture and requirement traceability. Table 11 shows how each tool measures in terms of their in-built support for rationale capture and traceability.

Table 11 Support provided by tools for requirement rationale capture and requirement traceability

Tools	In-built rationale support?	In-built traceability support?
Word processors & spreadsheets	No (unless using a customised template)	No (unless using a customised template)
VPA, STA, FFD	No	No
QFD	No	Yes
DOORS	Yes	Yes

Word processors and spreadsheets

Word processors and spreadsheets do not inherently support either rationale capture or requirement traceability, unless specifically designed templates are used. It has been observed that in some projects requirement rationales were captured along with requirements. And in other projects, requirements were seen labelled with IDs to facilitate traceability. However, these templates are not consistent, i.e. the same template has never appeared more than once.

VPA, STA, and FFD

VPA, STA, and FFD do not support requirement rationale capture nor requirement traceability.

QFD

QFD is not inherently designed to store rationale. Therefore, when a rationale-rich requirement is entered into a QFD, only the requirement statement part is retained. On the contrary, QFD does enable traceability through the duplication of requirement statements between two QFD tables. This kind of traceability is prone to synchronisation error when an update is not applied across all duplicates. Synchronisation error is revisited and discussed in greater detail in Section 8.2.2.

IBM DOORS

The DOORS tool supports rationale capture and has built-in traceability support. Rationale is captured when requirements are entered into DOORS using the company defined requirement input template, as shown on Figure 36. DOORS has been designed to support requirement traceability. In DOORS, every requirement has a unique addressable ID beginning with "doors://". This feature, coupled with hyperlinking mechanisms within DOORS, makes it very easy to link requirements thus making requirements navigable. In addition, every change made to a requirement is recorded in that requirement's log. With this feature, the history of requirement refinements is easily visible.

Despite the powerful features of DOORS, two factors confine its use to only after a set of requirements has been matured. First, DOORS is rarely used to capture requirement rationale and enable traceability at the point when requirements are created. This is because DOORS is not flexible enough as a tool for direct requirement capture and manipulation, such as capturing requirements live in a meeting. A lot of pre-requisites must be met before an engineer can operate within DOORS, including software installation, a stable connection to the DOORS server (e.g. not available on client site), and software training. Second, there exists a view that only fully-validated requirements can be input into DOORS (i.e. after requirements have been fully analysed). This means that unprocessed requirements elicited during a meeting may be considered unsuitable for DOORS.

4.3.3 Desired system attributes

On the basis of the discussions in the previous Section, four attributes of a system to improve existing practice were identified.

Attribute 1 – capture requirement rationale

One desired attribute is explicit support to capture requirement rationale in the requirement analysis process. Support can include formats to represent requirements in which rationale is an essential ingredient. Most of the processes and tools either reviewed in the literature or observed in the collaborating company do not support requirement rationale capture. An example where this has been achieved is the Volere template (Volere, 2014), which has a requirement attribute.

Attribute 2 – enable requirement traceability

Another desired attribute is explicit support to enable requirement traceability, particularly in source documents where requirements are created. Support should enable users to link requirements. Ramesh and Edwards (1993) state that requirement traceability should be bi-directional so that it is possible to see how requirements are arrived at, and be able to trace back to the original requirement at conception. Requirement bi-directional traceability is a concept proposed by Finkelstein (1991). Linked requirements can be navigated. Both QFD and DOORS already have this feature. However, the need to create traceability is not reinforced in the process that is followed to elicit and analyse requirements.

Attribute 3 – compatible with existing practices

Any tool that is to be introduced should not make engineers deviate from using the core flow-down workflow, nor prevent engineers from using the tools they already use. The existing company recommended practices (including the process and tools) have been refined over time and proven to be effective in situations that they are designed for. Therefore, any improvement to be proposed should only supplement, not replace, existing practices. Also from an implementation point-of-view, a supplementary proposal is more likely to be accepted.

Attribute 4 – suitable for use in requirement elicitation meetings

Any tool that is to be introduced should be suitable for use in requirement elicitation meetings, because this is the time when requirement rationales are most available. Two factors influence whether a tool is suitable for this environment: 1) complexity of setting up and operating the tool; 2) compatibility with other tools. DOORS is an example of a tool that is not flexible. Installation of DOORS is complex, and operating DOORS requires stable network access as well as operator expertise. As mentioned earlier, engineers require formal training before they can be productive with DOORS. Inflexibility would prevent DOORS being chosen as the preferred tool to use during requirement elicitation. QFD is also an example of a tool that is not flexible. Once requirements are transferred to the QFD format, it becomes impractical to analyse requirement in any other tool, because doing so would require a transformation of the requirements from QFD into a different format.

4.4 Proposing a workflow for requirement analysis

Workflow is a term used in this dissertation to describe the procedural steps and tools needed to undertake a business process (Maheshwari, 2008). In this project, a workflow is proposed that consist of a tool and a process. The proposed workflow is intended to improve requirement rationale capture and maintain requirement traceability. It shall be applied as a supplement to the current company practice. The proposal has been chosen in the form of a workflow because it is more readily applicable than recommending a tool on its own, without any guidance as to how to use it. By providing a workflow that integrates with existing practices, it is expected that engineers would find it easier to use it. Also, research has shown that making improvement holistically at a process level is much more effective than improving individual steps within that process (Sharp and McDermott, 2008). In this case, part of the workflow proposal involved examining the requirement analysis process as a whole.

4.4.1 The tool: Decision Rationale editor (DRed)

The tool chosen to support the proposed workflow is the Decision Rationale editor (DRed). The DRed software tool was developed to enable graphical capture of design rationale (Bracewell and Wallace, 2003; Bracewell et al, 2004; Bracewell et al., 2009a). A DRed map is a file that stores a network of connected nodes. The types of node relevant to this research are the issue node, the answer node (serve as requirement), the pro and con argument nodes, and the text node, see Table 12. Answer nodes can be in the statuses of neutral, accepted, or rejected. Nodes can be connected via arcs as well as by two additional types of link. The first is a transclusion link, it consists in creating a node that is a synchronised copy of another node (Bracewell et al, 2009). In DRed using the transclusion navigator transcluded nodes can be visited in a round-robin fashion. The second is tunnel link, which is essentially a bi-directional link. An example of rationale capture using DRed on an engine fan case internal acoustic liner is shown in Figure 37. Following its successful application for design rationale capture, it was found that the tool provided benefits also in laying out root cause analyses to solve engineering in-service problems (Bracewell et al., 2009b). Other uses of DRed include the capture of the functional interactions between the physical elements of a system (Aurisicchio et al, 2012) and requirement rationale capture (Dai and Aurisicchio, 2012). DRed differs from other IBIS-based tools as it does not need a dedicated database, which makes it compatible with existing document management practices. The overall development timeline of DRed is summarised in Figure 38. DRed is part of the standard toolset at the collaborating company. It has been installed on all engineers' computers.

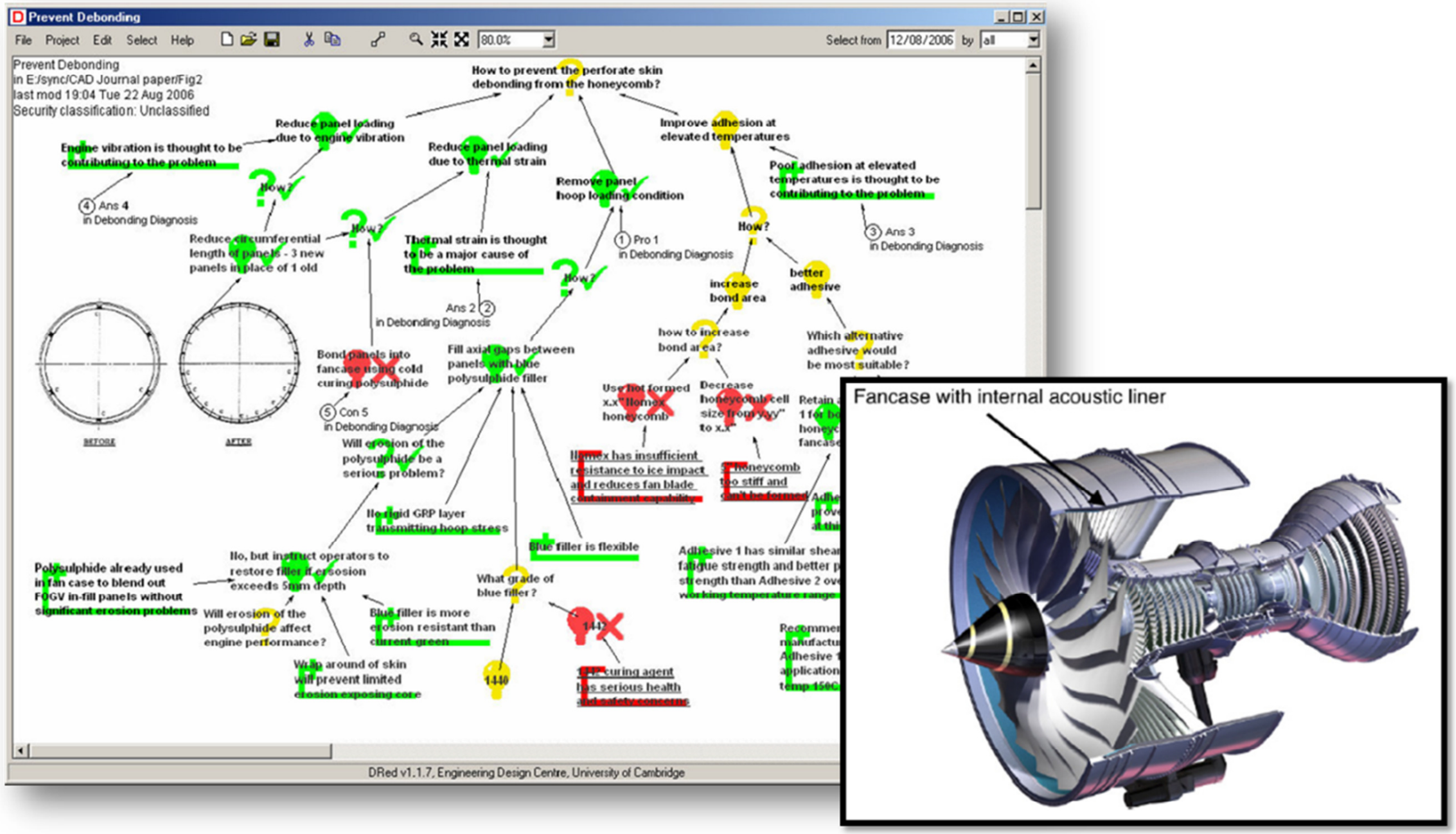

Prevent Debonding
File Project Edit Select Help
80.0%
Select from 12/08/2006 by all
Prevent Debonding
in E:/sync/CAD Journal paper/Fig2
last mod 19:04 Tue 22 Aug 2006
Security classification: Unclassified
How to prevent the perforate skin debonding from the honeycomb?
Reduce panel loading due to engine vibration
Engine vibration is thought to be contributing to the problem
Reduce panel loading due to thermal strain
Remove panel hoop loading condition
Improve adhesion at elevated temperatures
Poor adhesion at elevated temperatures is thought to be contributing to the problem
Ans 4 in Debonding Diagnosis
How?
Reduce circumferential length of panels - 3 new panels in place of 1 old
Thermal strain is thought to be a major cause of the problem
Pro 1 in Debonding Diagnosis
Ans 3 in Debonding Diagnosis
increase bond area
better adhesive
Ans 2 in Debonding Diagnosis
BEFORE
AFTER
Bond panels into fancase using cold curing polysulphide
Fill axial gaps between panels with blue polysulphide filler
how to increase bond area?
Which alternative adhesive would be most suitable?
Use hot formed x.x" Nomex honeycomb
Decrease honeycomb cell size from y.yy" to x.x"
Retain 1 for b honeyc fancase
Con 5 in Debonding Diagnosis
Will erosion of the polysulphide be a serious problem?
Nomex has insufficient resistance to ice impact and reduces fan blade containment capability
Honeycomb too stiff and can't be formed
Adhe prove at thi
No rigid GRP layer transmitting hoop stress
Blue filler is flexible
Adhesive 1 has similar shear fatigue strength and better strength than Adhesive 2 over working temperature range
Polysulphide already used in fan case to blend out FOGV in-fill panels without significant erosion problems
No, but instruct operators to restore filler if erosion exceeds 5mm depth
What grade of blue filler?
Will erosion of the polysulphide affect engine performance?
Blue filler is more erosion resistant than current green
1442
Recommen manufactu Adhesive 1 application temp 150C
Wrap around of skin will prevent limited erosion exposing core
1440
1442 curing agent has serious health and safety concerns
DRed v1.1.7, Engineering Design Centre, University of Cambridge
Fancase with internal acoustic liner

Figure 37 Using the Decision Rationale editor (DRed) to capture design rationale

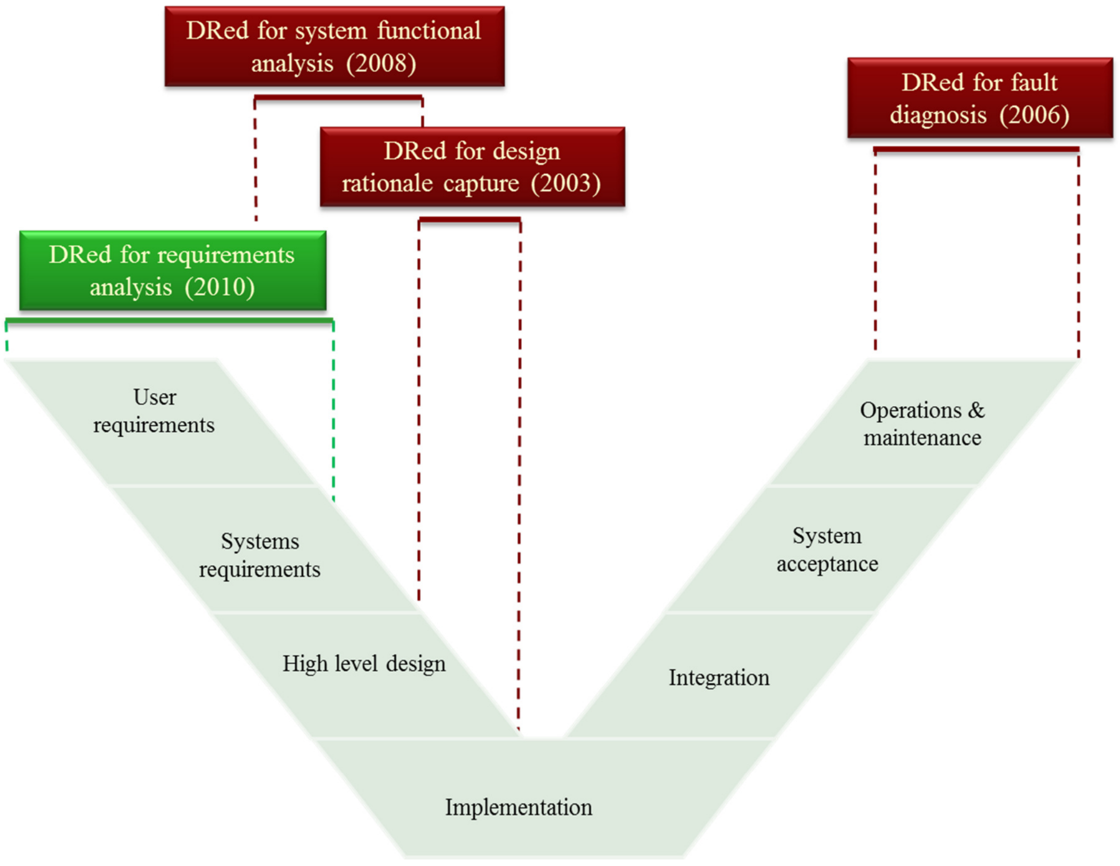

DRed for system functional analysis (2008)
DRed for design rationale capture (2003)
DRed for fault diagnosis (2006)
DRed for requirements analysis (2010)
User requirements
Systems requirements
High level design
Implementation
Integration
System acceptance
Operations & maintenance

Figure 38 Development timeline of the Decision Rationale editor (DRed)

Why the DRed tool satisfies the attributes?

DRed has been chosen as the tool for this workflow because it satisfies all four desired attributes defined earlier.

Attribute 1 – capture requirement rationale: DRed specialises in capturing rationale and can be used to capture other contextual metadata.

Attribute 2 – enable requirement traceability: DRed also has advanced hyperlinking ability such tunnels and transclusions (Bracewell et al. 2009b). Tunnels are bi-directional links that can connect requirements between two DRed maps. This can be useful to enable traceability between requirements. Transclusions are nodes which are linked and synchronised. It is suited to show duplicated requirements across multiple DRed maps. Nodes in DRed can also link into external documents (Bracewell et al., 2007) such as Microsoft Word and Excel. This can be useful to enable traceability between requirements captured in DRed and requirements captured using other tools.

Attribute 3 – compatible with existing practices: DRed is already recognised as one of the tools in the collaborating company's toolset. Engineers in the company have been using DRed for design related tasks (Aurisicchio and Bracewell, 2013).

Attribute 4 – suitable for use in requirement elicitation meetings: DRed can be used without prior training. The DRed software runs independently without a database, nor network access. DRed is versatile, and it has been demonstrated to be compatible with standard Microsoft Office tools such as Word and Excel (Bracewell et al., 2009a).

4.4.2 Process

The proposed process has three stages: 1) capturing requirements metadata; 2) structuring requirements; and 3) enabling traceability in requirements. These are now illustrated in turn.

Capturing requirement metadata

At the start of the requirement capture process, a new blank DRed map is created. As requirements are elicited, they should be captured along with any requirement metadata using the DRed notation in Table 12 in the format shown in Figure 39. This means that any requirement statement node can be attached with supporting or opposing rationale, background information, reference to other requirements or external files containing supporting evidence.

Table 12 DRed notations used for requirement and metadata capture

Type of information	DRed node
Requirement statement (undecided)	Neutral Answer
Requirement statement (accepted requirement)	Accepted Answer
Requirement statement (rejected requirement)	Rejected Answer
Rationale statement supporting a requirement	Pro
Rationale statement refuting a requirement	Con
A statement of background information (e.g. stakeholder who defined the requirement)	Text
Label for grouping purposes	Issue

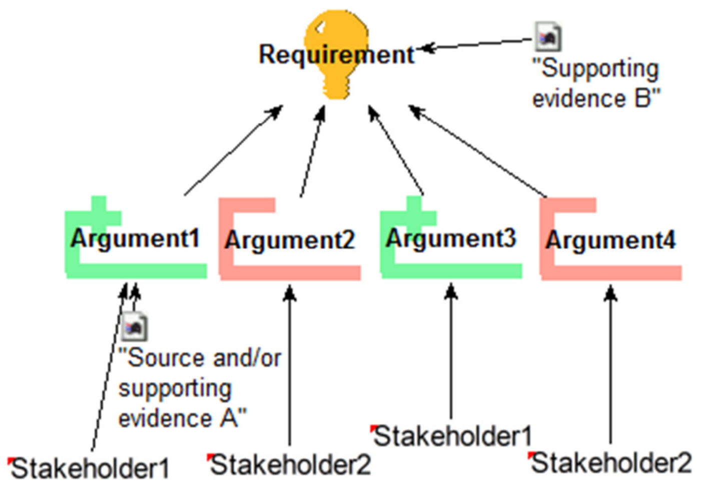


Figure 39 Requirement analysis representation

Once a DRed map has been populated, an optional technique can be used to help keep the map organised. A dotted line can be used to separate nodes that have been added during different requirement elicitation and analysis sessions. Each session is identified as a unique context consisting of a combination of time, themes explored and participants. Figure 40 shows a way to capture, juxtapose, check and agree requirement statements in a format in which it is possible to visually distinguish the different contexts within which a requirement or an argument was created. From Figure 40 it can be seen, for example, that argument 1 and argument 2 are related to requirement statement A and were elicited in different sessions.

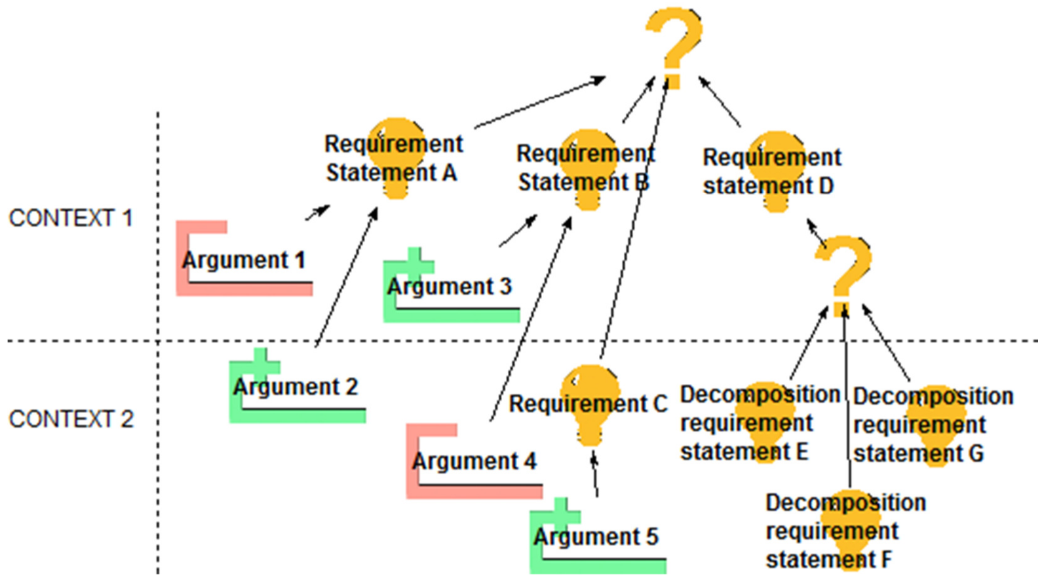


Figure 40 Requirement analysis and context representation

Structuring requirements

After the DRed map containing requirements and other metadata have been completed, a new DRed map is created for structuring. On this new DRed map, only the requirement nodes from the previous DRed map are transferred. These requirements shall be transferred via transclusion, so that synchronisation will be kept between the maps and synchronised nodes will be navigable. Structuring is achieved by grouping requirements and forming a hierarchy. Grouping is the process of identifying related requirements and letting a structure emerge. In Figure 41 the emergent structure is shown by grouping related requirements.

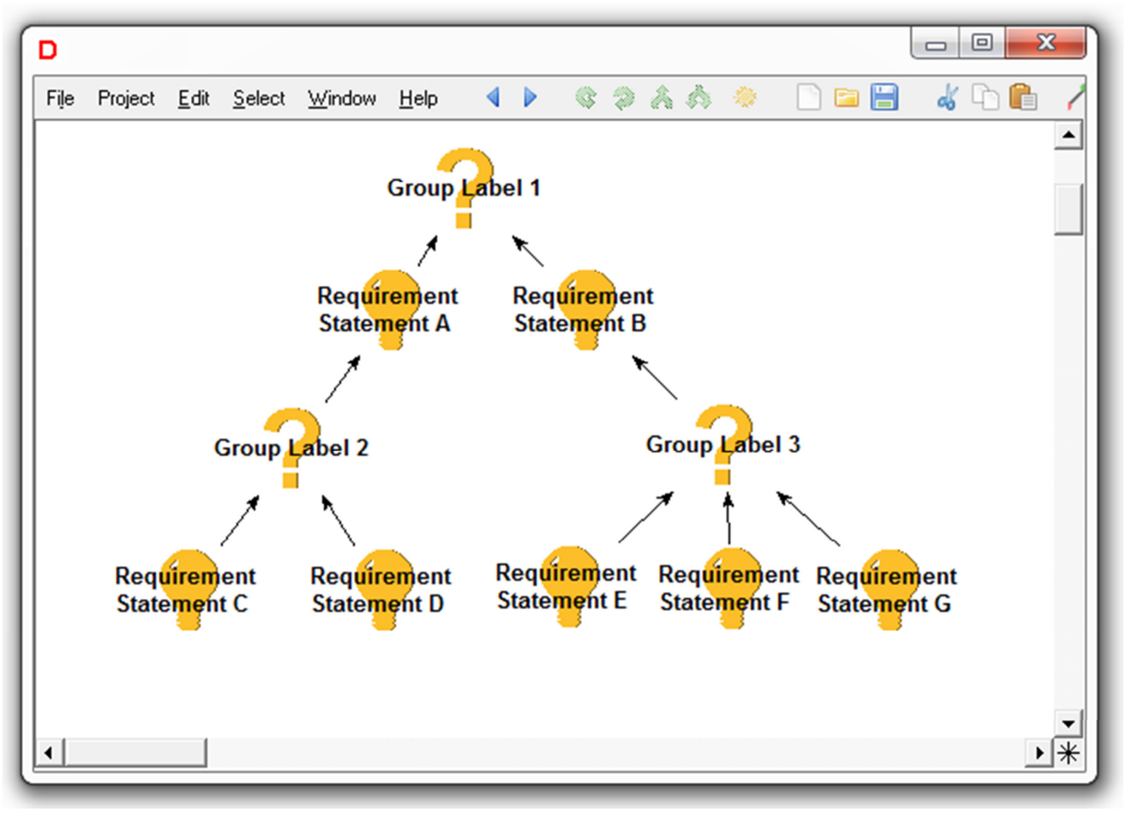


Figure 41 Requirement structuring representation

Enabling traceability in requirements

Requirement traceability is defined as "the ability to locate the requirement at any time within its life, both backwards and forwards" (Gotel and Finkelstein, 1994). This definition can be interpreted as every requirement being able to evolve into another requirement, and this evolution could be one-to-one, one-to-many, or many-to-one, as discussed in the literature review in Section 2.5.

Requirement evolution can happen both across documents and within a document. Evolution across multiple documents can take place when a set of requirements is transferred to another tool that specialise in a different kind of requirement analysis. Examples of transferring requirements for further analysis can be from DRed to QFD, or from DRed to DOORS. Evolution within a document can take place to capture the decomposition of a requirement (i.e. one-to-many evolution). An example of requirement decomposition is shown in Figure 40; it can be seen that the requirement statement D was decomposed into requirement statements E, F and G and that this operation was conducted as part of a separate session.

Enabling requirement traceability is defined as follows. Requirements shall be viewed as traceable if it is possible to interpret, from a requirement's metadata, where it has evolved from (source) and where it evolves to (destination). It is sufficient if it is possible to visually identify a requirement source and destination, e.g. on a traceability matrix. But a more user-friendly option would be to enable bi-directional navigation between source and destination requirements, e.g. using hyperlinks to connect the source and destination requirements.

Why the process satisfies the attributes?
This process has been chosen because it satisfies all of the four desired attributes defined earlier.

Attribute 1 – capture requirement rationale: the process deliberately places requirement rationale capture as the first stage; this is to enable requirement rationales to be captured when they are at most available.

Attribute 2 – enable requirement traceability: the process has been designed to take into account both scenarios that require within document requirement traceability and cross-document requirement traceability.

Attribute 3 – compatible with existing practices: the process has been designed with integration of other file formats in mind, such as QFD in Excel and IBM DOORS.

Attribute 4 – suitable for use in requirement elicitation meetings: the process has designated DRed as the tool to be used to capture requirements and rationale because it is a tool that is easy to learn and use.

The design of this process also aligns with the requirement analysis model defined in Section 2.3. The model divided requirement analysis as stages of checking, structuring, and evolution. The proposed process also defines activities for requirement analysis to be performed in that order. As part of the process, requirements are expected to be checked, structured into a hierarchy, and their evolution made traceable both within DRed maps and between DRed and other tools, such as QFD and DOORS.

4.5 Discussion

This research has presented an investigation of the requirement analysis practices of a global engineering company. The main results of the research are discussed below.

First, a picture of requirement analysis in the collaborating company has been drawn. The practice was found to consist of one process and a selection of software tools to support the process. The process is an instantiation of the Vee-model (Forsberg and Mooz, 1992) in systems engineering as discussed in Section 2.2.2. The process guides engineers to perform requirement validation as system-level requirements are flown-down to component-level requirements. The tools used to support the process had been previously discussed in the literature review in Section 2.4.

Second, two points for potential improvements of existing requirement analysis practice were identified. In particular, it was found that requirement rationale could be better captured, and requirement traceability could be more effectively supported.

Third, a workflow has been proposed to improve current requirement analysis practice. The proposed workflow has been designed to satisfy four specification attributes: 1) capture requirement rationale; 2) enable requirement traceability; 3) being compatible with existing practices; and 4) being suitable for use in requirement elicitation meetings. This workflow is similar to the proposal made by Ramesh and Dhar (Ramesh and Dhar, 1992) who have also advocated the use of IBIS to justify requirements. However, their proposal was based on the assumption that all stages of requirements analysis must be carried out using their tool and requirements must be converted to their REMAP format. Placing a restriction on the tool and format is not always practical in requirement analysis. As the investigations in this Chapter have demonstrated, the process of requirement analysis involves multiple tools in multiple formats because different tools specialise in different aspects of requirement analysis. For example affinity diagram facilitates structuring and systemic textual analysis facilitates checking. The workflow proposed in this Chapter removes any format-specific restriction by introducing requirement traceability. In this workflow, requirements can be analysed using any requirement analysis tool provided that the captured requirements are bi-directionally traceable to their origin.

Methodologies used to collect data about the current requirement analysis process include interviewing, document analysis, and shadowing. Interviewing systems engineers helped the researcher discover the desired practices to perform requirement analysis within the collaborating company. Document analysis and interviewing helped the researcher understand how requirement analysis is performed in practice. Eres et al. (2014) also applied the techniques of semi-structured interviewing and shadowing in an engineering and aerospace context. In this Chapter, a workflow for requirement analysis was proposed based on knowledge obtained in the literature review and research in industry. Hamraz et al (2013) applied a similar methodology in their research in which their proposal of a workflow is also based on literature survey and insights from industrial case studies.

4.6 Conclusion

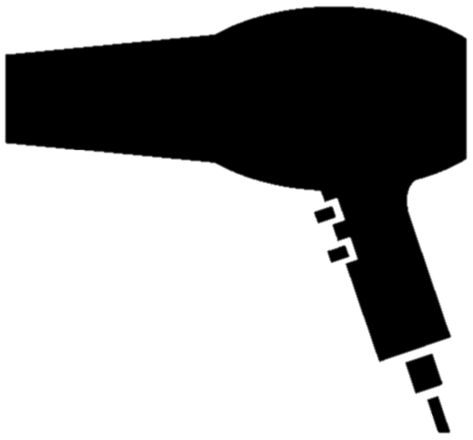
In this Chapter, a thorough understanding has been developed of the current requirement analysis practice within the collaborating company. Areas for improvements have been identified related to supporting requirement rationale capture and requirement traceability.

The methodology applied to capture data for this empirical study involved a combination of interviews, document analysis, and shadowing. This multi-channel approach has been effective at building a model of the current practice from various perspectives. Observations of overlapping concepts have helped cross-check the validity of the model. One limitation of the methodology is that not all engineers shadowed have been aware of company best practices; on average they had joined the company for less than a year. Although it was made stated that these engineers were often working under the guidance of more senior engineers who would make recommendations of the best practices available.

The outcome of the case study in this Chapter can be summarised in three points. First, robust and well-established processes and tools exist that guide engineers to perform requirement flow-down in a way that retains some rationale and maintain a degree of traceability. However, these are not fit to be used in the early stages of requirement elicitation and analysis. As a result, requirement rationale is not captured when it is most available, and rationale-rich requirements cannot be traced too. Second, a cause of the lack of requirement rationale capture, and low coverage of requirement traceability, lies in the process and tools available to support requirement analysis. There is insufficient emphasis on the need to capture requirement rationale and maintain requirement traceability. Also, some of the tools are unable to store rationale, nor any other type of requirement metadata. Third, many tools are not compatible with each other, which results in requirement captured by means of one tool having to be manually transferred to another. This need to transfer data can: discourage engineers from capturing requirement rationale in the first place; reduce the likelihood of transferring captured rationale, and maintaining requirement traceability during the transfer.

This Chapter contributes a workflow that can be potentially used to improve requirement rationale capture and enabling more requirement traceability. This proposed workflow is based on the model of requirement analysis described in Chapter 2. It contains elements of checking, structuring, and capturing requirement evolution. In the subsequent Chapters, this workflow shall be validated using empirical data.

Chapter 5 Evaluation of the workflow by reverse-engineering



Chapter outline

This Chapter presents a case study to validate the feasibility of the proposed workflow. The workflow is applied to the phases of requirement capture and analysis for a consumer product. The specific objectives of this Chapter are to demonstrate that the proposed workflow: 1) is feasible when applied to requirements of an engineering product, 2) can be incorporated into common existing requirement analysis practices, and 3) allow more requirement rationales to be captured and can improve requirement traceability.

5.1 Data collection and analysis

The data for this case study consists of requirements and rationale that were obtained by reverse engineering a consumer product. The data analysis was validated by two independent researchers to ensure coding reliability. This Section also presents an analysis to structure the acquired requirements and rationales according to the proposed workflow.

5.1.1 Reverse-engineering

Reverse-engineering has long been successfully used as a research method in engineering design (Otto and Wood, 2001). In this project the researcher applied reverse-engineering to a commercial hair dryer for the purpose of eliciting its requirements.

A consumer off-the-shelf hair dryer was bought, analysed as a whole, disassembled, and its parts examined to identify its functional and non-functional requirements and their metadata. Each disassembled component was studied to find out its characteristics, e.g. dimensions, weight, material, etc. From these analyses the non-functional requirements of the hair dryer were extrapolated. The disassembly also helped uncover functions satisfied by the product as a whole and its components. From these the functional requirements of the hair dryer were identified. No additional information was available on the design of the hair dryer, apart from the product itself and its manual. The reverse-engineering process produced a dataset of 15 functional requirements, 26 non-functional requirements, and 68 rationale statements.

5.1.2 Analysis to structure requirements

Following the elicitation of the requirements and rationales, the proposed workflow was applied to the data to analyse the requirements. The analysis resulted in four documents: 1) an IBIS map to capture and analyse newly elicited requirements (DRed); 2) an IBIS map to organise the requirements into a hierarchical structure (DRed); 3) a Quality Function Deployment table to study the relationship between requirements (Microsoft Excel); and 4) a list of requirements organised in a hierarchy (IBM DOORS). These documents were selected with the aim of representing as closely as possible the current practices at the collaborating company. Both QFD and DOORS are recommended support tools at the collaborating company. In addition, QFD was captured in Microsoft Excel which is another popular software format used to manage requirements at the collaborating company. The relationships between these four documents are shown in Figure 42. These documents are described in greater detail below.

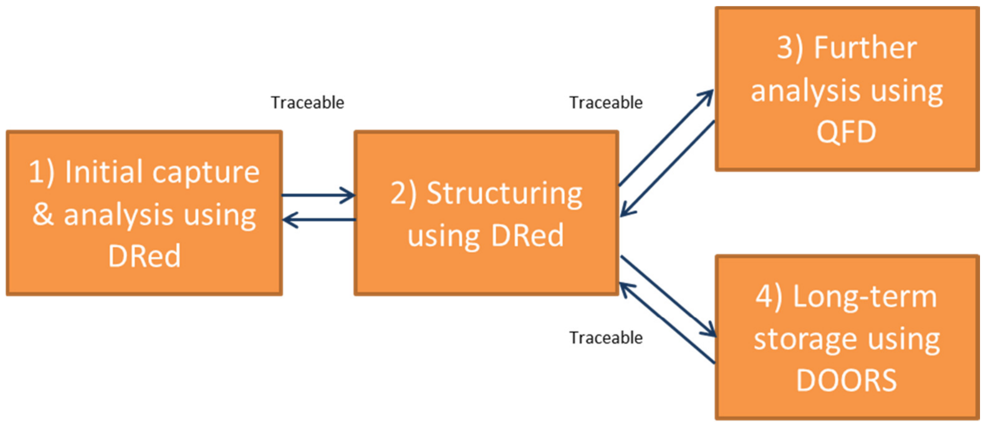

1) Initial capture & analysis using DRed
Traceable
2) Structuring using DRed
Traceable
3) Further analysis using QFD
Traceable
4) Long-term storage using DOORS

Figure 42 A graphical representation of the relationships between the four documents used to analyse requirements

IBIS map to capture and analyse newly elicited requirements

An IBIS map was used to capture the requirements and the metadata elicited through the reverse-engineering process, see Figure 43. The format defined in Section 4.4.2 was used to create this map. The requirements and metadata were mapped in DRed as if they emerged from two meetings. In the first meeting, participants, including customers and other stakeholders, gathered to express their needs. In the second meeting, the participants, including technical stakeholders from the company designing and selling the hair dryer, performed analysis on the requirements elicited during the previous session. The requirements and metadata elicited in the two meetings are divided by a dotted line. One type of metadata appearing frequently is requirement rationale which is captured as either a pro or con argument nodes in DRed. Most rationales were labelled with the stakeholders who created them. Depending on the meeting in which these rationales were elicited, they are either above or below the dotted line. Another type of metadata is the requirement source which is captured as a file node in DRed that refers to an external file. Most requirements share a common source that originates from a hypothetical interview. It is important to note that these requirements are unstructured, i.e. they were mapped and analysed as they were elicited.

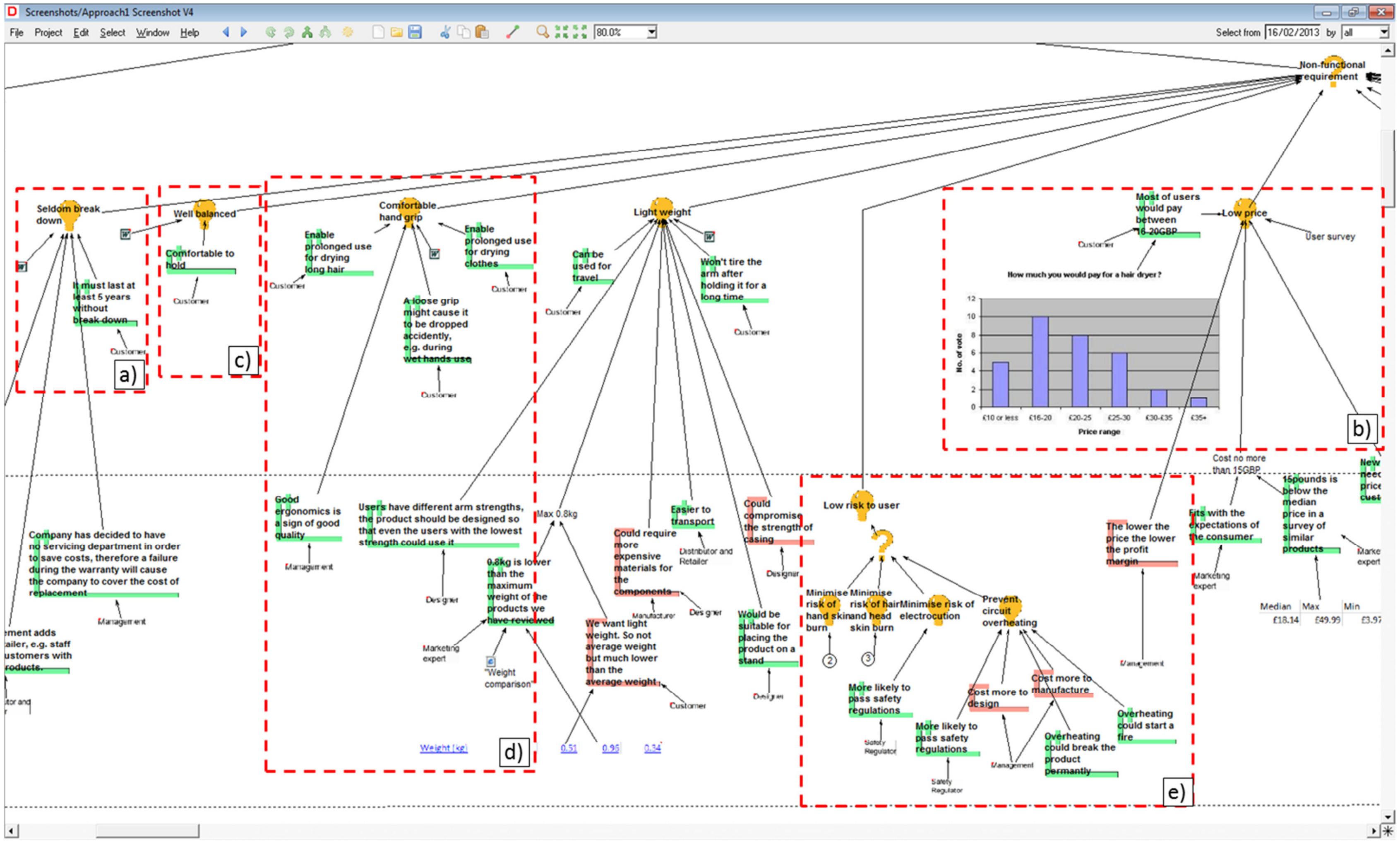

Screenshots/Approach1 Screenshot V4
File Project Edit Select Window Help
80.0%
Select from 16/02/2013 by all
Non-functional requirement
Seldom break down
It must last at least 5 years without break down
Customer
a)
Well balanced
Comfortable to hold
Customer
c)
Comfortable hand grip
Enable prolonged use for drying long hair
Customer
Enable prolonged use for drying clothes
Customer
A loose grip might cause it to be dropped accidently, e.g. during wet hands use
Customer
Good ergonomics is a sign of good quality
Management
Users have different arm strengths, the product should be designed so that even the users with the lowest strength could use it
Designer
0.8kg is lower than the maximum weight of the products we have reviewed
Marketing expert
"Weight comparison"
Weight (kg)
d)
Light weight
Can be used for travel
Customer
Won't tire the arm after holding it for a long time
Customer
Max 0.8kg
0.51
0.95
0.34
Could require more expensive materials for the components
Manufacturer
Designer
Easier to transport
Distributor and Retailer
We want light weight. So not average weight but much lower than the average weight
Customer
Could compromise the strength of casing
Designer
Would be suitable for placing the product on a stand
Designer
Company has decided to have no servicing department in order to save costs, therefore a failure during the warranty will cause the company to cover the cost of replacement
Management
Low price
Most of users would pay between 16-20GBP
Customer
User survey
How much you would pay for a hair dryer?
No. of vote
Price range
£10 or less
£16-20
£20-25
£25-30
£30-£35
£35+
b)
Cost no more than 15GBP
15pounds is below the median price in a survey of similar products
Median £18.14
Max £49.99
Min £3.97
Fits with the expectations of the consumer
Marketing expert
The lower the price the lower the profit margin
Management
Low risk to user
Minimise risk of hand skin burn
Minimise risk of hair and head skin burn
Minimise risk of electrocution
Prevent circuit overheating
More likely to pass safety regulations
Safety Regulator
More likely to pass safety regulations
Safety Regulator
Cost more to design
Cost more to manufacture
Management
Overheating could break the product permantly
Overheating could start a fire
e)

Figure 43 A proportion of the IBIS map. Red boxes with dotted lines indicate areas of this map that have been zoomed in further in subsequent figures

IBIS map to organise the requirements into a hierarchical structure
Following the initial capture and analysis of the requirements, an IBIS map was used to structure them, see Figure 44. This map contains only the requirement nodes from the first IBIS map. These requirement nodes have been transferred from the first IBIS map as linked copies, i.e. nodes related by the DRed transclusion link. After having transferred them, the nodes have been organised into a hierarchical tree using an affinity diagramming-like technique.

Quality Function Deployment table to study the relationship between requirements
A QFD table, see Figure 45, was created using the same set of requirements in the IBIS maps. This table was created to analyse the relationships between non-functional and functional requirements. It is noteworthy that the requirements in QFD are traceable to the requirements in DRed. As it can be seen in Figure 45 the requirements are underlined and this means that a live hyperlink allows navigation back to the DRed documents.

DOORS table to store requirements
A DOORS project, see Figure 46, was created using the same set of requirements in the DRed maps and QFD table. The DOORS project was created to show how the requirements would be stored. It is noteworthy that the requirements in DOORS are traceable to the requirements in DRed. As it can be seen in Figure 46, for each requirement a drop-down menu can be accessed which allows navigation back to the DRed files. DOORS is an IBM commercial product which was accessed through the IBM Academic Initiative licence.

