1 Introduction

1.1 Aim of Book

The aims of this book are to motivate successful improvements to requirements management, to promote understanding of requirements management as one of an interrelated set of systems engineering disciplines, and to understand these systems engineering disciplines and their interfaces to requirements processes.

1.2 Benefit to be gained from book

By understanding and following the guidance in this book you will be able to reap benefits of synergy between team members and across departmental boundaries by coordinating efforts in requirements management as part of your systems engineering activities.

We have seen organisations that as they have grown have developed to become a collection of independent departments. Too often these departments concentrate increasingly on achieving their departmental aims, eventually to the extent that their departmental aims become more important than the aims of the overall organisation. What we need is coordinated teamwork where each part of the team pulls in the same direction.

This book helps a team to understand the central role played by requirements in systems engineering projects. It shows that no one systems engineering discipline is more important that any other. It shows that all the systems engineering disciplines are interrelated and interdependent.

This book will establish the need for, and legitimise the use of requirements management and engineering.

Managing requirements consists of managing changes to requirements, managing various versions of requirements, managing multiple configurations of requirements, managing deliveries of requirements on time, in budget and to the correct quality without taking undue risks. And all the time ensuring that all those who need to know, know who is responsible for what. All of this requires communication and commitment.

Product and services produced to meet requirements must be checked against requirements to ensure that the specified and agreed requirements have been achieved

A perfectly optimized system is a set of suboptimal subsystems. If teams try to optimise each subsystem there will be conflict. Following the advice in this book teams will be inspired to see the big picture and be able to concentrate on getting the system built as required.

To introduce terms such as RM&E (requirements management and engineering) and relate to other nomenclature so that CMMI (Capability Maturity Model Integration) terms may be used throughout the book

1.3 Definition of terms

CMMI: Capability Maturity Model Integration. A framework for scoring an organisation's ability to work with systems engineering processes. CMMI comes from the Software Engineering Institute (SEI) of Carnegie Melon University in Pittsburgh U.S.A. Various trademarks and service marks of the SEI relating to CMMI are acknowledged.

HCM: HOOD capability model. A model for judging the quality of the implementation of a process mainly by considering the quality of its work products. Often used to support motivation of change programmes by measuring and publishing progress.

Process: (see also Software Process). A sequence of steps performed for a given purpose; for example, the software development process.

Requirement: A statement identifying a capability, physical characteristic, or quality factor that bounds a product or process need for which a solution will be pursued.

Requirements Definition: The process of producing documented and agreed requirements by means of elicitation, specification, analysis (quality check: judgment of requirements against quality criteria), and review (leading to acceptance, rejection, or return for rework) of requirements.

Requirements Development: The purpose of requirements development is to produce and analyze customer, product, and product-component requirements.

Requirements Engineering: See Requirements Development.

Requirements Management: The set of procedures that support the development of requirements including planning, traceability, impact analysis, change management and so on.

Requirements Management: The sum of the interfaces between requirements development and all other systems engineering disciplines such as configuration management and project management. The purpose of requirements management is to manage the requirements of the project's products and product components and to identify inconsistencies between those requirements and the project's plans and work products.

RM&E: Requirements management and engineering. The overall term used to include all requirements related processes.

Note to RM&E: In the 1990's the overall term used was requirements management. Then towards the end of the 1990's and early in the new millennium a trend gathered momentum to split the management of requirements from the development of requirements. Some organizations made the distinction along the lines that developing or defining requirements was requirements engineering. Others disagree. Some organisations use both terms requirements management and requirements engineering and consider that their understanding is the one and only true definition. Other organisations use definitions that completely contradict the understanding of others, and also consider that their understanding is the one and only true definition. Some use the CMMI definitions of requirements management and requirements development and combine these by using requirements engineering to encompass everything. So we use the term requirements management and engineering in an attempt to include all people, while acknowledging that there are a variety of definitions. At work we use whatever terminology our customers wish. There are more important battles to fight than who has the best words. People who get hung up on whose definition is correct, (or more normally the fight is who is incorrect!) should read A. A .Milne and learn from Winnie the Pooh; "We can use words to mean whatever we wish them to mean". As long as we understand each other we can work together. We advise the use of standards wherever possible. Where there is no single standard we must agree amongst ourselves.

Software Process: (see also Process). A set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products.

Stakeholder: A "stakeholder" is a group or individual that is affected by or in some way accountable for the outcome of an undertaking. Stakeholders may include project members, suppliers, customers, end users, and others.

