

Requirements change in complex technical systems: an empirical study of root causes

João Fernandes · Elsa Henriques · Arlindo Silva ·
Michael A. Moss

Received: 26 April 2013/Revised: 14 June 2014/Accepted: 17 July 2014/Published online: 1 August 2014
© Springer-Verlag London 2014

Abstract Capturing and managing requirements are crucial to the success of product design and development. This paper presents and analyzes the causes of requirements change during the design process of a complex aerospace system developed by Rolls-Royce during the course of 6 years. The study was part of a series of efforts ongoing at this original equipment manufacturer to improve requirements engineering practices and involved a group of engineers in the assessment of the root causes of change. A large data set was examined, containing 700 system requirements and more than 1,000 changes released during the project studied. The results showed that more than 80 % of the changes had internal root causes and change driven by the customer accounted for about 15 % of the total. The structured approach that was implemented to understand the causes of change allowed the group of engineers to identify and dissect various management guidelines aiming to further improve requirements engineering in practice. This paper thus also reports the guidelines captured at this aerospace manufacturer and intends to support practitioners across the industrial community.

Keywords Requirements engineering · Requirements change · Process improvements · Empirical study · Aerospace industry

Abbreviations

BRD	Business requirements document
CDR	Critical design review
CR	Concept review
CRD	Component requirements document
FP	First parts
FT	First test
MoU	Memorandum of understanding
PDR	Preliminary design review
PR	Production readiness
PRD	Product requirements document
SSRD	Sub-system requirements document

1 Introduction

Becoming truly effective at requirements engineering and management is challenging for organizations designing and developing complex systems. The process is inherently difficult for several reasons. A very large number of requirements typically need to be captured and managed at different levels of the product's physical and functional hierarchy (Weber and Weisbrod 2003). Multiple external and internal stakeholders are continuously involved in negotiation (Pohl 2010). And the engineers performing the process must act and take decisions in an environment containing technical uncertainty, semantic ambiguity (Tseng and Jiao 1998) and imperfect communication. Correctly and completely capturing external needs into

J. Fernandes (✉) · E. Henriques · A. Silva
IDMEC, Instituto Superior Técnico, Universidade de Lisboa,
Av. Rovisco Pais 1, 1049-001 Lisbon, Portugal
e-mail: joao.ventura.fernandes@tecnico.ulisboa.pt

E. Henriques
e-mail: elsa.h@tecnico.ulisboa.pt

A. Silva
e-mail: arlindo.silva@tecnico.ulisboa.pt

M. A. Moss
Knowledge Management, Research and Technology,
Rolls-Royce Plc, Moor Lane ML80, P.O. Box 31, Derby, UK
e-mail: michael.moss@rolls-royce.com

internal requirements for design and development activities, or deriving and flowing down new requirements from existing ones at higher levels of the product hierarchy, is highly dependent on skills, experience and organizational capabilities.

1.1 Research motivation

Our aim in this paper is to bring forward an empirical case study about the causes of requirements change during the product development process of complex technical systems. The study was performed at Rolls-Royce Plc, a world-leading provider of power systems and services for use on land, at sea and in the air which has established a strong position in different markets—civil aerospace, defense aerospace, marine and energy and nuclear (Rolls-Royce 2013). The 2011 financial results showed that this company has reached a revenue of £11.3 billion, employs over 40,000 people and has more than 59,000 gas turbines in service around the world (Rolls-Royce 2013).

A gas turbine is a *complex technical system*, containing many sub-systems, hundreds of sub-assemblies and thousands of components that need to be perfectly designed and integrated to deliver numerous functionalities. The design process of gas turbines is also inherently complex: It requires a large team of engineers, with a wide range of technical expertise, performing concurrent engineering typically during more than 5 years.

The authors engaged with Rolls-Royce in research about the causes of change since the company is continuously investing in improvements to requirements engineering and, generally speaking, to all of its systems engineering practices and processes. The study focused on the requirements evolution during the development process of a previous aerospace gas turbine project that took place during the course of 6 years. We were allowed to collect and analyze the project's requirements database, which contained more than 700 system requirements. These had evolved through more than 1,000 changes released during the development process.

Our investigation quantified the causes of changes in system requirements at *different phases* of the product development process. The assessment of the causes was performed by the engineers *actually* involved in the project. Using archival and qualitative research methods, we reconstituted the requirements engineering process dynamics and related it with the quantitative results about the causes of change. The quantitative and qualitative results allowed Rolls-Royce engineers to identify and dissect a number of management guidelines that aim to further improve requirements engineering in practice. These findings are reported in this paper. We specifically intend to shed light in this paper to three main research questions:

- How do requirements evolve in organizations developing complex technical systems?
- How much change is externally and internally driven?
- How do the causes of change vary during the different phases of the product development process?

1.2 Research contributions

Empirical studies around the topic of requirements engineering and management are rare in design literature. Chakrabarti et al. (2004) have researched the process and methods used by designers to identify, generate and apply requirements during design activities. A major empirical study containing a comprehensive overview of requirements management and of issues arising in practice in the automotive industry has been previously reported by Al-mefelt et al. (2006). A survey of requirements engineering capabilities in small-to-medium companies is also part of prior knowledge (Michael et al. 2007). And Sudin and Ahmed (2009) examined a data set of 271 specification change reports of an aero-engine and found most occurred during the manufacturing and testing phase due to the implementation of product improvements and correction of design errors.

Subsequent empirical studies addressed requirements management support for embedded systems development in the automotive industry using mainly qualitative research methods (Bergsjö et al. 2010; Čatić and Malmqvist 2010). Decision-making and organizational factors influencing the requirements change management process in practice have also been explored qualitatively by Sudin and Ahmed-Kristensen (2011) through a case study at a product development consultancy company, which showed change emerges as a result of internal activities and external factors arising from technology or market changes.

Related research contributions appear also in engineering change literature. Engineering change has been defined as an “alteration made to parts, drawings or software” released during product development (Jarratt et al. 2011), and various authors have investigated its causes. Pikosz and Malmqvist (1998) identified specification changes, design problems found during prototype development or production and product quality issues as key reasons. Fricke et al. (2000) reported changing or incorrect requirements, product innovation, error correction and change propagation as major causes of engineering change. Eckert et al. (2004) categorized the sources of change either as emergent (arising from the product) or initiated (arising from an outside source) and discussed case-study results showing that changes in customer and certification requirements, product innovations and design error correction identified during development, testing, fabrication

or service are important causes of engineering changes. A recent empirical research examining a large quantity of engineering change reports across two products showed that most change requests occurred during the manufacturing phase and confirmed that requirements change was one of the key causes of reporting during the development phase (Vianello and Ahmed-Kristensen 2012). In addition, change propagation methods developed to support decision-making and change management (Clarkson et al. 2004) have been used to understand the effects of changes in requirements on the product's architecture (Hamraz et al. 2012; Koh et al. 2012).

Considering prior research contributions, the authors argue that an in-depth and quantitative investigation to the root causes of requirements change in practice has not been reported yet. This paper intends to fill this research gap. Our understanding is also that prior work has not been able to access a large industrial dataset of requirements change such as the one examined in this paper. We thus argue that the findings presented in this paper showing in detail and quantitatively the evolution of requirements change and its causes in a complex technical system are original and unique. In addition, the results of this research allowed Rolls-Royce engineers to analyze various management guidelines for further improvements in their own requirements engineering process which are also useful for the broader industrial community of practitioners. This paper starts by presenting the context surrounding the case studied (Sect. 2). A detailed explanation of the research approach follows in Sect. 3. The main body of findings is then contained in Sect. 4. A discussion of improvement guidelines resulting from the research appears in Sect. 5. Section 6 highlights and summarizes the main conclusions from this research.

2 Background

We think of *requirements-oriented organizations* as enterprises which systematically develop new products according to the principle of validating a design solution against the requirements captured and engineered at all levels of the product's physical and functional hierarchy. Requirements elicited from the customer are typically decomposed and flown down to the system, sub-system and to the component levels. This principle and the idea of continuous decomposition and integration of a system's physical and functional architectures are cornerstones of the systems engineering approach to product development (Blanchard and Fabrycky 2006; NASA 2007).

Thousands of requirements must be captured, negotiated, agreed upon, managed and verified during the design of complex technical systems such as the ones developed

by Rolls-Royce. The collaborative and highly iterative process of identifying all stakeholders, eliciting, negotiating, documenting and validating requirements is known as *requirements engineering* or *requirements management* (Kotonya and Sommerville 1998; Robertson and Robertson 2006; Pohl 2010).

In addition, the classical paradigm of designing for "fixed" requirements has been abandoned, both in academia and in practice. The current paradigm is that requirements evolve during the design and development process (Lam et al. 1999) and some level of change should thus be expected by all firms. Change is truly inevitable for firms developing complex technical systems over long time scales, such as Rolls-Royce. It is normal that customers mature its own needs along the way and ask for changes to previous requirements. Moreover, the process shown in Fig. 1 of concurrently mapping requirements to solutions at different system levels is highly iterative and implies an internal maturation where both requirements and solution become understood with increasing confidence. This continuous rise of the understanding of both the problem and the solution space has been coined as design coevolution (Maher et al. 1996; Dorst and Cross 2001).