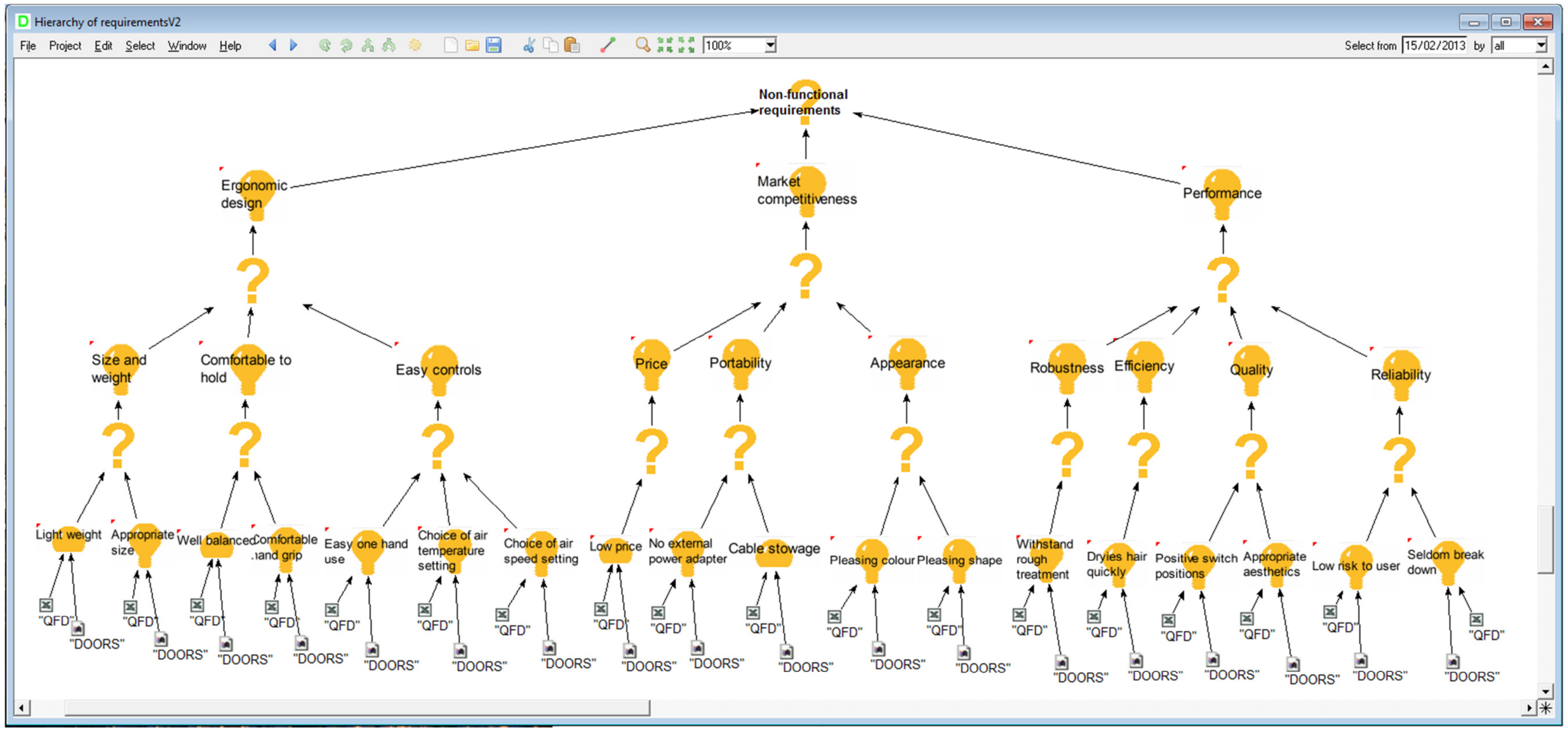
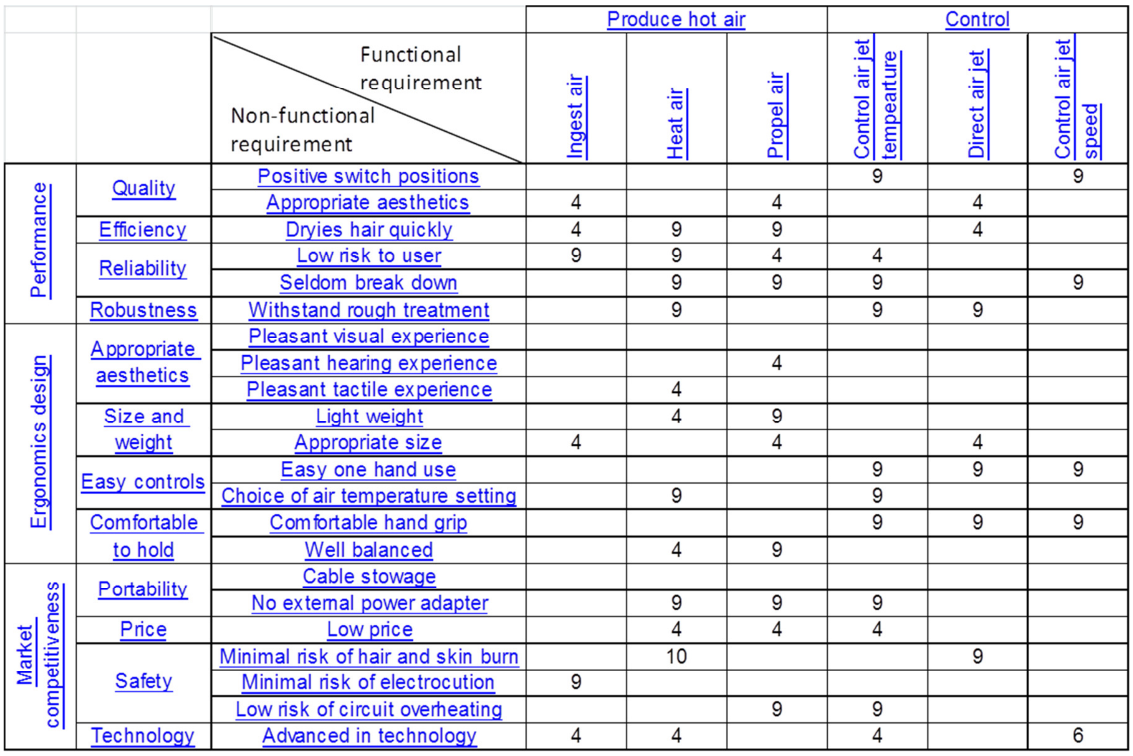
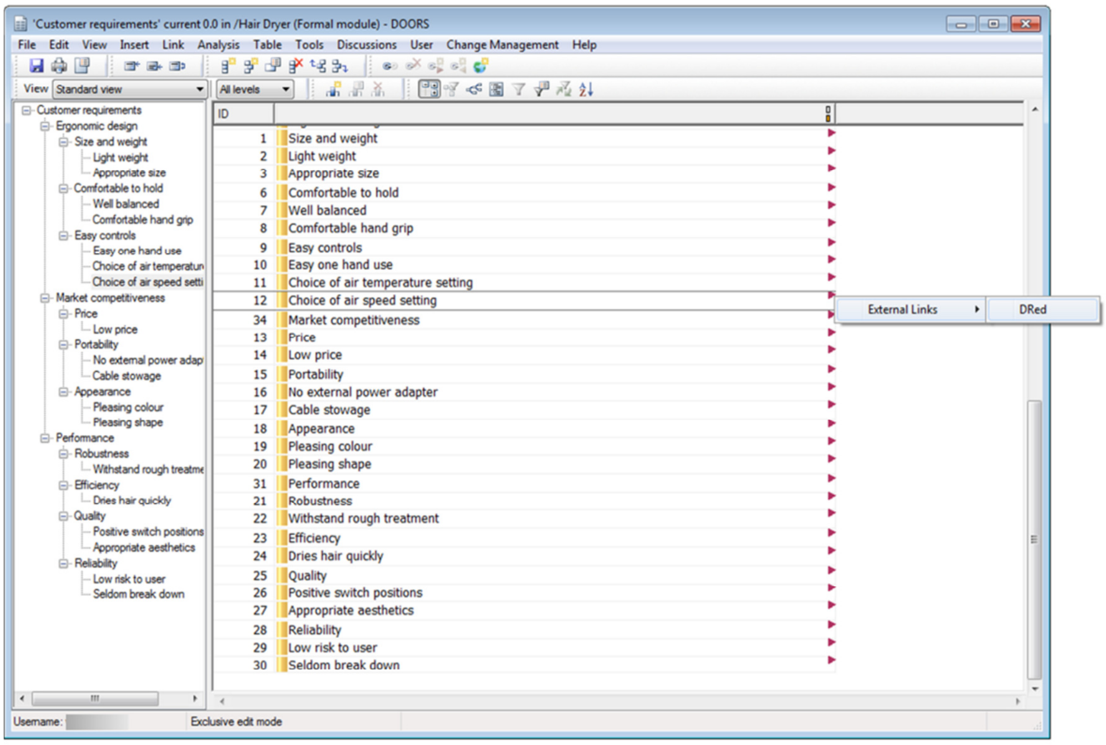


Figure 44 Structuring map



		Functional requirement / Non-functional requirement	Produce hot air			Control		
			Ingest air	Heat air	Propel air	Control air jet tempearture	Direct air jet	Control air jet speed
Performance	Quality	Positive switch positions				9		9
		Appropriate aesthetics	4		4		4	
	Efficiency	Dryies hair quickly	4	9	9		4	
	Reliability	Low risk to user	9	9	4	4		
		Seldom break down		9	9	9		9
	Robustness	Withstand rough treatment		9		9	9	
Ergonomics design	Appropriate aesthetics	Pleasant visual experience						
		Pleasant hearing experience			4			
		Pleasant tactile experience		4				
	Size and weight	Light weight		4	9			
		Appropriate size	4		4		4	
	Easy controls	Easy one hand use				9	9	9
		Choice of air temperature setting		9		9		
	Comfortable to hold	Comfortable hand grip				9	9	9
		Well balanced		4	9			
Market competitiveness	Portability	Cable stowage						
		No external power adapter		9	9	9		
	Price	Low price		4	4	4		
	Safety	Minimal risk of hair and skin burn		10			9	
		Minimal risk of electrocution	9					
		Low risk of circuit overheating			9	9		
	Technology	Advanced in technology	4	4		4		6

Figure 45 Requirement structured as a QFD table in Microsoft Excel



'Customer requirements' current 0.0 in /Hair Dryer (Formal module) - DOORS

File Edit View Insert Link Analysis Table Tools Discussions User Change Management Help

View: Standard view | All levels

- Customer requirements
  - Ergonomic design
    - Size and weight
      - Light weight
      - Appropriate size
    - Comfortable to hold
      - Well balanced
      - Comfortable hand grip
    - Easy controls
      - Easy one hand use
      - Choice of air temperatur
      - Choice of air speed setti
  - Market competitiveness
    - Price
      - Low price
    - Portability
      - No external power adap
      - Cable stowage
    - Appearance
      - Pleasing colour
      - Pleasing shape
  - Performance
    - Robustness
      - Withstand rough treatme
    - Efficiency
      - Dries hair quickly
    - Quality
      - Positive switch positions
      - Appropriate aesthetics
    - Reliability
      - Low risk to user
      - Seldom break down

ID	
1	Size and weight
2	Light weight
3	Appropriate size
6	Comfortable to hold
7	Well balanced
8	Comfortable hand grip
9	Easy controls
10	Easy one hand use
11	Choice of air temperature setting
12	Choice of air speed setting
34	Market competitiveness
13	Price
14	Low price
15	Portability
16	No external power adapter
17	Cable stowage
18	Appearance
19	Pleasing colour
20	Pleasing shape
31	Performance
21	Robustness
22	Withstand rough treatment
23	Efficiency
24	Dries hair quickly
25	Quality
26	Positive switch positions
27	Appropriate aesthetics
28	Reliability
29	Low risk to user
30	Seldom break down

Username: [redacted] Exclusive edit mode

Figure 46 Requirement in DOORS

Coding reliability

Applying the reverse-engineering process has resulted in requirements and matching rationales which were coded using the IBIS notation. In order to verify the reliability of the coding, an experiment was conducted. The experiment was designed to test if a requirement rationale matched its corresponding requirement and if a requirement rationale has pro and con arguments. The experiment was conducted through a survey and split into two parts. In part one, participants were presented with pairs of requirement and rationale which were either matched or mismatched. Participants were asked to establish if the link was correct or not. In part two, participants were presented with pairs of requirement and rationale in which the rationale was not classified as a pro or a con argument. The participants were asked to identify whether a rationale should be a pro or a con argument.

The survey was completed by two people, and the full survey and participants' response can be found in Appendix 1. In part one of the survey, more than 85% of the responses was in agreement with the originally elicited data. The remaining 15% of disagreements is possibly due to misinterpretation of rationale, or the interpretation being biased towards the participants' own field of research. In part two of the survey, all response agreed with the originally elicited data.

5.2 Results

The results of the research are presented in two Sections. Section 5.2.1 focuses on the requirement rationale capture aspect of the proposed workflow. Section 5.2.2 focuses on enabling requirement traceability.

5.2.1 Requirements rationale

The IBIS format in DRed has been used to capture requirements, requirement rationale and other requirement metadata. Rationale and other metadata have been used for requirement clarification, requirement justification, and requirement debate. Examples of these uses are provided below.

A case of requirement clarification is shown in Figure 47 in which a text node is used to clarify an answer node representing a requirement. A customer reinforces the meaning of "seldom break down" by quantifying that the hair dryer has to last at least 5 years. Note that there is a further text node linked to the node capturing the clarifying information which indicates the stakeholder who provided it. Also note the use of external hyperlinks in the form of a Microsoft Word icon in Figure 47, to point to a transcript that captures the stakeholder meeting where the requirement was elicited..

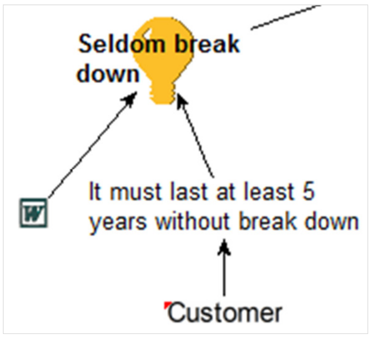


Figure 47 Requirement clarification (Figure 43a zoomed in)

Another example of requirement clarification is shown in Figure 48. A customer clarifies the meaning of "low cost" by stating that it has to cost less than £15. It is noteworthy that this time a file node appearing as an image is linked to the text node and it is used to justify why it is feasible. The requirement to produce a hair dryer with a retail cost below £15 is feasible as market research data (bar chart) reveals that other brands have managed to produce a hair dryer with even lower costs.

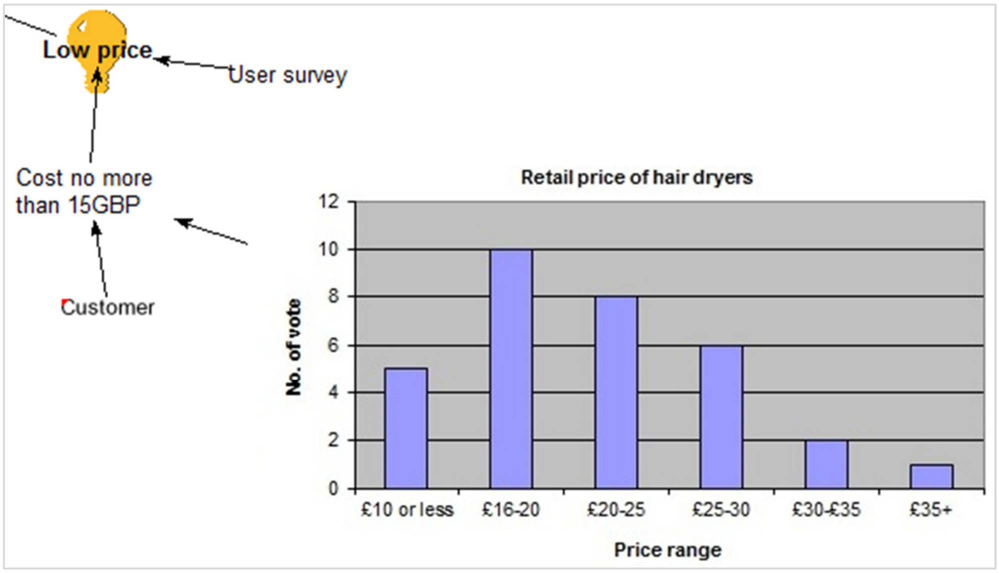
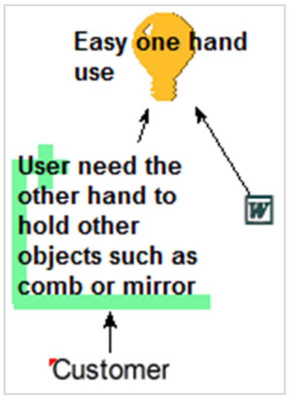


Figure 48 Requirement clarification and justification using graphical evidence (Figure 43b zoomed in)

A case of requirement justification is shown in Figure 49 in which a requirement statement is supported by a pro argument node. A customer defends the need for "easy one hand use" by arguing that the other hand has to be free to engage with other objects, e.g. a hair brush.



Easy one hand use
User need the other hand to hold other objects such as comb or mirror
Customer

Figure 49 Requirement justification using rationale (Figure 43c zoomed in)

An example of requirement debate is shown in Figure 50. The debate is captured and used to resolve possible conflicts. Pro argument nodes are initially used by the customer to justify the necessity for such requirement. In particular, the customer argues with two pro argument nodes that "light weight" is necessary as it makes the product suitable for travelling and less tiring. A text node is subsequently used to clarify the meaning of such requirement by quantifying it. Note that an upper limit of 0.8 Kg is specified. A pro argument node and a con argument node are then linked to such limit to express opinions on it. They show that current views on the proposed value are conflicting. At the same time other branches of the debate capture con argument nodes which raise risks with the "light weight" requirement and evaluate its impact on the "low cost" and "withstand rough treatment" requirements.

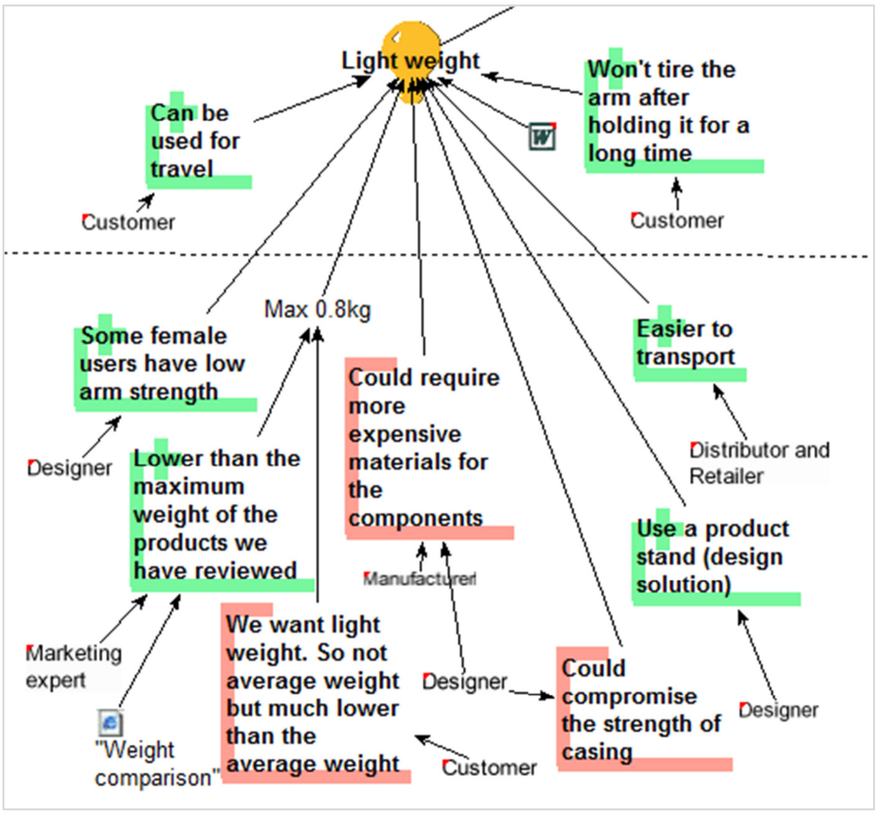

Light weight
Can be used for travel
Customer
Won't tire the arm after holding it for a long time
Customer
Max 0.8kg
Some female users have low arm strength
Designer
Easier to transport
Could require more expensive materials for the components
Distributor and Retailer
Lower than the maximum weight of the products we have reviewed
Use a product stand (design solution)
Manufacturer
Marketing expert
We want light weight. So not average weight but much lower than the average weight
Designer
Could compromise the strength of casing
Designer
"Weight comparison"
Customer

Figure 50 Requirement debate (Figure 43d zoomed in)

5.2.2 Requirement traceability

In Section 4.4.2, requirement traceability has been viewed as an important concept to enable both cross-document and within a document. In this case study there are both examples of cross-document traceability and within-document traceability.

Traceability across documents

Traceability across documents was enabled between DRed to DRed, DRed to QFD and DRed to DOORS.

Requirements across DRed files were linked with transclusion which is a bi-directional hyperlink. An example of where this link is used is between requirement nodes on IBIS map in Figure 43 and the requirements on the structuring map in Figure 44; this is also visualised on Figure 51. Transclusion enables the requirements at the source and destination to be navigable, and any changes to the source will be synchronised to the destination.

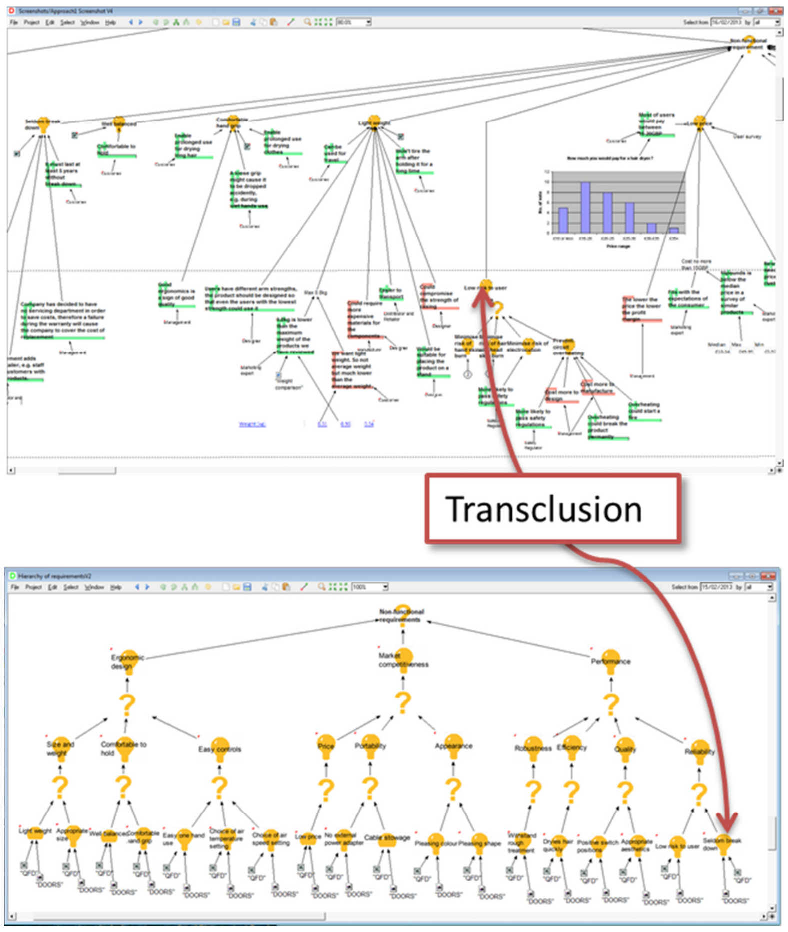
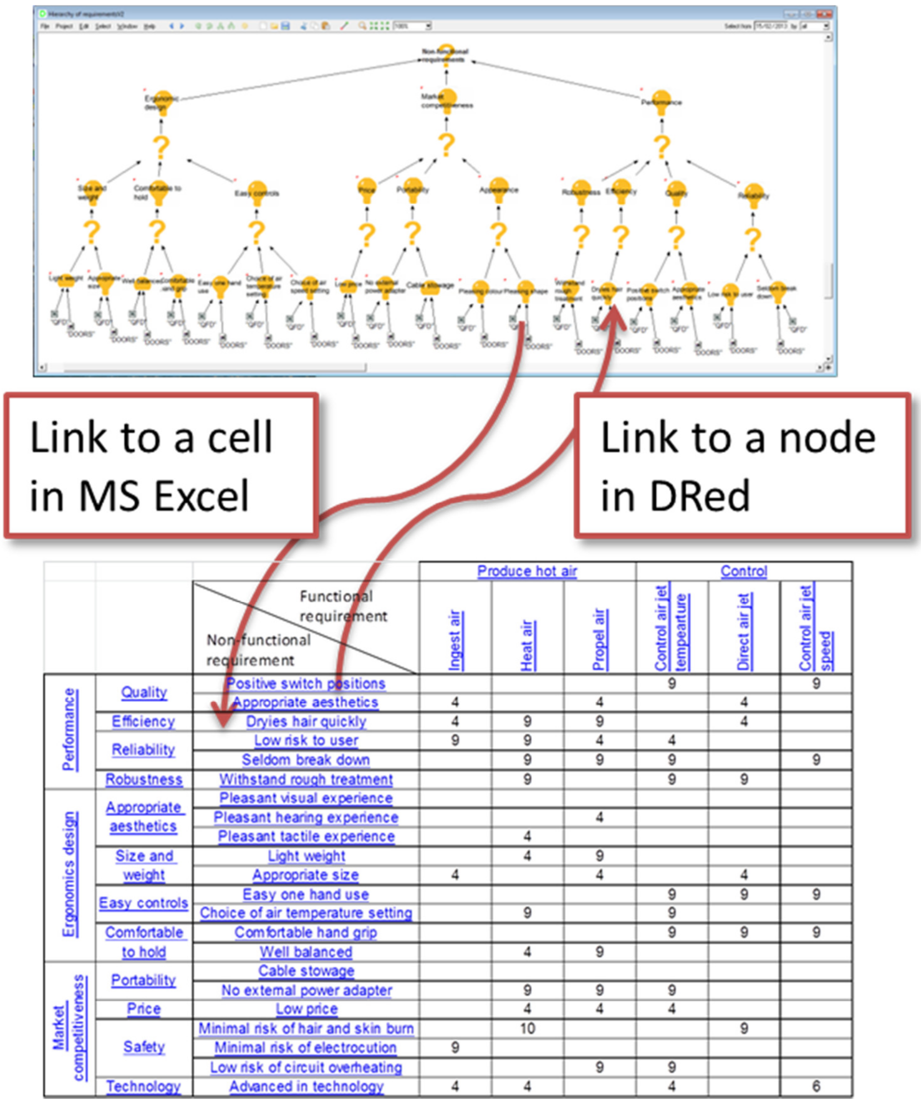


Figure 51 Transclusion link between two DRed documents

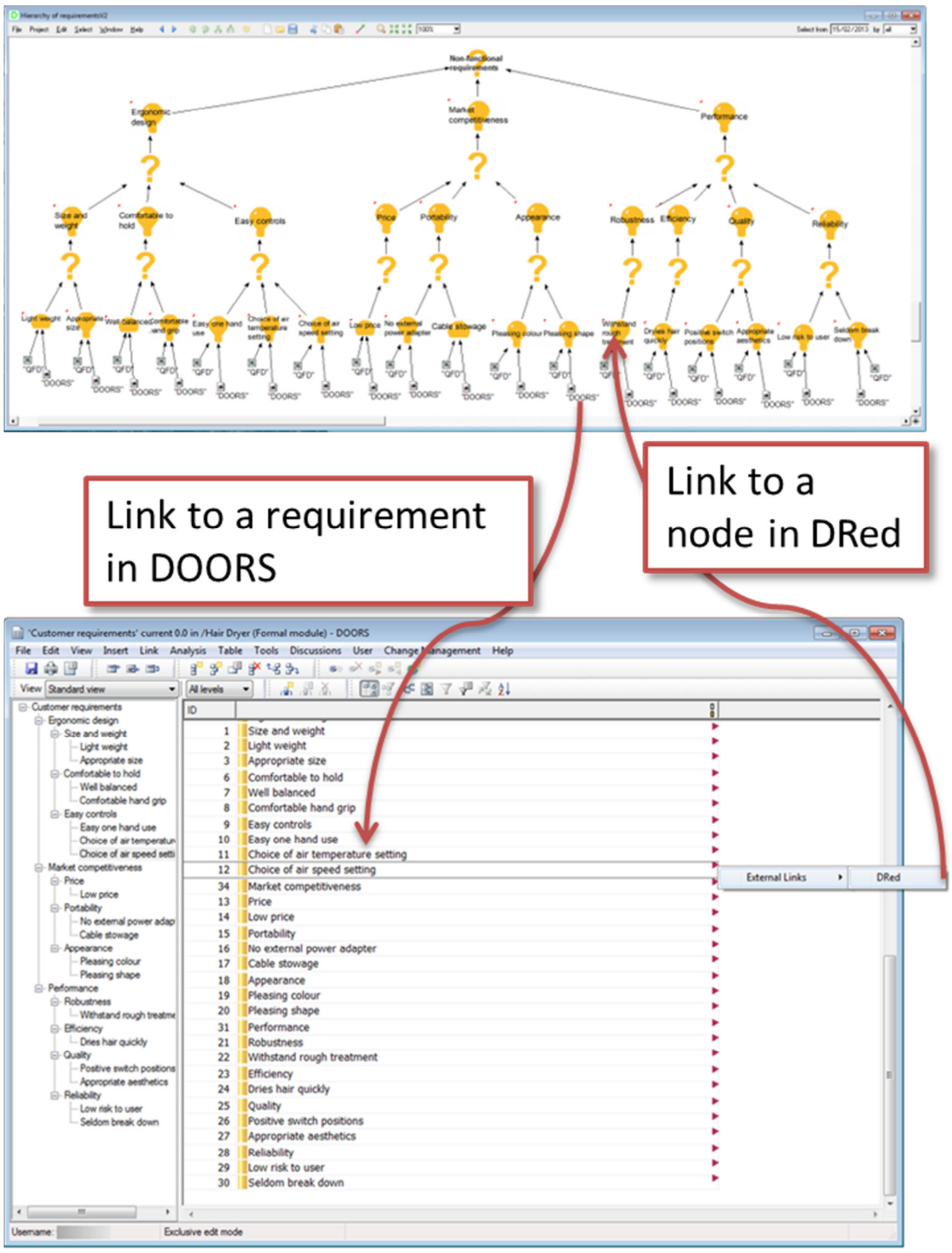
Requirements across DRed and QFD documents were linked with a bi-directional link between DRed and Microsoft Excel. An example of where this type of link is used is between requirement nodes on the structuring map in Figure 44 and the requirements on the QFD in Figure 45. The link can be visualised in Figure 52. To create a link from DRed to QFD, a file node is created in DRed that stores a reference to the corresponding requirement in QFD in Excel. This reference is constructed using a combination of the path to the Excel file and the alias to the corresponding Excel cell that stores the requirement. To create a reverse link from QFD to DRed, a requirement in an Excel cell stores a hyperlink to the corresponding requirement node in DRed. Any DRed node can be referenced because every node has a unique address by default, and this address is composed of the DRed file path and a unique node ID.



		Functional requirement / Non-functional requirement	Produce hot air			Control		
			Ingest air	Heat air	Propel air	Control air jet tempearture	Direct air jet	Control air jet speed
Performance	Quality	Positive switch positions				9		9
		Appropriate aesthetics	4		4		4	
	Efficiency	Dryies hair quickly	4	9	9		4	
	Reliability	Low risk to user	9	9	4	4		
		Seldom break down		9	9	9		9
	Robustness	Withstand rough treatment		9		9	9	
Ergonomics design	Appropriate aesthetics	Pleasant visual experience						
		Pleasant hearing experience			4			
		Pleasant tactile experience		4				
	Size and weight	Light weight		4	9			
		Appropriate size	4		4		4	
	Easy controls	Easy one hand use				9	9	9
		Choice of air temperature setting		9		9		
	Comfortable to hold	Comfortable hand grip				9	9	9
		Well balanced		4	9			
Market competitiveness	Portability	Cable stowage						
		No external power adapter		9	9	9		
	Price	Low price		4	4	4		
	Safety	Minimal risk of hair and skin burn		10			9	
		Minimal risk of electrocution	9					
		Low risk of circuit overheating			9	9		
	Technology	Advanced in technology	4	4		4		6

Figure 52 Hyperlink between DRed and Microsoft Excel documents

Requirements across DRed and DOORS were linked using bi-directional links between DRed and DOORS. An example of where this type of link is used is between requirement nodes on the structuring map in Figure 44 and requirements in DOORS in Figure 46. The link can be visualised in Figure 53. To create a link from DRed to DOORS, a file node is created in DRed that stores a reference to the corresponding requirement in DOORS. This reference is a DOORS address that is automatically generated when requirements are created in DOORS. To create a reverse link from DOORS to DRed, a requirement in DOORS can store the address to its corresponding requirement in DRed.


Link to a requirement in DOORS
Link to a node in DRed
'Customer requirements' current 0.0 in /Hair Dryer (Formal module) - DOORS
External Links
DRed

ID	
1	Size and weight
2	Light weight
3	Appropriate size
6	Comfortable to hold
7	Well balanced
8	Comfortable hand grip
9	Easy controls
10	Easy one hand use
11	Choice of air temperature setting
12	Choice of air speed setting
34	Market competitiveness
13	Price
14	Low price
15	Portability
16	No external power adapter
17	Cable stowage
18	Appearance
19	Pleasing colour
20	Pleasing shape
31	Performance
21	Robustness
22	Withstand rough treatment
23	Efficiency
24	Dries hair quickly
25	Quality
26	Positive switch positions
27	Appropriate aesthetics
28	Reliability
29	Low risk to user
30	Seldom break down

Figure 53 Demonstrating the ability to link to other tools, from a node in DRed to a requirement in DOORS

Traceability within a document through visualisation

Requirement decomposition is one type of evolution within a document. Figure 54 shows an example of decomposition. The requirement "appropriate aesthetics" was decomposed into the requirements "pleasant visual experience", "pleasant tactile experience" and "pleasant hearing experience". Other types of evolution within a document are requirement update and requirement re-composition.

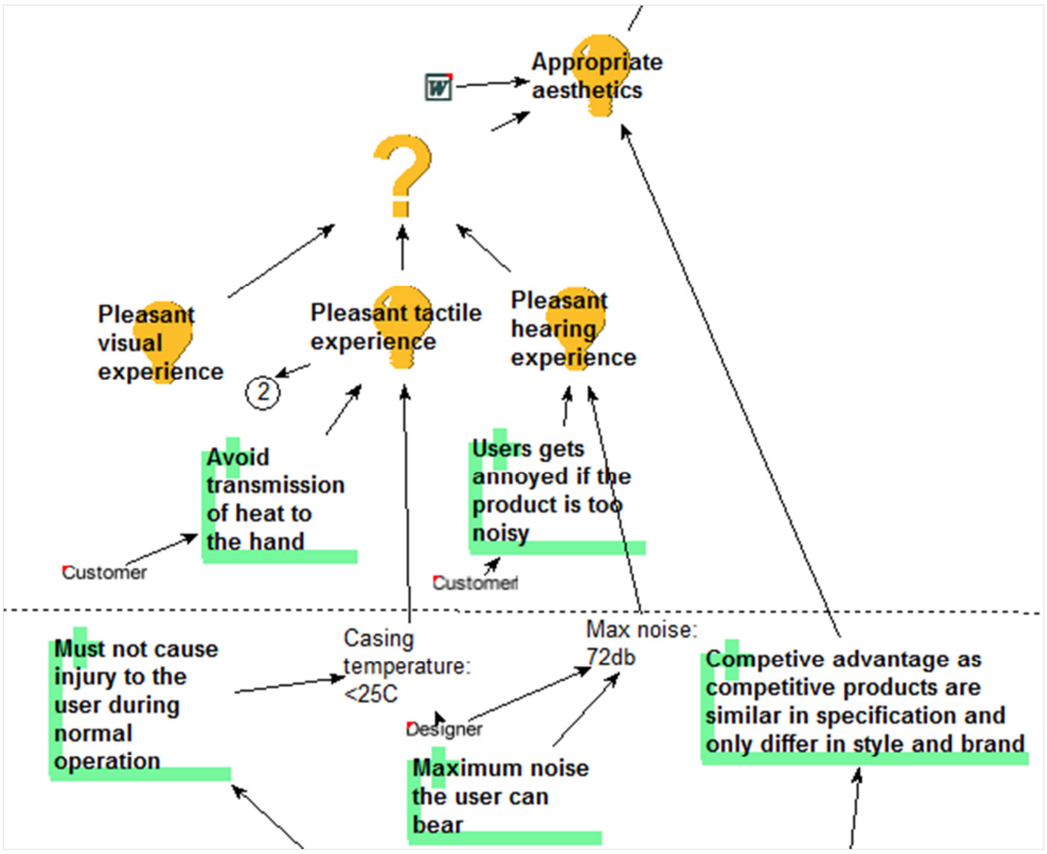

Appropriate aesthetics
Pleasant visual experience
Pleasant tactile experience
Pleasant hearing experience
2
Avoid transmission of heat to the hand
Users gets annoyed if the product is too noisy
Customer
Customer
Must not cause injury to the user during normal operation
Casing temperature: <25C
Max noise: 72db
Designer
Competive advantage as competitive products are similar in specification and only differ in style and brand
Maximum noise the user can bear

Figure 54 Requirement decomposition (Figure 43e zoomed in)

5.3 Discussion

This research has presented an application of the proposed workflow to a reverse engineered data set. The main results of the research are discussed below.

First, the case study has validated the feasibility of the proposed workflow, see Table 13. In particular, the case study has shown how requirements can be captured together with rich contextual information including argumentative statements and supporting documents. It was argued that explicit justification of requirements helps understand the reasons for requirements' existence and explains their importance. It was also shown how the requirements and the rationale developed by different groups of stakeholders can be laid out across designated areas of a DRed map enabling to trace the evolution of requirement analysis. The research has also proposed a solution to the problem of identifying the source of requirements and argumentation independently to whether this is an individual or a document. In line with the literature review, the results of the case study have shown that requirements analysis consists of capturing information to enable different types of check on requirements as well as define a hierarchical structure for requirements. The case study has also demonstrated how informal stakeholder requirements can be effectively documented, structured and transformed into a formal requirement specification. Specifically, the solution adopted consists of employing the IBIS notation, defining hierarchies of requirements and using DRed's transclusion link to follow requirements across DRed maps. Unlike managing hyperlinked requirements in DOORS, Compendium (Buckingham-Shum et al., 2006), and most other tools that require a database, DRed has a file-based architecture. Using a file architecture without the need for a database gives this workflow more portability, and therefore makes it more suitable to the informal early stages of requirement analysis.

Table 13 Coverage of validation

	Feasibility	Practicality	Scalability
Model requirements in IBIS	✓	✗	✗
Model requirement rationale in IBIS	✓	✗	✗
Enable requirement traceability	✓	✗	✗

Second, the workflow developed in this research is not intended to be used in isolation. Rather it is anticipated that it can co-exist with more formal requirement analysis and management tools such as QFD and DOORS. The case study has shown an example of how the result of structured requirements can be input in QFD and DOORS.

Third, used in the phase prior to the specification of system requirements, the workflow can fill in the gap left by current methods. QFD, for example, has no means to capture aspects of requirement analysis related to discussion and justification. The DOORS tool, on the other side, does not seem to have the flexibility to operate in this area. Therefore, the approach has the potential to extend the capture and traceability of requirements in early design, and trace a requirement to a DRed map where stakeholder rationale and other contextual information is stored. Analysis of engineering requirement would be undertaken on these maps prior to input into more formal tools such as QFD and DOORS.

Fourth, used in the phase after the specification of system requirements, the approach has also potential. It could in fact have the role of a platform to visualise and manipulate the structuring of requirements stored in QFD matrices or DOORS databases. Even when requirements are stored in formal repositories, it is unlikely that they will stay "frozen". As requirements are updated through design iterations, there is potential to adopt this approach to visualise decisions and rationale for changes.

Reverse-engineering was the principal method used to produce the data, consisting of requirements and their rationale. Further analysis was used to extrapolate more requirement metadata, such as requirement origin, graphs, and the requirement analysis sessions. Using reverse-engineering has been effective at generating a set of data that is realistic.

5.3.1 Limitations

This case study has two limitations. First, the practicality of the workflow has not been validated (see Table 13) as data for this case study was reconstructed by reverse engineering. This means that there is still a lack of empirical evidence to suggest that engineers in industry can follow this method of working during real projects. And scenarios, such as live capture during a requirement analysis meeting, are not yet verified. Second, the scalability of the proposed workflow has not been tested through this case study, see Table 13. The total number of requirements generated through reverse-engineering is small in comparison to the number of requirements for large projects such as a turbine engine. Theoretically, DRed can handle large number of nodes because its canvas is unbounded. Also, DRed has features such as tunnels (bi-directional hyperlinks) that can facilitate navigation of a large set of nodes spread across multiple DRed maps.

5.4 Conclusions

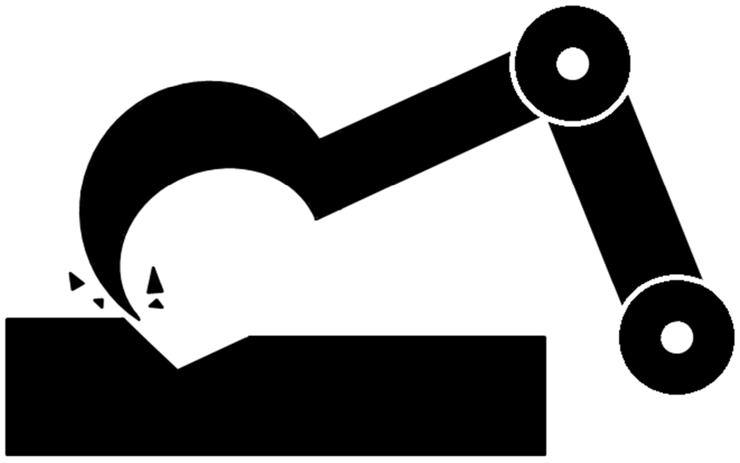
In this Chapter, the proposed workflow has been tested based on data generated through reverse-engineering a mechanical product. The results have shown that capturing and analysing requirements using IBIS-style maps is feasible, and it improves traceability capturing more contextual information than other approaches.

The methodology applied to capture data for this empirical study involved a combination of reverse-engineering, and analysis to extract data. Generating data through reverse-engineering has allowed the proposed workflow to be validated early in the research project. Early understanding of the feasibility of the workflow has encouraged further investigations into other aspects, including its practicality and scalability.

The outcome of the case study in this Chapter can be summarised in three points. First, this case study emulates typical requirement analysis scenarios. The size of the dataset is comparable to a small engineering project. Second, it has been demonstrated that DRed can be used to capture requirements, requirement rationale, and other requirement metadata. Third, it has been demonstrated how requirement traceability can be enabled; this includes both DRed-to-DRed files and DRed-to-other tools (such as QFD in Microsoft Excel and DOORS).

This Chapter contributes a case study to test the feasibility of the proposed workflow that can potentially be used to improve requirement rationale capture and enabling more requirement traceability. In the subsequent Chapters, this workflow shall be validated further for feasibility, practicality, and scalability.

Chapter 6 Evaluation of the workflow by graduate engineers



Chapter outline

This Chapter continues to validate the workflow proposed in Chapter 4 using a real engineering project as a case study. In the case study, the workflow is applied to the phases of requirement capture and analysis of a product developed by the collaborating company. The case study investigates requirement rationale capture and uncovers issues related to enabling requirement traceability across multiple documents. It differs from Chapter 5 in that the data was obtained from an industrial project, whereas Chapter 5 used a theoretical case study for validation. The specific objectives of this Chapter are to demonstrate that the proposed workflow: 1) is feasible when empirically obtained data is used, 2) can be incorporated into any existing requirement analysis practices, and 3) allow more requirement rationales to be captured and can improve requirement traceability.

6.1 Data collection and analysis

The data for this case study consists of requirements that originated from engineers who performed requirement analysis in an engineering project. The data includes project documents and the transcript of an interview with an engineer from the collaborating company.

6.1.1 Project document analysis

The case study presented in this Chapter is based on an engineering project undertaken in industry. This project was selected because the project team, keen to look for ways to improve their design workflow and the quality of documents produced, decided to apply methods for requirement analysis in DRed.

The background of the project was as follows. The project was scheduled for a period of twelve weeks, and was undertaken by a team of four newly joined graduate engineers. All four engineers had received training in the company’s systems engineering best practices, and this project was the first challenge where they could apply their systems engineering knowledge. The project consisted of designing a portable machine to take material samples off large pipes or chimneys in industrial power plants. An existing off-the-shelf machine had been available, but its capabilities were beyond what was required. Hence, it was envisaged that the unit cost could be reduced if a similar machine was designed without the unnecessary features. At the beginning of the project, the team was provided with three requirements. The project team conducted several requirement elicitation activities, including interviewing stakeholders, reusing requirements from the specification of the existing high-cost machine, and brainstorming. Subsequently, the team used various methods to analyse, refine, and elicit requirements.

The requirement analysis documents obtained consist of five requirement sets described in Table 14 and shown in Figure 55. They are an Issue Based Information System (IBIS) map, a Functional Analysis (FA) diagram, a Systemic Textual Analysis (STA) table, a Viewpoint Analysis (VPA) diagram, and a Quality Function Deployment (QFD) table. Each document corresponds to the application of a method and they were applied in the order presented in Table 14.