Alternative definition: People who will be affected by the project or can influence it but who are not directly involved with doing the project work. Examples are managers affected by the project, process owners, people who work with the process under study, internal departments that support the process, customers, suppliers, and financial department.

Alternative definition: People who are (or might be) affected by any action taken by an organization. Examples are: customers, owners, employees, associates, partners, contractors, suppliers, related people or located near by.

Alternative definition: Any group or individual who can affect or who is affected by achievement of a firm's objectives

Test: (See Validation and Verification). The activity of checking correctness.

Verification: Although "verification" and "validation" at first seem quite similar in CMMI models, on closer inspection you can see that each addresses different issues. Verification confirms that work products properly reflect the requirements specified for them. In other words, verification ensures that "you built it right."

Validation: Validation confirms that the product, as provided, will fulfill its intended use. In other words, validation ensures that "you built the right thing."

1.4 Structure of the Book

This book is divided into three parts. Part one is Requirements Management and Engineering: requirements management is greater than the sum of its parts. In this first part the aim is to introduce the book and to establish a common understanding of terminology with the reader.

Part two is Getting Down and Dirty: the low-down on systems engineering disciplines and their interfaces to requirements processes. In this part the aim is to discuss in detail the systems engineering disciplines, and specifically to define the interface between each systems engineering discipline and requirements development.

Part three is A Practical guide: helping to motivate and support successful implementation of requirements driven improvements with HOOD capability models (HCM).

Part one starts with Chapter 1 Introduction. This is the introduction to the book and describes the aims of the book and the benefits to be gained from reading the book. Some terms are defined that aid the understanding of the following chapters. Some terms are defined more than once quoting from various sources to show that the terms, although used often, are not standardized. If these definitions are contradictory it is made clear which definition is to be used in this book. The structure of the book is explained and a guide is given how to read this book to get the best advantage.

Chapter 2 Why Requirements Management and Engineering? In this chapter the need for and benefits of requirements management and engineering (RM&E) are explained. Terms such as RM&E and other nomenclature are explained so that CMMI terms may be used throughout the book. Definitions and detailed introduction of interfaces between systems engineering disciplines is not done here, but are investigated later in chapter 5.

Chapter 3 is Processes and Methods in Requirements Management and Engineering. The aim is to introduce the other systems engineering disciplines and process including those described by CMMI which will be explained in detail to form the structure of Part 2, the main part of this book.

Chapter 4 is Introduction to Requirements Engineering. The chapter introduces and defines requirements engineering, requirements development, and the HOOD requirements definition process.

Chapter 5 is Introduction to Requirements Management. Requirements management is firstly defined in terms of its activities and also in terms of its results. This is the introduction to a discussion that is the main part of this book. After the main discussion involving all the other disciplines, requirements management according to HOOD will be redefined in the more advanced, inclusive, and sophisticated way.

Part 2 starts with Chapter 6 Project Management. Project management is introduced, and its relationship to requirements development is investigated. The overlapping responsibilities between a project manager and a requirements manager are discussed, and the information common to both disciplines is exposed. The similarity and differences between tasks on a project plan and the requirements in a specification are explored.

Chapter 7 is Configuration Management. Configuration management is introduced and its relationship to requirements development is investigated. The overlapping responsibilities between the roles of configuration management and a requirements manager are discussed, and the information common to both disciplines is exposed. The idea of requirements and related information as configuration items and a set of requirements as a specification are explored.

Chapter 8 is Measurement and Analysis. Measurement and analysis is introduced, and its relationship to requirements development is investigated. The role of measurement and analysis is discussed, and the information common to measurement and analysis and requirements development is exposed. The similarity and differences between measurements of requirements and the requirements in a specification are explored. Particularly the need for measuring to support an aim rather than measuring just because a measurement is possible is emphasised.

Chapter 9 is Risk Management. Risk management is introduced, and its relationship to requirements development is investigated. The overlapping responsibilities between a risk manager and a requirements manager are discussed. The information common to both disciplines is exposed. The different types of risks and risk mitigation and their influence on the requirements in a specification are explored, as is the influence of requirements as a source of risk.

Chapter 10 is Test Management (Verification and Validation). Test management is introduced, and its relationship to requirements development is investigated. The overlapping responsibilities between a test manager and a requirements manager are discussed, and the information common to both disciplines is exposed. The similarity and differences between test cases in a test plan and the requirements in a specification are explored. The difficulties encountered when using a test plan as a requirements specification are recounted.

Chapter 11 is Change Management. Change management is introduced, and its relationship to requirements development is investigated. The overlapping responsibilities between a change manager and a requirements manager are discussed, and the information common to both disciplines is exposed. The similarity and differences between change requests and the requirements in a specification are explored.