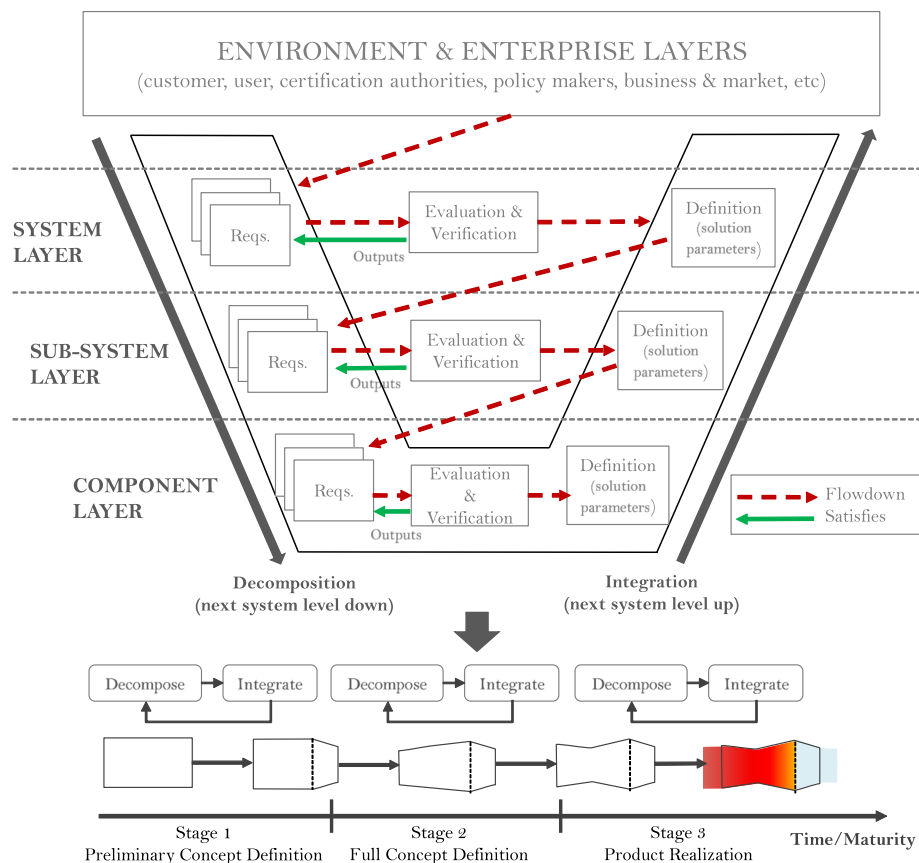
Although necessary, change to requirements must be managed since uncontrolled and late change is normally costly and thus undesirable. The authors view requirements change as modifications made to a previous version of requirements released in the organization during design, production or any other development stage. *Requirements change management* is the process of systematic analysis of change requests, identification of solutions, assessment of the change impact, selection and approval of requests by a company Change Board and monitorization of solution implementation (Kotonya and Sommerville 1998; Pohl 2010).

2.1 Organizational context

Improving requirements engineering practices is part of a wider strategy and efforts of continuous process improvement undertaken by engineering functions at Rolls-Royce. The requirements engineering process used within the company contains the activities recommended in state-of-the-art literature (Pohl 2010): stakeholder identification, requirements capture, prioritization and balancing, requirements flow down and documentation management. Additionally, it also provides a basis to ensure the execution of validation and verification activities.

The general capture principle defined in the process that we found contains simplicity and scalability, following the systems engineering V-model (Blanchard and Fabrycky 2006). The process is based on a top-down approach, starting in the beginning of any project at the highest level,

Fig. 1 General principles in requirements engineering embodied from the systems engineering approach to product development. Adapted from internal company sources



through the capture of high-level stakeholder needs into requirements for the whole system (Fig. 1). When it is verified through evidence that the proposed design solution satisfies the requirements, the solution parameters are flowed to the next system level as requirements and the process is then repeated for each subsequent lower level (Fig. 1). The process also standardizes the documentation landscape to be maintained by any product development project at Rolls-Royce.

Furthermore, projects are constantly monitored to ensure that the best practices are being followed and to search for improvements. Training is provided to staff. For instance, training material supports engineers writing good requirements. The company's aim is that all designers are proficient in requirement engineering and can work independently as requirements' authors. The role defined for systems engineering specialists in Rolls-Royce is then to support designers when needed, facilitating and driving requirements capture and understanding. For practical reasons, a smaller group of engineers may be appointed by the project as *requirements managers* at a particular system layer. These engineers work together with system, sub-system or component designers in requirements capture and are responsible for structuring and maintaining the requirements database during the project. In addition,

Rolls-Royce has standardized tools to conduct stakeholder needs analysis, resolve conflicts, populate and manage the requirements database and create and maintain the document landscape in product development projects.

2.2 The case studied

The case studied is the product development process of an aerospace gas turbine at Rolls-Royce. This project's time frame is represented in Fig. 2 together with key development milestones. The project departed from business conversations with the customer which resulted in the signature of a first Memorandum of Understanding (MoU) framing the development of a new system for a particular aerospace application. At that time, Rolls-Royce's advanced programs department was highly involved in concept development and selection.

The first product requirements document (PRD), containing the initial requirements for the *whole system*, was formally released one and a half years after the first business conversations with the customer. Two new versions of the PRD followed closely and came out during the second year of the project. Concept exploration, concept selection and preliminary design work lasted until the beginning of the third year, when a preliminary design review (PDR)

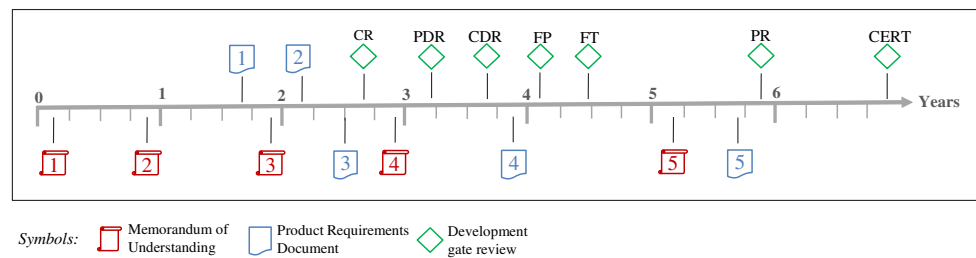


Fig. 2 Illustration of the project investigated, focusing in key milestones and document releases. CR Concept review, PDR preliminary design review, CDR critical design review, FP first parts, FT first test, PR production readiness, CERT certification

took place. Several interaction loops with the customer had occurred in parallel (Fig. 2).

Detailed design activities continued during the third year, and the design was initially “frozen” after passing the critical design review (CDR). Following the CDR, a fourth issue of the PRD was released. The first test of the new gas turbine occurred during the spring of the fourth year (Fig. 2). Several development prototypes were then tested during an extensive validation and verification program lasting one and a half years. By the end of the fifth year, the initial program requirements for certification were delayed by the customer. Additionally, major changes in some of the customer requirements also occurred. Around the same time, a fifth issue of the PRD was formally released by the project. The final stages of testing and development that lead to certification occurred close to the end of the project’s sixth year.

3 Research approach

Exploratory research was performed at Rolls-Royce with the purpose of understanding the context and selecting the case to study. This was conducted through a series of 14 semi-structured interviews to a heterogeneous sample of engineers and managers. Sample selection was purposive, but followed the method of maximum variation (Kuzel 1992; Robson 2002). The authors selected a group containing a wide range of expertise and experience, to quickly gain insight and allow comparison of answers. The sample included engineers involved in requirements engineering (2), systems engineering (2), sub-system integration (2), concept development (1), detailed design (1), design methods (3), engineering improvement and quality (2) and strategic management (1). The interviews followed the recommended structure *Introduction, Warm-up, Main body, Cool-off* (Robson 2002) and lasted between 45 min to 1 h with each interviewee. The guide consisted of 30 pre-defined open questions organized around the requirements engineering process, the organizational context, causes of change and practical information related to

archival access and information quality. The particular project that was selected was one of the main findings from this research stage.

3.1 Reasons for selecting a case-study strategy

Yin (1984) states that a case-study approach is a method adequate for an empirical investigation intending to study a “contemporary phenomenon within its real-life context, when the boundaries between phenomenon and context are not clearly evident and in which multiple sources of evidence are used.” It is suitable for investigations intending to provide “cause–effect” relationships and for explanatory types of research, dealing with “how” and “why” research questions (Lindvall 1997).

In our case, it was not possible to separate the phenomenon of interest—the causes of requirements change—from its technical and organizational context. The nature of our research questions was also intrinsically explanatory. We were interested in understanding how requirements evolve, how change is generated and how do the causes of change vary during the product development process. Explaining the causes behind observed changes in requirements is about understanding a cause–effect type of relationship. All of the above supports our choice of a case-study strategy. Moreover, it is rare that scientific investigations are allowed to study such phenomena in an industrial environment. Because such phenomena are typically inaccessible, the case study selected by the authors for this investigation can be considered a revelatory case (Yin 1984).

3.2 Methodology and data sources

The overall methodology followed during the study is presented in Fig. 3 and consisted of a six-step approach. The first step included the exploratory research based on the semi-structured interviews referred previously to understand the context and select the case.