Table 14 Overview of requirement sets

Requirement set name	Format of requirement set	Size of requirement set
Issue Based Information System	Tree of requirements structured in IBIS (Kunz and Rittel, 1970)	36
Functional Analysis	Tree of parts and functions	58
Systemic Textual Analysis	Table with operational requirements, functional requirements, non-functional system requirements, non-functional performance requirements, and non-functional implementation requirements	24
Viewpoint Analysis	Tree of functional requirements with non-functional requirements represented as adjacent bubbles	56
Quality Function Deployment	Table of requirements following QFD method (Akao, 2004)	39

The five requirement documents were initially analysed to research requirement evolution. Requirement evolution is defined as the *transformations* that requirements are subject to as a result of analysis. Requirement traceability is the ability to *navigate* the evolution of requirements. The analysis involved manually matching requirements that are similar in meaning across pairs of requirement documents.

The analysis of requirement evolution was performed on the requirement documents shown in Figure 55. The data in the five requirement documents was transferred into five columns of requirements in an Excel spreadsheet to allow comparing and contrasting requirements lists to investigate requirement evolution, see Figure 56. Then, the requirements from all the columns were compared to identify matching requirements. Finally, all sets of requirements were categorised into non-functional and functional in order to understand how these two fundamental types evolved. Figure 56 shows the final view of the spreadsheet at the end of the analysis process. Each column present the requirements generated through a requirement document, e.g. column three captures the requirements studied through the Systemic Textual Analysis. Each row of a column holds one requirement statement. The requirements with grey background are non-functional and the requirements with orange background are functional, see Figure 56. If a requirement was present in two columns, then it was stored in the same row indicating that it was also considered when the subsequent requirement document was created.

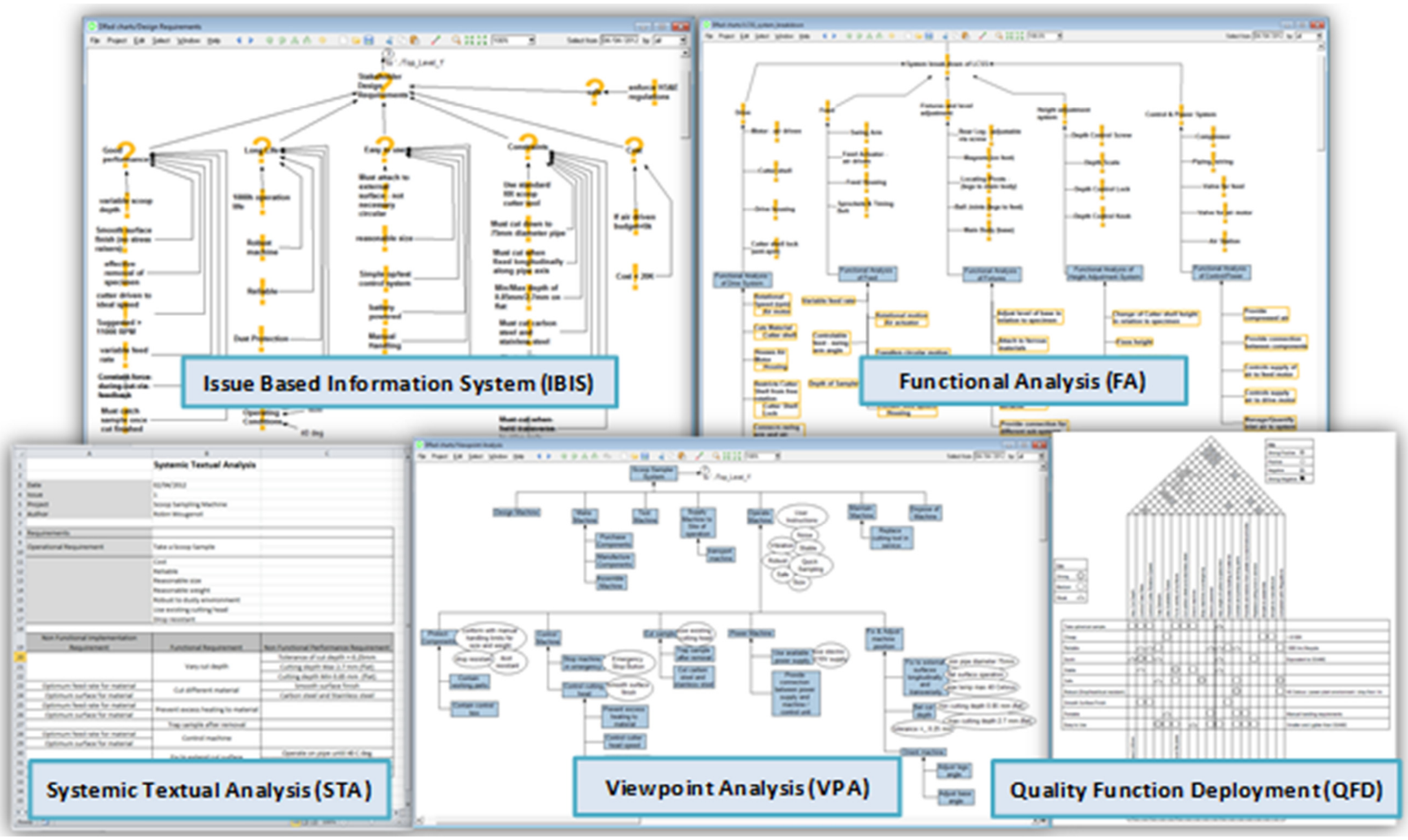

Issue Based Information System (IBIS)
Functional Analysis (FA)
Systemic Textual Analysis (STA)
Viewpoint Analysis (VPA)
Quality Function Deployment (QFD)

Figure 55 Requirement documents produced by the project team (pixelated intentionally due to confidentiality)

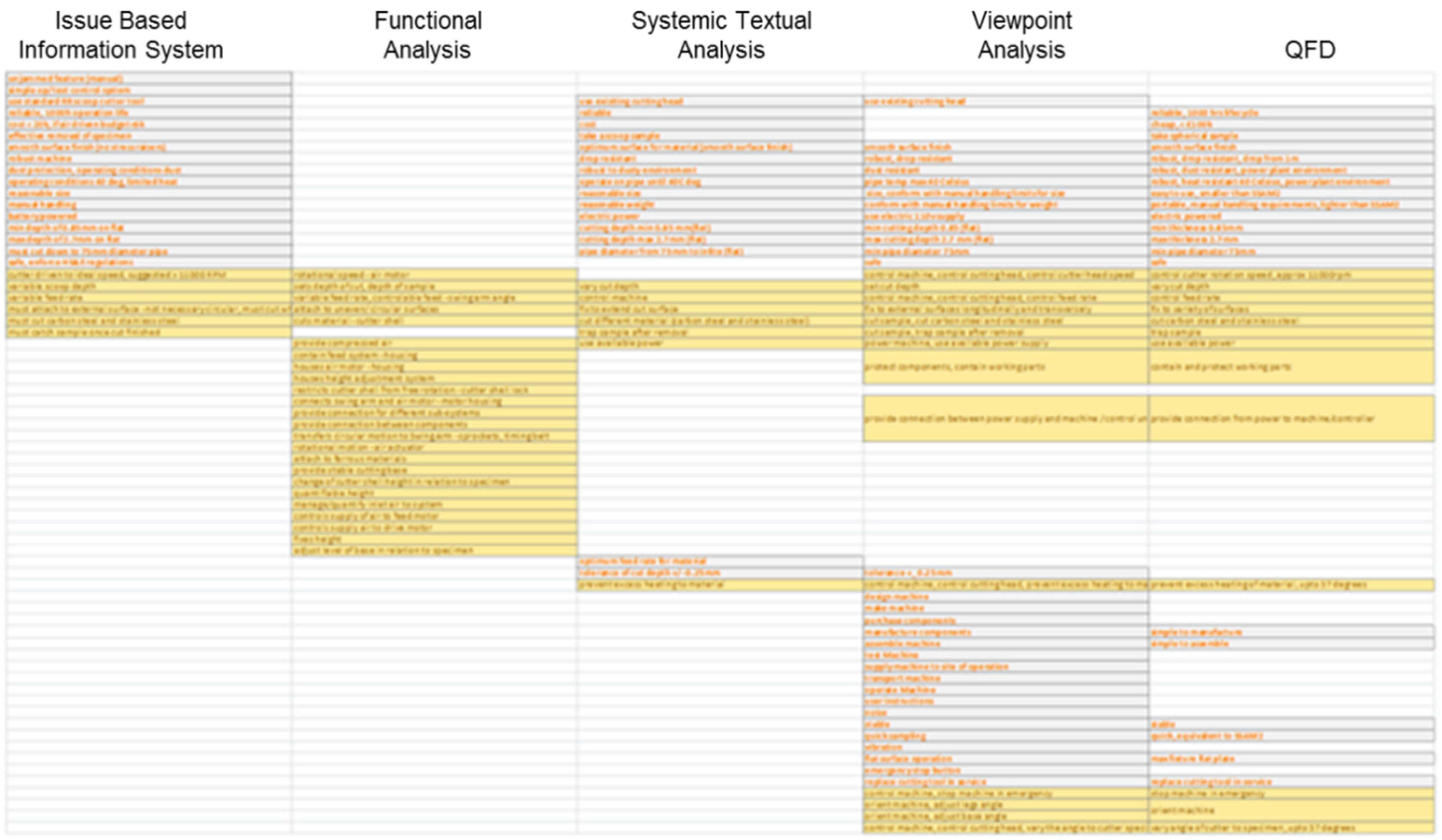

Issue Based Information System
Functional Analysis
Systemic Textual Analysis
Viewpoint Analysis
QFD

Figure 56 Excel spreadsheet showing requirement evolution across the five documents

6.1.2 Interview

Given that a preliminary analysis of the project documents raised doubts on the order in which they were produced and showed that the IBIS document was used without capturing rationale, an interview was conducted with a member of the project team to reconstruct this information. The reconstruction was deemed important to investigate requirement traceability, as well as to investigate how best to facilitate the capture of rationale.

Three people were involved in the interview. Two researchers were the interviewers and an engineer from the collaborating company was the interviewee. The interviewee was a member of the project team who generated the documents shown in Figure 55. During the twelve-week project, the interviewee took a leading role in all the requirement analysis sessions, and was responsible for producing any documentation. The interviewee was a male and he had been working in the company for one year before this project. During his employment, he went on a week-long systems engineering course. This course was approved and recommended by the collaborating company. The course was taught by an independent consultancy company.

The interview was conducted over the phone and the conversation was supported by the use of the project documents previously presented. Notes were taken during the interview. In addition, the entire interview was audio recorded, and later transcribed to backup any information that was not recorded during the interview.

The interview was semi-structured, i.e. a set of driving questions was pre-defined but new questions were asked depending on the evolution of the dialogue. It was divided in two parts. For the first part, the interviewee was asked to confirm the order in which the project requirement documents were constructed. Prior to the interview, the researchers had made a best guess of the order. For the second part, once the order was confirmed, attention was focused on one particular document, the IBIS map shown in Figure 55, which is the first requirement document used to capture the initial set of requirements. Within this document, every requirement in the set was examined together with the interviewee. He was asked to elicit any rationale associated with every requirement being examined. In order to help the interviewee elicit the rationale, prompts were used. The interviewee was asked questions such as: Why was the requirement necessary? Where did it come from? How important is the requirement, and why? How did the requirement arise? Did any debate take place around this requirement?

The data collected during the interview was used in two ways. First, the notes taken during the interview helped reconstruct and justify the order in which requirements documents were produced. Second, the interview transcript was fragmented into atomic requirement rationale statements. These were coded to distinguish whether the statements were supporting or confuting a requirement. Finally, the original IBIS map (shown on the top left corner of Figure 55) was filled in with these requirement rationales. Subsequent data analysis focused on the subject of the rationale statements. Lack of previous categorisations of requirement rationale required to undertake a bottom-up analysis to characterise the subject types.

6.2 Results

The results of the research are presented in four main Sections. The first focuses on the workflow employed by the team studied. The second focuses on how requirement rationale was captured using an interview. The third focuses on requirement evolution. The fourth proposes a method to enable requirement traceability based on the analysis of requirement evolution.

6.2.1 Project workflow reconstruction

The order in which the five requirement documents were created is shown in Figure 57. The order is Issue Based Information System, Functional Analysis, Systemic Textual Analysis, Viewpoint Analysis, and Quality Function Deployment.

The IBIS map is the first requirement set created by the project team. This set of requirements originated from the project brief, which contained the three requirements given by the customer. Subsequently, this set was expanded. The final set was formatted in a tree structure using the Decision Rationale editor (DRed). This file is called "Issue-Based-Information-System (IBIS) map" because the team of engineers to captured requirements using the IBIS derived notation REQUIREMENT-ANSWER-ARGUMENT – a notation that had been taught to them in previous training sessions. However, the requirements captured in this document lacked argument nodes.

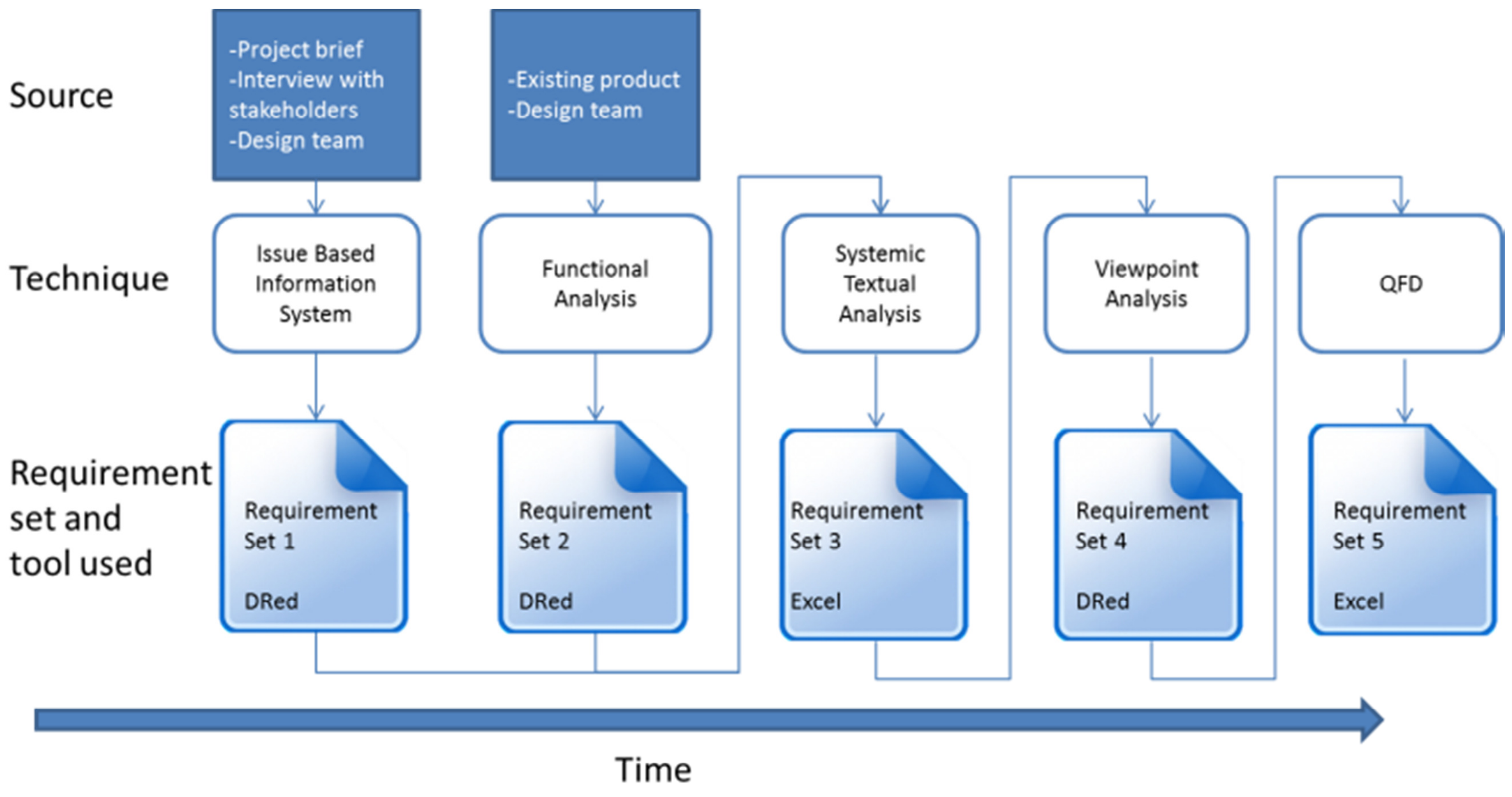


Figure 57 Workflow followed by the design engineering team

The FA diagram is the second document created. This document contains the functional requirements elicited by functional analysis of the existing machine. This exercise was conducted in addition to interview sessions with stakeholders in order to elicit additional requirements. This analysis was also captured using DRed, see Figure 55. The FA diagram has a section showing the decomposition of the parts of the existing machine, and another section showing the decomposition of the functions. A subset of the functions was later converted into functional requirements for the product to be designed as they had not been identified through the interviews with stakeholders. According to the interview with the requirement facilitator of the project team, FA was useful to identify, check and agree the complete functional behaviour and structural feasibility of the desired solution.

The STA table is the third document created. It was produced using the combined requirement lists from IBIS and FA as input. STA is a technique that is mainly used to check for missing requirements (Burge, 2004). STA has three steps: separate and sort identified stakeholder requirements; identify missing requirements; and clarify and refine requirements. STA is generally applied to unstructured requirements in order to categorise the requirements according to the requirement types defined by the Holistic Requirements Model (Burge, 2006), as introduced in Section 2.1. However, in this case the statements had previously been structured in the IBIS map. This means that the application of STA required a complete restructuring of the structured requirements from the IBIS map.

The VPA diagram is the fourth document created. It was produced using the requirements entirely from STA as input. VPA is a technique which facilitates checking the completeness of a set of requirements (Lamsweerde, 2000). A VPA diagram is created in two steps: create a hierarchical tree structure using functional requirements; and attach non-functional requirements as bubbles to the hierarchical tree.

QFD is a technique that can be used to show the evolution of customer requirements to system requirements. However in this case, the QFD was created using the template recommended by BurgeHughesWalsh (Burge, 2007). In their version, the Holistic Requirement Model has been mapped to the customer requirements and system requirements. Customer requirements are substituted by operational requirement and non-functional system requirements. System requirements are substituted by functional requirements, non-functional implementation requirements, and non-functional performance requirements. Using the Holistic Requirement

Model means that the same requirements identified in STA can be readily filled into the QFD. The QFD served the role of facilitating checking the completeness of the set of requirements.

6.2.2 Requirement rationale

Only the first document (the IBIS map) had features to support the capture of requirement rationale. Despite this, the IBIS method implemented in this document was not used as intended as rationales were reconstructed. Figure 58 shows a comparison of the IBIS map before and after rationales were populated based on the interview. Apart from populating requirement rationale, some of the requirements were also updated to reflect changes to the statuses of the requirements. In particular, the requirements which were no longer valid were marked as rejected, and those which were valid were left as open.

The rationale-populated IBIS map in Figure 58 contains thirty-six requirements divided into six groups. On this map, there are also forty-six arguments equally distributed between pro and con arguments. Pro arguments justify the need for the requirements that they are linked to, while con arguments confute it. Apart from two groups, which have very few requirements, the remaining four groups have requirements that are rich in rationale.

A typical requirement and its rationale can be seen in Figure 59. This particular requirement, concerning the method of powering the material sampling machine to be designed, was rejected as the team judged the battery pack as making the design too bulky. It is noteworthy that this requirement has a direct bearing on another requirement requesting conformance to manual handling legislation. Note that the con argument has an attachment showing who captured the argument. This notation aligns with that suggested in (Dai and Aurisicchio, 2012).

Further analysis of the IBIS map found that the team considered and evaluated alternative requirements. An example is shown in Figure 60. As it can be seen, fixed and variable feed rate alternatives were considered, and it can be seen that the rationale about the effectiveness of operation had a priority over that about complexity. This indicates that similarly to design solutions, alternative requirements were generated and carefully assessed.

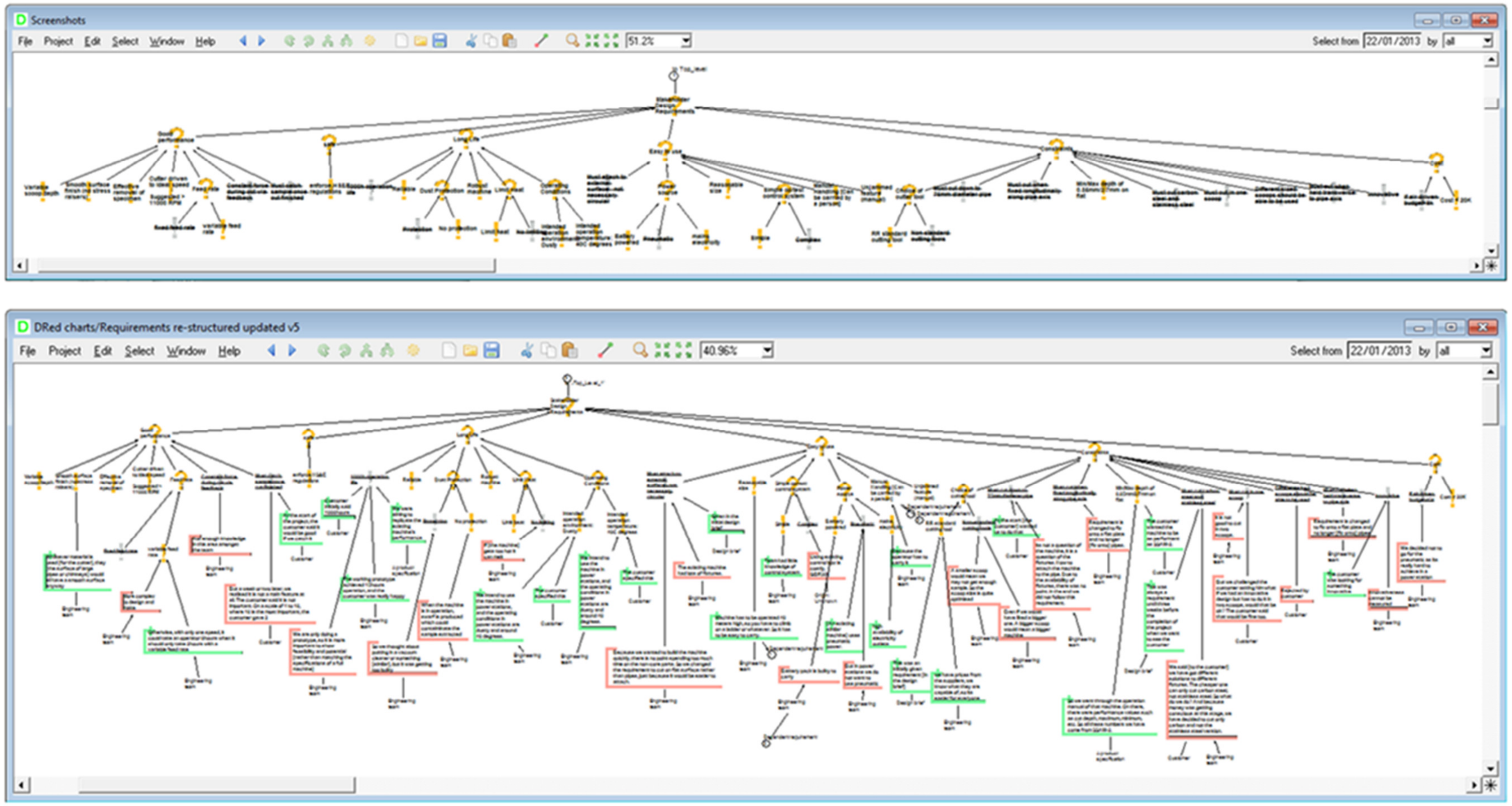

Screenshots
File Project Edit Select Window Help
51.2%
Select from 22/01/2013 by all
DRed charts/Requirements re-structured updated v5
File Project Edit Select Window Help
40.96%
Select from 22/01/2013 by all

Figure 58 IBIS map before (top) and after (bottom) requirement rationale capture

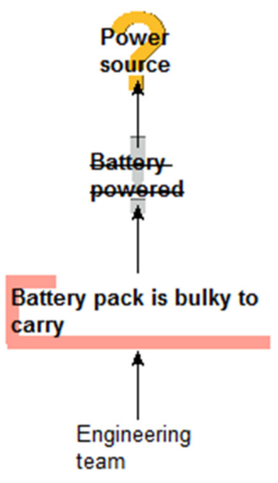


Figure 59 A typical requirement

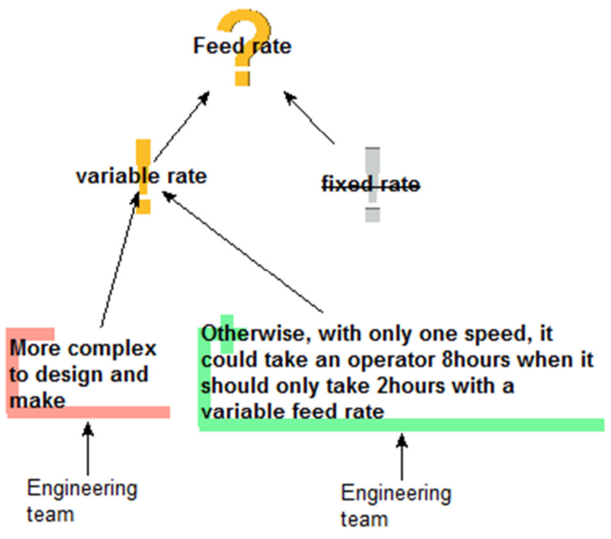

Feed rate
variable rate
~~fixed rate~~
More complex to design and make
Otherwise, with only one speed, it could take an operator 8hours when it should only take 2hours with a variable feed rate
Engineering team
Engineering team

Figure 60 Evaluation of alternative requirements

Analysis was also performed to understand the origin and subject of the rationale captured. The results showed that rationale originated from either the customer or the engineering team. In addition, requirement rationale was distinguished into the nine types shown in Table 15. At a high level the categories of rationales concerned *product design and use*, *precedents* and *project management*. The first set of subjects includes consideration of issues about product *design* and *usage*. An interesting finding is that the rationale of the type *usage* often consists of a description of an underlying engineering problem. For example, the rationale for the requirement to have dust protection describes the problem of swarf, which can cause sample contamination, see Figure 61. The second set focuses on off-the-shelf solutions and benchmarked products. Both types of rationale subjects consider existing solutions relevant to that being designed to shape how requirements are defined. The third set can be categorised as project management issues; it concerns the project scope, team expertise, time and cost. It is noteworthy that the rationale in this set often served the purpose of rejecting requirements. Finally, it can be seen that for some rationale statements the subject was classed as unknown. In this case a requirement was recommended by the customer, but the rationale was not made explicit.

Table 15 Types of rationale by category

Category	Subject	Explanation
Product design and use	**Design**	Rationale that relates to the consideration of challenges expected in the concept formulation of the product.
	Usage	Rationale that relates to the technical functioning of the solution and its use within the context of operation.
Precedents	**Off-the-shelf solutions**	Rationale that relates to the consideration of existing available solutions, and how the availability would affect a requirement.
	Benchmarked products	Rationale that relates to aligning to existing products.
Project management	**Project scope**	Rationale that relates to changes in the boundaries of the project.
	Team expertise	Rationale that relates to the effect of the project team's knowledge and skills.
	Time	Rationale that relates to the consideration of effects on the overall project duration.
	Cost	Rationale that relates to spending considerations.
Other	**Unknown**	Rationale which is unknown.

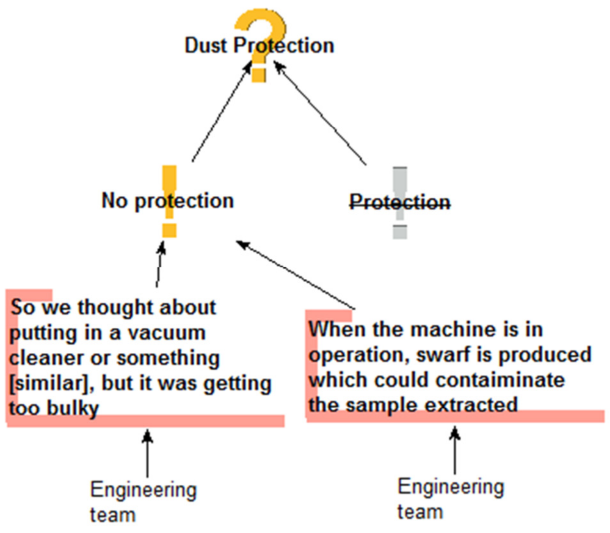

Dust Protection
No protection
~~Protection~~
So we thought about putting in a vacuum cleaner or something [similar], but it was getting too bulky
When the machine is in operation, swarf is produced which could contaiminate the sample extracted
Engineering team
Engineering team

Figure 61 Evaluation of alternative requirements

6.2.3 Requirement evolution

The documents in this project were also analysed to understand how requirements evolved and the impact of this on requirement traceability. Based on the analysis method described in 6.1.1, it was found that four types of operations were carried out on requirements: new introduction, transfer as-is, transfer with refinement, and drop. Figure 62 presents a model illustrating these operations. New requirements were introduced, for example, due to changes in the project scope and resource availability. Existing requirements were transferred as-is, for example, to reflect their recognised validity and suitability to be part of the next analysis stage. Transfer with refinement was found to be needed, for example, to clarify, quantify and decompose requirements as well as to make them conflict-free. Existing requirements were dropped when deemed unnecessary.

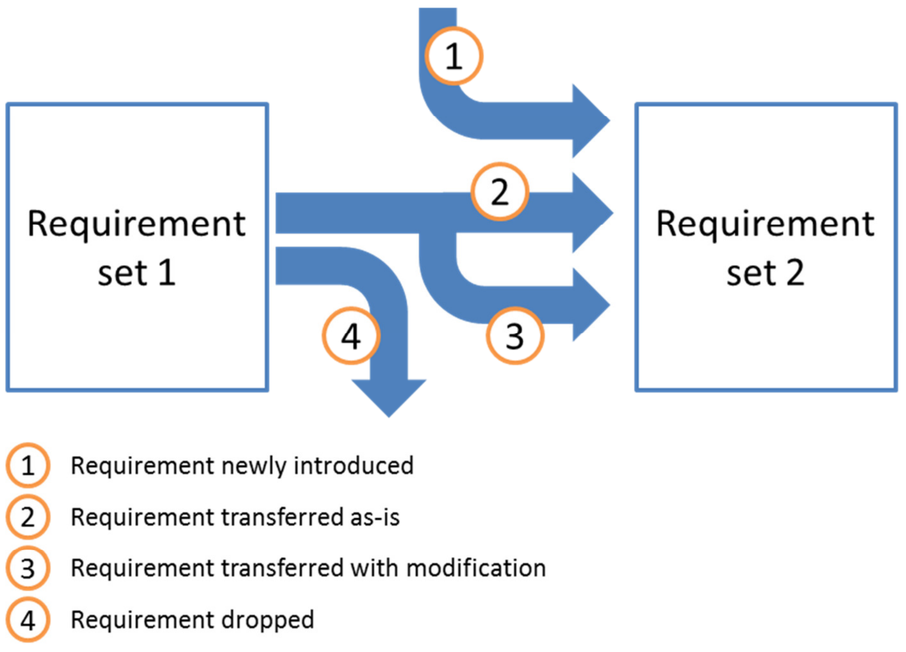
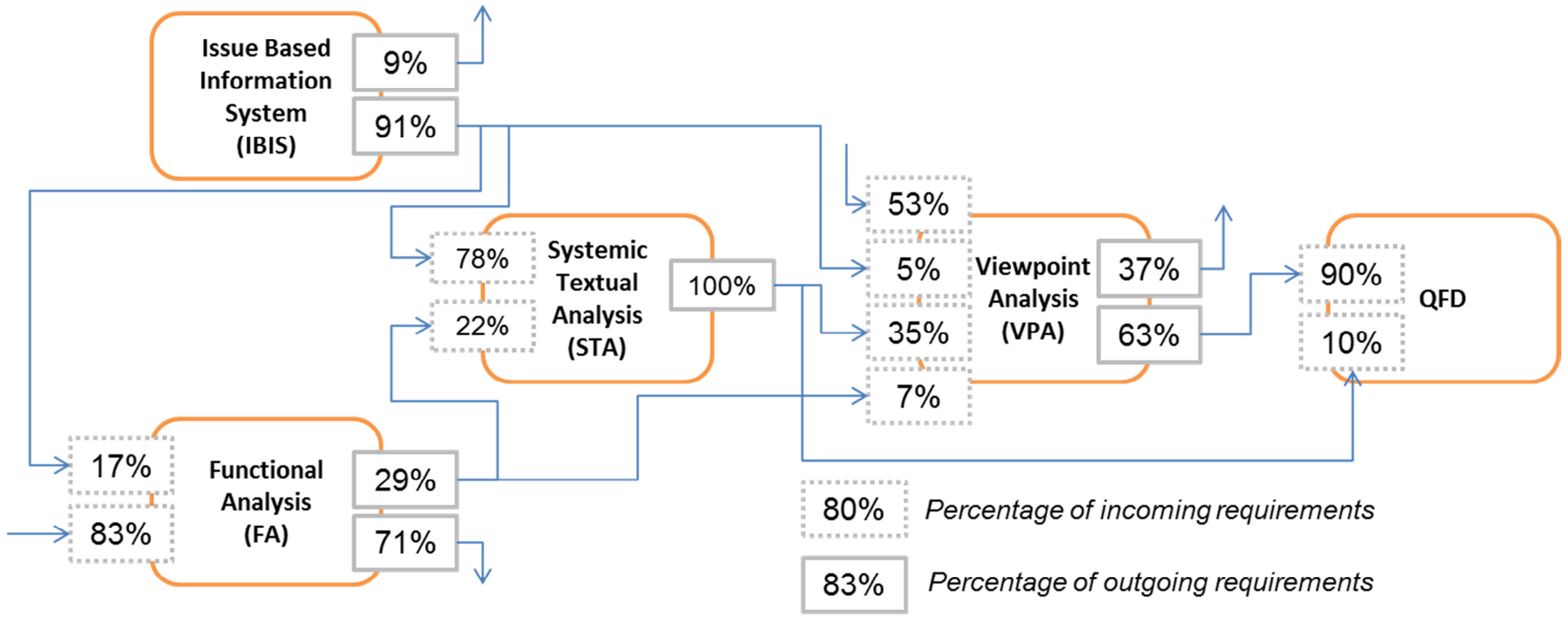

1
Requirement set 1
2
Requirement set 2
4
3
1 Requirement newly introduced
2 Requirement transferred as-is
3 Requirement transferred with modification
4 Requirement dropped

Figure 62 Model showing the evolution of requirements between consecutive documents

Based on the definition of the model of requirement evolution earlier, a quantitative analysis of requirement evolution is presented in Figure 63. The five requirement documents are shown ordered according to their sequence of creation. Each box has percentages to its left, showing proportions of incoming requirements, and percentages to its right, showing proportions of outgoing requirements.



Issue Based Information System (IBIS)
9%
91%
Systemic Textual Analysis (STA)
78%
22%
100%
53%
5%
35%
7%
Viewpoint Analysis (VPA)
37%
63%
90%
10%
QFD
17%
83%
Functional Analysis (FA)
29%
71%
80%
Percentage of incoming requirements
83%
Percentage of outgoing requirements

Figure 63 Diagram showing quantitative aspects of requirement evolution between project documents

In Figure 63 the proportions of transferred requirements are shown by the arrows linking a source to a destination requirement document, e.g. 78% of the requirements in the STA document were transferred from the IBIS document. The proportions of dropped requirements are shown by the arrows with no destination, e.g. 37% of the requirements in the VPA document were dropped and not used in the QFD document. Finally, the proportions of newly introduced requirements are shown by arrows with no origin, e.g. 53% of the requirements in the VPA document were newly introduced.

From Figure 63 three important findings emerged. First, there are always requirements in one document being transferred to another document. Second, there are always requirements being dropped. Third, there are at times newly introduced requirements. These findings are now presented and discussed in the context of the project studied.

From Figure 63 it can be seen that only 17% of the requirements in the IBIS document were also present in the FA document. Little overlap between the IBIS and FA documents confirms that these were two independent sources through which requirements were elicited. IBIS served the purpose of generating a first list of requirements. FA elicits a specific subset that consists of purely functional requirements from an existing product. Figure 63 also shows that the majority of requirements in the STA document were transferred from either the IBIS or FA documents. However, 71% of the requirements identified through the FA document were not carried over indicating that they were judged not suitable. Inspecting these requirements a range of reasons can be hypothesised for why they were left behind including, for example, their dependency on specific architectural design decisions made by the project team. If the rationale for dropping these requirements was captured it would have been easier for stakeholders to interpret the process followed by the project team. After the STA document, the team generated the VPA document, see Figure 63. Interestingly, even though the entirety of the STA requirements are input into the VPA, these requirements only make 35% of the VPA. The newly introduced requirements in the VPA come from its default template. Compared to the STA, the VPA recommends the project team to consider a broader set of design aspects including assembly, testing, and transportation. In addition, the requirements in the VPA document are very detailed including specification at component level. From Figure 63, it can also be seen that 90% of the requirements in the QFD document were transferred from previous work in the VPA document. It is noteworthy that these were only 63% of the VPA, indicating that the VPA is a larger set. The remaining 37% of the VPA requirements that was not transferred to the QFD includes mostly requirements about the broader design aspects previously mentioned, indicating that the QFD contains only requirements related to the product in operation.

Finally two issues important to understand the above model of requirement evolution are noted. The first, not observable from Figure 63, is that compared to the VPA, the QFD contains more quantified statements. A reason for this can be found in the fact that the QFD was used as the final set of requirements upon which further design and development was based. The second is that at times the requirements dropped after the use of a method were picked up again in the method after. For example, 5% of VPA are requirements in IBIS, which were dropped in STA but picked up again. Similarly, 10% of the requirements in QFD are dropped in VPA but picked up again in QFD. This occurrence indicates that sometimes requirements were not continuously considered. This is again a case where capture of design rationale could help clarify why they were dropped.

The operations in the model of requirement evolution are similar to those outlined by Heumesser et al. (2004) including definition of additional requirements, direct translation, and refinement. One difference is that this model describes specifically the transfer of requirements from one set to another on the same abstraction level, whereas Heumesser et al.'s model focuses on the derivation of requirements from a higher level. Another difference is that this model is based on the view that requirements reside in different sets which are generated by different tools, whereas Heumesser et al.'s model assume all requirements reside in the same set and are generated by the same tool. Therefore, it could be argued that this model is a variation of Heumesser et al.'s model that is more suited to requirement analysis when performed with the support of multiple tools.

6.2.4 Requirement traceability: a method to enhance support in DRed

In light of the requirement evolution model defined earlier, it is now worth asking how engineers involved in requirement analysis with multiple methods could be supported in visualising and tracing requirement evolution as well as capturing its rationales. The requirement evolution model has shown four types of evolution. In addition, these four types are indiscriminate between requirements within a document or across documents. This means that a method to enable requirements to be traceable must: a) indicate whether requirement evolution consists of a newly introduced, transferred as-is, transferred with modification, or dropped requirement; and b) tolerate evolution between requirements within a document or across documents.

Leffingwell and Widrig (2012), studying this problem, proposed to add a "traced from" and "traced to" note to every requirement. However, their solution was generic and it does provide any detail of implementation. In this research project, it is proposed that the "trace from" and "trace to" design could be implemented in DRed. DRed already supports mono- and bi-directional hyperlinking between information objects in its own files, and between its files and MS Office files (Word, Excel and PowerPoint) (Bracewell et al, 2007) as well as a form of hyperlinking, known as transclusion, which allows to create linked and navigable copies of information objects (Bracewell et al., 2009).

It is proposed that every requirement node would be made traceable by navigating hyperlinks. To achieve this, the DRed tool would require a small extension to the requirement node. The extension involves adding a "source" field and a "destination" field to a DRed requirement node. The source of a current requirement would be hyperlinked to the requirement where it has evolved from, and the destination of the "evolved from" requirement would be hyperlinked to the current requirement. Figure 64 shows an example of this proposal. In this example, the requirement "shall be battery powered" has evolved into the requirement "shall accept 2xAA batteries". The destination field of the former requirement is hyperlinked to the latter requirement; and the source field of the latter requirement is hyperlinked to the former requirement. The source and destination fields shall be hidden fields so that they will not clutter a requirement DRed map. But these fields can appear once the mouse pointer is hovered over the corresponding requirement node.

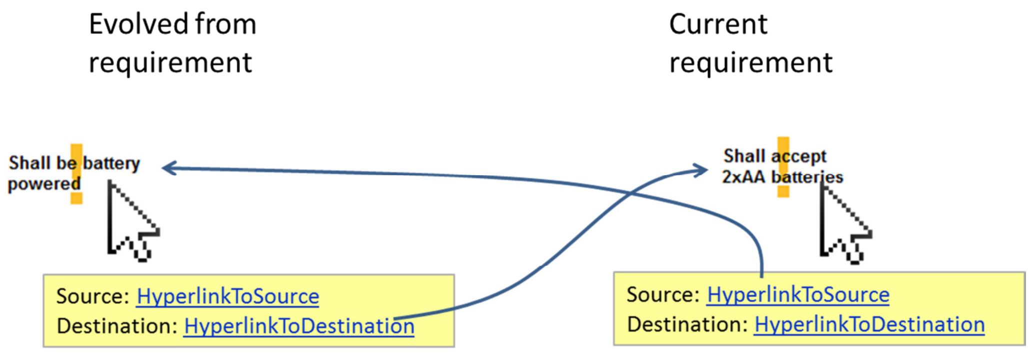
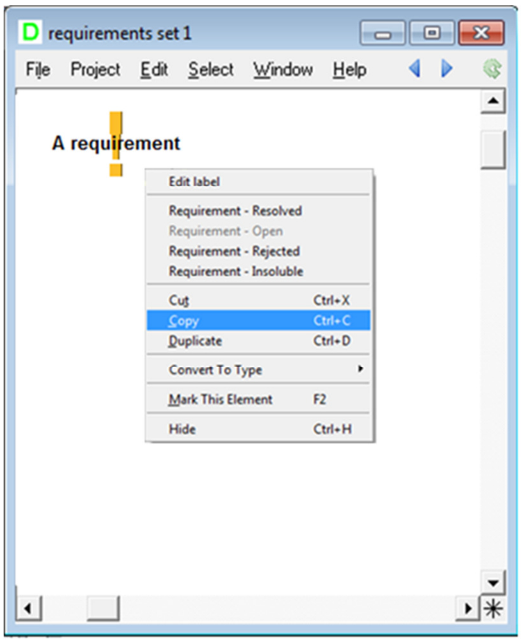

Evolved from requirement
Current requirement
Shall be battery powered
Shall accept 2xAA batteries
Source: HyperlinkToSource
Destination: HyperlinkToDestination
Source: HyperlinkToSource
Destination: HyperlinkToDestination

Figure 64 Proposing a method to enable requirement traceability in DRed

An additional modification to the DRed requirement node could be made to allow the user to create linked requirements conveniently. This modification would enable users to create traceable requirements using the same actions as a copy-paste procedure. For example, a user could right-click on a requirement to copy it, then *paste as evolution* to paste a traceable requirement node. This procedure would result in two requirements that are automatically linked such that the requirement at the source can be navigated to the requirement at the destination, and vice versa. Figure 65 demonstrates this procedure graphically.



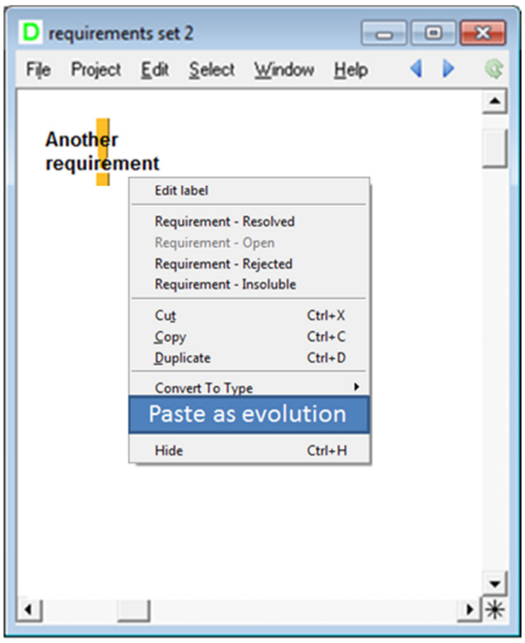


Figure 65 Create traceable requirements as easily as copy-paste

This proposal offers three benefits that contribute to enabling richer requirement traceability. First, it raises the awareness of the need to capture requirement evolution. There was no evidence to suggest that capturing requirement evolution was considered in both the project documents and during the interview in this project. Second, the proposal provides tool support for the process of enabling traceability when requirements evolve. Up to this point, many requirement analysis tools have been discussed; however, very few tools support requirement traceability. Third, the proposal reduces the overhead of linking requirements to a minimum. In tools that do support traceability, such as DOORS, it would be necessary to manually create hyperlinks to enable a source and a destination requirement to be bi-directionally navigable.