Chapter 12 is Advanced Requirements Management: the complete specification. This chapter is the summary of all previous chapters in Part 2, showing requirements management as a complete specification of the interfaces between requirements engineering and other systems engineering disciplines.

Part 3 is a practical guide, helping to motivate successful implementation of requirements driven improvements with HCM. Part 3 starts with Chapter 13 The HOOD capability models. This chapter provides motivation for supporting change. Psychological reasons why we need to measure and publish progress or lack of progress is described. Chapter 13 The HOOD capability models is an introduction to Chapters 14 and 15 which describe the HOOD capability models in detail.

Chapter 14 is HCM for Requirements Definition. The aim of this chapter is to introduce and define HCM for requirements definition. Help is offered for introducing improvements in practical terms. A stepwise introduction is recommended and supported, rather than present a theoretical treatise.

Chapter 15 is HCM for Requirements Management. The aim of this chapter is to introduce and define HCM for requirements management. Help is offered for introducing improvements in practical terms. A stepwise introduction is recommended and supported. This chapter pulls together the threads of all previous chapters and weaves the themes together to create a tapestry of all the ideas presented thus far. Each thread remains distinct but still takes its place in the overall picture.

1.5 How to read this book

There are many ways to use this book. Consider it a resource from which we may take as we please. The book may be read from beginning to end or it may be used as a reference book and dipped into time and again with random access as a project progresses.

The book is divided into three parts. Part 1 is an introduction and sets the scene for the rest of the book. Part 2 deals with systems engineering disciplines and their relevance to requirements management. This is the main part of the book that you will use as a reference for technical information within your project. Part 3 deals with the challenge of introducing the ideas of Part 2 into your organisation. Part 3 is the part of this book that you will use as a reference for management information to support your team during the introduction of improvements.

Reading from beginning to end is particularly useful not only for those beginning to grapple with the complexities of requirements management. Also, those with many years experience might enjoy seeing things explained that previously had been taken for granted. You may agree with some or all of the various opinions presented. We hope that if you at first disagree with some of the opinions that this will help you to respect the fact that there are many dearly held opinions, and that our industry is still in its infancy and there is much still to be standardised. We have tried to represent various views while being sure to tell you what we have found to work in practice. In cases where there is disagreement we fall back on the author's years of experience in engineering since 1977 and use our common sense. Remember that this subject is no longer a technical challenge; we are dealing with peoples' understanding and people's failure to understand. We are dealing with peoples' weaknesses and insecurities. So please be kind to those with a different opinion to yourself. As my Father said, "You can learn something from any idiot". Your challenge at work is to include others and to strengthen the team. We hope this book helps you better to construct and explain your arguments in order that you can convince. We hope this book helps you better to understand the opinions of others that you can benefit from a broader view.

Using the same principal as is used in good training courses, information will be introduced in context to be defined later in detail. This organisation of information enables and supports the reading of the book from beginning to end. Learning is supported and encouraged by introducing topics in a broad way. By using a technique similar to active listening, the reader is encouraged to ask questions. By asking questions the answers to the questions become much more relevant than if the information is presented in a series of unrelated and surprising facts. By removing the element of surprise as another piece of information is presented learning is supported. In this book when information is dealt with in detail there is no surprise as we have prompted the questions by the manner of the introduction.

Consider your brain to be a series of cupboards, drawers, and shelves. The stuff you store on shelves is easily visible and can be found without much structuring. You just have to scan the shelves for what you want, and when you see what you are looking for your search is over. But your shelf space is limited, and anyway, even if you could replace all of your drawers and cupboards with shelves the search would take too long. Only having shelves for storing information would be like the advice given in the comic Viz, "If you have trouble finding things, just thread all of your possessions onto a piece of string. If you lose anything, just follow the string from one end to the other and you will be sure to find what you are looking for." Now, that is a technical challenge!

So we need more than just shelving. We need more structure. If we have lots of things put away we need to file things in some order so that we can find them again. We need cupboards so that we can store different things in different places. The cupboards may contain shelves but these are only visible when we have opened the cupboard door. If we do not later shut the door properly the contents may fall out.

The early chapters can be considered to be an explanation of which cupboards we have. By explaining the aims of each chapter and each part of each chapter we open the cupboard doors so that you may file the information away in your brain. By summarising each chapter we help you to close the cupboard door to prevent the contents falling out. The aims at the beginning and the summary at the end of each chapter support learning.

The modular form of this book supports the reading of the book as a reference book so that detail is easy to find just when your project needs it.