The second step consisted in data collection through archival research. Our study started with the collection of

	1 Understand context	2 Collect data	3 Define causes	4 Design sample	5 Quantify causes	6 Capture improvement
Semi-structured interviews	✓		✓			
Archival research	✓	✓	✓		✓	
Statistical methods				✓	✓	
Expert knowledge elicitation sessions					✓	✓

Fig. 3 Six-step approach and methods used throughout the course of the research presented in this paper

system requirements and an analysis of its evolution which was embodied in the changes performed during the course of the development of the new gas turbine. Our belief was that an investigation to the causes of requirements change would allow us to identify potential improvements to the requirements engineering process. Prior work performed at Rolls-Royce by Pickard et al. (2010) and Nolan et al. (2011) reported that most requirements change in control systems' development is internally driven, but an in-depth and quantitative study focused on the development of the whole system had not been realized yet.

In the second step, the population of changes was collected from archival research in documents and from a commercial software tool used in the company to conduct requirements engineering and manage the requirements database. Each requirement contained a unique identification tag, which allowed following its evolution. The database provided also the author of the requirement, time of creation and time of change. We focused the study at system level, since the database was too large for a detailed study of all levels of the product's architecture within a reasonable research time frame.

After the data collection step, the third step consisted in the definition of a preliminary list of causes of change based on the observation of the requirements evolution. The company's requirements management software tool contained all records of changes, together with notes and comments from the engineers responsible for requirements engineering. It was thus possible to develop hypothesis about the root causes in many cases.

During the third step, the initial list was submitted to a process of revision and validation. The four requirements managers that had worked with the system's requirements and managed the changes during the course of the project were invited to participate in the research. They were initially interviewed during approximately 45 min each, using a prepared semi-structured guide. Interviews followed again the recommendations of Robson (2002). The interview guide focused on the history of the projects requirements engineering process and contained the proposed list of causes. The engineers were asked whether the list was

comprehensive and how they interpreted the definition of each cause of change. This step was performed to understand the capability of distinguishing in practice the different root causes and thus evaluate the risk of misclassification. Following this round of interviews, adjustments were made based on the feedback received to the list of causes and their definition to minimize that risk.

The data collected in the second step revealed more than 1,000 changes between *released versions* of product requirements documents (PRD_i). To deal with the size of the population, the fourth step consisted in the design of a sample representative of the population of changes between released versions of system requirements. Moreover, to ensure accurate and unbiased results, the assessment of each requirement change was performed by the engineer that had actually worked on it. The requirements managers had an inside knowledge about the full history of the requirement and had lived the project. Such knowledge cannot be easily captured from archival research. It had to be *elicited*.

The fifth step thus consisted in root cause assessment sessions with the requirements managers that had worked on each requirement change. We prepared a template containing the new version of the requirement, the previous version, the type of change and the list of causes of change, which was available from a drop down menu. Prior to each session, we prepared a small set of hypothesis about the cause of each requirement change from the history recorded in the requirements management software tool, to generate discussion with the engineers. This facilitated knowledge elicitation, making the requirements managers justify their assessment by remembering the history of the requirement, the sequence of events and the interactions with other individuals. This strategy intended to increase the confidence in the results. The *proportion* of each cause of change was then quantified in statistical terms from the sample designed in step four by the requirements managers.

Finally, the sixth step of the methodology consisted in the realization of a series of group working sessions with the engineers to discuss the statistical results characterizing

the requirements evolution and the causes of change during the project. This allowed the authors to capture from the engineers involved in the research a series of management guidelines that could be or were being implemented at Rolls-Royce to further improve requirements engineering practices.

Throughout the course of the investigation illustrated in Fig. 3, multiple sources of information have been used, from archival research, to semi-structured interviews and expert knowledge elicitation. Furthermore, the methodology makes use of quantitative methods and qualitative methods. Case-study construct quality (Yin 1984) has remained a concern, as demonstrated by the multiple validation loops performed and the triangulation of information from archives and documentation with the knowledge elicited from the requirements managers.

3.3 Defining causes of change

The methodology described in the previous section led us to the causes of change presented in Table 1. Although the list presented was derived in the context of the particular project studied at Rolls-Royce, we argue that it is applicable to most projects. Our belief is that the causes of change defined in this paper are comprehensive and cover most of the issues experienced during requirements engineering. Therefore, it complements literature on the requirements engineering topic (Kotonya and Sommerville 1998; Robertson and Robertson 2006; Pohl 2010). The issue of externally driven change is included in customer-, certification- and business-induced changes. Issues in requirements capture that may cause change are covered through missing requirements, incorrect capture, incomplete capture and incorrect or ambiguous language. Other capture related issues are described by requirements redundancy, decomposition and merger. Problems in the cascading process are described by incorrect and incomplete flow down. Inability to fulfill requirements or to verify them—change due to unachievable or unverifiable requirements—represents typical issues found during requirements validation and verification activities that often lead to change.

Finally, difficulties encountered during traceability management are also included in Table 1 as potential causes of change. Specific reasons not fitting in the above categories were grouped under the “Other” category (Table 1).

3.4 Sample design and representativeness

As previously referred, the statistical quantification of the requirements evolution was performed based on a representative sample. The study consisted in having the

Table 1 Comprehensive set of causes of change agreed with requirements engineers at Rolls-Royce

Cause of change	Definition
The customer changed the requirement	A change in customer needs, such as a change in the customer specifications, caused the requirement to change
Regulation or certification changed	A change in regulation or certification demands caused the requirement to change
Business requirements changed	A change in business needs caused the requirement to change
The requirement was missing	Although needed, the requirement was not previously captured and that caused the change
The capture was incorrect	The requirement was not correctly captured, due to wrong content or language, and that caused the change
The capture was incomplete	The requirement was captured but content was missing and that caused the change
The language was incorrect or ambiguous	Content was correctly captured but the requirement’s language or syntax was incorrect or unclear
The flow down was incorrect	The requirement was cascaded incorrectly from an upper level, e.g., misplaced, and that caused the change
The flow down was incomplete	Although needed, the requirement was not cascaded from an upper level and that caused the change
The requirement was decomposed	The requirement was decomposed into multiple requirements and that caused the change
The requirement was redundant	Awareness that the requirement was not needed, that content was repeated or that over-specification occurred, caused the change
The requirement was merged	Awareness that the requirement should be merged into another requirement caused the change
The requirement was unachievable	Awareness that the requirement was not achievable caused the change
The requirement was unverifiable	Awareness that the requirement could not be verified or validated caused the change
Traceability links or references changed	The content and the language were correct, but a traceability link or a reference mentioned in the requirement changed (e.g., updated and removed)
Other	The cause does not fit any of the above, and it was elicited by the expert’s own words

requirements managers from the project determining the cause of each change from a list of reasons. It thus equals to test each change against each possible cause and to verify whether the cause applies or not. According to

OBSERVED POPULATION				TOTAL (%)	SAMPLE DESIGNED				TOTAL (%)
	Additions	Modifications	Removals			Additions	Modifications	Removals	
Operational	A	B	C	8%	Operational	a	b	c	8%
Performance	D	E	F	14%	Performance	d	e	f	14%
Mass	3%	Mass	3%
Safety	Safety
...
Reliability	6%	Reliability	6%
Cost	W	X	Z	2%	Cost	w	x	z	2%
TOTAL (%)	45%	31%	24%	100%	TOTAL (%)	45%	31%	24%	100%
TOTAL SIZE				N	TOTAL SIZE				n

Fig. 4 Design principles used to arrive to a representative stratified sample. *Values are illustrative only*

sampling design methods (Lohr 2010), this case can be approximated by the case of a *proportion study*, where we are interested in determining the proportion p of the population that has some characteristic y_i . This variable is binary, taking $y_i = 1$ if individual i (a change in this case) has the characteristic (a possible cause) and $y_i = 0$ if it does not. Lohr (2010) shows that for a finite population of size N the sample size required for a chosen confidence interval z and margin of error e is a function of the proportion p :

$$n \geq \frac{Np(1-p)}{p(1-p) + \frac{e^2}{z^2}(N-1)} \quad (1)$$

The expression has a maximum when $p = 0.5$ which is recommended when no prior knowledge exists. Sample sizes were computed for a level of confidence of 95 % and a margin of error of 10 % in our study.

One of the main concerns of sampling design methods is to ensure that the samples designed are representative of the population. In this regard, the population of changes could be naturally divided into sub-groups of additions, modifications and removals, according to the *type of change*. Conversely, the population is also naturally grouped around *types of requirements*—operational requirements, performance requirements, mass requirements, etc.

We concluded through observation that large variations existed in the population and thus a simple random sample would be highly unrepresentative. A stratified sampling design method was consequently selected. In addition, stratified samples were designed using the proportional allocation method (Lohr 2010). Stratified samples were created to respect the two types of strata identified in the population, representing the distribution of change across the population according to the type of change *and* the type of requirements. The design principles used to engineer a representative stratified sample are illustrated in Fig. 4. Maximum differences in sample proportions relatively to

the population were approximately 1 %, due to rounding to the closest integer inside strata. Simple random sampling was then used inside the different strata to select the required number of changes from the population into the sample.