6.3 Discussion

This research has presented an application of the proposed workflow to an engineering project in industry which involved analysis of requirements through various system engineering methods. The main results of the research are discussed below.

First, the case study has validated the feasibility of the proposed workflow using data from a real project as well as the practicality in modelling requirements in the IBIS format, see Table

16. Although engineers independently documented requirements in the IBIS format, requirement rationale was not captured during the project. Rather it was reconstructed during the interview. In particular, the interview showed that the project team constructed justifications for their requirements at the time, even though these were not captured. Their justification consisted of both arguments in favour and against the proposed specifications. This suggests that the engineers developed reasons for and against requirements, and it validates the feasibility of using the IBIS notation as a knowledge representation scheme to model requirements. The IBIS notation used in this research (see Figure 55) differs from that proposed in Chapter 5 (summarised by (Dai and Aurisicchio, 2012)), because it employs the requirement element, instead of the answer element, to capture a requirement, and in this way it sets the dependency of a requirement element from an issue element. This notation does not represent an established convention in the collaborating company. Rather, it should be seen as an attempt to identify a useful notation for requirement capture involving use of the requirement element. This element was added to the DRed set only after the issue, answer and argument elements were, and it was initially made a subclass of the issue element. Hence, it can be questioned if linking it to an issue element is the most appropriate notation.

Table 16 Coverage of validation

	Feasibility	Practicality	Scalability
Model requirements in IBIS	✓	✓	✗
Model requirement rationale in IBIS	✓	✗	✗
Enable requirement traceability	✗	✗	✗

Second, the case study has shown that the tool chosen (i.e. DRed) for the proposed workflow is suitable for capturing requirements and metadata. In the case study, DRed was selected by the team, not only to implement the IBIS-based approach to requirement capture and analysis, but also to document function analysis and viewpoint analysis. This indicates that DRed has potential to be developed as a platform to support requirement capture and analysis beyond the IBIS method.

Third, data analysis has provided evidence that rich requirement rationale exists as illustrated from the results of the interview. The interview has allowed forty-six requirement rationales to be elicited for a set of thirty-six requirements that had no rationales before. It also found that alternative requirement options were considered and evaluated by the project team. This indicates that deliberations occurred on requirements as much as it generally happens on solutions. The investigation of the subject of requirement rationale showed that the team carefully considered reasons related to product design and use, precedents and project management. With respect to product design and use, the results showed that during the problem structuring stage consideration was given to concept formulation and the way the product would be operated. Precedents were used to form reasons to have certain features and requirements. Project management considerations such as project scope, team expertise, cost and time were critical to control the project. Often projects begin with an idealised state, but as soon as the design process develops and its factors face the reality of business, trade-offs have to be made, and changing the scope is an example of making a trade-off. This is reinforced by the observation that all scope rationales support the lifting of a constraint. Finally, the existence of rationales with unknown subject, which are simply because the customer said so, shows that customer's preference was not entirely understood. The rationale originating from both the customer and the engineering team suggests that debates took place. It is known that in this as much as in other types of design communication, shared understanding is a critical element to support the development of the design process. Real time documentation of requirement rationale would have probably supported such communication.

Fourth, this investigation identified the issue of lack of awareness to capture requirement rationale and other requirement metadata. Requirements were modelled in IBIS but the requirement set contained only requirements and no metadata, such as rationale and requirement origin. This issue can be resolved by applying the proposed workflow that advocates requirement rationale capture as a key stage in requirement analysis.

Fifth, the analysis of requirement evolution identified and modelled four key operations carried out on requirements when they evolve from one set to another set. The investigation has also shown that a large proportion of requirements that exists on the first requirement set (i.e. IBIS map) continue existing in requirement sets created at a later time. However, in the dataset studied these evolutions can only be traced manually, by individually matching source requirements to destination requirements. This manual process is not practical if the requirement set is large. Enabling requirement traceability is a key feature of the proposed workflow that can enhance requirement analysis practice. In addition, the results of this research have led to the development of a proposal to further extend the proposed workflow. The extension provides a solution towards enabling improved requirement traceability both within the same file (requirement set) and across files.

The methodology for data collection and analysis used in this case study is different to that used in the case study in Chapter 5, even though both case studies are used for the purpose of validating the proposed workflow. In Chapter 5, data was generated through reverse-engineering. In this Chapter, data from a real engineering project was collected and analysed. The methodology used in Chapter 5 was used to illustrate technical feasibility, whereas in this Chapter, the methodology has confirmed the feasibility when the workflow is applied to real-world data. This Chapter also justified that there is a need for the proposed workflow.

The methodology used in this case study was a combination of document analysis and interviewing. Document analysis was useful because it enabled the reconstruction of the workflow used by the project team and provided the data needed to visualise requirement evolution. Document analysis worked well because the researcher had prior knowledge of the systems engineering techniques used in this case study, and had access to the same reference material which the project team used to produce their documents. The phone interview was useful because it allowed requirement rationale to be extracted. One technique used in the interview was particularly effective at helping the interviewee focus. At the start of the interview, the interviewer recommended both the interviewee and the interviewer to have in front of them the DRed map of requirements. This technique allowed the researcher to re-direct the interviewee when he started talking about peripheral issues. This technique helped keep the interview concise such that the interviewee was engaged throughout the process.

6.3.1 Limitations

In terms of validation of the proposed workflow, the methodology used in this Chapter has four limitations.

First, only the feasibility of supporting requirement rationale capture is validated, not the practicality, see Table 16. However, the validation of feasibility is an improvement over the validation provided in Chapter 5. In this Chapter, the proposed workflow was applied to real engineering data, even though it was applied retrospectively. Requirement rationales were captured from the interview, unlike in Chapter 5 where requirements were extrapolated through reverse-engineering.

Second, the proposed workflow has not been validated for scalability, see Table 16. The project in this case study is relatively small when compared to a jet engine which is a typical core business project at the collaborating company. Scalability is an issue that is dealt with in Chapter 8.

Third, requirement traceability support in the proposed workflow has not been validated in this case study, see Table 16. Traceability was not enabled because the engineers had not been made aware of the workflow proposal at the time they were performing requirement analysis. Although traceability support was not investigated, an in-depth investigation was conducted to justify the need to create traceability. This need has been illustrated in this investigation by studying requirement evolution in the requirement data set.

Fourth, the number of data points used for validation is limited as the dataset comprises only one project. Nevertheless, the proposed workflow has already been validated using another project in Chapter 5. It will continue to be validated by other projects in subsequent Chapters.

6.4 Conclusion

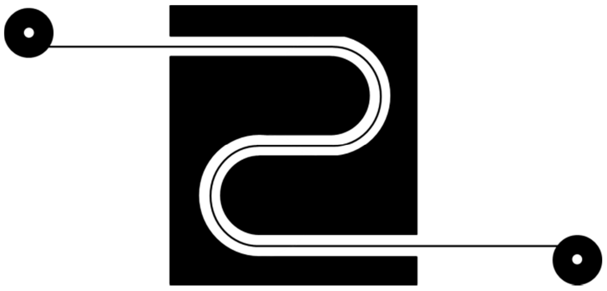
In this Chapter, the workflow for requirement capture and analysis has been tested based on data from a real engineering project in industry. Requirement rationales were elicited through an interview and were found to consist of reasons related to product design and use, precedents and project management. Requirement evolution was also analysed, and the investigation highlighted the problem of requirement loss and ambiguous requirement evolution.

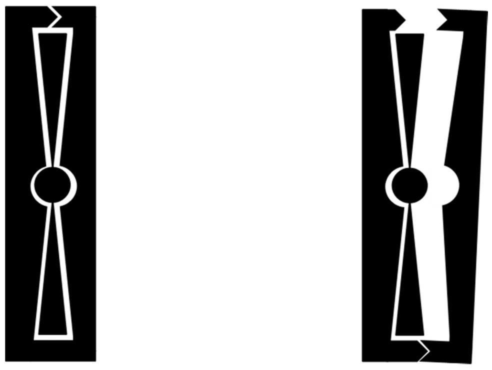
The methods of data collection used in this Chapter include project document analysis and interviewing. Document analysis enabled the researcher to understand requirement evolution. It showed that requirement traceability was not captured and documented. The use of interviewing was also an effective method as it provided the dataset to demonstrate the existence of requirement rationale.

The outcome of the case study in this Chapter can be summarised in five points. First, requirements and requirement rationales can be captured in the IBIS-format defined in the proposed workflow. Second, the results suggest that lack of requirements rationale and traceability information is because they are not captured. Third, the results have shown that four operations typify requirement evolution: requirements newly introduced; requirements transferred as they appeared previously; requirement transferred with modification; and requirements which are no longer used in subsequent analysis. Fourth, a mechanism to visualise and capture requirement evolution has been proposed. The mechanism involves the use of DRed and its hyperlinking features. Fifth, the investigation has confirmed the benefit of designing a requirement analysis workflow that has taken into account integration with other systems engineering tools.

This Chapter makes three contributions. First, using analysis of an engineering project in industry this study has demonstrated the feasibility of capturing requirement rationale with the IBIS format. However, more importantly this work has started to characterise requirement rationale distinguishing its subject types. Most literature mentions that requirement rationale is important, but it neither describes what rationale consists of, nor provides examples of real requirements rationale (Liang et al, 2010; Rooksby et al., 2006; Selvin et al. 2001; Mead, 2008). This research reported here is new because it presents an in-depth investigation into requirement rationale based on a real project in industry. Second, the study advances our understanding of engineering requirements by characterising how they evolve through empirical analysis of a design project in industry and discussing the implication for tool support. It also highlighted the opportunity to provide justification and clarification of the requirement analysis process. Third, this study can be seen as a step towards understanding the feasibility of capturing in real time requirements with *rich rationale*. It has influenced understanding of the concept as well as our ability to communicate it to the engineers that we worked with.

Chapter 7 Further evaluation of the workflow by graduate engineers





Chapter outline

This Chapter continues the validation the workflow proposed in Chapter 4 using two real engineering projects as a case study. In the case study, the proposed workflow has been applied by engineers, as is different from previous case studies which the workflow was applied to pre-existing data. The specific objectives of this Chapter are to demonstrate that the proposed workflow: 1) can be practically applied to engineering projects, 2) can be incorporated into any existing requirement analysis practices, and 3) allow more requirement rationales to be captured and can improve information traceability.

7.1 Data collection and analysis

This Section presents the approach used to validate the proposed workflow and the data analysed. The data upon which the research is based consists of project documents and interviews.

7.1.1 Participatory action research

Participatory action research is a method that asks users to apply a prescribed set of actions, and its principles of participatory research can be found in research by Reich (1994). In this case, the proposed workflow was introduced to the project teams before they began their requirement analysis process.

Two projects were identified as suitable for experimentation with their workflow. First, both projects were low-risk, i.e. core-business operations would be not be impacted by the effects of experimenting with the project workflow. Second, both project teams were not introduced to other company best practices, so they were more likely to accept changes to their natural workflow. All of the engineers in the teams were newly joined graduate engineers. Third, the projects ran in parallel and had a duration of 12 weeks which is short enough for the researcher to follow them closely from the beginning to the end. Fourth, the project teams already had access to the tool that is recommended for use in the proposed workflow, i.e. DRed, and they had received training to use the tool.

The researcher interacted three times with the project teams over the 12 week period. At week 2, the researcher introduced the proposed workflow to the project teams, with the intention to encourage the teams to adopt it. They were left to decide how best to integrate the workflow according to the nature of their projects. It is noteworthy that the researcher deliberately avoided making contact in week 1 in order to allow the project teams to acquaint themselves with their projects. At week 6, the researcher visited the two project teams again. During the visit, the researcher asked the engineers in the team for any issues that they had experienced, and answered questions about the usage of the workflow. At week 12, an interview was conducted to evaluate success in adopting the proposed workflow.

7.1.2 Project document analysis

Documents from both projects were collected for analysis at the end of the tasks. For ease of reference, one project shall be referred to as "Project 1: Cable routing project" and the other project shall be referred to as "Project 2: Fan blade mould cleaning project".

Project 1: Cable routing project

The aim of Project 1 was to design and build multiple test rigs that can reproduce the mechanical operation of control cables for valve actuation in various configurations. Control cables are typically found around the outer layer of an aircraft engine. The cables open and close air valves used for cooling. These commercially available cables feature hundreds of tiny bearing balls in a linear cage. Overtime, typically friction increases and this can affect performance of these cables. The team had the task to design test rigs that can reproduce the behaviour of control cables after 9 years of use within a week, by repeatedly operating the cables.

All requirements in Project 1 have been captured in DRed. In particular, 78 requirements and 42 requirement rationales have been captured across 64 DRed files. It is worth noting that not all the 64 DRed files contain requirements. Most files contain additional information types such as design rationale, design evidence and project management information. These files are distributed across several folders but they are interlinked to each other using DRed's bi-directional tunnel link. Apart from the DRed files, other files were referenced to DRed consisting of supporting documents, such as supplier specifications, emails concerning discussion of requirements, sketches and detailed prototype designs. There are seven folders used to organise the files. Five folders represent particular stages of the project. The folder names include "Project management", "Concepts", "Detailed design", "Component selection", and "Design of experiments". One folder is used to store materials from a past project that the current project builds on, and another one to store a single DRed map that shows an overview of the project. There is no one folder and file that captures all requirements. Instead, requirements are distributed across multiple DRed files, and mostly intertwined with DRed maps that describe conceptual design.

The researcher applied two types of analysis to the documents obtained in this project. First, the researcher created a thumbnail view of DRed maps (shown in Figure 66) to help understand the way requirements were captured. On Figure 66, blue lines represent bi-directional hyperlinks known as tunnels. Tunnels facilitate navigation from one location in a DRed map to another location in the same or different DRed map. Second, all documents were parsed, with reference to the thumbnail view, to identify requirements and requirement rationale from other types of information.

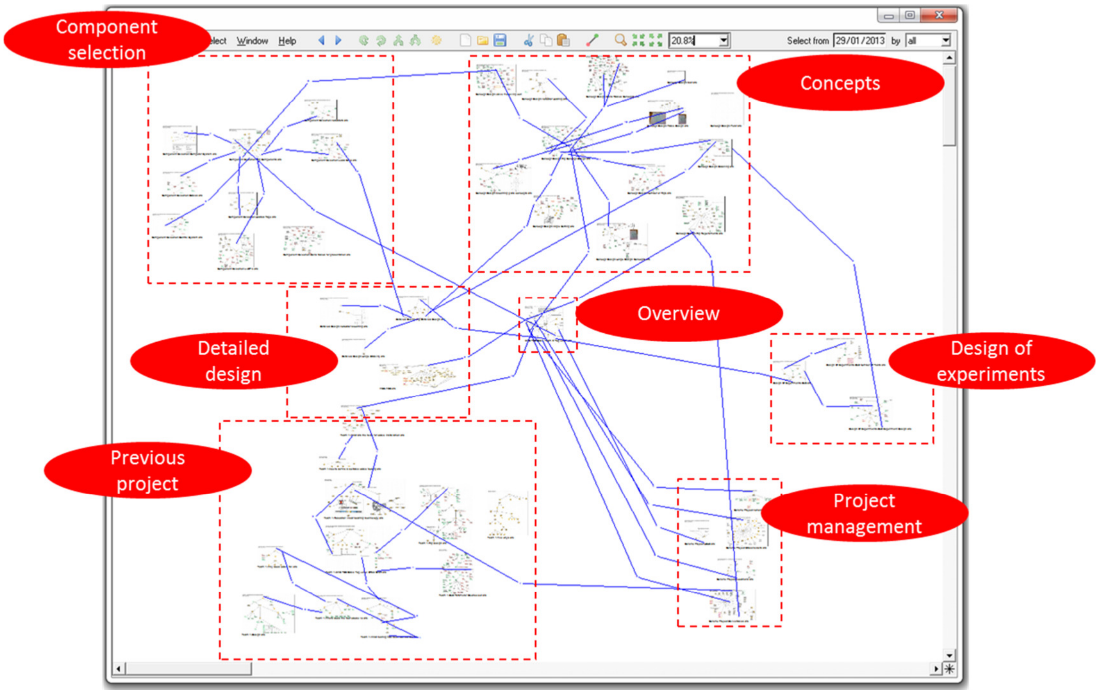


Figure 66 Thumbnail view of DRed maps for Project 1. Each cluster (red outline) is a folder on the hard drive. Blue lines indicate bi-directional tunnel links between DRed maps

Project 2: Fan blade mould cleaning project

The aim of Project 2 was to reduce the risk of turbine blade damage during the procedure to clean moulds that contain the blades by designing a solution to improve the procedure. Fan blades are created by running molten metal into a mould and cooled to room temperature. Then the mould is removed to reveal the blade. The current mould removal procedure requires manual filing which is prone to accidental damage to the fan blade.

The requirements in Project 2 were captured in DRed and Microsoft Excel. In this project, 20 requirements and 32 rationales were captured across two documents. The first document is a DRed map listing requirements and their rationale, see Figure 67. The second document is a Microsoft Excel table that evaluates potential solutions against the list of requirements, see Figure 68.

The researcher applied two types of analysis to the documents obtained in this project. First, the DRed map was rearranged to enhance interpretability. A previously scattered map of requirements was structured as a hierarchical tree. Figure 67 shows the result of the restructuring. Second, the researcher created bi-directional hyperlinks between the requirements on the DRed map and corresponding requirements in the MS Excel document. This was done to illustrate the potential of what could have been done to enable cross-file requirement traceability. Requirements on the left column of Figure 68 are mostly hyperlinked, as indicated by the blue text colour and text underline.

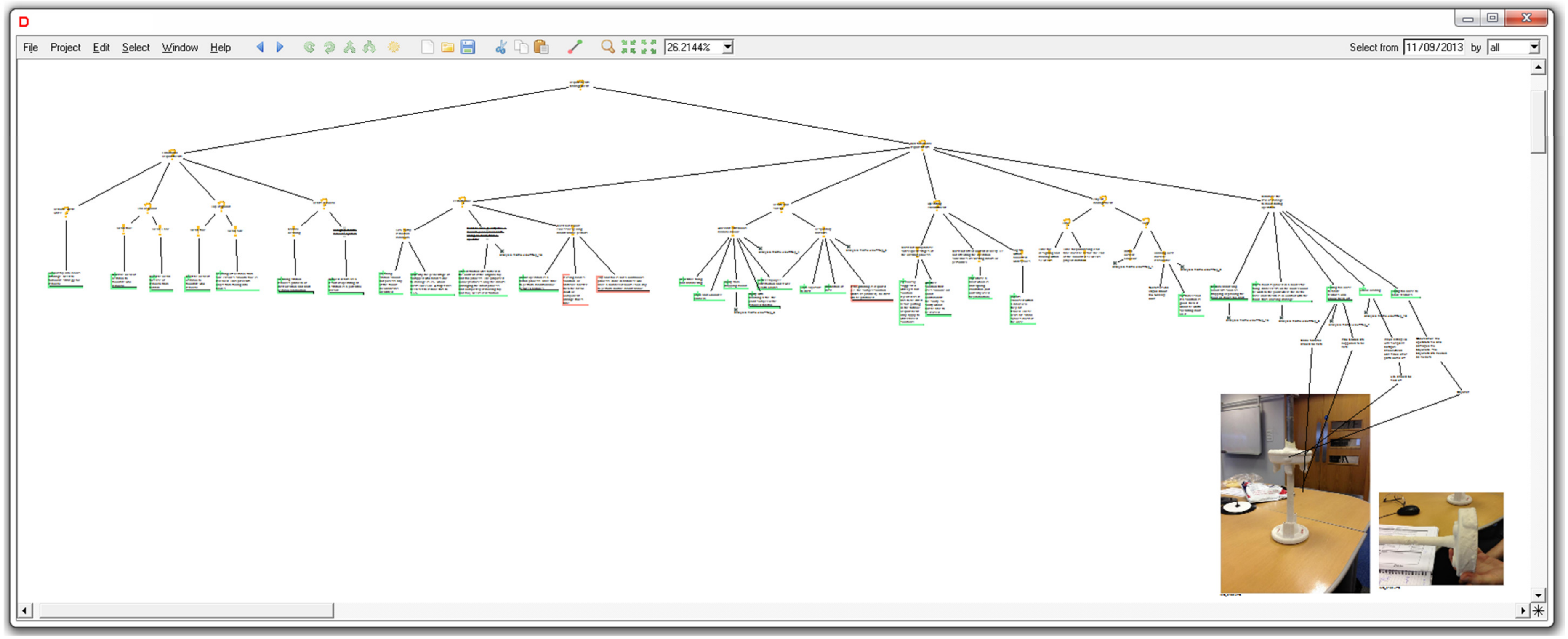


Figure 67 A DRed map from Project 2 showing captured requirements and requirement rationale

		Importance ranking	paint scraper	circular cutting tool	hacksaw	bandsaw	filing till top/ tail falls off	ring pull	air knives	blue tape	teflon	Si based water repellent	Al foil	abrasive blasting	inflatable airbags/ plates	overkill + offcut
Cost	Initial cost	1	5	2	5	4	3	3	3	5	4	5	5	1	4	4
	Running cost	2	5	2	5	2	3	2	3	2	4	2	3	2	2.5	3
	Durability of solution	1	5	3	3	3	3	3	5	5	4	5	5	2	3	3
HS&E	Alleviate strenuous manual	15	2	5	1	5	5	3	5	4	5	3	4	5	5	4
	Injuiry from handling tools and	16	3	4	1.5	3	5	5	5	5	5	5	5	5	3	5
	Dust Exposure	14	4	3	2	3	1	5	5	5	5	5	5	1	5	4
Scraps	Filing too close to bayonets	8	-	-	-	-	-	-		-	-	-				
	Filing too close to blade	13	-	-	-	-	-	-		-	-	-				
	Damage when placing moulds	11	-	-	-	-	-	-		-	-	-				
	Shock loading	11	2	5	2	5	4	1	5	5	5	5	5	4	3	3
	Damage to fragile pips on	3	-	-	-	-	-	-		-	-	-				
	Damage resulting from worn	9	3	5	2	5	4.5	4		5	5	5	5	5	5	3
	Damage from handling by	8	2	4	2	4	4	3	5	5	5	5	5	4.5	5	3
Time	Time for designing/making	5	5	2	5	2	3	4	1	5	4	5	5	4	3	3
	Cycle time	7	4	2.5	1	2	3	5	5	4	5	4	4	2	4	3
	Time for purchasing lead time	6	5	2	5	2	3	3	3	5	4	5	5	1	2	3
Complexity		7	5	2	4	2	3	3	4	5	5	5	5	2	4	3
	Total score:		187	178.5	175.5	179	181.5	181	186	197	197	196	198	175.5	185.5	181
			2nd choice						2nd choice		1st choice				2nd choice	

Figure 68 A matrix used in Project 2 to score potential solutions against requirements

7.1.3 Interviews

At the end of the project, interviews were conducted with both teams in the form of semi-structured discussions. Both project teams sent one representative to the interview. Each representative was interviewed in turn by the researcher. The representatives were the team project managers who were responsible for documentation. A summary of the context of the interview is described in Table 17.

Table 17 Summary of interview

Purpose of the interview			To understand how the teams had executed the process of requirement analysis, what they decided to capture and the reasons why
Participants	**Interviewer**		Researcher
	Interviewee	**Identifier**	Graduate engineer A
		Job title	Junior engineer
		Role in project	Requirement manager responsible for capturing documentation and facilitate requirement elicitation and analysis sessions
		Work experience	Graduate engineer who had joined company for less than a year
		Gender	Male
		Age (approx.)	20s
	Interviewee	**Identifier**	Graduate engineer B
		Job title	Junior engineer
		Experience (in this project)	Requirement manager who is responsible for capturing documentation and facilitate requirement elicitation and analysis sessions
		Experience (general)	Graduate engineer who had joined company for less than a year
		Gender	Male
		Age (approx.)	20s
Duration of interview			1 hour
Method of data capture			Semi-structured interview with notes taking
What questions were asked			What tools did your team use for requirement analysis? Why did your team choose these tools? What format did your team capture requirements in? Did your team use the IBIS format? What is the folder structure used to manage requirement documents? Can you explain what influenced your teams' decision to capture of requirement rationale? Has your team attempted to enable requirement traceability?
Method of data analysis after the interview			The knowledge gained in the interview was analysed and used to complement the understanding emerging from the document analysis

7.2 Results

The results of the research are presented in two Sections. The first focuses on requirement and requirement rationale capture. The second, instead, covers requirement traceability.

7.2.1 Requirement rationale and other metadata

Both teams captured requirement rationale, and both used the DRed tool. A summary of the number of requirements and requirement rationales captured in each project is shown in Table 18. As it can be seen Project 1 had a higher number of requirements but lower requirements to rationale ratio than Project 2. The higher number of requirements in Project 1 is a reflection of its wider scope. In Project 1, in addition to the requirements for the main solution, i.e. the cable-routing rigs, the team also recorded requirements on delivery constraints and requirements on tools that are needed to create the rigs.

Table 18 Summary of requirements and requirement rationales in each project

Project	Number of requirements	Number of requirement rationales	Requirement to rationale ratio
Project 1	78	42	0.54
Project 2	20	32	1.6

In Project 1, there is evidence that the engineers made an effort to capture a wide range of requirement contextual metadata, including requirement rationale, requirement source, debates on conflicting requirements, and clarification of requirements. An example of this can be seen in the DRed map titled "Rig requirements" shown in Figure 69 from the folder "Concepts". This DRed map captures almost a quarter of all the project requirements. It contains requirements for the test rigs to be designed. Each test rig would be able to hold a routing cable in a particular bending configuration. Requirements are captured in the IBIS format that aligns with the format recommended in Section 4.4.2.

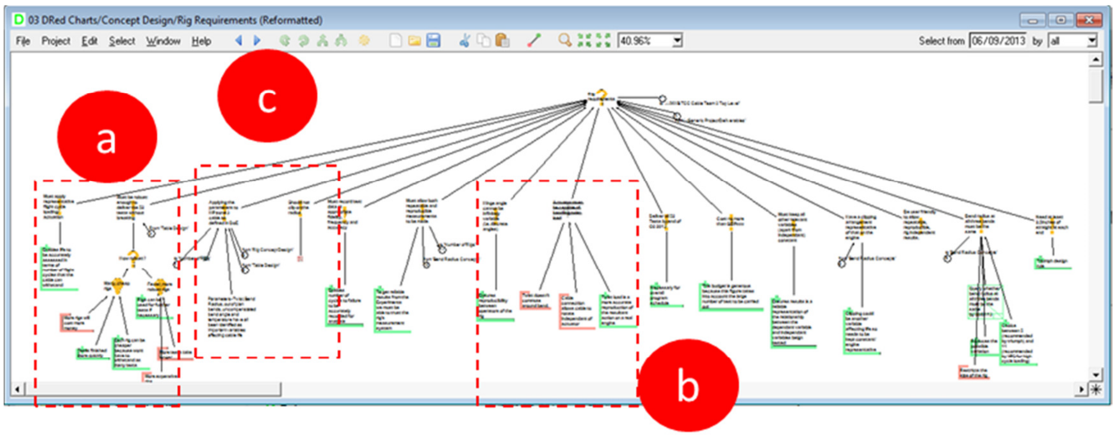

03 DRed Charts/Concept Design/Rig Requirements (Reformatted)
File Project Edit Select Window Help
40.96%
Select from 06/09/2013 by all
a
b
c

Figure 69 Rig requirements DRed map (Project 1)

Figure 70a shows an example of a debate between requirements. The debate was around a requirement regarding the "robustness needed to deliver 52 tests of routing cables" in different configurations. The issue being debated was the choice between aiming for a large number of inexpensive but less reusable rigs or for a smaller number of more expensive but more reusable rigs. From the content of the rationales, it can be clearly seen that there were considerations of cost and about being able to complete the project on time. Unlike the debate presented in Section 5.2.1 which regarded requirement necessity, the debate captured in this case study serves to clarify a requirement. The debate clarifies the definition of robustness, and shows that requirements on cost and time were constantly being considered.

Figure 70b shows an instance of requirement rejection, which is likely to be the result of another debate. The requirement is related to the "capability of the actuator to handle twist loads". As it can be seen the requirement was rejected because the argument in favour was overwhelmed by arguments refuting it. The rationales captured around the requirement help justify the reason as to why the requirement was rejected, i.e. twist loads do not transfer well around a bend even if an actuator can produce twist load.

Figure 70c shows the use of other types of requirement metadata for clarifying requirements. In this case, the requirement "applying the parameters to HP zone 3 cable as defined in DoE" has been elaborated. DoE stands for Design of Experiment and is a specification document which was linked to the requirement node. In particular, the parameters were listed in a text node attached to that requirement. The use of external files for clarification is also evident in the requirement "should not clip on the radius" in Figure 70c. This link points to an external file that contains additional supplier-defined requirements on how cables should be installed. Thus the link helps indicate that the requirement was derived from the supplier-defined requirement specification.

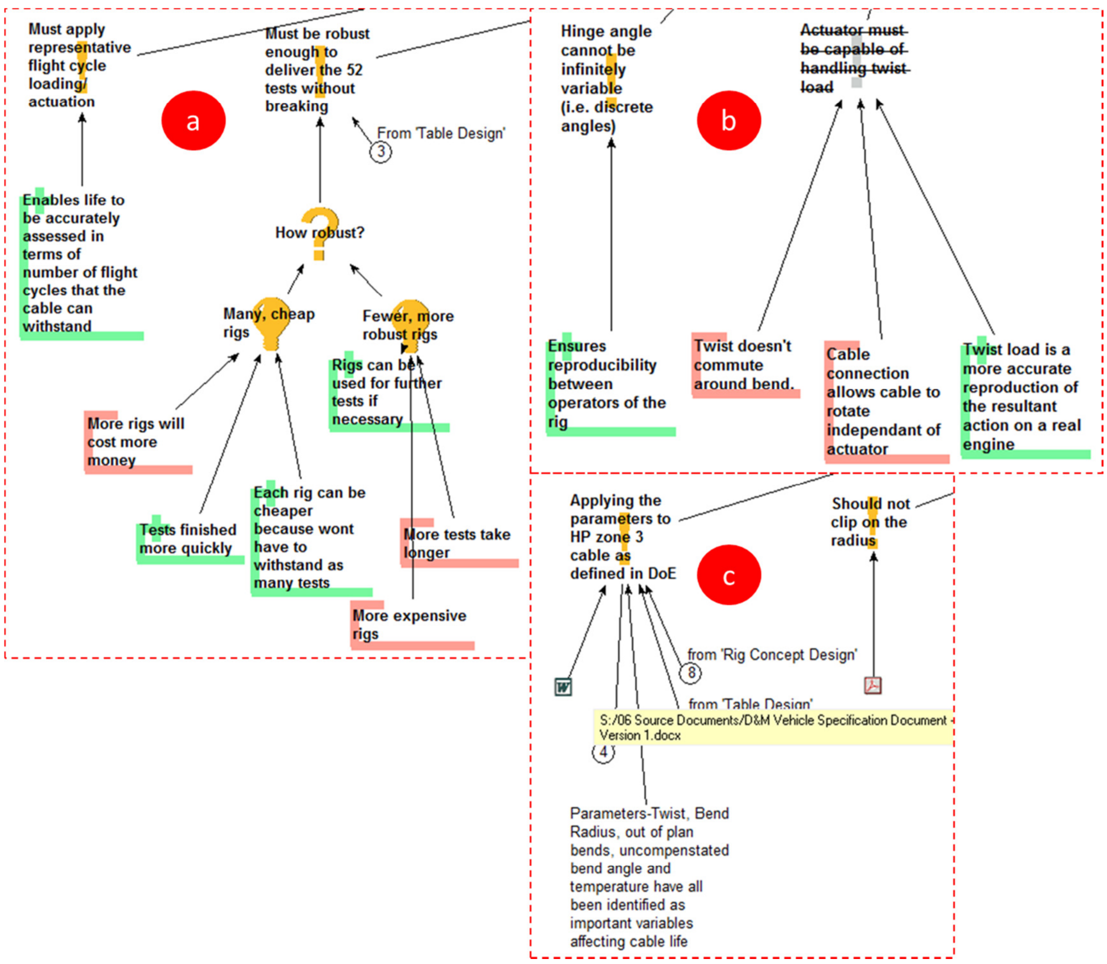

a
Must apply representative flight cycle loading/ actuation
Must be robust enough to deliver the 52 tests without breaking
From 'Table Design'
3
Enables life to be accurately assessed in terms of number of flight cycles that the cable can withstand
How robust?
Many, cheap rigs
Fewer, more robust rigs
Rigs can be used for further tests if necessary
More rigs will cost more money
Tests finished more quickly
Each rig can be cheaper because wont have to withstand as many tests
More tests take longer
More expensive rigs
b
Hinge angle cannot be infinitely variable (i.e. discrete angles)
~~Actuator must be capable of handling twist load~~
Ensures reproducibility between operators of the rig
Twist doesn't commute around bend.
Cable connection allows cable to rotate independant of actuator
Twist load is a more accurate reproduction of the resultant action on a real engine
c
Applying the parameters to HP zone 3 cable as defined in DoE
Should not clip on the radius
from 'Rig Concept Design'
8
from 'Table Design'
S:/06 Source Documents/D&M Vehicle Specification Document Version 1.docx
4
Parameters-Twist, Bend Radius, out of plan bends, uncompenstated bend angle and temperature have all been identified as important variables affecting cable life

Figure 70 Zoomed-in areas of the rig requirements DRed map (Project 1)

The types of requirement rationales found in Figure 70a, Figure 70b, and Figure 70c are in agreement with the list of rationale subject types defined in Section 6.2.2. As discussed earlier, Figure 70a shows rationales related to project delivery and cost. Figure 70b shows rationales with consideration of design and project scope, where a requirement has been rejected due to impracticality. Finally, Figure 70c shows a rationale related to product design and use.

The way requirement rationales are generated in this Chapter is different compared to the case studies presented in Chapter 5 and 6. In this Chapter, requirements were captured during analysis, instead of being re-created retrospectively. This result is particularly valuable to this research, as it demonstrates the practicality of this concept and it helps understand how rationale is generated and captured in real time.

One of the most notable observations from these projects is the evidence of the co-evolution of requirements and solution. Many requirement rationales were created as a result of considering available existing solutions. For instance, in Project 1, there was a function requirement on the solution that it must allow multiple components to be joined. The project team elicited more detailed requirements and rationale justifications by evaluating existing off-the-shelf solutions, mostly involving the use of tape for joining. This process is shown in Figure 71 where at the top requirements are elicited and justified, and at the bottom solutions are developed and rationalised. In particular, additional non-functional system requirements were elicited such as the tape must be thin, temperature-resistant, and short-lead time; and undesirable properties such as being too wide and too expensive. Furthermore, the team were able to identify five rationales for these requirements.

The use of existing solutions to elicit requirement rationale is also evident in Project 2. In Project 2, instead of considering existing off-the-shelf solutions, the team considered the current solutions in use, and evaluated them to elicit requirements and requirement rationales. Table 19 shows three instances of such observations. Project 2 was aiming to improve an existing manual process of removing moulds from fan blades.

Based on the interview data, the graduate engineers found requirement rationale beneficial. One interviewee claimed that the capture of rationale helped the team review requirements, and check for errors. Another interviewee claimed that the act of capturing rationale made his team to reflect deeper about the requirements, and evaluate more carefully whether a requirement is required. They also found that they took more care to word requirements in a way that requirement statements are easy to understand.

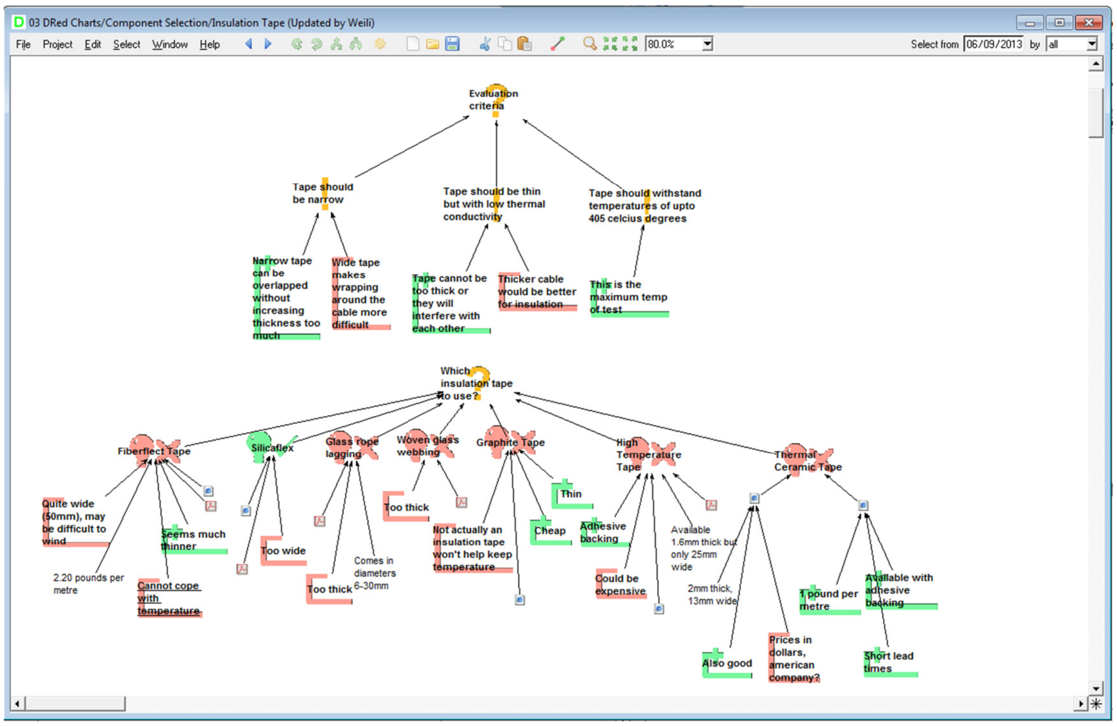

03 DRed Charts/Component Selection/Insulation Tape (Updated by Weili)
File Project Edit Select Window Help
80.0%
Select from 06/09/2013 by all
Evaluation criteria
Tape should be narrow
Tape should be thin but with low thermal conductivity
Tape should withstand temperatures of upto 405 celcius degrees
Narrow tape can be overlapped without increasing thickness too much
Wide tape makes wrapping around the cable more difficult
Tape cannot be too thick or they will interfere with each other
Thicker cable would be better for insulation
This is the maximum temp of test
Which insulation tape to use?
Fiberflect Tape
Silicaflex
Glass rope lagging
Woven glass webbing
Graphite Tape
High Temperature Tape
Thermal Ceramic Tape
Quite wide (50mm), may be difficult to wind
2.20 pounds per metre
Seems much thinner
Cannot cope with temperature
Too wide
Too thick
Comes in diameters 6-30mm
Too thick
Not actually an insulation tape won't help keep temperature
Thin
Cheap
Adhesive backing
Could be expensive
Available 1.6mm thick but only 25mm wide
2mm thick, 13mm wide
Also good
Prices in dollars, american company?
1 pound per metre
Available with adhesive backing
Short lead times

Figure 71 Evidence of requirements generated (top half) during concept exploration and solution evaluation (bottom half) (Project 1)

Table 19 Conversion of known problems into requirements and rationale (Project 2)

Existing solution	Requirement	Rationale
Revealing the fan blade by filing the mould	Prevent filing too close to the blades	Filing too close is very likely to cause critical damage to the blade
Loading and unloading moulds manually	Minimise shock loading	Moulds, and the content they holds, can become damaged if too much shock is applied
Clearing away mould chippings by brushing	Minimise dust produced during mould removal and cleaning process	Filing dust is a health and safety risk

7.2.2 Requirement traceability

Both projects provided evidence of the high potential of enabling requirement traceability. In Project 1, the use of DRed hyperlinking was observed. However they were not used for individual requirement traceability purposes, but for supporting navigation between requirement documents. The kind of hyperlink mechanism observed is the same type of hyperlink used to enable requirement traceability. These include bi-directional *DRed tunnel links*, and *DRed file nodes* that can store any type of hyperlink. Both the use of tunnels and DRed file nodes has been observed in use. Figure 70c shows two DRed tunnel links (represented as a digit in a circle) that connect the requirement shown to related requirements in other DRed maps. The tunnel links makes it easy to navigate from requirements to related requirements. Figure 70c also shows the use of two file nodes (represented by a Microsoft Word icon and an Adobe Reader icon) that stores hyperlinks to external specification documents that elaborates on the given requirement. In Project 2, it was observed that cross-file traceability can be enabled with minimal effort for some requirement documents, as currently used tools already support it. Project 2 consists of two sets of requirements of which one set is an evolution of another. Therefore, most of the requirements overlap between the two sets. The researcher made an attempt to hyperlink requirements that overlap in meaning. One set of the requirements is captured in a DRed file (Figure 67) the other set in Microsoft Excel (Figure 68). According to the interview with the project team, the DRed file was created first and subsequently used to create the Excel file. A bi-directional link is created for every matching pair of requirements using the same technique for linking requirements in DRed and requirements in QFD in Section 5.2.2. Any requirement in DRed is appended with a file node that is hyperlinked to the Excel file and corresponding Excel cell that stores the requirement. This cell is hyperlinked to the corresponding requirement in DRed using the destination DRed node's globally unique ID. Thus repeatedly applying this operation of linking from DRed to Excel, and Excel to DRed creates bi-directionally hyperlinked requirements between the two sets.

7.3 Discussion

This research has presented an application of the proposed workflow to two engineering projects by graduate engineers in industry. The main results of the research are discussed below.

First, the research has demonstrated that it is both feasible and practical to capture requirements along with rationale and other types of metadata, see Table 20. The high percentage of requirement to rationale seen in the documents, autonomously created by the graduate engineers, shows that the workflow proposed in Chapter 4 has been accepted. In both project interviews it was noted that the action of eliciting requirement rationale was beneficial to the requirement analysis process.

Table 20 Coverage of validation

	Feasibility	Practicality	Scalability
Model requirements in IBIS	✓	✓	✗
Model requirement rationale in IBIS	✓	✓	✗
Enable requirement traceability	✓	✓	✗

Second, the results in this Chapter reinforce the finding in Chapter 6 that DRed is a suitable tool for implementing the proposed workflow. Both project teams used DRed to capture requirements and other requirement metadata. In addition, the team working on Project 1 has documented their information predominantly in the form of DRed files that are interlinked using hyperlinking features in DRed.

Third, it has been shown that enabling cross-file requirement traceability is possible. The team working on Project 1 has made use of DRed's traceability features, while the team on Project 2 has produced two sets of requirements which are readily linkable. The reason why they were not linked is probably due to the fact that the project team was either not aware of the necessity to enable traceability or insufficiently familiar with the technical procedures to create links between DRed and Microsoft Excel. The former argument is deemed more likely, as team on Project 1 has captured all of their requirements in DRed and have demonstrated their knowledge of DRed's hyperlinking features. Therefore, they could have enabled traceable requirements had they been aware of the necessity to do so.

The methodology for data collection and analysis used in this Chapter includes participatory action research which has not been used in previous investigations. The use of participatory action research as a method to investigate tool use and acceptance has previously been used in a similar research in (Aurisicchio and Bracewell, 2013). Participatory action research allows the proposed workflow to be tested empirically; whereas in Chapter 6, potential improvements to an existing workflow were identified but not tested empirically. Similarly, compared to Chapter 5, the data generated in this Chapter is closer to reality as the proposed workflow has been applied by engineers rather than the researcher. In this

Chapter, two other data collection methodologies have been used: document analysis and interviews. Document analysis helped make sense of the reasons hindering use of the workflow. Interviews, particularly mid-project, helped prompt participants to engage in workflow use.

7.3.1 Limitations

The research presented in this Chapter has three limitations. First, the workflow was tested on case studies containing less than 100 requirements, which is small in comparison to the number of requirements for large projects such as a turbine engine. Second, the dataset is composed of only two projects. It is difficult to determine whether areas of commonality (such as the observation of requirement-solution co-evolution) between the two projects apply to similar mid-sized projects. Also, the level of adoption of the proposed workflow could be affected by whether the users are familiar with the DRed's features. Third, the case study participants had limited design experience, so their behaviour may not be representative of experienced line engineers. However, it can be counter-argued that, since the roll out of the proposed workflow has yet to be established, newly joined engineers may be the target audience for the proposed workflow, in which case inexperience would not be a limitation.