4 Presentation and discussion of findings

The analysis of the requirements evolution was performed at the time of *release* of new documents. After the initial capture, new released versions of the product requirements documents correspond to points in time where a major revision loop was finished and a complete set of updated system requirements was agreed by the project studied. Therefore, when we refer to requirements changes at a point in time, such value should be interpreted as the *accumulated amount* between two consecutive released versions (PRD_i and PRD_{i+1}).

4.1 Evolution of active requirements and types of change

Requirements evolve essentially through *modification* or *removal* of existing requirements and *addition* of new requirements. These can be thus considered different *types* of change. Lam and Shankararaman (1999) proposed the number or proportion of modifications, additions and removals in a given reporting period of a project as indicators of the requirements volatility (to distinguish between stable and unstable requirements) and the system's maturity. We have applied these concepts in our research to characterize and understand the evolution of the system's requirements in the project studied.

In addition, we define in this paper *active requirements* as the number or proportion of system requirements actually being *used* at a given point in time. The number of active requirements is given by the balance between the

requirements modified, added, removed and left unchanged relatively to the previous reporting period or document release:

$$\text{Active} = \text{No change} + \text{Modified} + \text{Added} - \text{Removed} \quad (2)$$

The evolution of the project's *system requirements* is presented in Fig. 5, built from the five major releases of product requirement documents that occurred during the development of the new gas turbine at Rolls-Royce. The time series found shows that the number of active system requirements roughly doubled during the course of the project, when compared with the number at the time of the initial capture.

Decomposition of the population of requirements that changed during the project according to the *type* of change is provided in Fig. 6. Analysis of Fig. 6 allowed us to observe a first stage dominated by modification of existing requirements following the initial requirements capture that occurred at PRD1. Although a significant amount of additions occurred at PRD2 (around 36 %), requirements modification was the dominant type of change both at PRD2 and PRD3, which were released during the preliminary design stage. Modifications were found in more than 50 % of changes in both cases (Fig. 6).

Between the third and fourth releases—a period that included the detailed design stage—Fig. 5 shows a slight decrease in the number of active system requirements. Figure 6 explains the reason behind this decrease, since the proportion of removals ($\approx 38\%$) is higher than the proportion of additions ($\approx 31\%$). Recalling that PRD4 was released shortly after the CDR and before the First Parts (FP) arrived (Fig. 5), it is also interesting to observe that the proportion of modifications reduced between the third and 4th releases and it is similar to the other types of change (around 30 %).

Figure 6 also demonstrates that the period comprised between the fourth and fifth releases was largely governed by additions (56 % of the total amount of changes at PRD5), which explains the increase in the number of active requirements in the later stages of the project that was observed in Fig. 5. Recalling the project's time frame, most of the physical testing and prototype development stage occurred during this time period. This effect was partly counterbalanced by removals, which accounted for about 19 % of changes as shown in Fig. 6. The requirements evolution found and the analysis of the type of requirements change found during the timespan of the project led us to the hypothesis that this requirements engineering process could be divided into three distinct phases:

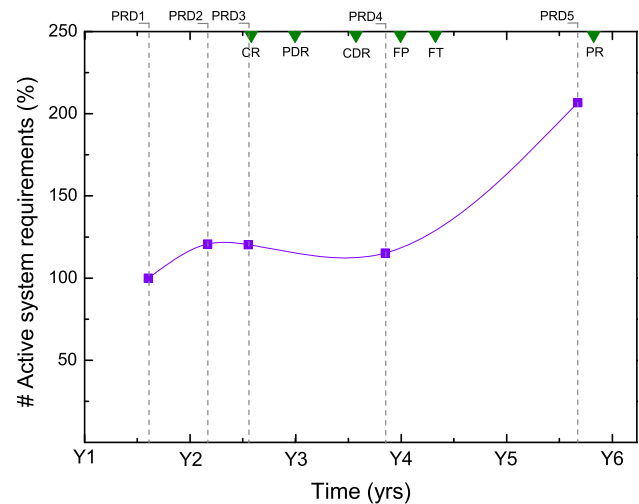


Fig. 5 Evolution of active system requirements throughout the course of the development of a new gas turbine. *Figures* correspond to values normalized by the number of requirements captured at the initial release of the PRD

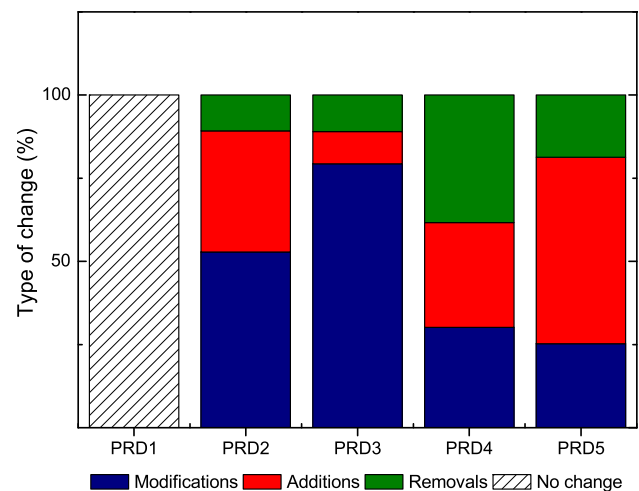
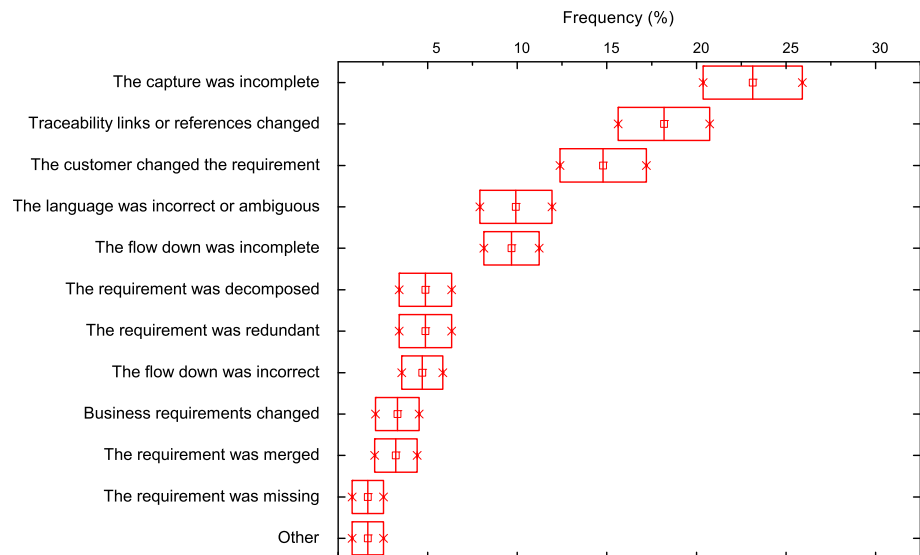


Fig. 6 Evolution of the types of change throughout the course of the project. *Figures* correspond to the accumulated proportion of requirements modification, addition and removal during the time period between two consecutive releases of the PRD

- Phase I—which started after the initial requirements capture and ended at PRD3—was mainly governed by modification of existing requirements;
- Phase II—which started after PRD3 release and ended at PRD4 release—was governed by a similar amount of additions, modifications and removals;
- Phase III—which started after PRD4 release and ended at PRD5 release—was governed largely by addition of requirements.

Fig. 7 Estimator of the mean proportion of each cause of change during the process phase dominated by modifications, between PRD2 and PRD3. A 50 % confidence interval of the mean proportion in the population is displayed



We supported this distinction in the observation that the initial and the late stages were substantially different and we assumed that the second phase corresponds to a transition between both stages. Our initial hypothesis was also that the population of system requirements changes during the initial and later stages of the process would also have very different root causes of change. This hypothesis defined the scope for the subsequent detailed study of causes of change.

It should be noted that these findings—evolution of active requirements and types of change—relate only to *one layer* of the requirements landscape (to system requirements in this case). This layer may be influenced by events occurring in other upstream or downstream layers. A typical example is related events occurring at the environmental layer that result from interactions with the customer, market, certification authorities, etc.

Since our findings characterize the requirements evolution found for the whole gas turbine system, they complement and significantly extend—in a quantitative manner—prior work performed within Rolls-Royce by Pickard et al. (2010) and Nolan et al. (2011), which reported high levels of requirements uncertainty during control systems' development and large amounts of requirements change between the CDR and Entry into Service of the product.

4.2 Causes of change

Due to the previous hypothesis, we focused our in-depth study of the causes of requirements change into phases I and III only. Samples representative of the population of changes observed at PRD3 and PRD5 were designed according to the principle illustrated in Fig. 4. The sample

size was computed according to Eq. 1 for a confidence level of 95 % and a margin of error lower or equal than 10 %. This resulted in representative samples containing 61 and 122 changes at PRD3 and PRD5, respectively.

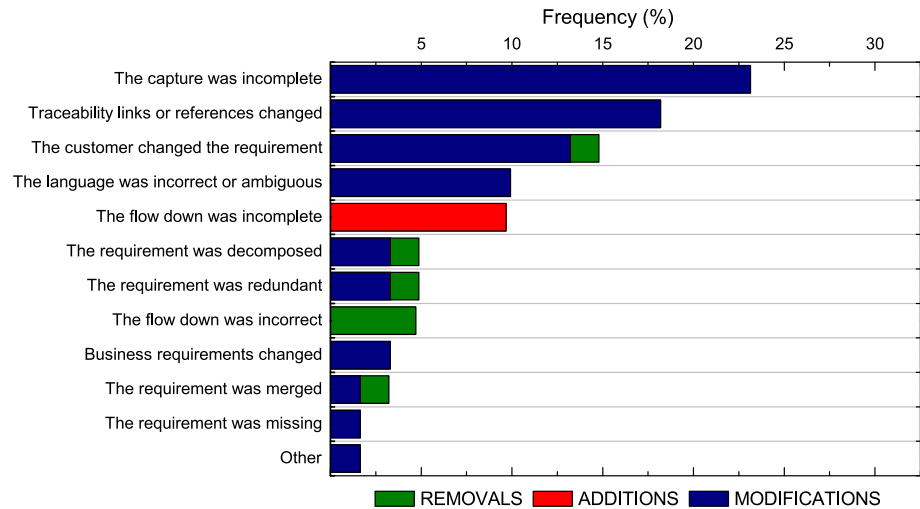
The assessment of the requirements managers then quantified the causes of changes relatively to previous releases, i.e., PRD2 and PRD4, respectively. Analysis of the statistical results presented throughout this section is based on three main statistical quantities: the estimator of the mean proportion in the population for each cause of change; the estimator of the proportion inside defined strata for each cause of change; and the standard error of the estimator of the mean proportion. Results follow sequentially for each of the two phases previously identified.

4.2.1 Modification-dominated phase

Statistical analysis showed that the most frequent cause of requirements change between PRD2 and PRD3 was incomplete capture. It was found in approximately 23 % of the changes, as presented in Fig. 7. We observed that the requirements content—targets, descriptive information, etc.—was progressively added into previously captured versions of the requirements during this early development phase. A typical example of a modification due to incomplete capture is provided below where all underlined text relates to changes at PRD3 compared with the same version at PRD2:

The system shall be capable of starting at X between Y°C (the hard limit of the oil temperature) and Z°C (the lower limit of the start environmental envelope as defined in Ref. A) with special starting procedures to raise the oil temperature. {Trace to: Business Requirements Document—...}

Fig. 8 Estimator of the mean proportion in the population according to the type of change, between PRD2 and PRD3



Change due to the modification of traceability information or references was the second most frequent cause, found in 18 % of the changes. These related essentially to modification of the system requirements' dependency information relatively to requirements captured in higher levels of the landscape, namely to business and customer requirements. Changes driven by the customer were the reason for approximately 15 % of the total number of changes, representing the third most frequent cause (Fig. 7).

Incorrect or ambiguous language and incomplete flow down appear as the fourth and fifth most frequent, respectively, with mean proportions around 10 % (Fig. 7). We found many examples of requirements modifications during this time period arising from semantics. An example that can be provided here is the modification at PRD3 of the expression “shall not hazard” by “shall not result in structural failure or hazardous system effects” in a particular system requirement.

All together, the top five causes accounted for more than 75 % of changes. Various other causes—business changes, missing requirements, incorrect flow down, redundancy, decomposition and merger—account for the remaining proportion, with individual mean values below 5 % (Fig. 7).

Decomposition of the data according to the type of change provides additional insight, which is presented in Fig. 8. The results demonstrate that all additions performed between PRD2 and PRD3 were caused by incomplete flow down of requirements. These relate to further decomposition of requirements from the environment layer—customer, business or certification requirements—down to the system level. Moreover, Fig. 8 also shows that incorrect flow down was the most frequent cause of removals during this period. This results from awareness that a particular requirement should be captured in or cascaded to another

level of the landscape. Change caused by the customer, requirements redundancy, decomposition and merger was also responsible for part of the removals (Fig. 8). Within the modification stratum, the most frequent causes of requirements change were identical to the ones found for the whole population. This result is expectable, recalling that during this time period modifications were the dominant type of change.

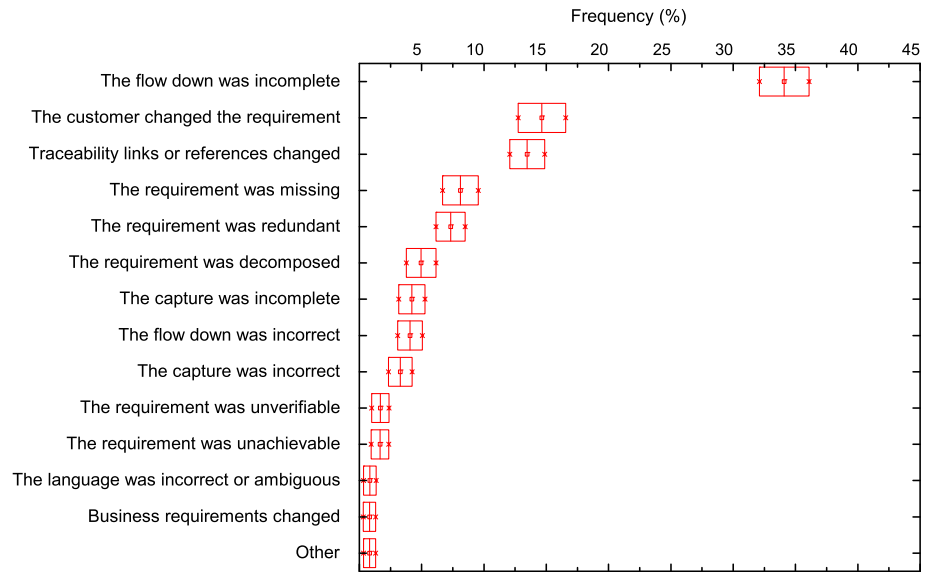
4.2.2 Addition-dominated phase

This period was noticeably different from the previous. Results support that incomplete flow down was in this case the most frequent cause of requirements change. Figure 9 shows that it was found in 34 % of the changes. Strata results presented in Fig. 10 demonstrated that more than 60 % of additions occurred due to incomplete flow down.

Our analysis revealed that requirements already existed in the environment layer, but they had not been formally cascaded and reengineered at downstream levels. We observed in strata results that the flow down was particularly intense into some types of requirements, such as into performance, structural, operational and life requirements. Performance relates to many functionalities that the gas turbine system must deliver at different operating conditions, structural requirements specify the level of integrity, the system must contain at all times, and life requirements specify the cyclic capabilities of the system under different operational conditions. We observed in the data that the flow down of certification requirements was also responsible for a proportion of the additions.

Change due to new customer requirements was found in almost 15 % of the cases (Fig. 9), becoming the second most frequent. The customer induced about 19 % of additions, approximately 17 % of removals and a relatively small proportion of modifications, about 3 % (Fig. 10). An

Fig. 9 Estimator of the mean proportion of each cause of change during the process phase dominated by additions, between PRD4 and PRD5. A 50 % confidence interval of the mean proportion in the population is displayed



interesting finding from our strata results relates to the distribution of customer-driven change. We found that it was more intense in performance requirements, which could be expectable considering that the gas turbine must deliver a high number of functionalities which have associated performance targets. The remaining proportion of change generated by the customer was similarly distributed throughout a variety of other types (program, mass, life, reliability and environmental requirements). The fact that a similar amount of change driven by the customer—about 15 %—was found both in the early and in the late stage of the process is a relevant finding.

Customer-induced change was followed by traceability generated changes, which accounted for approximately 13 % of the total amount of change accumulated between PRD4 and PRD5 (Fig. 9). These were the main bulk of modifications, as seen in Fig. 10. Figure 9 also shows that missing and redundant requirements became the fourth and fifth most frequent causes during this time period, respectively, with 8 and 7 %. Further strata analysis demonstrated that missing requirements appeared as additions (Fig. 10) and were concentrated in only a few types of requirements, such as in transportability or structural requirements. In the former, the time of this capture relates to increased understanding of transportability issues at this stage arising from the testing program of physical prototypes.

Redundancy was the most frequent cause of removals (about 39 %), as seen in Fig. 10. A deeper analysis revealed that in fact many removals due to redundancy were a direct result of addition of new requirements due to incomplete flow down. A *knock-on effect* was thus established between both types of change during this requirements engineering phase. Strata analysis also revealed that the majority of redundancies occurred in requirements

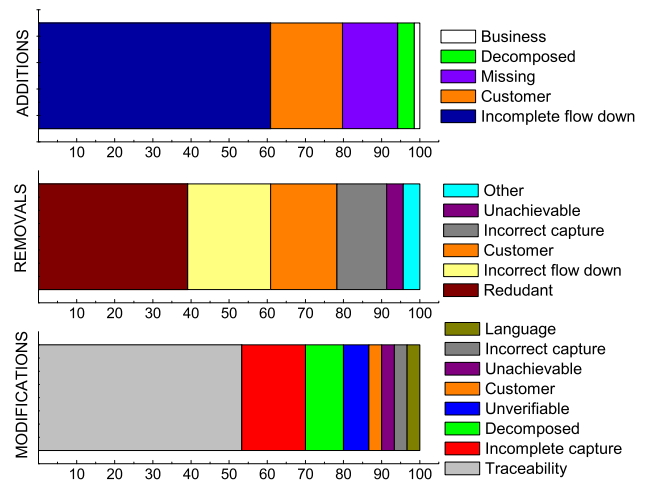


Fig. 10 Estimator of the proportion of each cause of change inside strata defined according to the type of change, between PRD4 and PRD5

previously captured for individual gas turbine product systems—e.g., oil systems and fuel systems.