7.4 Conclusion

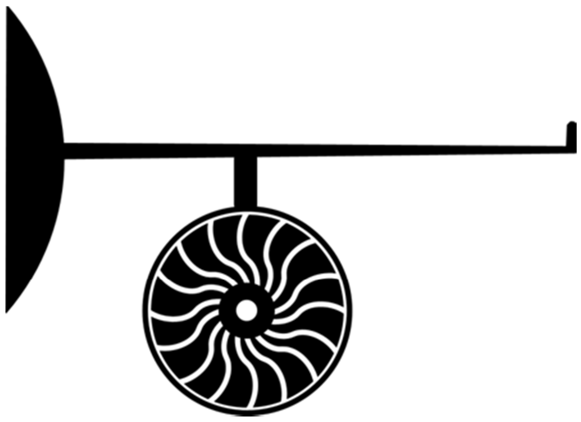
In this Chapter, the proposed workflow has been tested by graduate engineers in industry. It has been observed that requirement rationales were captured when engineers were left autonomously to create requirement documents. Although individual requirement traceability had not been enabled by engineers, it was shown that requirement traceability could have been added to existing requirements with minimum effort.

The methodology applied in this Chapter involved introducing the proposed workflow to two groups of engineers at the beginning of their projects, and analysing what they have captured at the end of the project. Overall, this methodology has been effective. This is shown by the high ratio of requirement to requirement rationale, and the popular uptake of the DRed tool as recommended in the proposed workflow.

The outcome of the case study in this Chapter can be summarised in three points. First, it is practical to capture requirements and requirement metadata in the IBIS-format defined in the proposed workflow. Second, it is practical to use the DRed tool to capture requirements and requirement metadata. Third, cross-document requirement traceability was readily achievable, although not implemented by participating engineers.

This Chapter makes a contribution to the overall research project in that it reaffirms that features of the proposed workflow are practical to implement. In next Chapters, this workflow shall be validated further for scalability.

Chapter 8 Evaluation of the workflow on a whole-engine development programme



Chapter outline

This Chapter validates the workflow proposed in Chapter 4 using a case study based on a whole-engine development programme. This large scale programme has a complexity that is many orders of magnitude greater than the projects in the previously presented case studies. The specific objectives of this Chapter are to demonstrate that the proposed workflow: 1) is scalable to a complex project, 2) can be incorporated into the company general practice, and 3) allow more requirement rationales to be captured and can improve information traceability.

8.1 Data collection and analysis

This Section presents the approach used to validate the proposed workflow and the data analysed. The research is based on data collected through project documents analysis, an interview to reconstruct the *existing* workflow, a questionnaire to validate the *improved* workflow, and an interview to reconstruct requirement rationales.

Definition of terms:

Proposed workflow: The workflow proposed in Section 4.4, which is validated by case studies in Chapters 5, 6, 7, and 8.

Existing workflow: The workflow used to analyse requirements in the existing engine project in this chapter.

Improved workflow: A workflow that is defined in this chapter through identifying improvements in the *existing* workflow and features of the *proposed* workflow.

8.1.1 Project document analysis

Document analysis consisted of researching requirement documents related to an engine development programme for a family of long range jet airliners. The documents, listed in Table 21, were obtained from the systems engineering community within the collaborating company. The analysis focused on reconstructing the existing workflow and identifying opportunities to enhance it. Based on the analysis, a set of recommendations to improve the existing workflow was developed. Features defined in the *proposed* workflow in Section 4.4 also contributed to the making of the recommendations. The recommendations define a new *improved* workflow that includes improvements for requirement metadata capture and requirement traceability.

Table 21 Overview of documents analysed

Document name	Document file format	Description of document
Viewpoint Analysis (VPA)	Microsoft Visio	• Pages: 1 • Structure: tree • Content: • 7 operational requirements • 38 functional requirements that are children of the operational requirements
Functional Flow Diagram (FFD)	Microsoft Visio	• Pages: 3 • Structure: directed graph (one per page consisting of nodes and directed edges) • Content: • 7 operational requirements • 10 functional requirements (a subset of VPA functional requirements) • 39 components • 189 uni-directional edges describing actions
Systemic Textual Analysis (STA)	Microsoft Excel	• Pages: 2 • Structure: table (one per page); o table 1 consists of non-functional systems requirements o table 2 consists of operational, non-functional implementation, functional, and non-functional performance requirements • Content: • 7 operational requirements (same as VPA and FFD) • 90 functional requirements decomposed from functional requirements on VPA) • 14 non-functional implementation requirements • 130 non-functional performance requirements • 109 non-functional system requirements
Product Requirement Document (PRD)	PDF extracted from IBM DOORS	• Pages: over 500 • Structure: tree (visualised by means of headings and sub-headings) • Content: • 14% functional requirements • 68% non-functional system requirements • 18% non-functional performance requirements
Presentation about the usage of other documents	Microsoft PowerPoint	• Pages: 15 • Structure: PowerPoint slides • Content: • Bullet points and diagrams to explain the workflow that was used to analyse requirements

8.1.2 Interview to reconstruct the existing workflow

To support the research to reconstruct the existing workflow, an interview was conducted with two systems engineers, see Table 22. They were responsible for ensuring that engineers in the collaborating company follow a common process to contribute requirements and, where necessary, provide guidance on the use of specific systems engineering tools.

Table 22 Context of the interview to reconstruct the existing workflow

Purpose of the interview			Reconstruct the existing workflow used to analyse requirements in the engine development programme under investigation
Participants	**Interviewer**		Researcher
	Interviewee	**Identifier**	System Engineer A
		Job title	Project Systems Engineer/Whole Engine Design Engineer
		Role in project	Requirement manager for this engine development programme with responsibility to update the requirement repository and facilitate requirement elicitation and analysis sessions
		Experience	Experienced in integrating diverse multi-functional systems on major engineering projects
		Gender	Male
		Age (approx.)	30s
	Interviewee	**Identifier**	System Engineer B
		Job title	Project Systems Engineer
		Role in project	Requirement manager for another engine development programme
		Experience	Experienced in leading improvement in systems engineering capability. Responsible for defining and executing the systems engineering strategy including requirements management
		Gender	Male
		Age (approx.)	20s
Duration of interview			1 hour
Method of data capture			Semi-structured interview with notes taking
What questions were asked			1. What methods and tools were used in this project to analyse requirements? 2. What is the order of usage of the methods and tools? 3. What were the roles of each method and tool? 4. Can you give examples of how requirements are analysed by each method or tool? 5. How familiar are engineers with these methods and tools?
Method of data analysis after the interview			Text analysis and workflow modelling

8.1.3 Questionnaire to validate the improved workflow

To validate the improved workflow a questionnaire was conducted with the same two system engineers. Prior to handing out the questionnaire, the researcher demonstrated the improved workflow to the participants. Then, both systems engineers were asked to evaluate the improved workflow. The context of the questionnaire is described in Table 23.

Table 23 Context of the questionnaire to validate the improved workflow

Purpose of the questionnaire	Validate the improved workflow
Respondent 1	Identical to System Engineer A in Table 22
Respondent 2	Identical to System Engineer B in Table 22
Time available for questionnaire	1 hour
Method of data capture	Structured questionnaire (see questionnaire format and completed questionnaire at Appendix 2)
What questions were asked	6. How important is it to capture rationale at the time when requirements are defined? 7. How important is it to capture contextual metadata (e.g. the source of a requirement, or the person who was responsible for defining a requirement) at the time when requirements are defined? 8. How valid is the proposal to capture rationale and contextual metadata using DRed together with the other System Engineering methods (Function Flow Diagram, Viewpoint Analysis, Systemic Textual Analysis, DOORS)? 9. How feasible is the proposal to capture requirements informally using DRed? 10. How useful is it to link requirements and make them traceable? 11. How valid is the proposal to link requirements and make them traceable using DRed together with the other System Engineering methods? 12. How feasible is the proposal to link requirements using DRed? 13. Have you seen or used similar proposals to enable the capture of rationale and the link and traceability of requirements? If so, please describe what they were and explain if you have found them effective? 14. Do you see ways in which this proposal could be improved? (Questions are elaborated with clarifying information in the actual questionnaire, see Appendix 2)
Method of data analysis after the questionnaire	Questionnaire responses analysis

8.1.4 Interview to reconstruct requirement rationales

In order to validate the requirement rationale capture aspect of the improved workflow, an interview was scheduled with a line engineer who was involved in the requirements specification for the engine project under study. One of the main features in the improved workflow is the use of DRed to capture requirements and requirement rationales. The purpose of this interview was to instantiate a specific example of requirement rationales, see Table 24. Although these requirement rationales were reconstructed retrospectively the session was designed to mimic a real scenario. In the session, which took place at the desk of the line engineer, he was asked to recall requirements that he had contributed to the project, and discuss rationales about those requirements. At the same time the interviewer captured live the requirements and requirement rationale using DRed. The requirements would all relate to a sub-system of the engine project in which the line engineer is a specialist. The sub-system has an operational requirement of preventing ice formation on the engine nacelle.

Table 24 Context of the interview to reconstruct requirement rationales

Purpose of the interview			Validate the workflow by instantiating it with an example of requirement rationales
Participants	**Interviewer**		Researcher
	Interviewee	**Identifier**	Line engineer
		Job title	Engineer
		Role in project	The engineer is a specialist in engine de-icing systems. He is responsible for defining and negotiating requirements with customers. He has contributed to a subset of requirements in this engine project
		Experience	Unknown
		Gender	Male
		Age (approx.)	30s
	Interviewee	**Identifier**	System engineer
		Job title	Project Systems Engineer/Whole Engine Design Engineer
		Role in project	Requirement manager for this engine project. He is responsible for updating the requirement repository. He also facilitates requirement elicitation and analysis session.
		Experience	Experienced in integrating diverse multi-functional systems on major engineering projects
		Gender	Male
		Age (approx.)	30s
Duration of interview			1 hour
Method of data capture			• Semi-structured interview with audio recording • Requirement capture using DRed (while requirements and rationales were elicited by the line engineer, the researcher captured them on the DRed map)
Roles of participants during the interview			The interview participants would perform the following tasks during the interview: • The line engineer – elicit/recall requirements which he defined before. He was encouraged to reason about the requirements elicited. • The researcher – capture requirements and any requirement metadata as they were being elicited by the line engineer. • System Engineer – give any background information about the engine project and the requirements elicited by the line engineer
Method of data analysis after the interview			Evaluate the workflow by modelling requirements and requirement rationales in DRed

8.2 Results

The results of the research are presented in two main Sections. The first focuses on requirement rationale. The second, instead, focuses on requirement traceability.

8.2.1 Requirement rationale

This Section describes an investigation into the requirement rationale capture aspect of the case study. It starts by presenting current practice, i.e. the existing workflow. On the basis of the understanding developed it raises first important issues for improvement and then introduces an improved workflow. After introducing the improved workflow, the rest of the Section reports feedback on the improved workflow and the results of the interview to reconstruct requirement rationales.

Current practice

A reconstruction of the existing requirement analysis workflow was performed based on document analysis and an interview with system engineers. An overview of the reconstructed workflow is shown in Figure 72. As it can be seen four main tools were used for requirement analysis, namely Viewpoint Analysis (VPA), Function Flow Diagram (FFD), Systemic Textual Analysis (STA) and DOORS. VPA and FFD are used to model the functional behaviour of the engine in order to facilitate engineers to elicit functional and non-functional performance requirements. The STA is used as a lookup table to indicate relationships between requirements of different types; for example, to record the relationship between a non-functional implementation requirement and a functional requirement. DOORS is used as a requirement repository to store all validated and accepted requirements. Figure 72 is annotated with the flows of function and non-functional requirements (as defined in Table 25). As it can be seen functional and non-functional requirements have different “entry points”, and this is an issue which shall be explained further in the next Section. At a high level it can be noticed that the first tool (VPA) and the second tool (FFD) were used just to model the functional requirements, whereas the non-functional requirements were considered only from the third tool (STA) onwards. The remainder of this Section presents a walkthrough of the existing requirement analysis workflow.

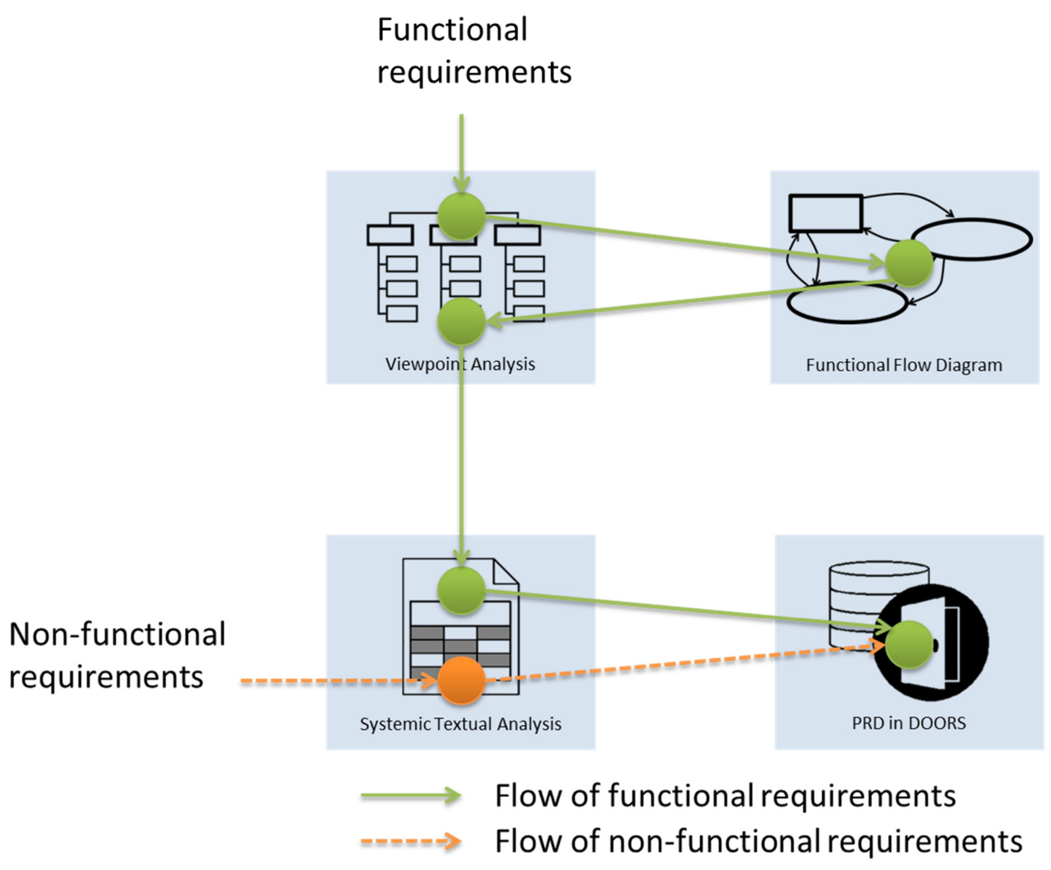

Functional requirements
Viewpoint Analysis
Functional Flow Diagram
Non-functional requirements
Systemic Textual Analysis
PRD in DOORS
Flow of functional requirements
Flow of non-functional requirements

Figure 72 Existing workflow used to analyse requirements

Table 25 Requirement categories

	Requirement types according to the Holistic Requirements Model (Burge, 2006)
Functional requirements	• Operational • Functional
Non-functional requirements	• Non-functional systems • Non-functional performance • Non-functional implementation

The Viewpoint Analysis (VPA) diagram was the first document created out of all the requirement documents, see Figure 73. An introduction to VPA was given in Section 2.4. In this engine project, the VPA was used to define the general functional behaviour of the engine, and the behaviour defined form the basis upon which requirements are grouped and structured. The VPA document is a hierarchical tree, see Figure 73. The first row from the top is a group header. The second row from the top contains operational requirements, such as deliver power, integrate auxiliary services, and control power. All subsequent rows are functional requirements, such as intake air, compress air, and distribute fuel. Conventionally, VPA should contain both functional and non-functional requirements (Burge, 2011b). The functional requirements would be made up of operational requirements on the second row from the top, and these would be broken down into functional requirements as tree branches extending from the operational requirements. In this method non-functional requirements are typically labelled as floating bubbles attached to the leaves on the branches. However, the VPA created in this engine development programme has functional requirements only. Non-functional requirements have been captured on separate documents.

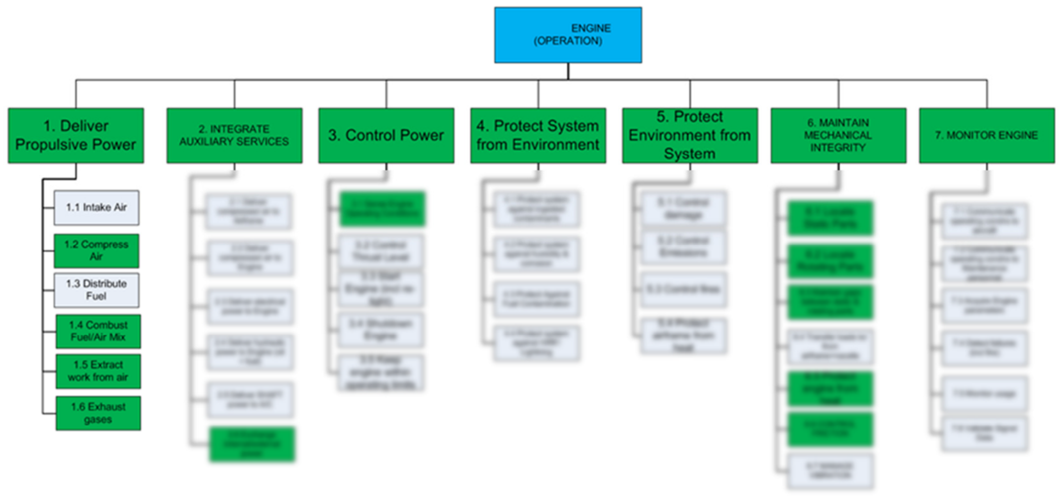

ENGINE (OPERATION)
1. Deliver Propulsive Power
2. INTEGRATE AUXILIARY SERVICES
3. Control Power
4. Protect System from Environment
5. Protect Environment from System
6. MAINTAIN MECHANICAL INTEGRITY
7. MONITOR ENGINE
1.1 Intake Air
1.2 Compress Air
1.3 Distribute Fuel
1.4 Combust Fuel/Air Mix
1.5 Extract work from air
1.6 Exhaust gases

Figure 73 A Viewpoint Analysis chart captured as a Microsoft Visio document (pixelated intentionally due to confidentiality)

The Function Flow Diagram (FFD) was the second document to be generated, see Figure 74. An introduction to FFD was given in Section 2.4. The FFD expands the requirements in the VPA. In the FFD, functions are used to indicate operational and functional requirements from the VPA. And the whole of the FFD is used to illustrate flows of input or output quantities (such as information, control, material, or energy) between these functions. For example, between the operational requirements “generate propulsive power” and “monitor engine" there is the flow of "monitoring parameters", see Table 26. The FFD contributes to requirement analysis in three ways. First, the process of modelling function flows can help engineers discover new functional requirements. Newly discovered functions can be translated into functional requirements. Second, the process of modelling function flows can also uncover non-functional implementation requirements. For example, if certain functionalities are already fulfilled by an existing component, a non-functional implementation requirement can be defined to make the use of that component a requirement. Third, the flows on the FFD can be useful for inferring performance requirements, as the flows can help engineers locate which performance values exist and must be converted into a requirement.

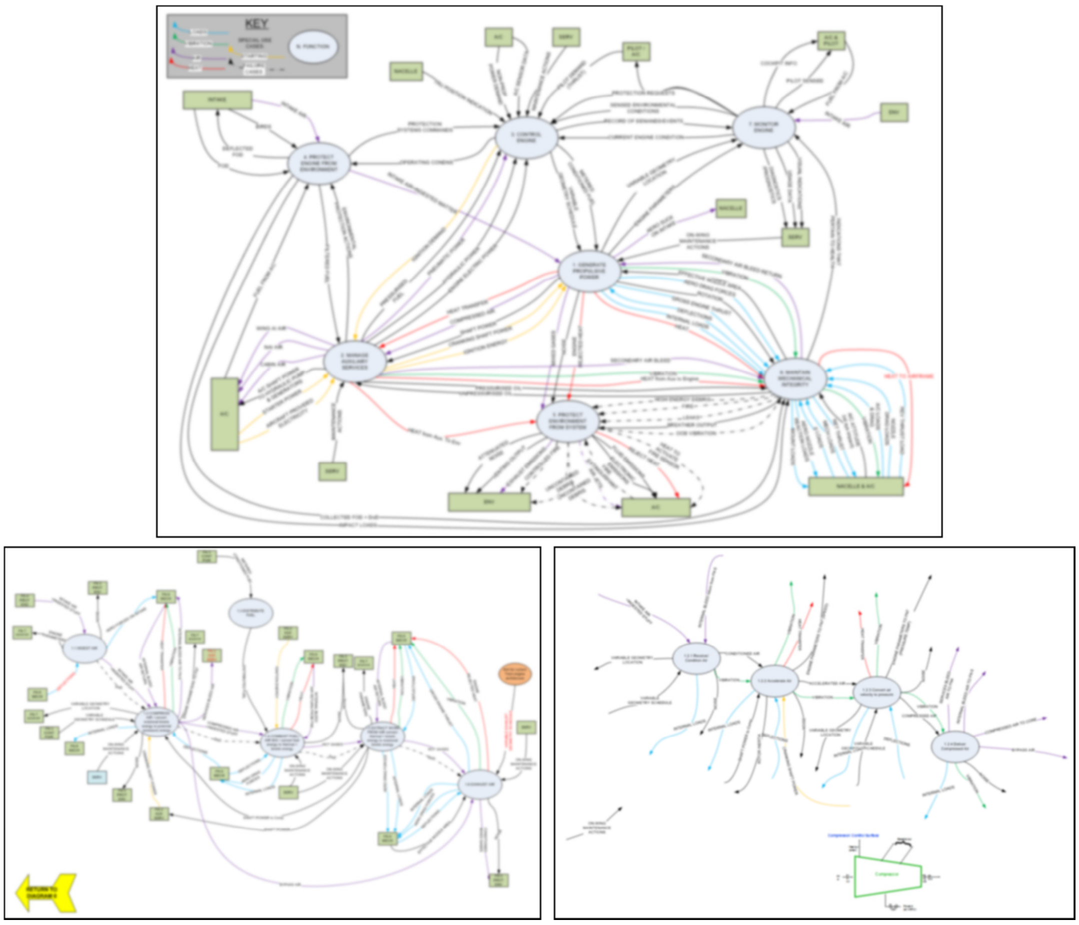


Figure 74 Function Flow Diagrams captured as Microsoft Visio documents (pixelated intentionally due to confidentiality)

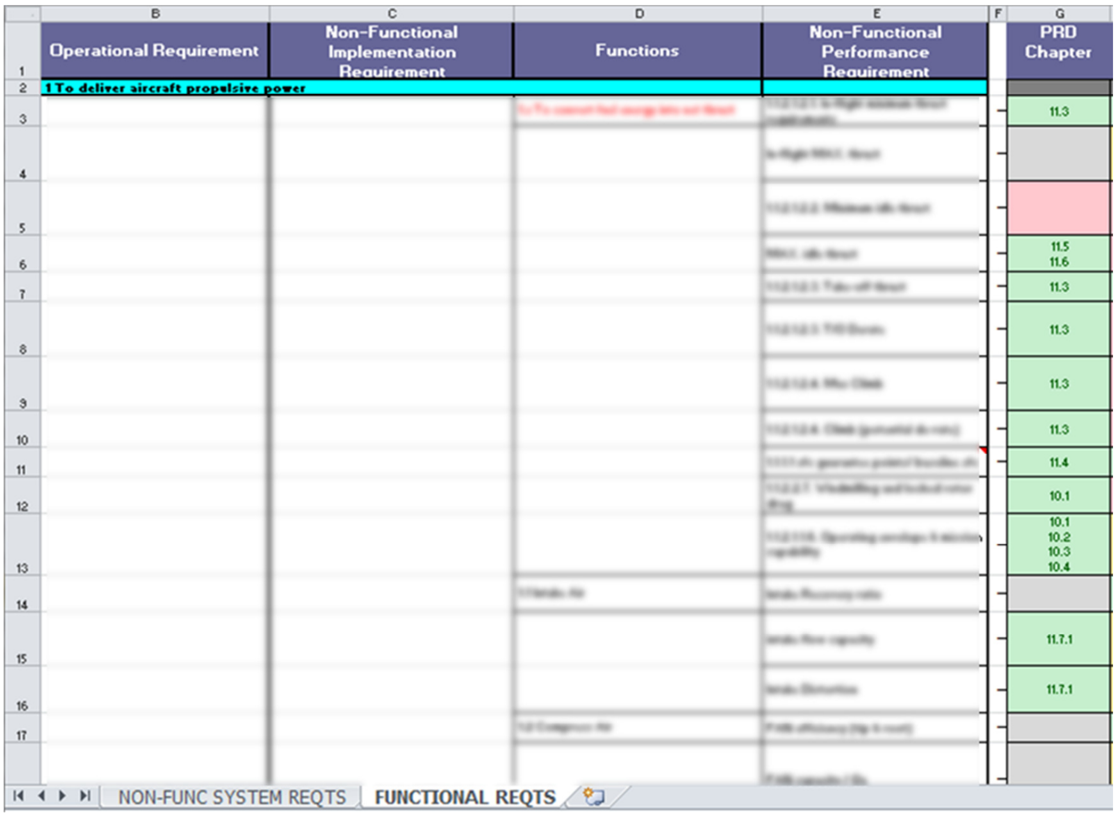
Table 26 Examples of functional interactions on the FFD

Operational requirement 1	Flow	Operational requirement 2
Generate propulsive power	Monitoring parameters	Monitor engine
Monitor engine	Parameters representing current engine condition	Control engine

The Systemic Textual Analysis (STA) is a lookup table that provides mappings between different types of requirements, see Figure 75. An introduction to STA was given in Section 2.4. It is composed of two spreadsheet tables. The first table contains operational, non-functional implementation, functional, and non-functional performance requirements, respectively presented in columns one, two, three, and four from left to right, see Figure 75 (top). The second table contains non-functional systems requirements only, see Figure 75 (bottom).

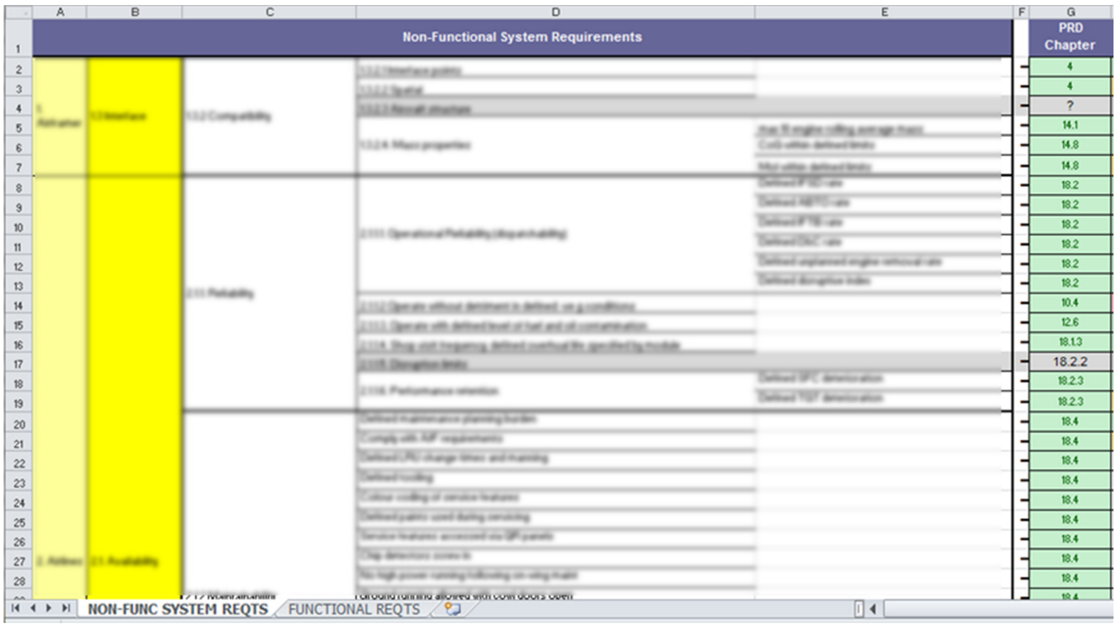
On the first table, the operational requirement and functional requirements originate from the VPA. The non-functional implementation requirements are linked to corresponding functional requirements. Non-functional performance requirements derived from functional requirements are also stored in the last method of the workflow, i.e. PRD in DOORS. In the VPA these are traceable to the PRD via Chapter headings. These Chapter headings populate the last column to the right of the STA table, see Figure 75 (top).

The second table stores the non-functional system requirements as a hierarchical tree that branch from left to right. The leaves of the tree are requirements which are mapped to Chapter headings in the PRD in DOORS. These Chapter headings populate the last column to the right of the table, see Figure 75 (bottom).



	Operational Requirement	Non-Functional Implementation Requirement	Functions	Non-Functional Performance Requirement	PRD Chapter
2	1 To deliver aircraft propelsive power				
3			[illegible]	[illegible]	11.3
4				[illegible]	
5				[illegible]	
6				[illegible]	11.5 11.6
7				[illegible]	11.3
8				[illegible]	11.3
9				[illegible]	11.3
10				[illegible]	11.3
11				[illegible]	11.4
12				[illegible]	10.1
13				[illegible]	10.1 10.2 10.3 10.4
14			[illegible]	[illegible]	
15				[illegible]	11.7.1
16				[illegible]	11.7.1
17			[illegible]	[illegible]	

NON-FUNC SYSTEM REQTS | FUNCTIONAL REQTS



	Non-Functional System Requirements	PRD Chapter
2	[illegible]	4
3	[illegible]	4
4	[illegible]	?
5	[illegible]	14.1
6	[illegible]	14.8
7	[illegible]	14.8
8	[illegible]	18.2
9	[illegible]	18.2
10	[illegible]	18.2
11	[illegible]	18.2
12	[illegible]	18.2
13	[illegible]	18.2
14	[illegible]	10.4
15	[illegible]	12.6
16	[illegible]	18.1.3
17	[illegible]	18.2.2
18	[illegible]	18.2.3
19	[illegible]	18.2.3
20	[illegible]	18.4
21	[illegible]	18.4
22	[illegible]	18.4
23	[illegible]	18.4
24	[illegible]	18.4
25	[illegible]	18.4
26	[illegible]	18.4
27	[illegible]	18.4
28	[illegible]	18.4

NON-FUNC SYSTEM REQTS | FUNCTIONAL REQTS

Figure 75 Systemic Textual Analysis tables captured in Microsoft Excel (pixelated intentionally due to confidentiality)

The Product Requirement Document (PRD) is the destination document for all validated and accepted requirements, see Figure 76. The PRD is the formal requirement specification for the project. It is stored in a requirement repository in the form of an IBM DOORS database. An introduction to the DOORS interface was given in Section 2.4. The PRD is a document that has 500 A4 pages, with an average of 6 requirements per page. The PRD document was given to the researcher for analysis as a PDF so that it is transferrable.

The PRD contains predominantly non-functional requirements. Most of the non-functional requirements are of the type performance requirements. An example requirement is "Each oil tank must have an expansion space of not less than xx% of the tank capacity". Another example is "a xxx engine shall achieve a minimum thrust level of xxx% of rated thrust within xxx seconds, at any altitude between xxx feet and xxx feet".

Every requirement in the PRD is defined in a format according to the template shown in Figure 77. The requirement template consists of an identifier that is automatically generated, the requirement body, and other contextual metadata. The requirement body can include text statements, tables of data, diagrams, and reference to other requirements stored in DOORS. Requirement metadata can include rationales, background information, definition of special terms, and a variable flag to indicate whether the requirement has changed since the last issue of the document.

The PRD was derived from Business Requirement Documents (BRD). The BRD is a combination of requirement sets agreed with external companies such as airframe manufacturers, regulatory bodies, and suppliers, as well as requirements defined by stakeholders internal to the company (e.g. requirements carried over from previous projects).

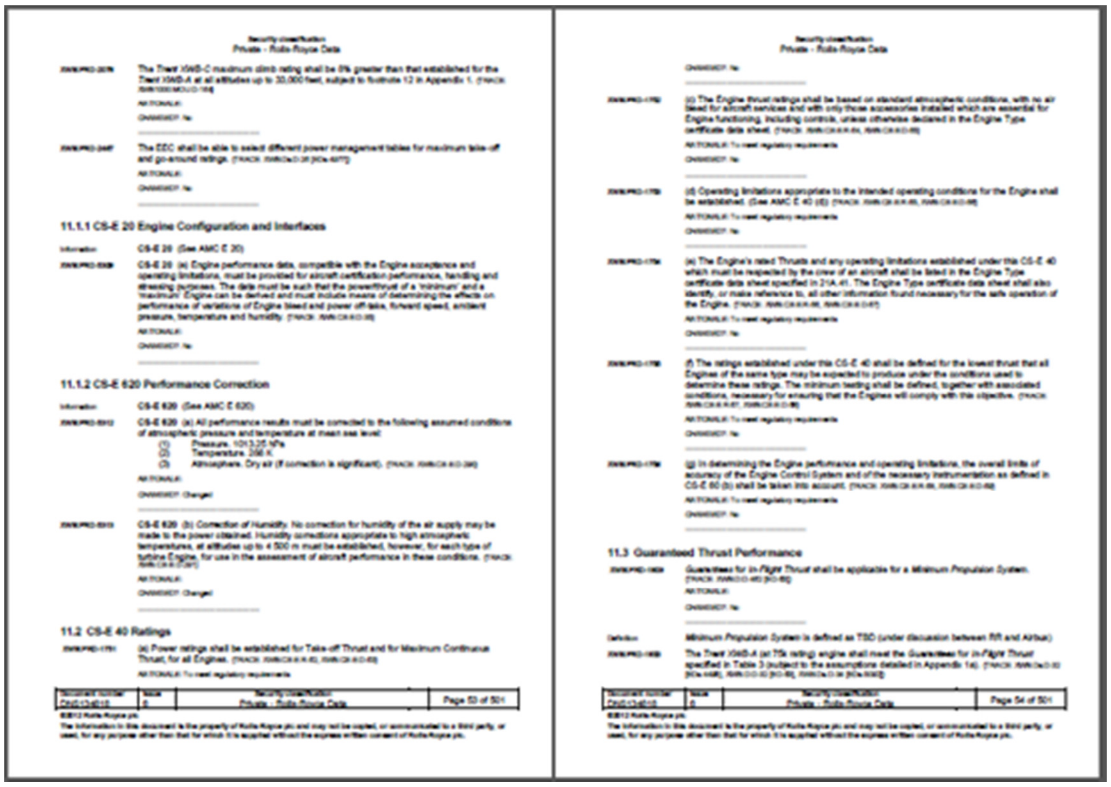


Figure 76 A page of Product Requirement Document in the form of a PDF file (pixelated intentionally due to confidentiality)

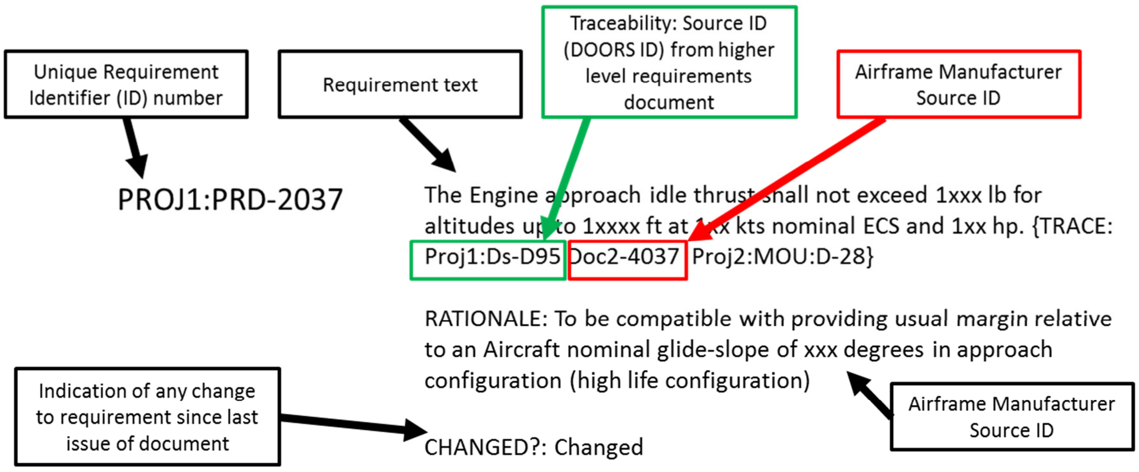

Unique Requirement Identifier (ID) number
Requirement text
Traceability: Source ID (DOORS ID) from higher level requirements document
Airframe Manufacturer Source ID
PROJ1:PRD-2037
The Engine approach idle thrust shall not exceed 1xxx lb for altitudes up to 1xxxx ft at 1xx kts nominal ECS and 1xx hp. {TRACE: Proj1:Ds-D95 Doc2-4037 Proj2:MOU:D-28}
RATIONALE: To be compatible with providing usual margin relative to an Aircraft nominal glide-slope of xxx degrees in approach configuration (high life configuration)
Indication of any change to requirement since last issue of document
CHANGED?: Changed
Airframe Manufacturer Source ID

Figure 77 Template for representing a requirement in PRD in DOORS

Issues with current practice

This Section discusses issues with current practice with a view to improve requirement rationale capture in the existing workflow.

Issue 1 – most of the tools chosen in the existing workflow are not designed to support requirement rationale capture. The VPA, FFD, and STA tools do not have requirement rationale as a data type in their dictionary. Furthermore, these tools have been designed to use as a means to elicit new requirements and checking for missing requirements. Introducing requirement rationale onto these tools may degrade their effectiveness.

Issue 2 – the PRD in DOORS is the only tool that has requirement rationale as a valid data type in its dictionary, but the PRD is also the final step in the existing workflow. Requirement rationale could easily be lost if it is only recorded long after it is thought of. Requirement rationale could also be lost if requirements are transferred from one document to another without the explicit transfer of rationale.

Issue 3 – not all tools are used to capture both functional and non-functional requirements. For example the VPA and FFD are used to analyse only functional requirements, see Figure 72. This is an issue if improvements are to be designed to support rationale capture throughout the entire spectrum of requirements as it calls for introducing such support after application of the VPA and FFD.

Proposal of an improved workflow (Requirement rationale capture)

This Section describes part one of a two-part proposal to improve the existing workflow. Part one describes improvement to requirement rationale capture. Part two, in Section 8.2.2, describes improvements to requirement traceability. An overview diagram of the proposal is shown in Figure 78. The proposal introduces a new stage at the beginning of the existing workflow. The new stage consists of using DRed as a medium to capture requirements and metadata as soon as they are elicited – this includes both functional and non-functional requirements.

The improvement addresses the three issues defined in the previous Section as argued below.

In response to issue 1, the improved workflow removes the need to modify all tools to support requirement rationale capture. It is sufficient that the initial tool (i.e. DRed) supports it, and the captured rationale can be traced to by navigating backwards from any other tool. In this way the IBIS notation is used to capture requirements and their justifications.

In response to issue 2, the improved workflow ensures that requirement rationale capture is not delayed until the final step. In fact it encourages the capture of requirement rationale at the point where it is the most abundant and still fresh in the mind of engineers.

In response to issue 3, the support for requirement rationale capture in the improved workflow is not affected by the fact that some tools only capture one type of requirement. The existing workflow is not replaced but extended. This means that users of the existing workflow can continue to use the tools that they are used to, as the demand on users to change their behaviour is minimised.

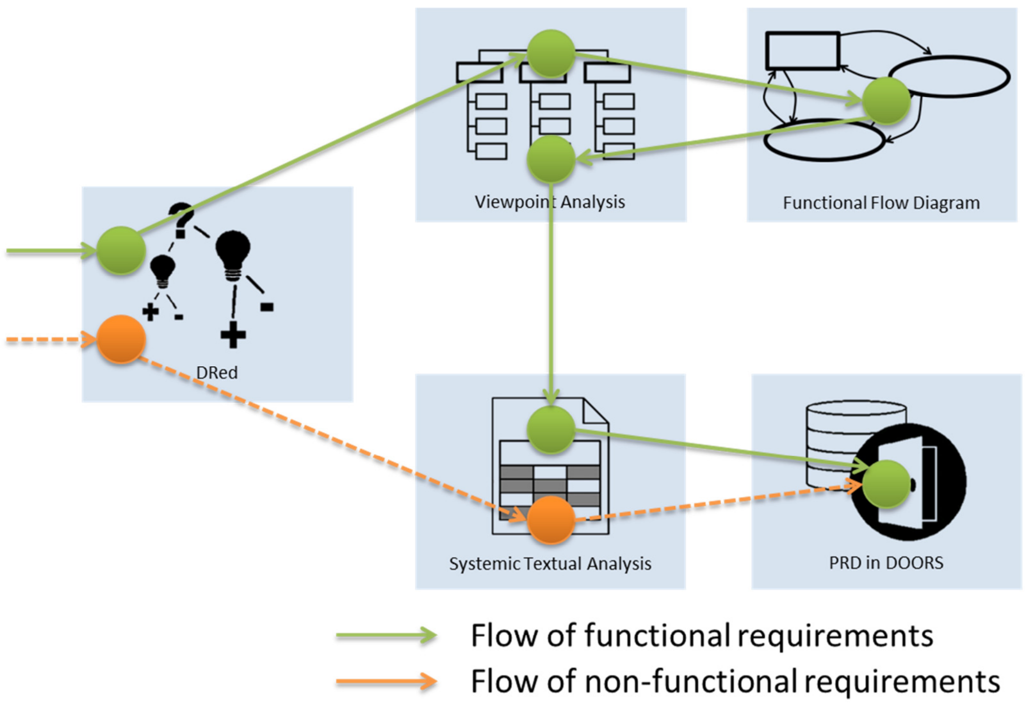

Viewpoint Analysis
Functional Flow Diagram
DRed
Systemic Textual Analysis
PRD in DOORS
Flow of functional requirements
Flow of non-functional requirements

Figure 78 Improved workflow with the introduction of the DRed tool to capture requirement rationale as soon as requirements are defined

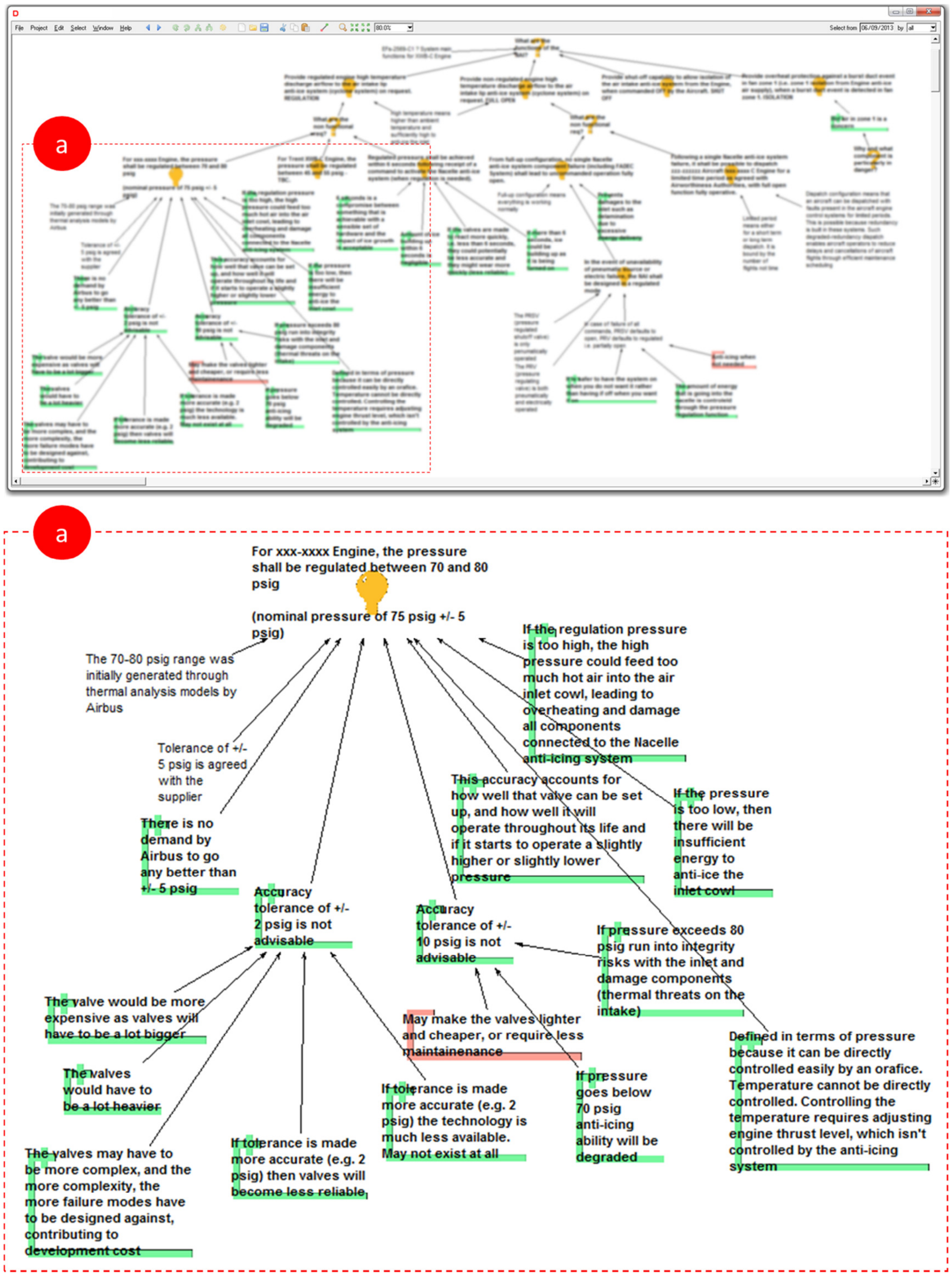
Feedback on the improved workflow

This Section describes the results from the questionnaire to evaluate the improved workflow. The overall feedback after presenting it to two systems engineers was very positive. Detailed questionnaire results can be found in Appendix 2.