These five causes led to more than 77 % of the total number of changes between PRD4 and PRD5. Decomposition was found in about 5 % of the causes, and incomplete capture and incorrect flow down were each found in about 4 %. Incorrect capture, unachievable and unverifiable requirements and business changes complete the set of causes found during this period with decreasing proportions (Fig. 9). It is interesting to notice that this small proportion of validation and verification related changes did not appear during the early process phase. Our analysis showed that they occurred essentially in program, performance and structural requirements.

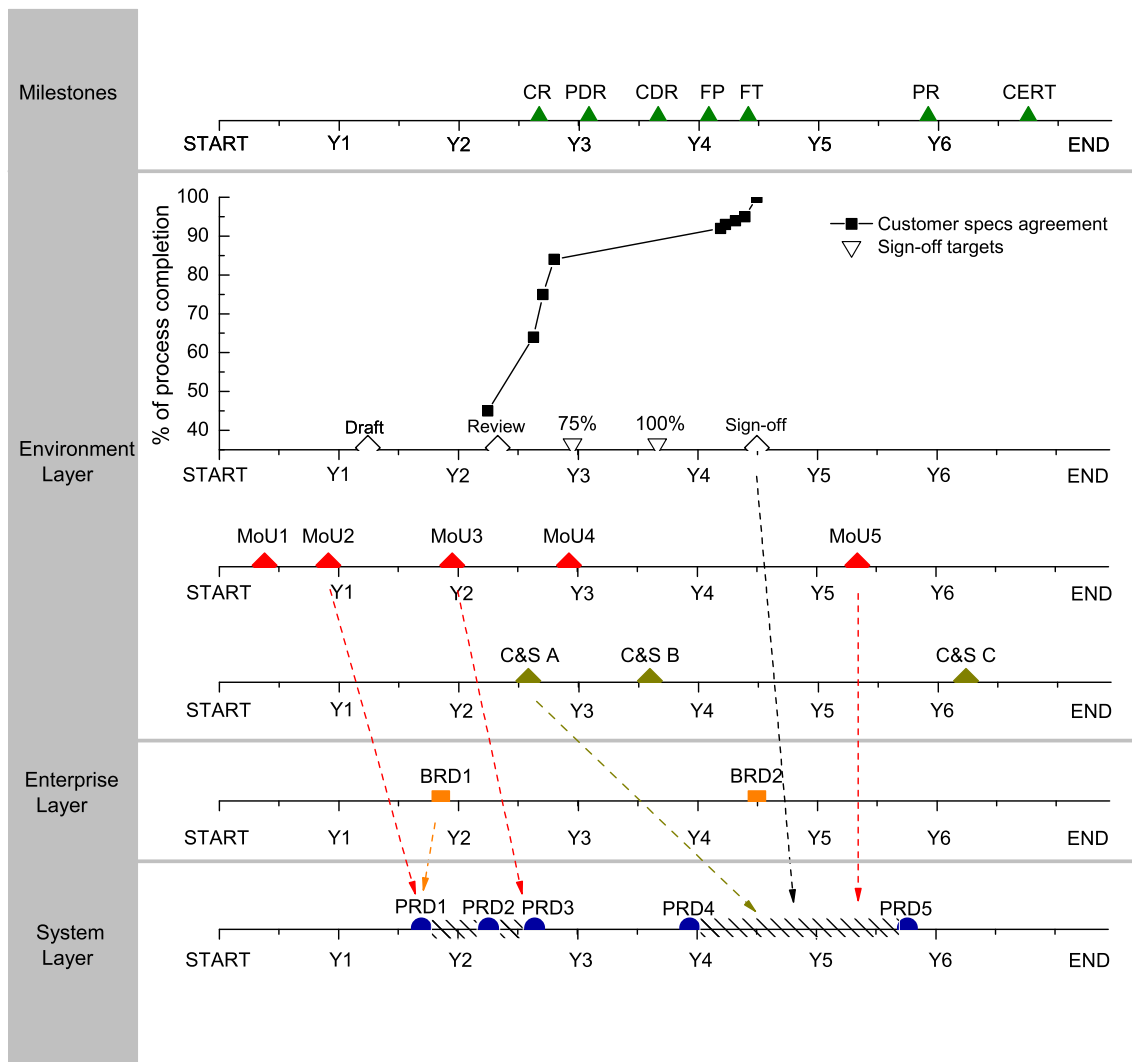


Fig. 11 Inter-layer analysis of the requirements engineering process during the development of a new gas turbine. *Arrows* represent flow down. *C&S* Certification and Standards, *BRD* business requirements document

4.3 Inter-layer analysis

Considering the previous results, we proceeded with an inter-layer analysis of the requirements engineering process performed during the development of this new gas turbine at Rolls-Royce. Our aim was to further understand the root causes of change, namely the causes arising from interdependencies between system requirements and requirements captured in other layers of the requirements landscape.

These interdependencies were visible in the amount of change triggered by incomplete flow down, for instance. The main findings generated from this inter-layer analysis are represented in Fig. 11. This figure joins the project’s time frame and the main events that occurred at each layer of the landscape which is upstream of the PRD. We have

included in this figure the *Environment* layer—containing requirements captured from the customer and from certification authorities—and the *Enterprise* layer, which contained requirements captured from internal business related functions.

We found that, at the time of initial capture, PRD1 captured customer requirements from the MoU (issue 2) and business requirements from issue 1 of the business requirements document (BRD) and engineered them into system requirements. The timing of the flow down process is illustrated in Fig. 11. During the modification-dominated phase of the project, part of the changes was triggered by the release of issue 3 of the MoU. This document contained new customer requirements and updated critical targets—such as constraints in maximum allowed weight or fuel consumption—that were incorporated into PRD3 in a

response to changing customer needs. This interaction is the origin of 15 % of the changes previously quantified between PRD2 and PRD3.

In parallel to the signature of the various MoUs, we found that a continuous interaction process was going on with the customer. This interaction consisted in the review and agreement process of the main bulk of customer specifications for the new gas turbine, consisting of *thousands* of technical specifications about the system.

Archival research demonstrated that the initial draft of the customer's technical specification documents was finalized at the end of the project's first year and that a review and negotiation process started between appointed customer and Rolls-Royce technical specialists. Data collected suggests that almost 50 % of the specifications were already agreed when the initial review finished. Moreover, as shown in Fig. 11, the agreement status increased rapidly, and by the end of the second year, about 75 % of the total amount was agreed (Fig. 11). This period coincides with the period comprised between PRD2 and PRD3 at system level.

Considering the sign-off targets that had been specified by the project, the agreement was on track at that time as illustrated in Fig. 11. However, the agreement rate slowed down during the third year. The process was at 90 % status at the targeted signature date, and the data collected showed that 9 months more were needed to complete the agreement process (Fig. 11). The official sign-off occurred during the second quarter of the fourth year.

Capture of the customer specifications occurred in the requirements database shortly after, and our investigation found that the systematic flow down and reengineering process of these specifications into system requirements started subsequently. This event explains a large share of additions due to incomplete flow down found during the period comprised between PRD4 and PRD5. It also means that requirements were *available earlier* but were only captured and engineered into downstream layers after the official signature.

In addition, the authors also found that certification specifications had been captured in the requirements management software tool during the project's second year (Fig. 11), but only begun to be flown down to the system level during the same time period. This second process also explains part of the new requirements found at PRD5 release.

Finally, a new MoU version (issue 5) was also signed late in the project which contained additional changes in customer needs. This new set was largely responsible for the amount of change caused by the customer (around 15 %) that was quantified during the previous study.

5 Discussion of improvement guidelines for management

Requirements change in complex technical systems is challenging for management. The previous results demonstrate that there are many root causes of change. Some—like customer-driven change—relate to external uncertainties which are typically outside the company's control. However, the vast majority is *self-induced* during the requirements engineering process itself due to a wide range of causes: incomplete capture, incomplete flow down, missing requirements, redundancy, decomposition, etc. Uncertainty and change triggers rework activities on requirements themselves and may also trigger repetition of many design activities, leading to higher product development costs and time-to-market.

An interesting question arising from these findings is the relationship between each cause of requirements change and the impact of the corresponding change implementation on program costs. Empirical experience suggests that types of change that implicate essentially carrying out the activity of requirements reengineering shall have significantly lower impact on development costs than those that imply the rework of component or even of whole subsystem design. For instance, requirements that are missing or incomplete and that are only discovered from testing problems (e.g., hardware malfunction) may have a large impact on program costs. Even if actual quantitative data about the impact of change are considered sensitive and need to remain protected, the findings reported in this paper provide valuable information for cost estimation through expert judgement, since our investigation has identified which types of change are most frequent and when they occur during the development process.

Regardless of the type of change, there are good reasons for desiring to reduce the share and the impact of internally driven requirements change. Our research methodology led us to conduct group sessions with the engineers responsible for managing system requirements during the development of the new gas turbine. The previous statistical and related findings were analyzed jointly. Several practical management guidelines aiming to further improve the requirements engineering process emerged from these sessions. These guidelines have the specific goal of reducing the *amount* or the *impact* of late requirements change. We present and discuss them in the following sections.