Three points can be summarised about the feedback. First, the systems engineers recognised the importance of rationale capture. On average they rated 8 out of 10 the importance of rationale with 0 not being important at all and 10 being extremely important. However, they have commented that it is not worthwhile to capture requirement rationale if it is obvious and incontestable. This indicates that a trade-off has to be found between capturing rationale and being able to progress a design task at the expected pace. Second, they both agreed that the concept of the improved workflow is sound and makes sense. On average they rated 7 out of 10 the validity of the proposal with 0 being not valid at all and 10 being extremely valid. Third, they both agreed that the concept can be implemented into current practice and envisaged that engineering work would benefit from it. On average they rated 7 out of 10 the feasibility of implementing the proposal with 0 not being feasible at all and 10 being extremely feasible.

Reconstructing requirement rationale

This Section describes the results from the interview to reconstruct the rationale for the requirement of a specific sub-system. During the interview, one DRed map was created capturing the rationale for 10 requirements which define a nacelle anti-icing system to be used on the gas turbine engines, see Figure 79. The nacelle is the casing on the outside of an aircraft. The main function of the nacelle anti-icing system is to prevent ice accretion when cold air enters the engine. Ice accretion on the nacelle leading edges can diminish aerodynamic performance and increase fuel consumption. The types of requirements and metadata in Figure 79 are also shown in Table 27. Four functional requirements and six non-functional requirements were captured. The four functional requirements refer to the functional behaviour of the anti-icing system in four states. These four states are: 1) regulation – the mode for normal state of operation; 2) full open – emergency mode in case of controller failure; 3) shut off – normal mode for controlled switch off to conserve energy; and 4) isolation – safe guard mode in the event of overheating. Out of the six non-functional requirements, three are derived from one functional-requirement, and the other three are derived from another functional requirement. Due to the limited time available, the interview focused on eliciting requirement justifications for non-functional requirements only. Table 27 also shows the distribution of rationale types for the non-functional requirements.


a
a
For xxx-xxxx Engine, the pressure shall be regulated between 70 and 80 psig

(nominal pressure of 75 psig +/- 5 psig)
The 70-80 psig range was initially generated through thermal analysis models by Airbus
If the regulation pressure is too high, the high pressure could feed too much hot air into the air inlet cowl, leading to overheating and damage all components connected to the Nacelle anti-icing system
Tolerance of +/- 5 psig is agreed with the supplier
This accuracy accounts for how well that valve can be set up, and how well it will operate throughout its life and if it starts to operate a slightly higher or slightly lower pressure
If the pressure is too low, then there will be insufficient energy to anti-ice the inlet cowl
There is no demand by Airbus to go any better than +/- 5 psig
Accuracy tolerance of +/- 2 psig is not advisable
Accuracy tolerance of +/- 10 psig is not advisable
If pressure exceeds 80 psig run into integrity risks with the inlet and damage components (thermal threats on the intake)
The valve would be more expensive as valves will have to be a lot bigger
May make the valves lighter and cheaper, or require less maintainenance
Defined in terms of pressure because it can be directly controlled easily by an orafice. Temperature cannot be directly controlled. Controlling the temperature requires adjusting engine thrust level, which isn't controlled by the anti-icing system
The valves would have to be a lot heavier
If tolerance is made more accurate (e.g. 2 psig) the technology is much less available. May not exist at all
If pressure goes below 70 psig anti-icing ability will be degraded
The valves may have to be more complex, and the more complexity, the more failure modes have to be designed against, contributing to development cost
If tolerance is made more accurate (e.g. 2 psig) then valves will become less reliable

Figure 79 DRed map produced during the interview with a portion zoomed in (a part of the DRed map is intentionally blurred to preserve confidentiality)

Table 27 Summary of requirements and requirement rationale captured during the interview

		Rationale			
	Number of requirements	Number of requirement rationales	Number of pro	Number of con	Number of neutral
Functional requirements	4	0	0	0	0
Non-functional requirements	6	30	21	2	7

To assess the value of the rationales reconstructed in Figure 79 the information available in the PRD document for the same set of ten requirements was analysed. The types of requirements and metadata captured in the PRD using the existing workflow are shown in Table 28. One significant result of the interview is that there has been far more requirement rationales elicited compared to what is available in the original dataset. Contrasting Table 27 and Table 28 it can be seen that 30 requirement rationales have been captured for 6 non-functional requirements rather than 8 requirement rationales in the PRD. In addition, more requirement rationales could have been captured in a real engineering project, as there would have been more stakeholders to contribute.

Table 28 Summary of requirements and requirement rationale captured in PRD

		Rationale			
	Number of requirements	Number of requirement rationales	Number of pro	Number of con	Number of neutral
Functional requirements	4	0	0	0	0
Non-functional requirements	6	8	3	0	5

One interesting result is the observation that requirement rationales captured in the interview served similar purposes as those captured in the projects presented in previous Chapters. During the interview, various use cases for requirement rationales were demonstrated. All of these use cases were seen previously. There was use of rationale for clarification. For example to clarify a requirement about the use of pressure for a valve that is used to control the air temperature within the anti-icing system, a rationale was captured to explain that pressure was used because it can be directly controlled by an orifice, while temperature cannot be directly controlled. There was also use of rationale for justification of numerical limits defined in requirements. For example, to justify the requirement that the lower boundary of pressure for an inlet duct must be 70 psig, a rationale was captured to explain that below this pressure the de-icing ability is degraded. There was rationale captured also to discuss conflicting requirements and explain how a compromise was reached. For example, the requirement of having a tolerance of +/- 5 psig is a value that was given after a compromise had been reached. The rationale against a requirement that demands a more accurate value such as +/- 2 psig is that the cost of valves would increase in order to reach that value. The rationale against a requirement that demands a less stringent tolerance of 10 psig is that the anti-icing ability of the system would degrade to a level that is unacceptable.

Another interesting result is about the subject of rationales. Four rationale categories emerged that align with those found in Section 6.2.2, see Table 29. For example, one rationale was related to the *design* of a valve inlet with consideration of the potential impact of excessive energy delivery.

Table 29 Subject of the rationales captured during the interview

Category of rationale	Definition of category	Example rationale in this project
Design	Rationale that relates to the consideration of challenges expected in the concept formulation of the product	***Requirement:*** In the event of a control component it shall fail in a safe manner ***Rationale:*** prevents damages to the inlet such as elimination due to excessive energy delivery ***Relationship to design:*** Consideration of excessive energy delivery causing damage to a component
Usage	Rationale that relates to the technical functioning of the solution and its use within the context of operation	***Requirement:*** [Same as above] ***Rationale:*** [Same as above] ***Relationship to usage:*** Consideration of the context of component failure and how components must react in that context
Project scope	Rationale that relate to changes in the boundaries of the project	***Requirement:*** Measure pressure with a tolerance of +/- 5psig ***Rationale:*** Tolerance more detailed than +/-5 psig is beyond the scope of the project ***Relationship to project scope:*** Considerations of whether a requirement is out of scope
Project cost	Rationale that relate to cost considerations	***Requirement:*** [Same as above] ***Rationale:*** Requirement on a higher tolerance value than +/-5 psig should be rejected because to achieve that level of tolerance, the cost of valves would increase beyond the budget for that component ***Relationship to project cost:*** Considerations of cost

Finally, as stated while presenting the data collection, the interview was designed to mimic a real scenario. Hence, another result of the interview is the proof-of-concept that it is feasible and practical to capture requirements and rationale during a live analysis session. The researcher doing the capture was able to keep up with the rate of elicitation of the line engineer. The capturer could have kept up with the pace even more comfortably if the capturer had been a domain expert, which would be the case if this workflow is adopted by an engineering team where the capturer is a project engineer. Furthermore, the rate of requirement elicitation would have been slower in a real project analysis session, as it would have taken more time if requirements were newly created compared to being recalled in this case. Requirements elicited in this session had already been defined in the PRD.

8.2.2 Requirement traceability

This Section describes an investigation into the requirement traceability aspect of the case study. It starts by presenting current practice, i.e. the existing workflow. On the basis of the understanding developed it raises first important issues for improvement and then introduces the second part of the improved workflow. After the introduction of the workflow, the rest of the Section provides feedback on the improved workflow.

Current practice

The investigation of requirement traceability begins by analysing cross-document traceability. In particular, *traceability mechanisms* were analysed. The concept of traceability mechanism was touched upon in previous Chapters; it is viewed as the technique used to capture requirement evolution such that it is possible to interpret where a requirement originates from and what a requirement evolves into.

In order to analyse cross-document traceability mechanisms, it is necessary to determine the order of document creation so that it is clear which document contains the original set of requirements that other requirements evolved from. As can be seen in Figure 80, the requirement documents were created in the following order: Viewpoint Analysis (VPA), Function Flow Diagram (FFD), Systemic Textual Analysis (STA), and Product Requirement Document (PRD). Data analysis showed that requirements in each of these documents are made traceable to requirements in other documents using two types of traceability mechanisms: text description duplication and manually generated ID. These mechanisms are evaluated in the next Section.

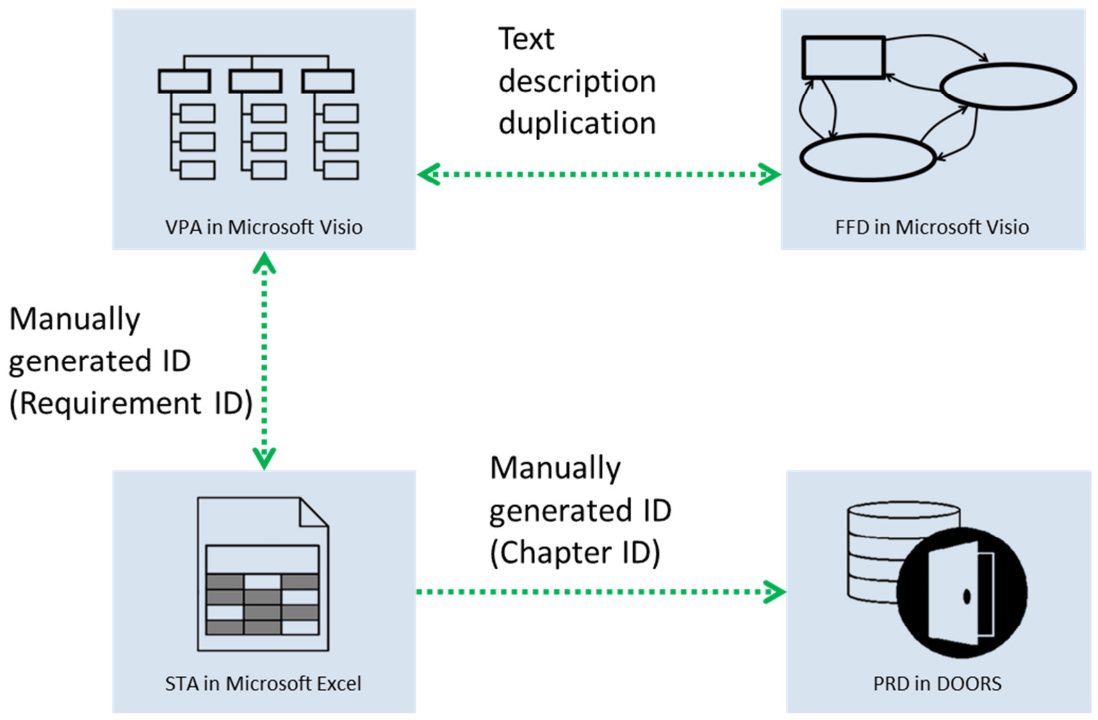

Text description duplication
VPA in Microsoft Visio
FFD in Microsoft Visio
Manually generated ID (Requirement ID)
Manually generated ID (Chapter ID)
STA in Microsoft Excel
PRD in DOORS

Figure 80 Existing traceability mechanisms in the requirement documents of the engine project

Issues with current practice and other traceability mechanisms

At this point, it is worth reviewing the four types of traceability mechanisms that were encountered throughout this Chapter and project data in other Chapters. These four ways to link requirements to enable traceability can be distinguished as either trace-by-duplication or trace-by-reference:

- Trace-by-duplication
  - Text description duplication
  - Manually generated ID
- Trace-by-reference
  - Matrix
  - Hyperlinking

Trace-by-duplication refers to traceability links that rely on a value at the source requirement and destination requirement being exactly the same. Trace-by-reference refers to traceability links where the source requirement has a pointer to the destination requirement, and the destination requirement also has a pointer to the source requirement.

Text description duplication (shown graphically in Figure 81) refers to when the same or similar textual descriptions are used at the source and destination requirements. It is a trace-by-duplication link. This type of traceability link is used between VPA and FFD, see Figure 80.

This technique has two advantages. First, it is convenient as it is supported by any tool. Second, there is no need to capture any supplementary information in addition to the requirement statement.

This technique has also three disadvantages. First, due to the fact that the traceability link is implied by the source requirement and destination requirement describing the same thing, the links can become ambiguous as the size of the requirement set increases. This becomes increasingly problematic when there are multiple requirement contributors and multiple requirement readers. Second, this kind of traceability link is only feasible for one-to-one requirement evolutions. So the decomposition or combination of requirements cannot be made traceable. Third, it is prone to losing synchronisation. If changes to the source requirement are not mirrored on the destination requirement, then the two requirements will lose traceability. The problem of text duplication and losing synchronisation is documented by Krottmaier (2002) as the problem of Cut-and-Paste.

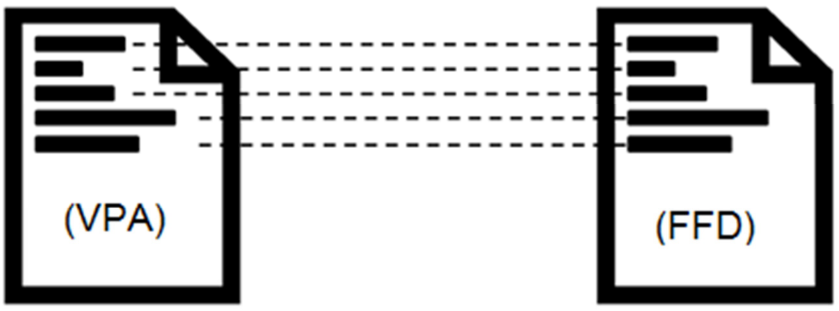


Figure 81 Graphical illustration of text description duplication

Manually generated ID (shown graphically in Figure 82) refers to when the source and destination nodes share the same ID. It is also a trace-by-duplication link. This type of traceability link is used between the VPA and STA, see Figure 80, and to some extent, between the STA to PRD, see Figure 80, if chapter headings in the PRD are considered as uniquely generated IDs. But note that between the STA and PRD, requirements are only traceable in one direction (from the STA to PRD).

This technique has three advantages. First, like text description duplication, the manually generated ID technique is not restricted to specific tools, as long as a unique ID is always present along with requirements statements. Second, this flexibility is particularly useful when the source requirement and destination requirement are in different formats. For example, when requirements are initially captured in Microsoft Word they could be assigned ID's. Third, this technique eliminates ambiguity, which is an advantage over *text description* *duplication* traceability link. Traceability exists as long as the source and destination requirements share the same ID, and the content of the requirement can vary independently.

This technique has four disadvantages. First, the maintenance of ID generation must be managed. IDs must be generated so that they are globally unique for all requirements across a project. Otherwise, two different requirements with the same ID will cause ambiguity. This limitation on ID generation means that this type of traceability link may not be practical for large-scale project requirements because it requires all the people involved to generate requirements from one source. Secondly, there is the inconvenience that the IDs must always stay with the requirement. In some situations, this may clutter a requirements document. Third, the use of IDs is not compatible with one-to-many or many-to-one. In the case of one-to-many, it could be incorrect to assign the destination requirements with the ID of their source requirement, because the destination requirements are not the same as the source, rather they are derivatives. In the case of many-to-one, source requirements may have different IDs, but the destination can only have one ID. Fourth, there is the restriction that once a requirement is assigned an ID, that ID cannot be changed; otherwise source and destination requirements will lose synchronisation.

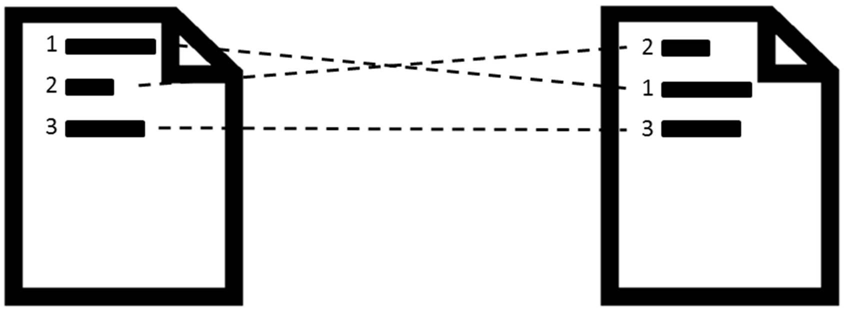


Figure 82 Graphical illustration of manually generated ID

Matrix (shown graphically in Figure 83) refers to a lookup matrix of traceability links, where any source node can be looked up to find its destination node. It is a trace-by-reference link. This type of traceability link is present in the case study in the form of the STA in Figure 80. The STA provides a lookup matrix between functional requirements and non-functional requirements.

This technique has three advantages. First, it is useful in scenarios that demands visualising requirement traceability. The matrix can display all traceability links that exist between the source requirement set and destination requirement set. Second, it can support one-to-many and many-to-one requirement evolutions. Third, the source and destination requirement(s) can vary independently without affecting requirement traceability.

This technique has two disadvantages. First, the matrix is an additional document to maintain, and there needs to be one matrix between every two requirements sets (i.e. source set and destination set). So for projects which contain several requirement sets, such as that in the case study, maintaining multiple traceability matrices is impractical. Second, the matrix is time consuming to setup. One has to ensure that only one instance of the matrix exists, and that the matrix is accessible to all requirement contributors.

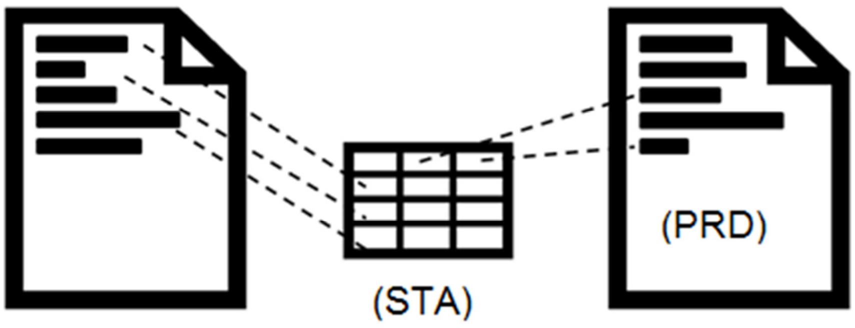


Figure 83 Graphical illustration of matrix

Hyperlinking (shown graphically in Figure 84) refers to a bi-directional hyperlink between the source node and destination node. It is a trace-by-reference link. The hyperlink must be bi-directional in order to satisfy the condition of both backwards and forward traceability (Gotel and Finkelstein, 1994). This type of traceability link is not present in the case study, but was proposed in Section 6.2.4.

This technique has four advantages. First, requirements evolution can be easily navigated, because both the source and destination are linked. Second, requirement contents can vary without risking to loose synchronisation. Third, this kind of traceability link can be scaled. It can be used on small projects and large projects alike. And unlike manually generated ID technique, this technique does not need to maintain a unique ID generator. Fourth, many-to-one, one-to-many, and many-to-many requirement evolutions can be supported.

This technique has two disadvantages. First, every requirement evolution has to be explicitly linked. Second, not many requirement capture tools support bi-directional hyperlinking. In order to support hyperlinking, the tool must be able to create requirements which are: a) uniquely addressable; b) support bi-directional hyperlinks; and c) support storage of multiple outwards hyperlinks. For example, in one-to-many requirement evolution, the source requirement needs to store hyperlinks to all of its destination nodes.

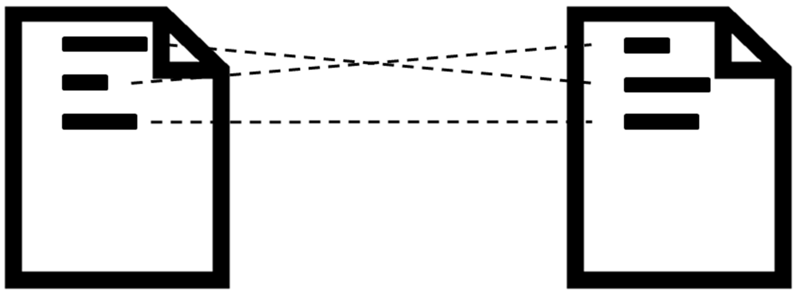


Figure 84 Graphical illustration of hyperlinking

Proposal of an improved workflow (Requirement traceability)

This Section describes part two of the two-part proposal to improve the existing workflow. Part two describes improvement to requirement traceability. When the two high-level mechanisms discussed previously are compared, it can be seen that the trace-by-reference link is more preferable than the trace-by-duplication link. Trace-by-reference is tolerant to requirement update (i.e. linked source and destination requirement(s) can vary independently without causing ambiguity), whereas trace-by-duplication suffers the potential problem of requirements loosing synchronisation.

Out of the two trace-by-reference options, hyperlinking is more practical than the matrix because it demands little maintenance, and it is scalable for projects of any size. Therefore, it has been proposed that hyperlinking shall be introduced in the improved workflow as the preferred traceability mechanism between all requirement documents, whenever the requirement capturing tool can support it. The overall strategy for the improved workflow is based on the concept that DRed is the central tool for requirement analysis, replacing where needed other software, and that all the documents will be traceable via the DRed document for requirement rationale capture proposed in Section 8.2.1. The DRed document shall hold the original requirement set and all the other requirement sets will evolve from it, see Figure 85. The aim of this strategy is to ensure that all requirements can be traced from the point of conception (i.e. captured in the DRed document) through their analysis, until the point when they are archived in a more stable form. This strategy fully utilises DRed's many hyperlinking mechanisms. These mechanisms include external bi-directional hyperlinks, tunnels, and transclusion. External bi-directional hyperlinks consist of linking a source requirement in DRed to a destination requirement in another software tool. To achieve this, the source requirement node would be attached a DRed file node, and the file node would store a hyperlink to a destination requirement node. This mechanism is only available if the tool that stores the destination requirement can allow requirements to be addressed by a hyperlink. External bi-directional hyperlinking is introduced between DRed and STA in Microsoft Excel, and between DRed and the PRD in IBM DOORS, see Figure 85. A demonstration of implementing hyperlinking from DRed to Excel and from DRed to DOORS was already given in Section 5.2.2.

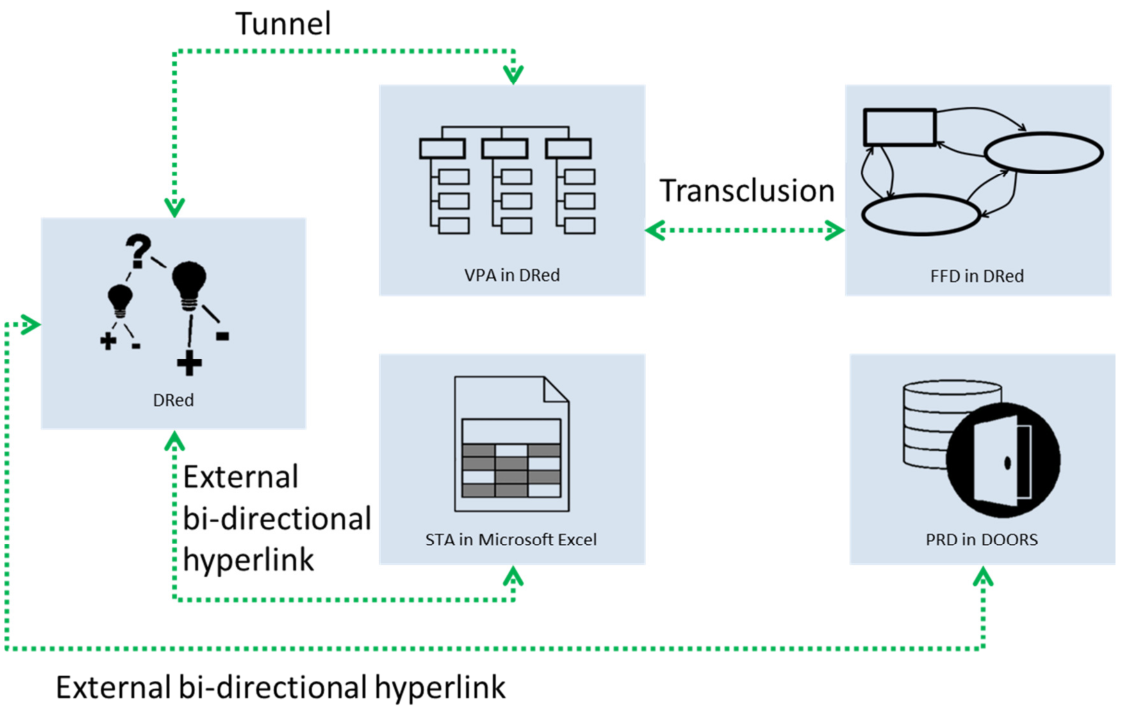


Figure 85 Improved traceability to existing workflow

Tunnelling is a DRed-specific feature that can be applied between two DRed nodes. The nodes can be on different DRed maps or on the same DRed map. The tunnel enables bi-directional navigation. Tunnelling is preferable to external bi-directional hyperlinks because it requires less mouse-click operations and is visually concise. To achieve traceable requirement nodes via tunnelling, the source requirement would need to be linked to a destination requirement via a tunnel link. Tunnelling is introduced between DRed and VPA, see Figure 85.

Transclusion is also a DRed-specific feature that can be applied between two DRed nodes. The nodes can be on different DRed maps or on the same DRed map. Transclusion enables bi-directional navigation. It further enables node synchronisation so that if either the source or destination requirement changes, the other would update to reflect the change. Transclusion is preferable to tunnelling when the source requirement and destination requirement are always the same, as the content synchronisation ability would relief the user from duplicating content. Transclusion is introduced between the VPA and FFD, see Figure 85.

Note that the improved workflow requires a shift from analysing requirements in multiple software tools to using the DRed as the central requirement analysis tool. This is so that advanced traceability mechanisms (i.e. tunnelling and transclusion) can be used, as currently they are only available between DRed to DRed documents. However, the conversion to DRed is not mandatory because the same effect can be achieved with other tools supporting bi-directional hyperlinking. In the improved workflow, two documents are transformed, see Figure 85. The VPA, previously in Microsoft Visio (as shown in Figure 73), is now a DRed-based document (as shown in Figure 86), and the FFD, previously as separate Microsoft Visio documents (as shown in Figure 74), are also converted to DRed-based documents (shown in Figure 87).

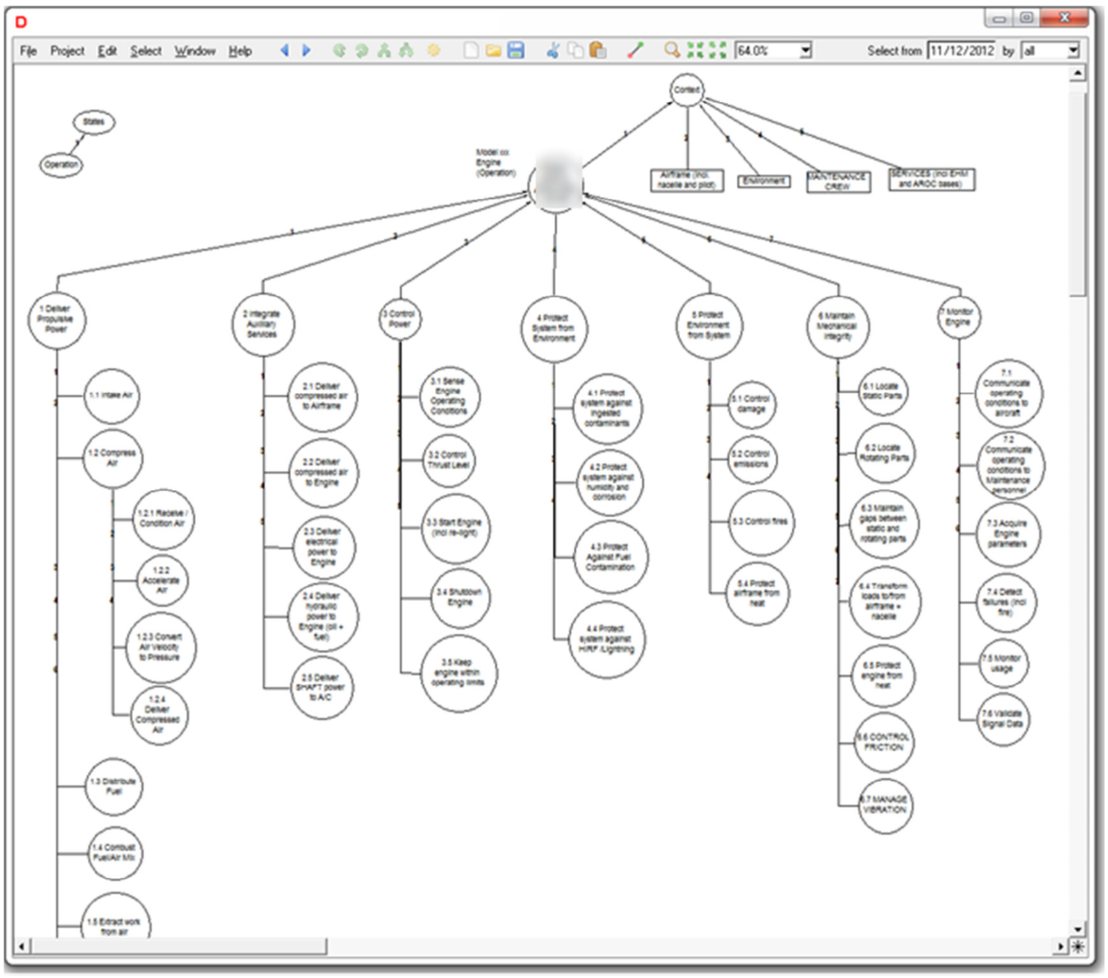

File
Project
Edit
Select
Window
Help
64.0%
Select from 11/12/2012 by all
States
Operation
Model xx Engine (Operation)
Context
Airframe (incl. nacelle and pilot)
Environment
MAINTENANCE CREW
SERVICES (incl EHM and AROC bases)
1 Deliver Propulsive Power
1.1 Intake Air
1.2 Compress Air
1.2.1 Receive / Condition Air
1.2.2 Accelerate Air
1.2.3 Convert Air Velocity to Pressure
1.2.4 Deliver Compressed Air
1.3 Distribute Fuel
1.4 Combust Fuel/Air Mix
1.5 Extract work from air
2 Integrate Auxiliary Services
2.1 Deliver compressed air to Airframe
2.2 Deliver compressed air to Engine
2.3 Deliver electrical power to Engine
2.4 Deliver hydraulic power to Engine (oil + fuel)
2.5 Deliver SHAFT power to A/C
3 Control Power
3.1 Sense Engine Operating Conditions
3.2 Control Thrust Level
3.3 Start Engine (incl re-light)
3.4 Shutdown Engine
3.5 Keep engine within operating limits
4 Protect System from Environment
4.1 Protect system against ingested contaminants
4.2 Protect system against humidity and corrosion
4.3 Protect Against Fuel Contamination
4.4 Protect system against HIRF /Lightning
5 Protect Environment from System
5.1 Control damage
5.2 Control emissions
5.3 Control fires
5.4 Protect airframe from heat
6 Maintain Mechanical Integrity
6.1 Locate Static Parts
6.2 Locate Rotating Parts
6.3 Maintain gaps between static and rotating parts
6.4 Transform loads to/from airframe + nacelle
6.5 Protect engine from heat
6.6 CONTROL FRICTION
6.7 MANAGE VIBRATION
7 Monitor Engine
7.1 Communicate operating conditions to aircraft
7.2 Communicate operating conditions to Maintenance personnel
7.3 Acquire Engine parameters
7.4 Detect failures (incl fire)
7.5 Monitor usage
7.6 Validate Signal Data

Figure 86 VPA in DRed

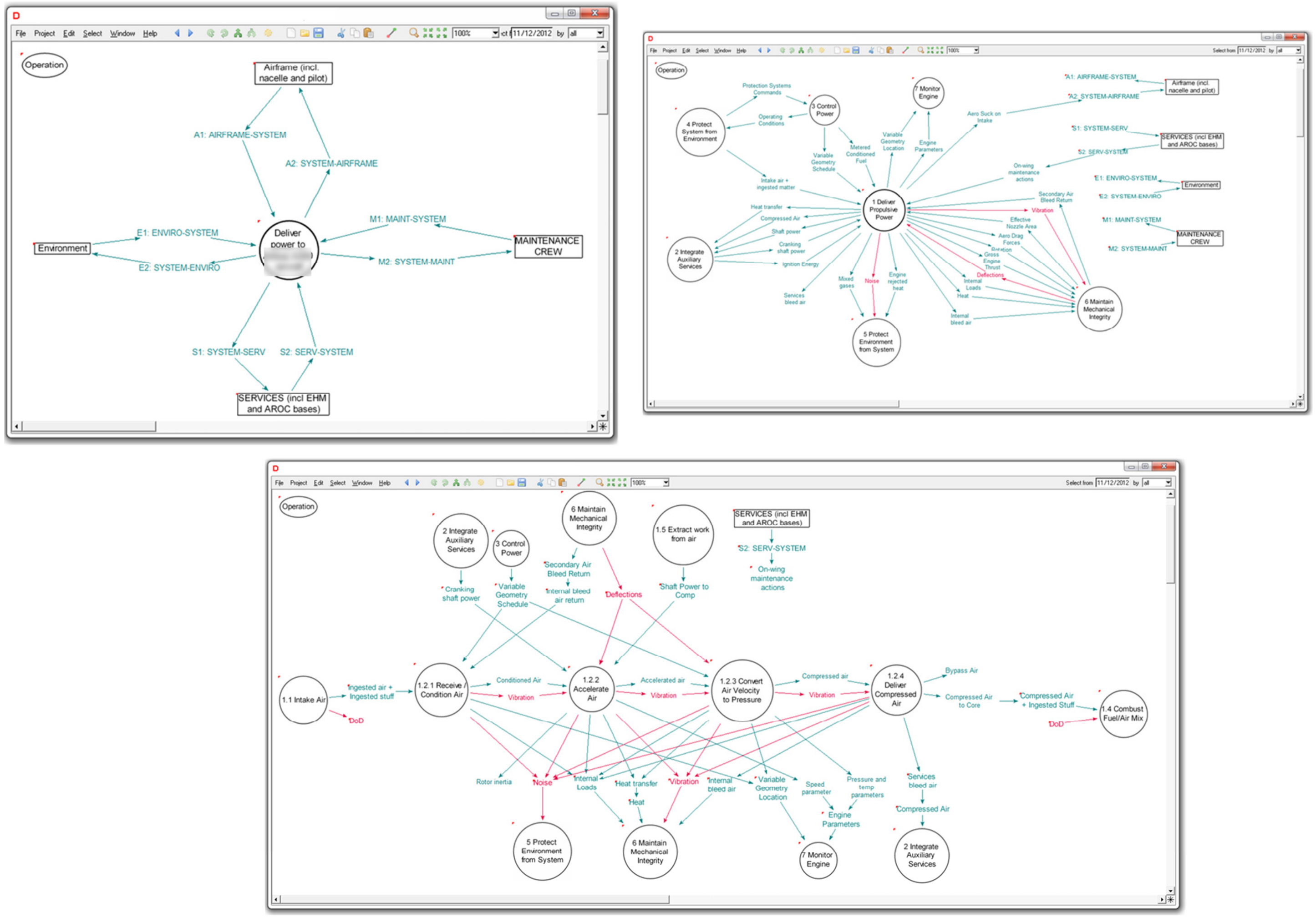

Operation
Airframe (incl. nacelle and pilot)
A1: AIRFRAME-SYSTEM
A2: SYSTEM-AIRFRAME
Deliver power to
Environment
E1: ENVIRO-SYSTEM
E2: SYSTEM-ENVIRO
M1: MAINT-SYSTEM
M2: SYSTEM-MAINT
MAINTENANCE CREW
S1: SYSTEM-SERV
S2: SERV-SYSTEM
SERVICES (incl EHM and AROC bases)
1 Deliver Propulsive Power
2 Integrate Auxiliary Services
3 Control Power
4 Protect System from Environment
5 Protect Environment from System
6 Maintain Mechanical Integrity
7 Monitor Engine
1.1 Intake Air
1.2.1 Receive / Condition Air
1.2.2 Accelerate Air
1.2.3 Convert Air Velocity to Pressure
1.2.4 Deliver Compressed Air
1.4 Combust Fuel/Air Mix
1.5 Extract work from air
Ingested air + Ingested stuff
DoD
Conditioned Air
Accelerated air
Compressed air
Vibration
Bypass Air
Compressed Air to Core
Compressed Air + Ingested Stuff
Secondary Air Bleed Return
Internal bleed air return
Deflections
Shaft Power to Comp
Cranking shaft power
Variable Geometry Schedule
On-wing maintenance actions
Rotor inertia
Noise
Internal Loads
Heat transfer
Heat
Internal bleed air
Variable Geometry Location
Speed parameter
Pressure and temp parameters
Engine Parameters
Services bleed air
Compressed Air

Figure 87 FFD in DRed

Feedback on the improved workflow

After presenting the requirement traceability part of the improved workflow to two system engineers for evaluation, the overall feedback received was positive. Detailed questionnaire results can be found in Appendix 2.

The system engineers strongly agreed that enabling traceability in requirements is essential and on average they rated 8 out of 10 the importance of this concept with 0 being not important at all and 10 being extremely important.

The system engineers raised two questions about the advantages of using DRed over other tools well-known to them. The first question was aimed at understanding DRed capabilities compared to existing tools. This question was raised because the system engineers believed that existing tools, such as IBM DOORS and Artisan Studio for editing SysML, could already achieve rationale capture and enable traceability. This point should be clarified in two ways. First, it was not the intention of the *improved* workflow to replace any existing tools, rather to act as an extension to the requirement analysis process such that requirements with rich rationale can be captured at first instance. Second, both IBM DOORS and SysML have the restriction that all requirements have to be captured in one format only. This restriction already contradicts the original workflow of using different tools (VPA, STA, and FFD) to analyse requirements.

The second question, instead, was aimed at understanding what support there is in DRed to enable requirement traceability in a user-friendly way. This question was raised because DRed currently does not have a user-friendly interface to enable requirement traceability. The process to link two requirements is a multi-step manual process. However, this point should also be clarified in two ways. First, supporting requirement traceability in DRed is still a concept proposal, not an end product. If the proposal is deemed useful, then the usability of this feature in DRed could be improved. Second, it is not compulsory to use DRed for the *proposed* or the *improved* workflow, although DRed is found to be the most suitable. One of the aims of the *proposed* workflow is to highlight the issue that cross-document traceability is often ambiguous and subject to synchronisation error. The *improved* workflow is an attempt to address this issue, but it may not be the final chosen solution.

8.3 Discussion

This research has presented an application of the *proposed* workflow to a large engine development programme. This consisted of creating a fully integrated information space of requirements, which is forward and backward traceable. The main results of the research are discussed below.

First, the research has shown that the *proposed workflow* is still feasible when scaled to a large project, see Table 30. The *proposed* workflow integrates well with the workflow used in the case study, i.e. the *existing* workflow. The integration was achieved with minimal disruption to the existing workflow. The existing workflow was extended instead of being replaced. Hence, if the *existing* workflow could support the management of large number of requirements, so would the *proposed* workflow.

Table 30 Coverage of validation

	Feasibility	Practicality	Scalability
Model requirements in IBIS	✓	✓	✗
Model requirement rationale in IBIS	✓	✓	✗
Enable requirement traceability	✓	✓	✗

Second, it was shown that experts support the goals of the *proposed* workflow. Two experienced system engineers strongly agreed with the need to capture requirement rationale and the need to enable requirement traceability. A similar view is shared by Ramesh and Dhar (1992). They argued for the need to capture rationale along with requirements in large-scale development efforts because as time passes the justifications for decisions may be no longer available when needed.

Third, it has been shown that the use of DRed as the first tool to capture requirements and rationale has also made it possible to enable cross-file traceability. The introduction of DRed, and the use of its hyperlinking capabilities, helped improve cross-file traceability to make it non-ambiguous and more tolerant to requirement updates without losing synchronisation. In addition, fully-traversable sets of requirements enable users to navigate from requirements captured in the final requirement specification to requirements captured in DRed where contextual metadata would be richly documented.

The methodology for data collection and analysis used in this case study is different to that used in the case studies in Chapter 7. In Chapter 7, data was generated by engineers applying the proposed workflow. In this Chapter, requirements (and other metadata) had already been generated; hence the investigation began by reviewing an *existing* workflow used to analyse requirements, then improvements were proposed, and an *improved* workflow was tested for feasibility and scalability. This is a step beyond the method of analysis used in Chapter 6. In Chapter 6, potential improvements to an existing workflow were identified but not tested empirically. The method for data collection used in this Chapter is also significantly more robust than that used in Chapter 5. Compared to Chapter 5, the data in this Chapter are real requirements rather than extrapolated ones. In addition, the target project size is significantly more complex, and the workflow has been reviewed by experts as well as a trial run by expert users.

The data collection methods used in this case study include project document analysis, workshop with system engineers, and an interview with a line engineer. Project documents helped reconstruct the original workflow used for requirement analysis in this engine project. The availability of original requirement documents allowed the researcher to examine the documents in detail and identify areas of potential improvement. Conducting a workshop was effective at receiving feedback directly from experienced system engineers about the *improved* workflow. Finally, the method of interviewing was useful as a way to collect real requirement rationale examples.

8.3.1 Limitations

This investigation has five limitations. First, although the participants to the interviews and the survey questionnaire were experienced, the number of participants is low. There were only two system engineers as the participants in the interview to reconstruct the existing workflow and the questionnaire. The low number of participants may limit the amount of variation in the participants' experiences. For example, both participants are experienced IBM DOORS users; and in their questionnaire responses, both of them compared the proposed workflow to DOORS. In addition, their familiarity with DOORS may have affected their preference of using DOORS to create requirement traceability instead of using DRed. This may explain the significantly higher score they both gave to the requirement rationale aspect of the proposed workflow compared to the traceability aspect. In the interview to reconstruct requirement rationale, only one line engineer participated in the requirement analysis replay session. This prevented the occurrence of scenarios such as requirement debates, as would have happened when there is more than one stakeholder in a real-world requirement analysis session.

Second, although the choice to use DRed has been justified, there was little exploration of alternative software solutions. However, it should be clarified that the purpose of the improvement was to demonstrate the potential of the solution. DRed was chosen as a proof-of-concept, not a final solution. It has been chosen because it is a readily available tool in the collaborating company that is capable of capturing requirement rationales and enabling requirement traceability.

Third, there is insufficient metadata in the existing dataset. Hence, it is difficult to appreciate the importance and value of capturing requirement rationale. But the interview to reconstruct requirement rationale has uncovered a small portion of these requirement rationales that could have been captured.

Fourth, the interview to reconstruct requirement rationales was conducted on an engineer who was recalling requirements, and rationale, rather than eliciting new information. Therefore, the interview scenario may not fully represent a typical requirement analysis session. However, the types of information being captured during the reconstruction interview should be the same as the types likely to be encountered in a real requirement analysis session.

Fifth, there is a large proportion of ambiguous links in the existing dataset, which made it difficult to measure the amount of traceability already existing in the dataset and appreciate the scale of the issue of untraceable requirements. With ambiguous links, it is difficult to tell if two requirements are related or not. These ambiguous links are caused by requirement traceability relying on trace-by-duplication instead of trace-by-reference. The presence of ambiguous traceability links prevents an accurate measure of untraceable requirements; and hinders communicating the benefits of the proposed workflow to potential users.

8.4 Conclusion

In this Chapter, the proposed workflow was validated against data generated from a large-scale engineering project. The validation involved applying the proposed workflow to the existing workflow used in the engineering project, to demonstrate the scalability of the proposed workflow. In addition, improvements to the existing workflow were suggested. The improvement allows rich requirement rationale to be captured when they are most abundant. Requirement traceability was another aspect that was improved. The types of traceability links in the original workflow were converted from trace-by-duplication types to trace-by-reference types. Also, all traceability paths have been redirected to the newly introduced first stage to take advantage of advanced traceability features.

The methodologies applied in this Chapter include project document analysis, workshop with systems engineers, and an interview with a line engineer. Document analysis has been effective at helping the researcher understand the existing workflow used to analyse requirements. From the analysis, the researcher was able to identify areas for improvement and use the proposed workflow to define a new improved workflow. The workshop has been effective at retrieving feedback about the improved workflow from experts. The interview has been effective at validating the improved workflow for feasibility and practicality.

The outcome of the investigation presented in this Chapter can be summarised in three points. First, the proposed workflow was shown to be feasible and practical when scaled to a large project. Second, questionnaire response shows that systems engineering experts agree with the view that rationale and traceability are important elements of a requirement analysis workflow. Third, it has been shown that there is scope to improve requirement rationale capture and requirement traceability; and the solutions to achieve that can be very practical.

This investigation contributes to the overall research project in three ways. First, it demonstrates the scalability of the proposed approach. Second, it has highlighted that capturing requirement rationale and enabling requirement traceability are issues that exist in projects regardless of size and resources. Compared to projects in previous case studies, the whole-engine project has significantly larger amount of resources dedicated to project requirement management. For example there is at least one systems engineer whose full-time role is to facilitate requirement analysis. Yet, the need for more requirement rationale and requirement traceability is still valid. Third, it provides justification that the proposed workflow is addressing issues that are significant and worth pursuing. Industry experts have confirmed the value of the proposed approach. They reaffirmed the need to increase the amount of requirement rationale and requirement traceability, which are the aims of the proposed approach.

Chapter 9 Discussion and conclusions

Chapter outline

This Chapter presents concluding remarks on the research project and it consists of six Sections. Section 1 explains how the aim and objectives have been met. Section 2 discusses the implications and the significance of the research for the wider engineering and research domain. Section 3 highlights the main contributions of this research. Section 4 describes the overall limitations of the research. Section 5 presents a discussion on future research opportunities to extend this research. Finally, Section 6 draws the main conclusions.

9.1 Summary of main results

This dissertation has investigated the field of requirement analysis focusing on engineering design projects. The aim of the research was to investigate how to improve requirement analysis and documentation by supporting computer-based capture of requirement rationale and enabling improved requirement traceability.

In this project, a workflow was proposed with the intention to make requirement rationale capture an important component of the requirement analysis process and enable greater traceability of requirements. The workflow was proposed on the basis of a literature review and observations made in industry about current requirement analysis practice. The workflow has been validated against data from four case studies.

The main results of this research are now presented as answers to the research objectives.

- **Objective A: Develop an understanding of requirements, requirement metadata, the requirement analysis process, and methods and tools to support requirement analysis.**

In Chapter 2, literature relating to requirement analysis was reviewed. The literature review presented relevant work on requirement types, and requirement metadata. The literature review also uncovered that a very limited number of publications exist that provide a comprehensive description of the process of requirement analysis. Most publications view the process as a sub-process of design engineering, systems engineering, or requirements engineering. The process that is the closest representation of the requirement analysis process is Pre-Requirement Specification in requirements engineering, consisting of validation, negotiation, and requirement transformation. Therefore, in this dissertation, a model of the requirement analysis process was defined to consolidate existing knowledge. The model defines requirement analysis as a process through which stakeholder requirements converge into system requirements by requirement checking, requirement structuring, and requirement evolution. The model has been used in subsequent research phases of this project as a definition reference of the requirement analysis process.

The literature review has also analysed popular tools to support requirement analysis. The evaluation found that all tools support either one or more aspects of checking, structuring, and evolution as described by the model of requirement analysis defined earlier. However, there is a gap that is not fulfilled by any of the tools, i.e. the support of requirement rationale capture and traceability in the early phases of requirement analysis. Both requirement rationale and traceability improve the richness of requirement contextual data, and can lead to higher quality requirement documentation. Requirement rationales record the results of checking and structuring requirements, e.g. by recording debates during a check for requirement conflicts, and recording the reasoning to change the structure of a requirement by decomposition. Requirement traceability records the evolution of requirements and allows engineers to navigate requirement evolutions.

- **Objective B: Develop an understanding of requirement analysis in existing engineering practice.**

In Chapter 4, an investigation was conducted to understand the requirement analysis practices of a global engineering organisation. Data was obtained by the researcher actively participating in sessions where engineers elicited and analysed requirements, interviewing systems engineers, and analysing existing requirement documents from various projects. It was found that the current requirement analysis practice consists of a process and a range of associated tools. The process is defined as a series of steps to capture requirements, validate requirements, resolve conflicts in requirements, and flow-down (decompose) requirements. The tools include word processors and spreadsheets, the Function Flow Diagram, the Viewpoint Analysis diagram, the Systemic Textual Analysis table, the Quality Function Deployment table, and IBM DOORS. A particularly significant finding is that requirements captured in one format are frequently transformed into another format in order to facilitate certain requirement analysis tools to be applied.

Analysis of existing practice further confirmed the lack of support for requirement rationale and requirement traceability. Rationale is important because it can support the checking and structuring aspects of the requirement analysis process. It can enhance team communication (Hooks and Farry, 2000) and reduce the risk of starting design based on ambiguous requirements (Ramesh and Dhar, 1992). Requirement traceability is important because it can support the evolution aspect of requirement analysis. It can also enhance team collaboration and improve the quality of requirement documentation.

- **Objective C: Develop a workflow to support the capture of requirement rationale and to enable improved requirement traceability whilst integrating with existing practices.**

In Chapter 4, a workflow for requirement analysis was proposed based on the results from the literature review as well as the investigation of requirement analysis practice in industry. The workflow aims to integrate with existing practices and enable existing tools to support requirement rationale capture and traceability. The workflow advocates requirements to be captured with their rationales as soon as they are elicited. The workflow also advocates requirements to be made traceable from the point they are captured to every point where the requirement has been used in other tools. To realise the workflow, a tool that is capable of supporting both rationale capture and enabling requirement traceability is needed. The tool that satisfies these needs is an IBIS-derivative known as DRed. DRed was chosen because it already supports decision rationale capture in the collaborating company and it has advanced hyperlinking capabilities to enable cross-document traceability.

- **Objective D: Validate the workflow in a range of contexts of increasing complexity.**

In Chapter 5, the feasibility of the proposed workflow was validated employing a reverse-engineering approach. The case study showed that the IBIS-based format and the hyperlinking functionality at the core of the workflow were effective at capturing requirements and rationale, and enabling cross-document requirement traceability. In particular, it was shown how requirements captured in DRed were made traceable to QFD and DOORS.

In Chapter 6, the feasibility of the proposed workflow was further validated. The case study demonstrated that the DRed tool can be used to capture requirements and rationale in the IBIS-format and requirement rationale. Furthermore, analysis of the rationales created by the graduate engineers led to the identification of a set of requirement rationale categories, namely product design and use, pre-existing rationale, and project management. In addition, It was found that in order to fully support requirement traceability, there is a need to develop a system which can capture and trace four types of requirement transformations, namely newly introduced requirements, copied requirements, updated requirements, and deleted requirements.

In Chapter 7, the practicality of the proposed workflow was validated. The case study has shown that two project teams accepted the proposed workflow. In particular, different aspects of the proposed workflow were used by graduate engineers to integrate into their projects with the intention to improve their requirement analysis process. Requirement rationales captured by the graduate engineers were found to align with the rationale categories defined in Chapter 6.

In Chapter 8, the feasibility and scalability of the proposed workflow were validated. The investigation has four main findings. First, an understanding was gained regarding to the existing requirement analysis workflow; it was found that the workflow could benefit from improvement in requirement rationale capture and requirement traceability. Second, the proposed workflow was found to be applicable to the existing workflow, which demonstrates the feasibility and scalability of the proposed workflow. Third, the necessity and aims of the proposed workflow were reaffirmed through questionnaire feedback. Fourth, it was found to be feasible to use an IBIS-based structure to capture requirements and rationale during a requirement analysis session, which is the usage scenario envisioned for the proposed workflow.

9.2 Discussion

This Section discusses four points regarding to the implications of this research on the wider research domain and the justifications for the value in this research.

First, this research advocates a change of perspective on the process of requirement analysis. The research suggests that a shift from documentation of requirements late in the design process to communication and negotiation of requirements early in the design process is possible. As Ramesh and Dhar (1992) stated “in the requirements analysis phase, much of it would involve discussions or deliberations aimed at polishing an initially fuzzy set of requirements into a precise one, and that of making decisions about what is to be modelled and how”. In order to support this shift, the proposed workflow in this research is designed to shift requirement rationale capture and requirement traceability from requirement specification into the phases which gave rise to those requirements. This shift was envisaged by Finkelstein (1991) but it has not materialised until now. An attempt was made in (Rooksby et al., 2006) to introduce an IBIS-based workflow to support requirement analysis.

But the focal point of their research was activities that precede requirement elicitation, rather than requirement analysis. In their research, the goal was to take pre-emptive action to resolve all issues that may lead to requirement conflicts and ambiguity. The proposed workflow improves contemporary tools which often only provide support to document the *result* of requirement analysis sessions rather than the *process* of analysis including reasons for accepting or rejecting requirements, arguments made during requirement debates, and using additional contextual information to clarify requirements. For example, VPA, STA, SysML, and QFD all serve to record the results of performing requirement analysis. In contrast to the contemporary tools, the workflow proposed in this research put emphasis on the capture of processes of to clarify and negotiate requirements between stakeholders during a requirement analysis session.

Second, an aspect of the proposed workflow making it unique is that it is an extension to existing support rather than a replacement. Existing support, such as SysML and DOORS, require requirements to be converted into their respective formats before rationale can be added. This may not always be practical. For example, requirements captured in the SysML format may not be accepted by regulations and compliance authorities (Ferrari et al., 2011). And in case of the DOORS tool, it may not always be possible to access the tool. The proposed workflow can be used harmoniously with existing support without the need to for conversion.

Third, the workflow proposed in this research can be used to increase the quality of documented requirements. The proposed workflow supports requirement rationale capture and requirement traceability, both of which have been found to improve requirement quality (see Section 2.5). Jarke (1998) stated that a set of well-documented requirements allows stakeholders to easily create shared understanding of the issues involved in realising the system vision, such as its functionality, non-functional properties, intended and unintended side effects. Well-documented requirements contribute to meeting the five challenges outlined at the beginning of this dissertation, see Table 31.

Table 31 The five challenges faced by large engineering systems manufacturers

Challenge	Research implications
Challenge 1: Project management complexity and cost How to identify requirement errors early (i.e. what is being built is different to what the stakeholder had wanted)?	Well-documented requirements can help engineers understand the reasoning behind requirements, therefore allowing requirement errors to be identified early
Challenge 2: Evolutionary products and redesign How to increase knowledge re-use?	Well-documented requirements are easier to understand, therefore increase reusability of requirements
Challenge 3: Global product development How to enable teams to collaborate effectively over multiple-geographical locations?	Well-documented requirements can make requirements more self-explanatory, therefore eliminating the overhead of having to consult other people
Challenge 4: Increased outsourcing How to ensure suppliers deliver to specification?	Well-documented requirements ensure suppliers have as much information as they can to understand the specification to avoid requirement errors
Challenge 5: Increasingly transient workforce How to enable greater knowledge transfer?	The creation of well-documented requirements demands requirement contributors to making available their internal knowledge in a transferrable form

Fourth, this research has highlighted the need to capture requirement metadata; the greater availability of requirement metadata opens opportunities to use software to assist the requirement analysis process. An existing branch of requirement engineering research is already exploring ways to use software algorithms to process requirements. Requirements are made recognisable in software by applying language notations in formal logic to requirements to standardise their definitions (Dubois et al, 1998; Abrial, 1996; Spivey, 1988; Lamsweerde, 2001). Software algorithms can already provide assistance in checking for requirement conflicts and ensuring functional requirements are logically valid. For example, the KAOS model (Knowledge Acquisition in autOmated Specification) (Lamsweerde 1991; Lamsweerde, 2009) describes a requirement in terms of “goal”, “object”, “agent”, “action”, or “event” so that logical operations (AND/OR) can be applied to requirements.

9.3 Contributions

This dissertation makes six main contributions.

First, a model of the requirement analysis process was proposed (Chapter 2). The model was formulated through consolidation of the literature related to the design engineering, systems engineering, and requirement engineering processes. The model is new in that it combines commonly agreed components of requirement analysis, and explicitly categorises requirement activities as three types, namely checking, structuring, and evolution. Most processes discussed in the literature only refer to one or two of the three types. For example Kotonya and Sommerville (1998) and Robertson and Robertson (1999) refer to checking only; Hales (2004), Pahl and Beitz (1996), and Pugh (1991) refer to evolution only; and the Vee-model by Forsberg and Mooz (1992) refers to both checking and structuring, while evolution is only implied.

Second, a workflow to conduct requirement analysis was proposed based on the model of requirement analysis, the IBIS concept and hyperlinking mechanisms, and validated through multiple case studies. The workflow establishes the need to extend requirement traceability to the stages prior to requirement specification, and stresses the importance of capturing rationale in order to address the lack of "adequate information about the context and underlying intent of requirements" (Andersson et al., 2003). The workflow extends current practice by introducing the use of the IBIS format to capture requirement rationale and that of advanced hyperlinking technology to enable traceability of requirements.

Third, the data collected in this research enabled the characterisation of the requirement analysis practices of a global engineering company. The case studies in Chapters 4, 6, 7, and 8 contributed empirical examples of processes and tools being used to support requirements analysis. The tools include requirement templates in Microsoft Word and Excel, the Viewpoint Analysis (VPA) diagram, the Systemic Textual Analysis (STA) table, the Quality Function Deployment (QFD) matrix, the Function Flow Diagram (FFD), and the IBM DOORS tool. The processes consist of procedures to use the tools in various ordering configurations.

Fourth, new knowledge was developed of requirement rationales. In Chapter 6, the data demonstrated the existence of rejected requirements, showing that a debate had taken place to invalidate a previously suggested requirement. In addition, requirement rationales have been grouped into three common categories: rationale related to design and operational challenges of the solution to be developed, rationale concerning previously developed products, and rationale about project management issues. In Chapter 7, the data confirms that clarification and debate of requirements has taken place. In Chapter 8, the data also showed that when rationales are elicited, they serve the purpose of clarification, justification, and resolving requirement conflicts.

Fifth, requirement traceability support and new knowledge was developed about requirement evolution. In Chapter 6, the analysis of traceability helped define a model of requirement evolution. In order to fully support requirement traceability, it emerged that it is important to support the capture of four types of requirement transformations: newly introduced requirements, copied requirements, updated requirements, and deleted requirements. In Chapter 7, the data showed scenarios to which requirement traceability could be added. In Chapter 8, the data showed existing practices of enabling requirement traceability.

Sixth, this research has developed an understanding of the requirement analysis process for physical systems. The majority of publications related to supporting requirement rationale and traceability are based on applications to software systems (Burge et al. 2008; Ramesh and Jarke, 1999; Nuseibeh and Easterbrook, 2000; Dutoit et al., 2006; Davis et al., 2006). As an example, out of all the nine publications from the international workshop on Empirical Requirements Engineering 2012, eight are based on investigations of requirements engineering in the software domain (Gross et al., 2012; Bjarnason et al, 2012; Ernst and Murphy, 2012; Knauss et al., 2012; Vara et al., 2012; Morales-Ramirez et al. 2012; Massacci et al., 2012; Hussain et al., 2012). A more specific example is the SEURAT (Software Engineering Using RATionale) system (Burge, 2007) that enables rationale capture within the software programming tool of Eclipse. In comparison, all the case studies in this research investigate mechanical engineering based systems.

9.4 Limitations

This Section addresses some of the general limitations in this research, as the previous Chapters have already reported limitations specific to each case study.

First, the data upon which the research is based come from one company; hence the data could be seen as having limited variation. However, it could be argued that there is a sufficiently high degree of variation in the data because of four reasons. First, the collaborating company is large and it develops a diverse range of engineering products. Second, the sources of data collected in this investigation originate from projects in different domains (including aerospace, power generation, and robotics), and people from different departments. Third, engineers from the company often need to collaborate with outside companies that are also engineering firms, such as airframe manufacturers, power generation providers, and engineering suppliers. Fourth, systems engineers from the company frequently exchange ideas with the wider systems engineering community. Therefore, the data collected in this investigation can be considered as representative of the general requirement analysis in the engineering domain.

Second, the data sets used to validate the workflow are always based on the DRed tool, even though the workflow is not intended to be tool-specific. The workflow has been designed to be compatible with any IBIS-based tool that has the ability to create hyperlinks. Alternative software tools include designVUE (2014) and Compendium (Buckingham-Shum et al., 2006). The reason DRed was chosen to demonstrate the workflow is because of its availability across all computers owned by the collaborating company. DRed had been approved as part of the company's standard toolset.

Third, this investigation has limited data to show the practicality of the workflow, i.e. data describing the applications of the workflow by engineers. One factor contributing to this limitation is that the workflow has not fully materialised into a software tool, due to resource limitations. As a result of the absence of a demonstrable tool, extra effort was needed to attract interest from engineers to participate. An example of a feature that can be made into software is the procedure to create a bi-directional traceability link for a requirement between DRed and QFD. Currently to create such a link, a user would have to perform at least four separate steps, including creating an alias in Excel for a target cell, create a file node in DRed, add hyperlink to Excel cell to DRed file node, add hyperlink to DRed file node to Excel cell. Section 6.2.4 presents a proposal that describes how an automated linking procedure could be achieved in software.

9.5 Future research

This project has produced a workflow that would improve the support for requirement rationale capture and requirement traceability. The workflow has been thoroughly validated for feasibility and scalability by applying it to engineering projects in varying sizes and degree of complexity.

Further work could be undertaken to:

- Understand and validate the assumption that increased requirement rationale and traceability would lead to improved requirement interpretability. This research has been partially based on the assumption that increased requirement rationale and traceability have a positive contribution to the quality of requirement documentation. Although this assumption has been reinforced by similar researches in literature, it remains to be a research gap to put a measurable attribute on the causal relationship between requirement rationale (and traceability) and improved requirement traceability. This research would directly benefit from any reaffirmations in the value of capturing requirement rationale and enabling requirement traceability.

- Validate the practicality of applying the proposed workflow in real-time. The method of validating the proposed workflow has, so far, been mostly centred on using retrospectively generated data.
- Develop the workflow further into a software tool. As noted in Section 9.4, a factor that is currently limiting the attractiveness of the workflow is its accessibility. Users would need some prior knowledge of IBIS, IBIS-based tools, and hyperlinking mechanisms. However, as the proposal in Section 6.2.4 shows, with further investment into packaging this workflow as a software tool, the hurdles can be significantly reduced to attract more users.

9.6 Conclusions

A key challenge in the design of complex engineered systems is that of enabling engineers to capture and analyse requirements from the early phases of a design project and to link such information to more formal representations of system requirements. This would allow manufacturing organisations to create large corporate repositories of requirements which are easily interpretable and highly traceable. In addition, research of existing requirement analysis practices has indicated that there are opportunities for improvement linked to requirement rationale capture and traceability.

This dissertation described a workflow to process stakeholder requirements into systems requirements with an emphasis on capturing requirement metadata. The workflow provides a way to facilitate the capture of requirement rationale, the visualisation and manipulation of requirement structures, and the cross-document traceability of requirements. The application of this workflow ensures that by the time requirements are entered into corporate repositories, they are well-documented and highly traceable.

The proposed workflow has been validated in terms of feasibility, practicality, and scalability. It was demonstrated to be a useful extension of existing requirement analysis support as a way to add requirement rationale capture and enable requirement traceability in existing methods and tools. The collaborating company has shown interest in exploring further this workflow to understand how to integrate it into their current practice. It is, however, important to state that this workflow is not the only workflow being considered, and does not necessarily represent the direction that the collaborating company will pursue in the future.

In recent years, the growth of mobile-based and cloud-based applications have raised the bar on the design of software supporting domain-specific workflows. It is expected that the research in this topic can provide a valuable insight into the design of next generation of requirement analysis supporting tools.

List of publications

Dai W, Aurisicchio M, Armstrong G, An IBIS based approach for the analysis of non-functional requirements, ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers (ASME), 2012.

Dai W, Aurisicchio M, An empirical investigation of requirement evolution in an industrial project, 19th International Conference on Engineering Design (ICED13), 2013.

References

Abrial, J. (1996) The B-Book: assigning programs to meanings. Cambridge: Cambridge University Press.

AIRBUS (2012, May) A350 XWB [Online]. Available: http://www.airbus.com/aircraftfamilies/passengeraircraft/a350xwbfamily. Last accessed 15th July 2014.

Akao, Y., (2004) Quality function deployment: integrating customer requirements into product design, 1st edition, New York, NY: Productivity Press.

Alexander, I. and Stevens, R. (2002) Writing Better Requirements. Harlow: Pearson Education Limited.

Alexander, I. and Beus-Dukic, L. (2009) Discovering Requirements: How to Specify Products and Services. West Sussex: Wiley.

Almefelt, L. (2003) Requirements Management in Theory and Practice - From Requirements to a Balanced Product Concept, Chalmers University of Technology, Göteborg, Sweden.

Andersson, F., Sutinen, K. and Malmqvist, J. (2003) Product Model for Requirements and Design Concept Management: Representing Design Alternatives and Rationale, in International Conference on Systems Engineering.

Aurisicchio, M. and Bracewell, R. (2013) Capturing an integrated design information space with a diagram based approach, in Journal of Engineering Design, vol. 24, no. 6.

Aurisicchio, M., Bracewell, R. and Armstrong, G. (2012) The Function Analysis Diagram, in ASME 2012 International Design Engineering Technical Conferences.

Bieniawski, Z. (1993) Principles and methodology of design for excavations in geologic media, Research in Engineering Design, pp. 49–58, 1993.

Bjarnason, E., Svensson, R. B. and Regnell, B. (2012) Evidence-based timelines for project retrospectives — A method for assessing requirements engineering in context, in Workshop on the Empirical Requirements Engineering.

Blessing, M. and Chakrabarti, A. (2002) DRM: A Design Research Methodology, in Proceedings of International Conference on The Science of Design. London: Springer-Verlag.

Boehm, B., Grunbacher, P., Briggs, O., (2001) Developing groupware for requirements negotiation: lessons learned, Software, IEEE , vol. 18, no. 3, pp. 46-55.

Browne, J. and Zhang, J. (1999) Extended and Virtual Enterprises - Similarities and Differences, International Journal of Agile System and Management, vol. 1, no. 1, pp. 30–36.

Buede, D. (2009) The Engineering Design of Systems: Models and Methods, 2nd edition.

Buckingham-Shum, J., Selvin, M., Haley, B., Nuseibeh, B. and Mistrik, I. (2006) Hypermedia Support for Argumentation-Based Rationale: 15 Years on from gIBIS and QOC, in Rationale Management in Software Engineering, (Eds.) Allen H. Dutoit, Raymond McCall, Ivan Mistrik, and Barbara Paech. Berlin: Spring-Verlag.

Burge, E. (2007) Supporting Requirements Traceability with Rationale, in Proceedings of International Symposium on Grand Challenges in Traceability. pp. 67–75.

Burge, E., Carroll, M., McCall, R., and Mistrík, I. (2008) Rationale and requirements engineering, in Rationale-Based Software Engineering, Berlin: Springer.

Burge, S. (2004) Systemic Textual Analysis (STA), Burge Hughes Walsh.

Burge, S. (2006) Holistic Requirements Model (HRM), Burge Hughes Walsh.

Burge, S. (2007) A Functional Approach to Quality Function Deployment, Burge Hughes Walsh.

Burge, S. (2011a) The Systems Thinking Tool Box: Affinity Diagram (AD), Burge Hughes Walsh.

Burge, S. (2011b) Viewpoint Analysis, Burge Hughes Walsh.

Burge, S. (2011c) Functional Modelling, Burge Hughes Walsh.

Brace, W. and Cheutet, V. (2011) A framework to support requirements analysis in engineering design, Journal of Engineering Design, vol. 23 no. 123, pp. 876-904.

Bracewell, R. and Wallace, K. (2003) A tool for capturing design rationale, in 14th International Conference on Engineering Design (ICED'03), pp. 185–186.

Bracewell, R., Ahmed, S. and Wallace, K. (2004) DRed and design folders: a way of capturing, storing and passing on, knowledge generated during design projects, in ASME International Design Engineering Technical Conferences, pp. 235-246.

Bracewell, R. , Gourtovaia, M., Wallace, K. and Clarkson, P. (2007) Extending Design Rationale to Capture an Integrated Design Information Space, in International Conference on Engineering Design, ICED'07, pp. 85-96.

Bracewell, R., Gourtovaia, M., Moss, M., Knott, D., Wallace, K. and Clarkson, P. J. (2009a) DRed 2.0: a method and tool for capture and communication of design knowledge deliberated in the creation of technical products, in 17th International Conference on Engineering Design ICED'09, pp. 223–234.

Bracewell, R. Wallace, K., Moss, M. and Knott, D. (2009b) Capturing design rationale, Computer-Aided Design, vol. 41, no. 3, pp. 173–186, Mar.

Christel, M. and Kang, K. (1992) Issues in Requirements Elicitation, Carnegie Mellon University.

CIRI (2011) Quality Function Deployment, Creative Industries Research Institute, Mar-2011. [Online]. Available: www.ciri.org.nz/downloads/Quality Function Deployment.pdf.

Cleland-Huang, J. (2005) Toward improved traceability of non-functional requirements, in Proceeding TEFSE '05 Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering, pp. 14 – 19.

Conklin, J. (2005a) Dialogue Mapping: Building Shared Understanding of Wicked Problems. John Wiley & Sons.

Conklin, J. (2005b) Wicked problems and health promotion: reflections on learning, in Dialogue Mapping: Building Shared Understanding of Wicked Problems. In Health promotion journal of Australia: official journal of Australian Association of Health Promotion Professionals, vol. 22, no. 2, pp. 83-4.

Corbett, M. (2010) The Outsourcing Revolution: Why it Makes Sense and How to Do it Right. New York: Simon and Schuster.

Crow, K. (2011) Quality Function Deployment, DRM Associates. [Online]. Available: http://www.ieee.li/tmc/quality_function_deployment.pdf. Last accessed 14th July 2014.

Dai, W., Aurisicchio, M. and Armstrong, G. (2012) An IBIS based Approach for the Analysis of Non Functional Requirements, in ASME 2012 International Design Engineering Technical Conferences.

DAU (1991) Systems engineering fundamentals, Systems engineering fundamentals. Defense Acquisition University.

Davis, A, Dieste, O., Hickey, A. Juristo, N. and Moreno, A. (2006) Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review, 14th IEEE International Requirement Engeering Conference, pp. 179–188.

Davis, A., (2005) Just Enough Requirements Management: Where software Development Meets Marketing. New York: Dorset House.

designVUE (2014, July) Imperial College Design Engineering Group, designVUE. [Online]. Available: http://www3.imperial.ac.uk/designengineering/tools/designvue. Last accessed 15th July 2014.

Dick, J. and Chard, J. (2004) The System Engineering Sandwich: Combining requirements, models and design. Telelogic AB. [Online]. Available: http://www.telelogic.com/download/paper/SystemsEngineeringSandwich.pdf. Last accessed 12th June 2011.

Dick, J. (2005) Design traceability, Software, IEEE, vol. 22, no. 6, pp. 14–16.

Dowlatshahi , S. (1992) Product Design in a Concurrent Engineering Environment - An Optimization Approach, International Journal of Agile System and Management, vol. 30, no. 8, pp. 1803–1818.

Dubois, E.,Yu, E.and Petit, M. (1998) From Early to Late Formal Requirements: a Process Control Case Study. In Proceedings of 9th International Workshop on Software Specification and Design.

Dutoit, A., McCall, R., Mistrik, I. and Paech, B. (2006) Rationale Management in Software Engineering, pp. 34-43.

Easterbrook, S. and Nuseibeh, B. (1995) Managing inconsistencies in an evolving specification, in Proceedings of 2nd International Symposium on Requirements Engineering (RE '95), pp. 48–55.

Eres, M., Bertoni, M., Kossmann, M. and Scanlan, J. (2014) Mapping customer needs to engineering characteristics: an aerospace perspective for conceptual design, Journal of Engineering Design, vol. 25, no. 1-3, pp. 1–24, 2014.

Ernst, N. and Murphy, G. (2012) Case studies in just-in-time requirements analysis, in Workshop on the Empirical Requirements Engineering. Chicago: IEEE, pp. 25-32.

Ferrari, A. , Magnani, G. , Grasso, D., Fantechi, A. and Tempestini, M. (2011) Adoption of model-based testing and abstract interpretation by a railway signalling manufacturer, IJERTCS, vol. 2, no. 2, pp. 42–61.

Finkelstein, A. and Fuks, H. (1989) Multi-Party Specification, in International Workshop on Software Specification & Design, 1989, pp. 185–195.

Finkelstein, A. (1991) Tracing Back from Requirements, in Tools and Techniques for Maintaining Traceability During Design IEE Colloquium, Computing and Control Division.

Forsberg, K. and Mooz, H. (1992) The relationship of systems engineering to the project cycle, Engineering Management Journal , vol. 4, no. 3, pp. 36–43.

French, M. J. (1999) Conceptual Design for Engineers, 3rd edition, London: Springer.

Friedenthal, S., Moore, A. and Steiner, R. (2008) A Practical Guide to SysML: The Systems Modelling Language. Waltham: Morgan Kaufmann.

Friedenthal, S., Moore, A. and Steiner, R. (2009) OMG Systems Modeling Language (OMG SysML) Tutorial, in INCOSE. [Online] available: http://www.omgsysml.org/INCOSE-OMGSysML-Tutorial-Final-090901.pdf. Last accessed 15th July 2014.

Friedenthal, S., Moore, A. and Steiner, R. (2011) A Practical Guide to SysML, 2nd ed. Waltham: Morgan Kaufmann.

Génova, G., Fuentes, J., Llorens, J., Hurtado, O., and Moreno, V. (2013) A framework to measure and improve the quality of textual requirements, Journal of Requirement Engineering, vol. 18, no. 1, pp. 25-41.

Glass, R. (2003) Facts and Fallacies of Software Engineering. Boston: Addison-Wesley Professional

Glazier, L. (2011) Deployment of Requirements Management in Rolls-Royce , INCOSE UK Annual Systems Engineering Conference.

Gotel, O. and Finkelstein, A. (1994) An analysis of the requirements traceability problem, in Proceedings of the First International Conference on Requirements Engineering, 1994, pp. 94–101.

Grady, J., (1993) System Requirements Analysis. Burlington: Elsevier.

Gross, A., Jurkiewicz, J., Doerr, J. and Nawrocki, J. (2012) Investigating the usefulness of notations in the context of requirements engineering. In Workshop on the Empirical Requirements Engineering, pp. 9-16.

Hales, C. (2004) Managing engineering design. 2nd edition. London: Springer.

Hamraz, B., Caldwell, N., Wynn, D. and Clarkson, P. (2013) Requirements-based development of an improved engineering change management method, Journal of Engineering Design, vol. 24, no. 11, pp. 765–793.

Hause, M. and Francis, T. (2008) Building Bridges Between System and Software With SysML and UML, in INCOSE International Symposium.

Herzwurm, G. and Schockert, S. (2006) What are the Best Practices of QFD?, in Transactions from the 12th Int. Symposium on Quality Function Deployment, vol. 49, no. 0.

Heumesser, N., de Mets, A., Demeestere, L., Omasreiter, H. Tavakoli, R. Houdek, F., Weisbrod, J. and Zink, T. (2004) Framework for Requirements.

Hooks, I. and Farry, K. (2000) Customer Centered Products: Creating Successful Products Through Smart Requirements Management, 1st ed. New York: AMACOM/American Management Association.

Hove, D., Goknil, A., Kurtev, I., Berg, K. and Goede, K. (2008) Change impact analysis based on formalization of trace relations for requirements, in Proceedings of the ECMDA Traceability Workshop, pp. 59-75.

Hull, E., Jackson, K., and Dick, J. (2010) Requirements Engineering, 3rd edition. New York: Springer.

Hussain, I., Ormandjieva, O. and Kosseim, L. (2012) LASR: A tool for large scale annotation of software requirements, in Workshop on the Empirical Requirements Engineering, pp 57 – 60.

IBM (2009) Requirements Engineering for Product Development. [Online] Available: ftp://ftp.software.ibm.com/software/plm/solutions/Requirements_Engineering.pdf. Last accessed 15th July 2014.

IBM (2011, May) Rational DOORS. [Online] Available: http://www-03.ibm.com/software/products/en/ratidoor. Last accessed 15th July 2014.

IEEE STD 1220-1998 (1998), Standard for Application and Management of the Systems Engineering Process. New York: IEEE.

INCOSE (2011) Guide for Writing Requirements. Requirements Working Group INCOSE. [Online] availabile: http://www.incose.org/ProductsPubs/products/guideforwritingrequirements.aspx. Last accessed 10th June 2013.

INCOSE (2012 Nov) OMG Systems Modelling Language (OMG SysML) Tutorial, OMG SysML, 2006. [Online]. Available: http://www.omgsysml.org/SysML-Tutorial-Baseline-to-INCOSE-060524-low_res.pdf. Last accessed 10th June 2013.

ISO-15288, AP233 ISO 15288, Systems and Software Engineering - System Life Cycle Processes (15288).

Jarke, M. Informatik, V., Aachen, R. and Bubenko, J. (1993) Theories underlying requirements engineering: an overview of NATURE at genesis, in Proc. of the 1st IEEE Symposium on Requirements Engineering, IEEE Press, New York, pp. 19–35.

Jarke, M. (1998) Requirements Tracing, Communications of the ACM, pp. 32–36.

Knauss, A., Borici, A., Knauss, E. and Damian, D. (2012) Towards understanding requirements engineering in IT ecosystems, in Workshop on the Empirical Requirements Engineering, pp. 33-36.

Koski, J., Parvinen, P., Dean, A., Kontio, J. and Business, S. (2003) Quality Function Deployment in Requirements Engineering: A Review and Case Studies. Published MBA thesis. Executive School of Business. Helsinki University of Technology.

Kossmann, M., Gillies, A., Odeh, M. and Watts, S. (2009) Ontology-driven requirements engineering with reference to the aerospace industry, in ICADIWT ’09. Second International Conference on the Applications of Digital Information and Web Technologies, pp.95–103.

Kotonya, G. and Sommerville, I. (1998) Requirements Engineering: Processes and Techniques, Chichester: John Wiley & Sons.

Krottmaier, H. (2002) Transcluded Documents: Advantages of Reusing Document Fragments, in Proceedings of the 6th International Conference on Electronic Publishing (ELPUB), pp. 359–367.

Kunz, W. and Rittel, H. (1970) Issues as Elements of Information Systems, in Issues as Elements of Information Systems, Working Paper No. 131 (July 1970); Studiengruppe für Systemforschung, Heidelberg, Germany (reprinted, May 1979).

Lamsweerde, A., Dardenne, A. Delcourt, B. and Dubisy, F. (1991) The KAOS Project: Knowledge Acquisition in Automated Specification of Software, in Proc. AAAI Spring Symp. Series, pp. 59–62.

Lamsweerde, A. (2000) Requirements engineering in the year 00: a research perspective, in Proceedings of the 22nd international conference on Software engineering, 2000, pp. 5–19.

Lamsweerde, A. (2001) Goal-Oriented Requirements Engineering: A Guided Tour, in Proceedings 5th International Symposium on Requirements Engineering, pp. 1–13.

Lamsweerde, A. (2009) Requirements engineering: from system goals to UML models to software specifications. New York: Wiley.

Leffingwell, D. and Widrig, D. (2002) The Role of Requirements Traceability in System Development, the rationale edge, e-zine for the rational community, the rational edge. [Online] available: http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep02/TraceabilitySep02.pdf. Last accessed 15th July 2014.

Liang, P., Avgeriou, P. and He, K. (2010) Rationale Management Challenges in Requirements Engineering, in 3rd International Workshop on Managing Requirements Knowledge (MaRK'10), pp. 16-21.

Maffin, D. J. B., (1996) Engineering Design and Product Development in a Company Context, University of Newcastle upon Tyne. Published PhD thesis. Mechanical, Materials and Manufacturing. Newcastle University.

Maheshwari, K. (2008) Design and implementation of workflow for content management system. Published PhD thesis. San Diego State University.

Massacci, F., Nagaraj, D., Paci, F., Ming Sang, T. Le and Tedeschi, A. (2012) Assessing a requirements evolution approach: Empirical studies in the Air Traffic Management domain, in Workshop on the Empirical Requirements Engineering. pp. 49-56.

Mazur, G. (1993) QFD for Service Industries: From Voice of Customer to Task Deployment, in The Fifth Symposium on Quality Function Deployment, pp. 1-17.

Mead, N. (2008) Requirements Elicitation Case Studies Using IBIS , JAD , and ARM. Department of Homeland Security. [Online] available: https://buildsecurityin.us-cert.gov/articles/best-practices/requirements-engineering/requirements-elicitation-case-studies-using-ibis-jad-and-arm. Last accessed 10th July 2014.

Miles, B. and Swift, K. (1998) Design for Manufacture and Assembly, Manufacturing Engineer, vol. 77, no. 5, pp. 221–224.

Morales-Ramirez, I., Vergene, M., Morandini, M. Sabatucci, L., Perini, A. and Susi, A. (2012) Revealing the obvious?: A retrospective artefact analysis for an ambient assisted-living project, in Workshop on the Empirical Requirements Engineering, pp. 41-48.

Nguyen, L. and Swatman, P. (2003) Managing the requirements engineering process, Journal of Requirements Engineering, vol. 8, issue 1, pp. 55–68.

Nuseibeh, B. and Easterbrook, S. (2000) Requirements engineering: a roadmap, in Proceedings of the Conference on the Future of Software Engineering, vol. 1, pp. 35–46.

Olivier, W. (2009) Fundamentals of Systems Engineering: Requirements Definition, MIT Open Courseware. [Online] available: http://ocw.mit.edu/courses/aeronautics-andastronautics/16-885j-aircraft-systems-engineering-fall-2005/readings/sefguide_01_01.pdf. Last accessed 15th July 2014.

OMG (2008) OMG Systems Modeling Language (OMG SysML) Hybrid SUV Non-Normative Example. [Online] available http://www.omg.org/ocsmp/HSUV.pdf. Last accessed 15th July 2014.

OMG (2012) OMG Systems Modeling Language Version 1.3. [Online] available: http://www.sysml.org/docs/specs/OMGSysML-v1.3-12-06-02.pdf. Last accessed 15th July 2014.

Otto, K. and Wood, K. (2001) Product design: Techniques in reverse engineering and new product development. Upper Saddle River, NJ: Prentice Hall.

Paetsch, F. (2003) Requirements Engineering in Agile Software Development. In Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 308-313.

Pahl, G. , and Beitz, W. (1996) Engineering Design, editted by Wallace, K. M., 2nd edition, London: Springer-Verlag.

Pohl, K. (1993) The Three Dimensions of Requirements Engineering, 5th International Conference on Advanced Information Systems Engineering, CAiSE'93 Paris, France, pp. 275-292.

Pugh, S. (1991), Total design: integrated methods for successful product engineering. Harlow, Essex: Addison-Wesley.

Pulham, J. (2008) Visualizing DOORS Information and Traceability with DOORS/Traceline, Innovation2008 Telelogic User Group Conference. [Online] available: http://download-na.telelogic.com/download/ugcagenda/Jared_Pulham_Visualizing_DOORS_Information_and_Traceability.pdf. Last accessed 15th July 2014.

Ramesh, B. and Dhar, V. (1992) Supporting Systems Development by Capturing Deliberations During Requirements Engineering, IEEE Transactions on Software Engineering - Special issue on knowledge representation and reasoning in software development, vol. 18, no. 6, pp. 498–510.

Ramesh, B. and Edwards, M. (1993) Issues in the Development of a Requirements Traceability Model, in Requirements Engineering, 1993., Proceedings of IEEE International Symposium on, pp. 256–259.

Ramesh, B. and Jarke, M. (1999) Towards reference models for requirements traceability, Software Engineering, IEEE Transactions on Software Engineering, vol. 27, no. 1, pp. 58–93.

Reich, Y. (1994) What is wrong with CAE and can it be fixed?, in Preprints of Bridging the Generations: An International Workshop on the Future Directions of Computer-Aided Engineering, pp. 237–242.

Robertson, S. and Robertson, J, (1999) Mastering the Requirements Process. 2nd edition. Boston, MA: Addison Wesley.

Rooksby, J. Sommerville, I. and Pidd, M. (2006) A Hybrid Approach to Upstream Requirements : IBIS and Cognitive Mapping, in Rationale management in software engineering, A. H. Dutoit, R. McCall, I. Mistrik, and B. Paech, Eds. Springer Berlin Heidelberg, 2006, pp. 137–154.

Rzepka, W. (1989) A requirements engineering test bed: concept, status and first results, in System Sciences, 1989. Vol.II: Software Track, Proceedings of the Twenty-Second Annual Hawaii International Conference on Systems Science, vol.2, pp. 339–347.

Sage, A. P. (1992) Systems Engineering. New York, NY: Wiley-Blackwell.

Selvin, A., Shum, S., Sierhuis, M., Conklin, J., Zimmermann, B., Palus, C., Drath, W., Horth, D., Domingue, J., Motta, E. and Li, G. (2001) Compendium : Making Meetings into Knowledge Events, in Knowledge Technologies, pp. 1–13.

Sharp, A. and McDermott, P. (2008) Workflow Modeling: Tools for Process Improvement and Applications Development, 2nd revision. Norwood, MA: Artech House.

Spivey, J. (1988) Understanding Z. Cambridge: Cambridge University Press.

Stoller, R. (1988) Tracer: a tool for tracing and control in Engineering Management Conference, 1988. Engineering Leadership in the 90's, pp. 27–36.

SysML.org (2006) Systems Modelling Language (SysML) Specification. [Online] available: http://www.sysml.org/docs/specs/SysMLv1.0a-051114R1.pdf. Last accessed 15th July 2014.

Tague, N. (2005) The Quality Toolbox, 2nd ed. Milwaukee, WI: ASQ Quality Press.

Vara, J., Hoyo, L., Collado, E. and Sabetzadeh, M. (2012) Towards customer-based requirements engineering practices, in Workshop on the Empirical Requirements Engineering, pp. 37-40.

Volere (2014, May) Volere Requirements Snowcard. [Online]. Available: http://www.volere.co.uk/snowcard.pdf. Last accessed 15th July 2014.

Wiegers, K. (1999a) Automating Requirements Management, Journal of Software Development, vol. 7, no. 7, pp. S1–S5.

Wiegers, K. (1999b) Software Requirements. Redmont: Microsoft Press, Redmont.

Wiegers, K. (2000) When Telepathy Won't Do: Requirements Engineering Key Practices, Cutter IT Journal. STQE magazine produced by STQE Publishing. [Online] available http://www.cs.nott.ac.uk/~jds/teaching/archive/RequirementsTraps.pdf. Last accessed 15th July 2014.

Wynn, D. and Clarkson, J. (2005) Models of designing, in Design Process Improvement - A review of Current Practice, Clarkson, P. and Eckert, C., pp 34-59. In Design process improvement: A review of current practice. London: Springer London.

Young, R. (2003), The Requirements Engineering Handbook. Norwood, MA: Artech House.

Glossary of terms

Existing workflow: The existing workflow used to analyse requirements in the engine project in Chapter 8.

Feasibility: Demonstration of the conceptual soundness of the workflow by testing it with an engineering design dataset.

Graduate engineers: Engineers who have recently joined the company as a graduate. They are distinguished from line engineers who typically have much more industry experience.

Improved workflow: A workflow that is defined in Chapter 8 through identifying improvements to the existing workflow and borrowing features from the proposed workflow.

Line engineers: Engineers who have industry experience and are typically expected to work on business-critical projects.

Practicality: Demonstration of the conceptual soundness and industrial relevance of the workflow by testing through practicing engineers applying the workflow.

Proposed workflow: The workflow proposed in Section 4.4, which is validated by case studies in Chapters 5, 6, 7, and 8.

Requirement engineer: Any person who took part in requirement analysis can be considered as a requirement engineer. This engineer is usually involved in the creation or modification of requirements.

Requirement evolution: The transformations that requirements are subject to as a result of analysis. During a transformation an existing requirement is updated and refined. Requirement evolution is related to requirement traceability. Traceability is the ability to navigate the evolution of requirements.

Requirement metadata: Any supplementary information enriching requirements. Requirement rationale and traceability information are considered as types of requirement metadata.

Requirement rationale: Justification to support or oppose a requirement as well as detailing alternative requirements considered in the design process.

Requirement traceability: The ability to navigate forward and backward requirement models by means of capturing information to trace the evolution of requirements.