5.1 Front-load the requirements engineering process

Our investigation observed that the flow down process of many technical specifications into system requirements aiming to support design and development activities only occurred late in the process. Several reasons may justify this

fact. Firstly, complex technical systems, such as the one developed by Rolls-Royce, can have *very long* negotiation processes over customer specifications. Organizational policies may dictate that requirements should only be flown down when the process is completed to avoid wasting engineering resources. In such cases, the organization is *knowledgeable* of what the product needs to deliver but chooses only to engineer the requirements when it is sure of its content. Constraints over human resources available to perform the requirements engineering job may also justify the postponement.

However, the outcome of the group discussion between the requirements managers at Rolls-Royce favored *front-loading* the process to reduce the impact of late changes and the risk of proceeding with extensive testing programs without having all requirements engineered and in place. The cost of hardware problems in such cases was perceived to be *much higher* than the cost of wasted engineering resources during earlier product development stages due to potentially larger amounts of change. The aim of this strategy is thus to reduce the *impact* of requirements change through a reduction in the amount of change occurring late, but not necessarily through a reduction in the total amount of change over the project's lifespan.

The effects of front-loading the requirements engineering process are represented in Fig. 12. The process envisioned is one capable of *stabilizing* the number of customer requirements captured and the number of system, sub-system and component requirements engineered before the design freeze, which is planned to occur at the CDR.

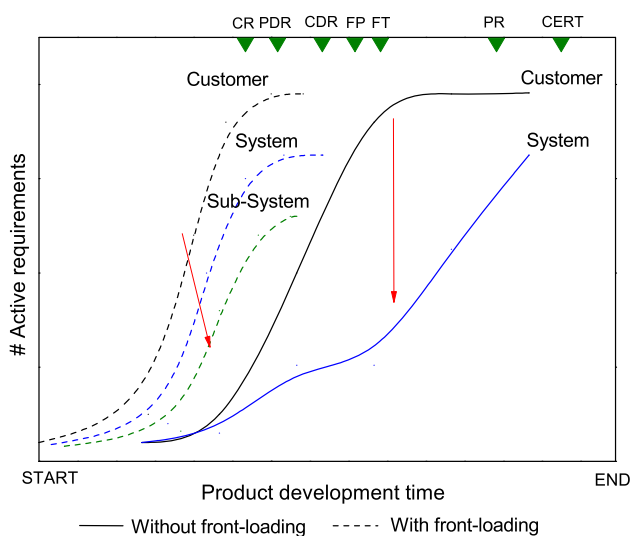


Fig. 12 Simplified representation of the effects of front-loading the requirements engineering process. *Arrows* represent flow down

5.2 Increase the level of concurrent requirements engineering

Front-loading the requirements engineering process demands a great deal of concurrent requirements engineering. It came out from the analysis of the statistical results by the requirements managers that the flow down process of requirements that have been *reviewed and agreed* can occur in parallel to the negotiation and agreement process with the customer. This principle of concurrency can be extended downstream across all layers of the landscape. Concurrency in requirements engineering is illustrated in Fig. 12 by the small delay in the stabilization of the number of active requirements across different levels of the landscape.

The group of requirements managers discussed the level of coordination and integration between requirements engineering teams that is needed in organizations such as Rolls-Royce to implement this practice. Concurrent processes must be anchored in *robust* change management processes which standardize how day-to-day procedures must be performed. These procedures specify how a team responsible for a higher-level requirement—e.g., a system requirement—should engage with a team responsible for a dependent lower-level requirement—e.g., a sub-system requirement—to communicate, evaluate, sign-off and implement a requirements change. Monitoring change implementation across system levels demands also a joined structure between the requirements database and the solution definition documentation, which points to additional developments in the IT infrastructure in place in the organization.

Furthermore, an increase in the level of process concurrency demands also an increase in the frequency of regular coordination meetings between the teams involved. These add organizational complexity and costs, but an increase in the level of requirements engineering process concurrency was perceived as an improvement guideline for management.

5.3 Allocate more resources earlier

Under normal market conditions, organizations developing complex technical systems typically struggle to find enough engineering resources to perform the design and development activities which are needed. However, concurrent processes are certainly difficult to implement without the right amount of resources allocated to requirements engineering.

In complex technical systems containing thousands of higher-level specifications, the process of flowing down and reengineering these specifications involves a *huge* amount of background engineering work to ensure that the

new requirements are complete, independent, atomic, unambiguous, etc. It is known from practice that under-resources teams will tend to serialize requirements engineering activities. Because of that, serializing the process increases the chance that the flow down process of many requirements into system, sub-system and component requirements occurs late in the process.

Allocating more resources is thus a management guideline that follows from the idea of increasing concurrency in the requirements engineering process. In addition, front-loading the process demands that resource allocation ramps-up during the preliminary design stage at all layers of the landscape, so that the capture and flow down process is completed before the CDR. It thus means that more resources need to be allocated to requirements engineering activities *earlier* in the process.

5.4 Invest in practical training

The previous strategies search for a reduction in the impact of late change. In addition, management guidelines aiming to reduce the *amount* of requirements change also emerged from the group analysis of the causes of change reported in this paper.

The statistical results demonstrated that a significant proportion of change arises also from incorrect capture or flow down, ambiguous language, issues with traceability, redundancy, etc. Discussion of many examples with the requirements managers pointed out that the capture of a single requirement is often a quite challenging process since it involves several areas of expertise that engage into negotiation about its content, language, dependencies, etc. This process is essentially *iterative*, which helps to explain the previous types of change. Because of that, a lot of “learning while doing” occurs.

Since the process is highly iterative and practical, the capability of mastering the requirements engineering process and reducing the amount of internally generated change depends on personal experience and skills which are typically accumulated *over time*. This knowledge may be lost when people move jobs in the organization, a situation that occurs often in large firms such as Rolls-Royce. A management guideline that came out from this discussion was the need to invest in *practical* requirements engineering training in organizations, in opposition to the traditional training model which tends to remain at a high level—explaining the general principles, the main stakeholders, the document landscape, the deployment process, etc. Conducting practical training—one relying heavily on example-based learning—across the organization was a guideline recommended also to ensure that requirements engineering knowledge is preserved and consequently

future projects are able to reduce the amount of self-induced change.

5.5 Set-up validation and verification systems

We also found that a part of the requirements was discovered to be unachievable or unverifiable during later stages of the process. The capture process can often occur without enough time being spent on understanding and documenting how particular requirements should be validated and verified. This typically occurs due to time and budget pressures that organizations face. However, organizations such as Rolls-Royce recognize that validation and verification efforts promote higher requirements quality and stability in the longer term.

Setting-up validation and verification systems was thus pointed out as a guideline to address this cause of requirements change. This system can be designed to best fit the organizational needs of each firm. Developing and using computational algorithms that test whether the requirement is complete, semantically correct and traceable is one possible approach to perform requirements validation. Introducing a standardized template attached to each requirement that allows the engineer to document how the requirement will be satisfied (through inspection, computational analysis and simulation, physical testing, etc.) is also one possible avenue to conduct requirements verification. The system in place should also allow metrics calculation so that management can inspect the amount of requirements that have been specified with validation and verification checks in product development projects.

5.6 Effects of the guidelines in the process

Front-loading, increasing the level of requirements engineering concurrency, allocating more resources earlier, investing in practical training and setting-up validation and verification systems were various management guidelines identified and discussed. The effects envisioned on the requirements engineering process are illustrated in Fig. 13. This qualitative representation of the requirements change evolution aims to point out that there are two main improvements targeted by these guidelines. The first is a reduction in late requirements change, and the second is a reduction in the total amount of change observed during product development projects.

6 Conclusions

This paper contributes with an increased understanding of the requirements evolution and the causes of change found *at different phases* of the product development process of

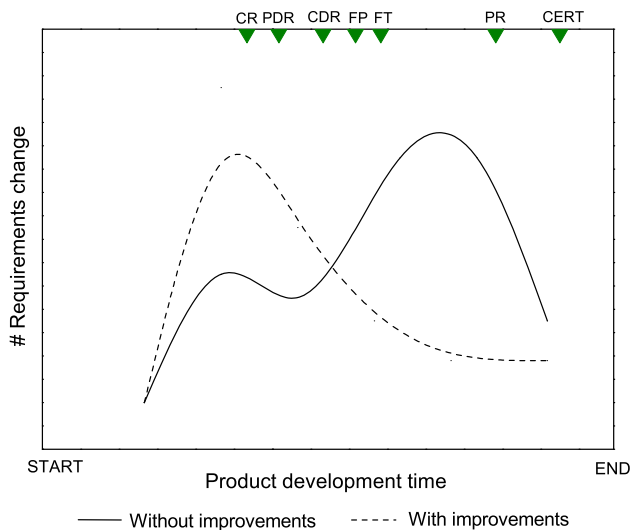


Fig. 13 Simplified representation of the effects of the management guidelines captured in the paper on the evolution of requirements change during the product development process

complex technical systems. We present findings from an empirical study performed at Rolls-Royce which investigated a new gas turbine developed by this manufacturer during the course of 6 years. We have collected and studied the requirements database of this system. It contained 700 system requirements and its evolution was embodied in more than 1,000 changes, which occurred during the product development process. Our research approach generated original quantitative results about requirements change during the product development process which extend current knowledge on the topic of requirements engineering and management.