Scalability: Demonstration of the conceptual soundness, industrial relevance and complexity process of the workflow by testing it on large engineering programme in industry.

Support tool: Support for requirement analysis in the form of a package that consists one (or a combination of) method of analysis, format of requirement capture, and software.

System engineers: System engineers are trained and have knowledge of the latest company best practices. They are the facilitators when line engineers elicit and analyse requirements, effectively acting as the "glue" between different engineering teams. The system engineers also manage all the requirements agreed by line engineers, ensuring the requirements are readily accessible and up-to-date.

Workflow: A term used to describe the procedural steps and tools needed for each step in a business process.

Appendix 1

Hair-dryer case study coding reliability

The contents of this appendix show the questionnaire given to participants to verify coding reliability of data generated through reverse-engineering. The methodology of this investigation is shown in Section 5.1.1. The content that follows is structured as:

- Questionnaire Part 1 (Questions)
- Questionnaire Part 1 (Solution)
- Questionnaire Part 1 (Response 1)
- Questionnaire Part 1 (Response 2)
- Questionnaire Part 2 (Questions)
- Questionnaire Part 2 (Solution)
- Questionnaire Part 2 (Response 1)
- Questionnaire Part 2 (Response 2)

Questionnaire Part 1 (Questions)

This research focuses on the capture of requirements and the rationale to justify them. To validate a case study I worked on I have developed a questionnaire, which I am inviting you to fill in. In the case study, a hair-dryer was reverse engineered to identify the requirements that would have been defined by the design team. For each requirement, one or more rationale statements were also identified. The questionnaire is divided into two parts each of which will take you approximately 10 minutes to complete.

The specific aim of part 1 of the questionnaire is to validate if the rationale statements are *related* to their corresponding requirements. A rationale is *related* to a requirement if any of the following is true:

- The rationale justifies the necessity of the requirement (i.e. the rationale provides evidence that a requirement is needed). An example of this could be a rationale statement that describes a use case.
- The rationale justifies the feasibility of the requirement (i.e. the rationale provides evidence that it can be satisfied and the constraints are achievable). An example of this could be a rationale statement that describes an existing product which already satisfies the requirement.
- The rationale clarifies the requirement (i.e. the rationale supplies additional information to explain the requirement). An example of this could be a rationale statement that adds contextual information including the person responsible for the definition of the requirement, and the requirement source.

For this questionnaire, there exists a list of requirement-rationale pairs, see Table 1. The first column of Table 1 lists the requirements and the second column lists the rationales.

For each requirement-rationale pair, the rationale is supposed to justify or clarify the requirement. Your task is to validate the requirement-rationale pairs in Table 1 by accepting or rejecting the existence of a relationship. This can be done by writing YES (accept relationship) or NO (reject relationship) in the third column in Table 1.

As an example take the first requirement “seldom breakdown” related to the hair dryer in consideration. It implies that the hair dryer must not breakdown frequently. The rationale linked to it is “low breakdown rate is good for the manufacturer’s reputation”. This rationale can be regarded as related to the requirement because it justifies the necessity of the requirement, and so a **YES** is filled in the third column.

Table 1 Requirement-rationale pairs, third column is to be completed

Requirements	**Rationales**	**Is the rationale related to the requirement? (if YES fill in the column to the right, if NO proceed to next line)**
Seldom breakdown	Low breakdown rate is good for the manufacturer’s reputation	YES
Seldom breakdown	Low breakdown rate reduces the cost of servicing for the manufacturer to replace broken down products, i.e. staff time to serve customers with broken down products	
Seldom breakdown	Maximise accessibility to the product. Increases the range of customers this product is available to. Users have different arm strengths, the product should be designed so that even the users with the lowest strength could use it.	
Seldom breakdown	It must last at least 5 years without break down, because the company has committed to a warranty period of 5 years. A breakdown within the warranty period will incur the manufacture additional cost of providing a replacement	
Well balanced	Well balanced product is comfortable to hold, giving users a good experience	
Comfortable hand grip	Comfortable grip enables prolonged use for drying long hair, giving users a good experience	
Comfortable hand grip	This is good ergonomics, and is a sign of good quality which increases the reputation of the product	
Comfortable hand grip	Could compromise the strength of casing, and ultimately affect the durability of the product, giving users a bad experience, thus a bad reputation for the manufacturer	
Comfortable hand grip	Comfortable grip enables prolonged use for drying clothes without tiring the arm. This is a sign of good design, and gives the user a good experience.	
Light weight	More portable and can be used for travel. Portability is a selling point for this product.	
Light weight	Low breakdown rate reduces the cost for the manufacture, as there is less	

	broken down products to replace. Company has decided to have no servicing department in order to save costs, therefore a failure during the warranty will cause the company to cover the cost of replacement	
Light weight with a max weight of 0.8kg	0.8kg is considered light weight as it is lower than the average weight of similar products on the market	
Light weight with a max weight of 0.8kg	0.8kg may be difficult and costly to achieve considering only a few other manufacturers have achieved it	
Light weight	Could require more expensive materials for the components. Less profit for the manufacturer.	
Light weight	Easier to transport for the distributor and retailers	
Light weight	Would be suitable for placing the product on a stand. Potentially saving the retailer the cost of making a customised stand.	
Light weight	Without this requirement, uncomfortable hand grip could cause the user to loosely grip the product, which might cause it to be dropped accidentally, e.g. during wet hands use. Persistent dropping could damage the product, and give users a bad experience, thus bring negative reviews of the product.	
Light weight	Low price is needed to attract customers, especially considering this is a new product model with no previous reputation	
Minimise risk of electrocution	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	
Prevent circuit overheating	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	
Prevent circuit overheating	Cost more to design	
Prevent circuit overheating	Cost more to manufacture	
Prevent circuit overheating	Without this requirement, overheating could break the product permanently, giving users a bad experience and decrease the reputation of the	

	manufacturer	
Prevent circuit overheating	15GBP is considered a low price by the general consumers	
Low price	Most of users would pay between 16-20GBP	
Low price	The lower the price the lower the profit margin for the manufacturer	
Low price, cost no more than 15GBP	Without this requirement, overheating could start a fire, and potentially give the manufacturer a bad reputation	
Low price, cost no more than 15GBP	15GBP is low price statistically as it is below the median price in a survey of similar products	
Low price	Will not tire the arm after holding it for a long time. This is a sign of good design, and gives the user a good experience.	
Advanced in technology	Advanced in technology is more competitive, therefore more likely to outsell against competing products	
Advanced in technology	Advanced in technology may be costly to manufacture, and therefore conflict with the requirement of low price	
Advanced in technology	Customers may not find advanced in technology more preferable than a simple to operate product	

Questionnaire Part 1 (Solution)

Requirements	**Rationales**	**Is the rationale related to the requirement? (if YES fill in the column to the right, if NO proceed to next line)**
Seldom breakdown	Low breakdown rate is good for the manufacturer's reputation	YES
Seldom breakdown	Low breakdown rate reduces the cost of servicing for the manufacturer to replace broken down products, i.e. staff time to serve customers with broken down products	YES
Seldom breakdown	Maximise accessibility to the product. Increases the range of customers this product is available to. Users have different arm strengths, the product should be designed so that even the users with the lowest strength could use it.	NO
Seldom breakdown	It must last at least 5 years without break down, because the company has committed to a warranty period of 5 years. A breakdown within the warranty period will incur the manufacture additional cost of providing a replacement	YES
Well balanced	Well balanced product is comfortable to hold, giving users a good experience	YES
Comfortable hand grip	Comfortable grip enables prolonged use for drying long hair, giving users a good experience	YES
Comfortable hand grip	This is good ergonomics, and is a sign of good quality which increases the reputation of the product	YES
Comfortable hand grip	Could compromise the strength of casing, and ultimately affect the durability of the product, giving users a bad experience, thus a bad reputation for the manufacturer	NO
Comfortable hand grip	Comfortable grip enables prolonged use for drying clothes without tiring the arm. This is a sign of good design, and gives the user a good experience.	YES
Light weight	More portable and can be used for travel. Portability is a selling point for this product.	YES
Light weight	Low breakdown rate reduces the cost for the manufacture, as there is less	NO

	broken down products to replace. Company has decided to have no servicing department in order to save costs, therefore a failure during the warranty will cause the company to cover the cost of replacement	
Light weight with a max weight of 0.8kg	0.8kg is considered light weight as it is lower than the average weight of similar products on the market	YES
Light weight with a max weight of 0.8kg	0.8kg may be difficult and costly to achieve considering only a few other manufacturers have achieved it	YES
Light weight	Could require more expensive materials for the components. Less profit for the manufacturer.	YES
Light weight	Easier to transport for the distributor and retailers	YES
Light weight	Would be suitable for placing the product on a stand. Potentially saving the retailer the cost of making a customised stand.	YES
Light weight	Without this requirement, uncomfortable hand grip could cause the user to loosely grip the product, which might cause it to be dropped accidentally, e.g. during wet hands use. Persistent dropping could damage the product, and give users a bad experience, thus bring negative reviews of the product.	NO
Light weight	Low price is needed to attract customers, especially considering this is a new product model with no previous reputation	NO
Minimise risk of electrocution	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	YES
Prevent circuit overheating	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	YES
Prevent circuit overheating	Cost more to design	YES
Prevent circuit overheating	Cost more to manufacture	YES
Prevent circuit overheating	Without this requirement, overheating could break the product permanently, giving users a bad experience and decrease the reputation of the	YES

	manufacturer	
Prevent circuit overheating	15GBP is considered a low price by the general consumers	NO
Low price	Most of users would pay between 16-20GBP	YES
Low price	The lower the price the lower the profit margin for the manufacturer	YES
Low price, cost no more than 15GBP	Without this requirement, overheating could start a fire, and potentially give the manufacturer a bad reputation	NO
Low price, cost no more than 15GBP	15GBP is low price statistically as it is below the median price in a survey of similar products	YES
Low price	Will not tire the arm after holding it for a long time. This is a sign of good design, and gives the user a good experience.	NO
Advanced in technology	Advanced in technology is more competitive, therefore more likely to outsell against competing products	YES
Advanced in technology	Advanced in technology may be costly to manufacture, and therefore conflict with the requirement of low price	YES
Advanced in technology	Customers may not find advanced in technology more preferable than a simple to operate product	YES

Table 2 Requirement-rationale pairs, third column is to be completed

Requirements	**Rationales**	**Is the rationale a pro or a con with respect to the requirement?**
Seldom breakdown	Low breakdown rate is good for the manufacturer's reputation	Pro
Seldom breakdown	Low breakdown rate reduces the cost of servicing for the manufacturer to replace broken down products, i.e. staff time to serve customers with broken down products	
Seldom breakdown	Low breakdown rate reduces the cost for the manufacture, as there is less broken down products to replace. Company has decided to have no servicing department in order to save costs, therefore a failure during the warranty will cause the company to cover the cost of replacement	
Seldom breakdown	It must last at least 5 years without break down, because the company has committed to a warranty period of 5 years. A breakdown within the warranty period will incur the manufacture additional cost of providing a replacement	
Well balanced	Well balanced product is comfortable to hold, giving users a good experience	
Comfortable hand grip	Comfortable grip enables prolonged use for drying long hair, giving users a good experience	
Comfortable hand grip	This is good ergonomics, and is a sign of good quality which increases the reputation of the product	
Comfortable hand grip	Without this requirement, uncomfortable hand grip could cause the user to loosely grip the product, which might cause it to be dropped accidentally, e.g. during wet hands use. Persistent dropping could damage the product, and give users a bad experience, thus bring negative reviews of the product.	
Comfortable hand grip	Comfortable grip enables prolonged use for drying clothes without tiring the arm. This is a sign of good design, and gives the user a good experience.	
Light weight	More portable and can be used for travel. Portability is a selling point for this product.	
Light weight	Maximise accessibility to the product. Increases the range of customers this	

	product is available to. Users have different arm strengths, the product should be designed so that even the users with the lowest strength could use it.	
Light weight with a max weight of 0.8kg	0.8kg is considered light weight as it is lower than the average weight of similar products on the market	
Light weight with a max weight of 0.8kg	0.8kg may be difficult and costly to achieve considering only a few other manufacturers have achieved it	
Light weight	Could require more expensive materials for the components. Less profit for the manufacturer.	
Light weight	Easier to transport for the distributor and retailers	
Light weight	Would be suitable for placing the product on a stand. Potentially saving the retailer the cost of making a customised stand.	
Light weight	Could compromise the strength of casing, and ultimately affect the durability of the product, giving users a bad experience, thus a bad reputation for the manufacturer	
Light weight	Will not tire the arm after holding it for a long time. This is a sign of good design, and gives the user a good experience.	
Minimise risk of electrocution	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	
Prevent circuit overheating	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	
Prevent circuit overheating	Cost more to design	
Prevent circuit overheating	Cost more to manufacture	
Prevent circuit overheating	Without this requirement, overheating could break the product permanently, giving users a bad experience and decrease the reputation of the manufacturer	
Prevent circuit overheating	Without this requirement, overheating could start a fire, and potentially give the manufacturer a bad reputation	
Low price	Most of users would pay between 16-20GBP	
Low price	The lower the price the lower the profit margin for the manufacturer	
Low price, cost no more than 15GBP	15GBP is considered a low price by the general consumers	
Low price, cost no	15GBP is low price statistically as it is	

more than 15GBP	below the median price in a survey of similar products	
Low price	Low price is needed to attract customers, especially considering this is a new product model with no previous reputation	
Advanced in technology	Advanced in technology is more competitive, therefore more likely to outsell against competing products	
Advanced in technology	Advanced in technology may be costly to manufacture, and therefore conflict with the requirement of low price	
Advanced in technology	Customers may not find advanced in technology more preferable than a simple to operate product	

Questionnaire Part 1 (Participant 1)

Requirements	**Rationales**	**Is the rationale related to the requirement? (if YES fill in the column to the right, if NO proceed to next line)**
Seldom breakdown	Low breakdown rate is good for the manufacturer's reputation	YES ✔
Seldom breakdown	Low breakdown rate reduces the cost of servicing for the manufacturer to replace broken down products, i.e. staff time to serve customers with broken down products	YES ✔
Seldom breakdown	Maximise accessibility to the product. Increases the range of customers this product is available to. Users have different arm strengths, the product should be designed so that even the users with the lowest strength could use it.	YES ✘
Seldom breakdown	It must last at least 5 years without break down, because the company has committed to a warranty period of 5 years. A breakdown within the warranty period will incur the manufacture additional cost of providing a replacement	YES ✔
Well balanced	Well balanced product is comfortable to hold, giving users a good experience	YES – BUT IT DEPENDS ON THE PRODUCT. ✔
Comfortable hand grip	Comfortable grip enables prolonged use for drying long hair, giving users a good experience	NOT – LONG PERIODS MAKE THE USER FEEL TIRED ✘
Comfortable hand grip	This is good ergonomics, and is a sign of good quality which increases the reputation of the product	YES ✔
Comfortable hand grip	Could compromise the strength of casing, and ultimately affect the durability of the product, giving users a bad experience, thus a bad reputation for the manufacturer	NOT – DURABILITY IS A BIG ISSUE, IF THE PRODUCT IS GOOD DURABILITY IS IMPORTANT BUT IF IT IS NOT GOOD IT IS NOT IMPORTANT ✔
Comfortable hand grip	Comfortable grip enables prolonged use for drying clothes without tiring the arm. This is a sign of good design, and gives the user a good experience.	NOT – GOOD USER EXPERIENCE MAY BE RELATED TO USING THE PRODUCT LESS ✘
Light weight	More portable and can be used for travel. Portability is a selling point for	YES - BUT MAKING IT PORTABLE AFFECT OTHER

	this product.	ASPECTS OF THE PRODUCT ✓
Light weight	Low breakdown rate reduces the cost for the manufacture, as there is less broken down products to replace. Company has decided to have no servicing department in order to save costs, therefore a failure during the warranty will cause the company to cover the cost of replacement	NOT – IF YOU INVEST IN THE QUALITY OF THE PRODUCT THE MANUFACTURING IS MORE EXPENSIVE. THE SERVICE OF REPARING IS ANOTHER THING ✓
Light weight with a max weight of 0.8kg	0.8kg is considered light weight as it is lower than the average weight of similar products on the market	YES ✓
Light weight with a max weight of 0.8kg	0.8kg may be difficult and costly to achieve considering only a few other manufacturers have achieved it	YES ✓
Light weight	Could require more expensive materials for the components. Less profit for the manufacturer.	YES ✓
Light weight	Easier to transport for the distributor and retailers	YES ✓
Light weight	Would be suitable for placing the product on a stand. Potentially saving the retailer the cost of making a customised stand.	YES – BUT SOMETIMES THE PRODUCER WANTS BIG PACKAGES, E.G. LEGO ✓
Light weight	Without this requirement, uncomfortable hand grip could cause the user to loosely grip the product, which might cause it to be dropped accidentally, e.g. during wet hands use. Persistent dropping could damage the product, and give users a bad experience, thus bring negative reviews of the product.	YES ✗
Light weight	Low price is needed to attract customers, especially considering this is a new product model with no previous reputation	NOT – IT DEPENDS OF HOW THE PRODUCT IS TARGETED ✓
Minimise risk of electrocution	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	YES ✓
Prevent circuit overheating	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	YES ✓
Prevent circuit overheating	Cost more to design	PROBABLY YES – IT DEPENDS ON THE DESIGN SOLUTION ✓

Prevent circuit overheating	Cost more to manufacture	PROBABLY NO – IF IT IS WELL DESIGNED ✘
Prevent circuit overheating	Without this requirement, overheating could break the product permanently, giving users a bad experience and decrease the reputation of the manufacturer	YES ✔
Prevent circuit overheating	15GBP is considered a low price by the general consumers	YES ✘
Low price	Most of users would pay between 16-20GBP	YES ✔
Low price	The lower the price the lower the profit margin for the manufacturer	USUALLY YES - BUT IF YOU SELL LOTS OF PRODUCTS THEN THE PROFIT IS HIGH ✔
Low price, cost no more than 15GBP	Without this requirement, overheating could start a fire, and potentially give the manufacturer a bad reputation	NOT – SAFETY IS IMPORTANT AND MAY HAVE IMPACT ON THE COST ✔
Low price, cost no more than 15GBP	15GBP is low price statistically as it is below the median price in a survey of similar products	YES ✔
Low price	Will not tire the arm after holding it for a long time. This is a sign of good design, and gives the user a good experience.	NOT TIRING THE ARM IS A GOOD SIGN, BUT IT CAN AFFECT THE PRICE ✔
Advanced in technology	Advanced in technology is more competitive, therefore more likely to outsell against competing products	YES IF USED CORRECTLY, FOR EXAMPLE DRYING THE HAIR QUICKLY ✔
Advanced in technology	Advanced in technology may be costly to manufacture, and therefore conflict with the requirement of low price	YES ✔
Advanced in technology	Customers may not find advanced in technology more preferable than a simple to operate product	YES – NEWER MEANS THAT THEY HAVE TO LEARN TO USE A PRODUCT, WELL-KNOWN IS SOMETHING LEARNED ✔

25 Agreed with solution

6 Disagreed with solution

Questionnaire Part 1 (Participant 2)

Requirements	**Rationales**	**Is the rationale related to the requirement? (if YES fill in the column to the right, if NO proceed to next line)**
Seldom breakdown	Low breakdown rate is good for the manufacturer's reputation	YES
Seldom breakdown	Low breakdown rate reduces the cost of servicing for the manufacturer to replace broken down products, i.e. staff time to serve customers with broken down products	YES ✓
Seldom breakdown	Maximise accessibility to the product. Increases the range of customers this product is available to. Users have different arm strengths, the product should be designed so that even the users with the lowest strength could use it.	NO ✓
Seldom breakdown	It must last at least 5 years without break down, because the company has committed to a warranty period of 5 years. A breakdown within the warranty period will incur the manufacture additional cost of providing a replacement	YES ✓
Well balanced	Well balanced product is comfortable to hold, giving users a good experience	YES ✓
Comfortable hand grip	Comfortable grip enables prolonged use for drying long hair, giving users a good experience	YES ✓
Comfortable hand grip	This is good ergonomics, and is a sign of good quality which increases the reputation of the product	YES ✓
Comfortable hand grip	Could compromise the strength of casing, and ultimately affect the durability of the product, giving users a bad experience, thus a bad reputation for the manufacturer	YES ✗
Comfortable hand grip	Comfortable grip enables prolonged use for drying clothes without tiring the arm. This is a sign of good design, and gives the user a good experience.	YES ✓
Light weight	More portable and can be used for travel. Portability is a selling point for this product.	YES ✓
Light weight	Low breakdown rate reduces the cost for the manufacture, as there is less	NO ✓

	broken down products to replace. Company has decided to have no servicing department in order to save costs, therefore a failure during the warranty will cause the company to cover the cost of replacement	
Light weight with a max weight of 0.8kg	0.8kg is considered light weight as it is lower than the average weight of similar products on the market	YES ✔
Light weight with a max weight of 0.8kg	0.8kg may be difficult and costly to achieve considering only a few other manufacturers have achieved it	NO ✘
Light weight	Could require more expensive materials for the components. Less profit for the manufacturer.	YES ✔
Light weight	Easier to transport for the distributor and retailers	YES ✔
Light weight	Would be suitable for placing the product on a stand. Potentially saving the retailer the cost of making a customised stand.	NO ✘
Light weight	Without this requirement, uncomfortable hand grip could cause the user to loosely grip the product, which might cause it to be dropped accidentally, e.g. during wet hands use. Persistent dropping could damage the product, and give users a bad experience, thus bring negative reviews of the product.	YES ✘
Light weight	Low price is needed to attract customers, especially considering this is a new product model with no previous reputation	NO ✔
Minimise risk of electrocution	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	YES ✔
Prevent circuit overheating	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	YES ✔
Prevent circuit overheating	Cost more to design	YES ✔
Prevent circuit overheating	Cost more to manufacture	YES ✔
Prevent circuit overheating	Without this requirement, overheating could break the product permanently, giving users a bad experience and decrease the reputation of the	YES ✔

	manufacturer	
Prevent circuit overheating	15GBP is considered a low price by the general consumers	NO ✔
Low price	Most of users would pay between 16-20GBP	YES ✔
Low price	The lower the price the lower the profit margin for the manufacturer	YES ✔
Low price, cost no more than 15GBP	Without this requirement, overheating could start a fire, and potentially give the manufacturer a bad reputation	NO ✔
Low price, cost no more than 15GBP	15GBP is low price statistically as it is below the median price in a survey of similar products	YES ✔
Low price	Will not tire the arm after holding it for a long time. This is a sign of good design, and gives the user a good experience.	NO ✔
Advanced in technology	Advanced in technology is more competitive, therefore more likely to outsell against competing products	YES ✔
Advanced in technology	Advanced in technology may be costly to manufacture, and therefore conflict with the requirement of low price	YES ✔
Advanced in technology	Customers may not find advanced in technology more preferable than a simple to operate product	YES ✔

27 Agreed with solution

4 Disagreed with solution

Questionnaire Part 2 (Questions)

The specific aim of part 2 of the questionnaire is to validate whether the rationale statement related to its corresponding requirement is pro or con.

A pro rationale is an argument which agrees to the necessity or feasibility of a given requirement. In other words, for any pro rationale one should be able to complete any of the following sentences:

- “this requirement is necessary because… [rationale describing the necessity of the requirement]”;
- “this requirement is necessary because in its absence… [rationale describing a detrimental effect resulting from the absence of the requirement]”;
- “this requirement is feasible because…[rationale describing the feasibility of the requirement]”.

A con rationale is an argument which disagrees the necessity or feasibility of a given requirement.

For this questionnaire, there exists a list of valid requirement-rationale pairs (you may remember that some pairs were not valid in part 1 of the questionnaire, these pairs have now been corrected), see Table 2. The first column of Table 2 lists the requirements and the second column lists the rationales.

Your task is to recognise the rationale type by classifying it as a pro or a con in relation to the requirement. This can be done by writing PRO or CON in the third column in Table 2.

As an example, take the first requirement “seldom breakdown” related to the hair dryer in consideration. It implies that the hair dryer must not breakdown frequently, and the rationale to justify the necessity of this requirement is the “manufacturer’s reputation is at risk”. This rationale can be regarded as a **pro** argument.

Table 2 Requirement-rationale pairs, third column is to be completed

Requirements	Rationales	Is the rationale a pro or a con with respect to the requirement?
Seldom breakdown	Low breakdown rate is good for the manufacturer’s reputation	Pro
Seldom breakdown	Low breakdown rate reduces the cost of servicing for the manufacturer to replace broken down products, i.e. staff time to serve customers with broken down products	
Seldom breakdown	Low breakdown rate reduces the cost for the manufacture, as there is less broken down products to replace. Company has decided to have no servicing department in order to save costs, therefore a failure	

	during the warranty will cause the company to cover the cost of replacement	
Seldom breakdown	It must last at least 5 years without break down, because the company has committed to a warranty period of 5 years. A breakdown within the warranty period will incur the manufacture additional cost of providing a replacement	
Well balanced	Well balanced product is comfortable to hold, giving users a good experience	
Comfortable hand grip	Comfortable grip enables prolonged use for drying long hair, giving users a good experience	
Comfortable hand grip	This is good ergonomics, and is a sign of good quality which increases the reputation of the product	
Comfortable hand grip	Without this requirement, uncomfortable hand grip could cause the user to loosely grip the product, which might cause it to be dropped accidentally, e.g. during wet hands use. Persistent dropping could damage the product, and give users a bad experience, thus bring negative reviews of the product.	
Comfortable hand grip	Comfortable grip enables prolonged use for drying clothes without tiring the arm. This is a sign of good design, and gives the user a good experience.	
Light weight	More portable and can be used for travel. Portability is a selling point for this product.	
Light weight	Maximise accessibility to the product. Increases the range of customers this product is available to. Users have different arm strengths, the product should be designed so that even the users with the lowest strength could use it.	
Light weight with a max weight of 0.8kg	0.8kg is considered light weight as it is lower than the average weight of similar products on the market	
Light weight with a max weight of 0.8kg	0.8kg may be difficult and costly to achieve considering only a few other manufacturers have achieved it	
Light weight	Could require more expensive materials for the components. Less profit for the manufacturer.	
Light weight	Easier to transport for the distributor and retailers	
Light weight	Would be suitable for placing the product on a stand. Potentially saving the retailer	

	the cost of making a customised stand.	
Light weight	Could compromise the strength of casing, and ultimately affect the durability of the product, giving users a bad experience, thus a bad reputation for the manufacturer	
Light weight	Will not tire the arm after holding it for a long time. This is a sign of good design, and gives the user a good experience.	
Minimise risk of electrocution	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	
Prevent circuit overheating	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	
Prevent circuit overheating	Cost more to design	
Prevent circuit overheating	Cost more to manufacture	
Prevent circuit overheating	Without this requirement, overheating could break the product permanently, giving users a bad experience and decrease the reputation of the manufacturer	
Prevent circuit overheating	Without this requirement, overheating could start a fire, and potentially give the manufacturer a bad reputation	
Low price	Most of users would pay between 16-20GBP	
Low price	The lower the price the lower the profit margin for the manufacturer	
Low price, cost no more than 15GBP	15GBP is considered a low price by the general consumers	
Low price, cost no more than 15GBP	15GBP is low price statistically as it is below the median price in a survey of similar products	
Low price	Low price is needed to attract customers, especially considering this is a new product model with no previous reputation	
Advanced in technology	Advanced in technology is more competitive, therefore more likely to outsell against competing products	
Advanced in technology	Advanced in technology may be costly to manufacture, and therefore conflict with the requirement of low price	
Advanced in technology	Customers may not find advanced in technology more preferable than a simple to operate product	

Questionnaire Part 2 (Solution)

Requirements	**Rationales**	**Is the rationale a pro or a con with respect to the requirement?**
Seldom breakdown	Low breakdown rate is good for the manufacturer's reputation	Pro
Seldom breakdown	Low breakdown rate reduces the cost of servicing for the manufacturer to replace broken down products, i.e. staff time to serve customers with broken down products	Pro
Seldom breakdown	Low breakdown rate reduces the cost for the manufacture, as there is less broken down products to replace. Company has decided to have no servicing department in order to save costs, therefore a failure during the warranty will cause the company to cover the cost of replacement	Pro
Seldom breakdown	It must last at least 5 years without break down, because the company has committed to a warranty period of 5 years. A breakdown within the warranty period will incur the manufacture additional cost of providing a replacement	Pro
Well balanced	Well balanced product is comfortable to hold, giving users a good experience	Pro
Comfortable hand grip	Comfortable grip enables prolonged use for drying long hair, giving users a good experience	Pro
Comfortable hand grip	This is good ergonomics, and is a sign of good quality which increases the reputation of the product	Pro
Comfortable hand grip	Without this requirement, uncomfortable hand grip could cause the user to loosely grip the product, which might cause it to be dropped accidentally, e.g. during wet hands use. Persistent dropping could damage the product, and give users a bad experience, thus bring negative reviews of the product.	Pro
Comfortable hand grip	Comfortable grip enables prolonged use for drying clothes without tiring the arm. This is a sign of good design, and gives the user a good experience.	Pro
Light weight	More portable and can be used for travel. Portability is a selling point for this product.	Pro
Light weight	Maximise accessibility to the product. Increases the range of customers this	Pro

	product is available to. Users have different arm strengths, the product should be designed so that even the users with the lowest strength could use it.	
Light weight with a max weight of 0.8kg	0.8kg is considered light weight as it is lower than the average weight of similar products on the market	Pro
Light weight with a max weight of 0.8kg	0.8kg may be difficult and costly to achieve considering only a few other manufacturers have achieved it	Con
Light weight	Could require more expensive materials for the components. Less profit for the manufacturer.	Con
Light weight	Easier to transport for the distributor and retailers	Pro
Light weight	Would be suitable for placing the product on a stand. Potentially saving the retailer the cost of making a customised stand.	Pro
Light weight	Could compromise the strength of casing, and ultimately affect the durability of the product, giving users a bad experience, thus a bad reputation for the manufacturer	Con
Light weight	Will not tire the arm after holding it for a long time. This is a sign of good design, and gives the user a good experience.	Pro
Minimise risk of electrocution	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	Pro
Prevent circuit overheating	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	Pro
Prevent circuit overheating	Cost more to design	Con
Prevent circuit overheating	Cost more to manufacture	Con
Prevent circuit overheating	Without this requirement, overheating could break the product permanently, giving users a bad experience and decrease the reputation of the manufacturer	Pro
Prevent circuit overheating	Without this requirement, overheating could start a fire, and potentially give the manufacturer a bad reputation	Pro
Low price	Most of users would pay between 16-20GBP	Pro
Low price	The lower the price the lower the profit margin for the manufacturer	Con
Low price, cost no more than 15GBP	15GBP is considered a low price by the general consumers	Pro
Low price, cost no	15GBP is low price statistically as it is	Pro

more than 15GBP	below the median price in a survey of similar products	
Low price	Low price is needed to attract customers, especially considering this is a new product model with no previous reputation	Pro
Advanced in technology	Advanced in technology is more competitive, therefore more likely to outsell against competing products	Pro
Advanced in technology	Advanced in technology may be costly to manufacture, and therefore conflict with the requirement of low price	Con
Advanced in technology	Customers may not find advanced in technology more preferable than a simple to operate product	Con

Questionnaire Part 2 (Participant 1)

Requirements	**Rationales**	**Is the rationale a pro or a con with respect to the requirement?**
Seldom breakdown	Low breakdown rate is good for the manufacturer's reputation	Pro ✓
Seldom breakdown	Low breakdown rate reduces the cost of servicing for the manufacturer to replace broken down products, i.e. staff time to serve customers with broken down products	Pro ✓
Seldom breakdown	Low breakdown rate reduces the cost for the manufacture, as there is less broken down products to replace. Company has decided to have no servicing department in order to save costs, therefore a failure during the warranty will cause the company to cover the cost of replacement	Pro ✓
Seldom breakdown	It must last at least 5 years without break down, because the company has committed to a warranty period of 5 years. A breakdown within the warranty period will incur the manufacture additional cost of providing a replacement	Pro ✓
Well balanced	Well balanced product is comfortable to hold, giving users a good experience	Pro ✓
Comfortable hand grip	Comfortable grip enables prolonged use for drying long hair, giving users a good experience	Pro ✓
Comfortable hand grip	This is good ergonomics, and is a sign of good quality which increases the reputation of the product	Pro ✓
Comfortable hand grip	Without this requirement, uncomfortable hand grip could cause the user to loosely grip the product, which might cause it to be dropped accidentally, e.g. during wet hands use. Persistent dropping could damage the product, and give users a bad experience, thus bring negative reviews of the product.	Pro ✓
Comfortable hand grip	Comfortable grip enables prolonged use for drying clothes without tiring the arm. This is a sign of good design, and gives the user a good experience.	Pro ✓
Light weight	More portable and can be used for travel. Portability is a selling point for this product.	Pro ✓
Light weight	Maximise accessibility to the product. Increases the range of customers this product is available to. Users have different arm strengths, the product should be designed so	Pro ✓

	that even the users with the lowest strength could use it.	
Light weight with a max weight of 0.8kg	0.8kg is considered light weight as it is lower than the average weight of similar products on the market	Pro ✔
Light weight with a max weight of 0.8kg	0.8kg may be difficult and costly to achieve considering only a few other manufacturers have achieved it	Con ✔
Light weight	Could require more expensive materials for the components. Less profit for the manufacturer.	Con ✔
Light weight	Easier to transport for the distributor and retailers	Pro ✔
Light weight	Would be suitable for placing the product on a stand. Potentially saving the retailer the cost of making a customised stand.	Pro ✔
Light weight	Could compromise the strength of casing, and ultimately affect the durability of the product, giving users a bad experience, thus a bad reputation for the manufacturer	Con ✔
Light weight	Will not tire the arm after holding it for a long time. This is a sign of good design, and gives the user a good experience.	Pro ✔
Minimise risk of electrocution	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	Pro ✔
Prevent circuit overheating	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	Pro ✔
Prevent circuit overheating	Cost more to design	Con ✔
Prevent circuit overheating	Cost more to manufacture	Con ✔
Prevent circuit overheating	Without this requirement, overheating could break the product permanently, giving users a bad experience and decrease the reputation of the manufacturer	Pro ✔
Prevent circuit overheating	Without this requirement, overheating could start a fire, and potentially give the manufacturer a bad reputation	Pro ✔
Low price	Most of users would pay between 16-20GBP	Pro ✔
Low price	The lower the price the lower the profit margin for the manufacturer	Con ✔
Low price, cost no more than 15GBP	15GBP is considered a low price by the general consumers	Pro ✔
Low price, cost no more than 15GBP	15GBP is low price statistically as it is below the median price in a survey of similar products	Pro ✔
Low price	Low price is needed to attract customers,	Pro ✔

	especially considering this is a new product model with no previous reputation	
Advanced in technology	Advanced in technology is more competitive, therefore more likely to outsell against competing products	Pro ✓
Advanced in technology	Advanced in technology may be costly to manufacture, and therefore conflict with the requirement of low price	Con ✓
Advanced in technology	Customers may not find advanced in technology more preferable than a simple to operate product	Con ✓

Questionnaire Part 2 (Participant 2)

Requirements	Rationales	Is the rationale a pro or a con with respect to the requirement?
Seldom breakdown	Low breakdown rate is good for the manufacturer's reputation	Pro ✓
Seldom breakdown	Low breakdown rate reduces the cost of servicing for the manufacturer to replace broken down products, i.e. staff time to serve customers with broken down products	Pro ✓
Seldom breakdown	Low breakdown rate reduces the cost for the manufacture, as there is less broken down products to replace. Company has decided to have no servicing department in order to save costs, therefore a failure during the warranty will cause the company to cover the cost of replacement	Pro ✓
Seldom breakdown	It must last at least 5 years without break down, because the company has committed to a warranty period of 5 years. A breakdown within the warranty period will incur the manufacture additional cost of providing a replacement	Pro ✓
Well balanced	Well balanced product is comfortable to hold, giving users a good experience	Pro ✓
Comfortable hand grip	Comfortable grip enables prolonged use for drying long hair, giving users a good experience	Pro ✓
Comfortable hand grip	This is good ergonomics, and is a sign of good quality which increases the reputation of the product	Pro ✓
Comfortable hand grip	Without this requirement, uncomfortable hand grip could cause the user to loosely grip the product, which might cause it to be dropped accidentally, e.g. during wet hands use. Persistent dropping could damage the product, and give users a bad experience, thus bring negative reviews of the product.	Pro ✓
Comfortable hand grip	Comfortable grip enables prolonged use for drying clothes without tiring the arm. This is a sign of good design, and gives the user a good experience.	Pro ✓
Light weight	More portable and can be used for travel. Portability is a selling point for this product.	Pro ✓
Light weight	Maximise accessibility to the product. Increases the range of customers this	Pro ✓

	product is available to. Users have different arm strengths, the product should be designed so that even the users with the lowest strength could use it.	
Light weight with a max weight of 0.8kg	0.8kg is considered light weight as it is lower than the average weight of similar products on the market	Pro ✓
Light weight with a max weight of 0.8kg	0.8kg may be difficult and costly to achieve considering only a few other manufacturers have achieved it	Con ✓
Light weight	Could require more expensive materials for the components. Less profit for the manufacturer.	Con ✓
Light weight	Easier to transport for the distributor and retailers	Pro ✓
Light weight	Would be suitable for placing the product on a stand. Potentially saving the retailer the cost of making a customised stand.	Pro ✓
Light weight	Could compromise the strength of casing, and ultimately affect the durability of the product, giving users a bad experience, thus a bad reputation for the manufacturer	Con ✓
Light weight	Will not tire the arm after holding it for a long time. This is a sign of good design, and gives the user a good experience.	Pro ✓
Minimise risk of electrocution	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	Pro ✓
Prevent circuit overheating	More likely to pass safety regulations, and is less likely to cause harm to the user. Protects the manufacturer's reputation.	Pro ✓
Prevent circuit overheating	Cost more to design	Con ✓
Prevent circuit overheating	Cost more to manufacture	Con ✓
Prevent circuit overheating	Without this requirement, overheating could break the product permanently, giving users a bad experience and decrease the reputation of the manufacturer	Pro ✓
Prevent circuit overheating	Without this requirement, overheating could start a fire, and potentially give the manufacturer a bad reputation	Pro ✓
Low price	Most of users would pay between 16-20GBP	Pro ✓
Low price	The lower the price the lower the profit margin for the manufacturer	Con ✓
Low price, cost no more than 15GBP	15GBP is considered a low price by the general consumers	Pro ✓
Low price, cost no	15GBP is low price statistically as it is	Pro ✓

more than 15GBP	below the median price in a survey of similar products	
Low price	Low price is needed to attract customers, especially considering this is a new product model with no previous reputation	Pro ✓
Advanced in technology	Advanced in technology is more competitive, therefore more likely to outsell against competing products	Pro ✓
Advanced in technology	Advanced in technology may be costly to manufacture, and therefore conflict with the requirement of low price	Con ✓
Advanced in technology	Customers may not find advanced in technology more preferable than a simple to operate product	Con ✓

Appendix 2

Engine project questionnaire

Requirement rationale	Question 1	How important is it to capture rationale at the time when requirements are defined? *Please rank importance on a scale of 1 to 10 (10 being the most important) and provide a reason.*
	Responder 1	8 – rationale or "Evidence" to justify a requirement is essential for knowledge management, particularly in the long-term as changes are often made with different people involved who don't have the original understanding to hand. It also allows reviewers and other stakeholders to understand the reasoning behind a requirement to allow them to agree to disagree with it in a more efficient manner. The only other consideration is the time and cost of capturing the rationale needs to be minimised to avoid unnecessary burden on the business. As such it may be the case that for some requirements the added value of capturing rationale is negligible so not worth the effort as they are obvious and incontestable, e.g. regulatory requirements.
	Responder 2	8 – rationale helps communicate the context of the requirement and can make up for a poorly written requirement. It also allows sensible challenges to be made against the requirements as captured in order to arrive at the best solution. A requirement without a rationale is still more useful than no requirement at all however, and so the extra effort spent to capture rationale should not jeopardise the opportunity to capture a requirement.
	Question 2	How important is it to capture contextual metadata (e.g. the source of a requirement, or the person who was responsible for defining a requirement) at the time when requirements are defined? *Please rank importance on a scale of 1 to 10 (10 being the most important) and provide a reason.*
	Responder 1	10 – source and owner of requirements is crucial to effective requirements management. Otherwise change control and change history interrogations are not possible or useful. Other forms of meta-date may be of less value. Industry-standard requirements management software (e.g. DOORS) already allows this via links and attributes.
	Responder 2	8 – as above. Metadata can give an indicator of the quality of a requirement, or provide a traceable link to more rationale.
	Question 3	How valid is the proposal to capture rationale and contextual metadata using DRed together with the other System Engineering methods (Function Flow Diagram, Viewpoint Analysis, Systemic Textual Analysis, DOORS)? Validity refers to whether the concept is sound, makes sense and does not show flaws. *Please rank validity on a scale of 1 to 10 (10 being the most valid) and provide a reason.*

Requirement rationale	Responder 1	6 – the tool does show some promise and is certainly a step forward from existing methods where data is held in various disparate and unconnected software tools (e.g. Excel or Visio). There remain some concerns about the basic use-ability of DRED, e.g. being able to export from DRED to PowerPoint or Word, or even to print diagrams. If these issues can be addressed and some user-tests are completed successfully then it should offer a good near-term solution for our needs. The longer-term strategy will likely be towards more formal MBSE techniques and tools like SysML.
	Responder 2	8 – Highly valid proposal, the only issues are with the implementation (tailoring of syntax, training of users, roll-out of software etc)
	Question 4	How feasible is the proposal to capture requirements informally using DRed? Feasibility refers to whether the concept can be implemented into current practice and it is envisaged that an engineer would benefit from it. *Please rank feasibility on a scale of 1 to 10 (10 being the most feasible) and provide a reason.*
	Responder 1	7 – subject to the usability concerns above being addressed and DRed installations being made readily available through corporate IT, plus some training/communication to the user base, then there should be no reason why this should not be used and benefit the engineering population.
	Responder 2	7 – Thinking about capture alone (i.e. from brain to written text) it has benefits over paper methods because it is electronic and therefore easy to manipulate and transfer. There is a (small) barrier to entry compared to other electronic tools like Word/Excel/PowerPoint etc, but is easier than DOORS. If you consider capture to include the manipulation of data and reporting then again it isn't as good as Word/Excel/PowerPoint.
Requirement traceability	Question 5	How useful is it to link and make traceable requirements? *Please rank usefulness on a scale of 1 to 10 (10 being the most useful) and provide a reason.*
	Responder 1	10 – it is essential. Our current processes specify that all requirements should have traceability, except at the highest (Environment) level. This has been poorly adhered to in the past but it remains a firm policy and objective and efforts are being made to improve.
	Responder 2	6 – Traceability is one of several quality measures for requirements, it is also important to have a good written requirements as well as good validation/verification. Also in the Civil Aviation industry, structured verification and traceability is not a contractual obligation.
	Question 6	How valid is the proposal to link and make traceable requirements using DRed together with the other System Engineering methods? Validity refers to whether the concept is sound, makes sense and does not show flaws. *Please rank validity on a scale of 1 to 10 (10 being the most valid) and provide a reason.*
	Responder 1	1 - This functionality is currently adequately provided by DOORS so there is no benefit to using another, separate tool. DRed is not a requirements management tool.

Requirement traceability	Responder 2	5 – Nice idea, but I personally struggle to see people spending the time to make the links, particularly since DOORS is the corporate too for requirements management.
	Question 7	How feasible is the proposal to link requirements using DRed? Feasibility refers to whether the concept can be implemented into current practice and it is envisaged that an engineer would benefit from it. *Please rank feasibility on a scale of 1 to 10 (10 being the most feasible) and provide a reason.*
	Responder 1	1 - This functionality is currently adequately provided by DOORS so there is no benefit to using another, separate tool. DRed is not a requirements management tool.
	Responder 2	As above.
Additional questions	Question 8	Have you seen or used similar proposals to enable the capture of rationale and the link and traceability of requirements? If so, please describe what they were and explain if you have found them effective?
	Responder 1	DOORS already provides traceability, and with some limitations, could offer a more comprehensive rationale capture if the necessary attributes, views and GUIs were created. Also, SysML tools like Artisan Studio can offer much more comprehensive structuring and linking of various data-types, including requirements. The only limitation to this is its cost and complexity at present.
	Responder 2	SysML (Artisan Studio)
	Question 9	Do you see ways in which this proposal could be improved?
	Responder 1	See above comments on the need to improve basic usability in DRed and the need for some real user testing and feedback before wider roll-out.
	Responder 2	A fully integrated solution is a difficult task to achieve in one step. What is useful, is to enable data transfer into and out of DRed in a simple and flexible way so that people can still use their Microsoft-Office software and/or DOORS for specific tasks (e.g. to produce a presentation or run an FMEA session) without having to commit to a single tool from the outset.