The issue of result generalization always arises from case-study approaches (Yin 1984). However, we argue that the main findings contained in this paper are generally representative of complex product development processes. It is common that large technical systems require long development processes which make them more vulnerable to external uncertainties and long and laborious negotiation processes with multiple stakeholders about high-level technical specifications. Requirements engineering processes are thus highly challenging for organizations developing such systems. The case study showed that most requirements change is self-induced due to a variety of different causes, being incomplete capture and incomplete flow down some of the most frequent. Considering that Rolls-Royce is one of the most competitive companies in the world, we argue that the same finding is likely to be encountered in many organizations developing complex technical systems.

Looking back to the questions we presented in Sect. 1 which motivated this research, the authors argue the

findings described in the paper answer them in the following manner:

- Relatively to *how requirements evolve in organizations developing complex technical systems*, we concluded that the number of active requirements roughly doubled during the course of the development process. Furthermore, the evolution showed that the stabilization of the number of requirements can occur late in the process, typically during prototype development and testing.
- The question concerning *how much change is externally and internally driven* has been clearly answered. We found that more than 80 % of changes generated during the development of the complex system studied in this paper had internal root causes. We have also observed that the amount of change generated from the customer remained relatively constant during the development process, around 15 %, which is an interesting finding. Our belief is that this trend is general, but the exact proportion of external versus internal change may vary across projects and industries.
- In the context encountered, our results show *how do the causes of change vary during the different phases of the development process*. We concluded that the preliminary design stage was dominated by requirements modifications due to incomplete capture. Traceability issues, customer-driven change and incorrect or ambiguous language were also relevant during this stage. Conversely, we found that late development stages—namely during prototype development and testing—were governed by requirements addition due to incomplete flow down. Customer-induced change, missing requirements, traceability and redundancy were also frequent during this phase. Knock-on effects between some of the causes of change were also observed.

The conclusions stated above constitute major contributions from this empirical research. This knowledge constitutes a general academic contribution from this paper, which complements previous empirical studies (Lindvall 1997; Almfelt et al. 2006; Michael et al. 2007) and recognized literature (Kotonya and Sommerville 1998; Robertson and Robertson 2006; Pohl 2010) on the topic of requirements engineering and management.

In addition, these findings led the group of engineers involved in the research to analyze various management guidelines aiming to improve requirements engineering in practice. Front-loading the requirements engineering process was proposed as a way to reduce the impact of late change. Increasing the level of concurrency in requirements engineering processes was discussed to ensure that the number of requirements captured and engineered

stabilizes roughly at the same time and before design freeze at all levels of the landscape. Allocating more engineering resources earlier in the process was a management guideline that followed from the previous two. Moreover, investing in example-based training across the organization and setting-up validation and verification systems were recommended to ensure that future projects are able to reduce the amount of internally generated change. The previous management guidelines ultimately aim to attain processes that generate less changes and changes with lower potential impact since they occur earlier in the development process. We argue that the management guidelines captured in this paper are also useful for other organizations performing product development and practicing requirements engineering.

In conclusion, it is true that effective requirements management *alone* does not ensure success in the development of complex technical systems, such as the one studied in this paper. However, the authors believe that it surely helps.

Acknowledgments The authors would like to thank Rolls-Royce Plc and all engineers and managers that have been involved in this study. We are grateful for their enthusiasm throughout the research and their effort to support us with all their knowledge and experience. We specifically thank the engineers from the project studied in this paper for many hours of their time that they have spent with us in the statistical assessments, group discussions and reviewing written material. Finally, funding from the Portuguese Foundation of Science and Technology under the doctoral Grant SFRH/BD/51107/2010 is also gratefully acknowledged.

References

- Almefelt L, Berglund F, Nilsson P, Malmqvist J (2006) Requirements management in practice: findings from an empirical study in the automotive industry. *Res Eng Des* 17:113134
- Bergsjö D, Almefelt L, Malmqvist J (2010) Supporting requirements management in embedded systems development in a lean-influenced organization. In: Proceedings of the 11th international design conference (DESIGN 2010), Dubrovnik, Croatia, pp 1025–1034
- Blanchard BS, Fabrycky WJ (2006) Systems engineering and analysis, 4th edn. Prentice Hall, Englewood Cliffs
- Ćatić A, Malmqvist J (2010) Requirements management when introducing new mechatronic sub-systems: managing the knowledge gaps. In: Proceedings of the 11th international design conference (DESIGN 2010), Dubrovnik, Croatia, pp 661–672
- Chakrabarti A, Morgenstern S, Knaab H (2004) Identification and application of requirements and their impact on the design process: a protocol study. *Res Eng Design* 15(1):2239
- Clarkson PJ, Simons C, Eckert C (2004) Predicting change propagation in complex design. *J Mech Des* 126(5):788–797
- Dorst K, Cross N (2001) Creativity in the design process: co-evolution of problem solution. *Des Stud* 22(5):425–437
- Eckert C, Clarkson P, Zanker W (2004) Change and customisation in complex engineering domains. *Res Eng Des* 15(1):1–21
- Fricke E, Gebhard B, Negele H, Igenbergs E (2000) Coping with changes: causes, findings, and strategies. *Syst Eng* 3(4):169–179
- Hamraz B, Caldwell NHM, Clarkson PJ (2012) A multidomain engineering change propagation model to support uncertainty reduction and risk management in design. *J Mech Des* 134(10):100,905
- Jarratt T, Eckert C, Caldwell N, Clarkson P (2011) Engineering change: an overview and perspective on the literature. *Res Eng Des* 22(2):103–124
- Koh E, Caldwell N, Clarkson P (2012) A method to assess the effects of engineering change propagation. *Res Eng Des* 23(4):329–351
- Kotonya G, Sommerville I (1998) Requirements engineering: processes and techniques, 1st edn. Wiley, Chichester
- Kuzel A (1992) Sampling in qualitative inquiry. In: Doing qualitative research: research methods for primary care (vol 3). Sage, Beverley Hills
- Lam W, Shankaraman V (1999) Requirements change: a dissection of management issues. In: Proceedings of the 25th EUROMI-CRO conference, Milan, Italy
- Lam W, Loomes W, Shankaraman V (1999) Managing requirements change using metrics and action planning. In: Proceedings of the third European conference on software maintenance and reengineering, Amsterdam, Netherlands
- Lindvall M (1997) An empirical study of requirements-driven impact analysis in object-oriented software evolution. PhD thesis, Linköping University, Institute of Technology, Sweden
- Lohr SL (2010) Sampling: design and analysis, 2nd edn. Books Cole, Boston, MA
- Maher M, Poon J, Boulanger S (1996) Formalising design exploration as co-evolution. In: Gero J, Sudweeks F (eds) Advances in formal design methods for CAD, IFIP, the international federation for information processing. Springer, New York, pp 3–30
- Michael S, Kilian G, Lucienne B (2007) Managing requirements or being managed by requirements—results of an empirical study. In: Proceedings of the 16th international conference on engineering design (ICED07), Paris, France
- NASA (2007) Systems engineering handbook, 1st edn. National Aeronautics and Space Agency, USA
- Nolan A, Abrahao S, Clements P, Pickard A (2011) Managing requirements uncertainty in engine control systems development. In: IEEE 19th international requirements engineering conference, Trento, Italy, pp 259–264
- Pickard A, Nolan A, Beasley R (2010) Certainty, risk and gambling in the development of complex systems. In: INCOSE international symposium, Chicago, USA
- Pikosz P, Malmqvist J (1998) A comparative study of engineering change management in three Swedish engineering companies. In: Proceedings of ASME design engineering technical conferences, Atlanta, GA, USA, pp DETC98/EIM-5684
- Pohl K (2010) Requirements engineering: fundamentals, principles and techniques, 1st edn. Springer, Berlin
- Robertson S, Robertson J (2006) Mastering the requirements process, 2nd edn. Addison-Wesley, Amsterdam
- Robson C (2002) Real world research: a resource for social scientists and practitioners, 2nd edn. Blackwell Publishing, London
- Rolls-Royce (2013) <http://www.rolls-royce.com/>
- Sudin MN, Ahmed S (2009) Investigation of change in specifications during a product's lifecycle. In: Proceedings of the 17th international conference on engineering design (ICED09), Paolo Alto, CA, USA, pp 371–380
- Sudin MN, Ahmed-Kristensen S (2011) Change in requirements during the design process. In: Proceedings of the 18th international conference on engineering design (ICED11), Denmark, Copenhagen, pp 200–208

- Tseng M, Jiao J (1998) Computer-aided requirement management for product definition: a methodology and implementation. *Concurr Eng* 6(3):145160
- Vianello G, Ahmed-Kristensen S (2012) A comparative study of changes across the lifecycle of complex products in a variant and a customised industry. *J Eng Des* 23(2):99–117
- Weber M, Weisbrod J (2003) Requirements engineering in automotive development: experiences and challenges. *IEEE Softw* 20(1):1624
- Yin RK (1984) *Case study research design and methods*. Sage, Beverly Hills