The Significance of Configuration Management in the Context of Systems Engineering

Stephen Watts[1], Mario Kossmann[2]
AIRBUS
[1]stephen.watts@airbus.com
[2]mario.kossmann@airbus.com

INCOSE, International Symposium 2010

Abstract

Systems Engineering (SE) is widely recognised to be the key discipline to develop complex products in many industries. There is strong evidence that requirements related issues are at the heart of the majority of project failures, delays or budget overruns. Yet Requirements Engineering (RE) frequently does not seem to be sufficiently interfaced or integrated with other engineering disciplines such as design, manufacturing, finance, procurement and customer services. Configuration Management (CM) is widely accepted as a key discipline to establish this traceability and enable controlled baseline evolutions or iterations within the development process. However, the traditional scope of CM in the context of SE more often than not excludes requirements in the early phases of the development process. The present paper will give a brief introduction into the importance of RE in the SE context and the significance of CM to establish traceability between the requirements, design and product, before highlighting in more detail the potential contributions of the CM discipline with an emphasis on the early phases of the SE development life cycle.

Introduction

Systems Engineering (SE) is widely recognised to be the key discipline to develop complex products in many industries. SE can be understood to comprise a set of distinctive concepts, methodologies, organizational structures etc. that are developed to meet the challenges of engineering non-functional and functional physical systems of highly complex products and systems. An aircraft like the Airbus A380 is a prime example of a highly complex SE product.

Initially starting in software development, the last decades brought about tremendous increase in the use of Requirements Engineering (RE) in almost all technology driven industry sectors such as the IT, aerospace, defence, pharmaceutical, automotive, and telecommunications industries, just to name a few. The scope of requirements dealt with has been constantly widened and does not only include performance, safety and functional requirements, but also increasingly target cost, quality, confidentiality, environmental and other non-functional requirements. There are many publications about projects that went wrong for various reasons, the majority of which are connected in one way or another to the lack of or bad implementation of RE [1, 2]. Also, there seems to be consensus amongst many that requirements need to be properly elaborated and taken as the basis of the development work for any complex system [e.g. 3, 4, 5, 6, 7]. However, it seems that a significant number of project managers or senior managers still think of RE as something that takes place with limited resources in the beginning of a project only. Some may even well see the advantages of sound RE but feel tempted not to see their yearly bonus reduced by over-spending their budget for requirements work, if they can under-spend it by not doing it.

There is strong evidence that if you know what your customers need, if you keep in touch with them and allow for changes in their needs over time, and if you have traceability to your requirements and design so that you can actually analyse the impact of changes, then you are in a much better position to develop successful systems, earn money doing so and gain sustainable competitive advantage over time [8].

Alexander and Stevens (2002) argue that ‘if a system designer reaches for the wrong goal, everything they do will be wrong. A single wrong requirement is likely to create a shower of design mistakes’ [1].

Hooks (1990) argues that programme cost overruns and schedule delays are to a large part associated with requirements problems, ranging from ‘incomplete, inconsistent, and incomprehensible requirements to the complexities of the change management process’ [9]. She identifies two basic underlying causes of these problems, i.e. ‘inability to write good requirements and a lack of understanding about the importance of requirements’. Both causes are ‘compounded by the scope, complexity, and long life cycle of major programs’ [9].

Chatzoglou and Macaulay (1995) found that RE is an iterative process (18% of the projects that were considered performed just one iteration, 32% performed two, and 50% performed 3 or more iterations). They also found that the more time is spent on the RE stage, the less time is spent in the entire development process. And similarly, the higher the cost of the RE stage, the lower the cost of the whole development process [10].

Bahill and Henderson (2005) offer good and bad examples of famous systems where requirements development, validation and verification were done correctly and incorrectly respectively, in order to highlight the importance of these activities for the successful realisation of systems [2].

Lucchetta, Baroni, Delaire and Bariani (2008) investigated the importance of RE in the context of aircraft development, using the example of Airbus. They conclude that the design cycle of an aircraft is longer than the fast changing needs of customers within the air transportation environment and that the ‘robustness of architectural aircraft design during the concept phase against moving requirements throughout the product development is critical for the final, long-term success of the product’ [11]. Fricke and Schulz (2005) had come to similar findings in the automotive industry and offer an approach to cope with changeability in their context [12].

Being able to conceive a brilliant idea that will reshape the world and develop that idea into an affordable product or system is one thing. However, it is quite another having the ability to know how you managed to design, develop, produce, manufacture, distribute, operate, maintain, support, phase-out and finally dispose of this product or system; yet this is vital if you intend to make more than one such product. Configuration Management (CM) is the discipline that focuses on establishing and maintaining consistency of a product or system by ensuring all related items and their components are known, documented, controlled and tracked. CM concerns everybody involved in developing, producing, delivering and operating a given product or system all along the entire life cycle of that product or system. This is not only important for regulatory purposes, but for instance airlines need to know the configuration of each of the individual aircraft they operate, as do the maintainers that need to work on the aircraft. Also, the aircraft manufacturer needs to know the configuration as any changes to an aircraft have to be analysed and carefully documented, not to mention the need of being able to conduct systematic change impact analysis in case of proposed modifications.

Traditionally, in the aerospace context, CM has covered well the traceability between the design and the product baselines. However traceability between the requirements and the design – especially in the early phases of the development life cycle – has only recently started to be perceived as highly beneficial as it facilitates re-use and helps to prevent unnecessary rework, all of which seems to have tremendous potential to reduce development time and costs. CM can be argued to be a pre-requisite for increased levels of re-use of requirements together with existing design solutions that have already been verified against those previously validated requirements. There is a lot of documented but unpublished evidence that supports the above.
CM, however, due to its formal character, has been perceived to be very heavy and a burden to the creative aspects of early development work. Therefore, it seems crucial to find the right degree of formality in CM depending on the stage of a given programme or project. In the following, the paper will look at some relevant CM aspects in more detail with a clear emphasis on the early phases of the development life cycle, specifically the feasibility and concept phases.

Overview of Configuration Management
In the context of Systems Engineering

The **system lifecycle** in SE concerns a 'system or proposed system that addresses all phases of its existence to include system design and development, production and/or construction, distribution, operation, maintenance and support, retirement, phase-out and disposal' [13].
The application of good Configuration Management becomes mandatory if your product is to be released to market. The SE participants required to elicit the customer needs and requirements will be the same actors as the CM participants for this product. Those participants in CM activities potentially come from many different domains and disciplines (see Figure 1).

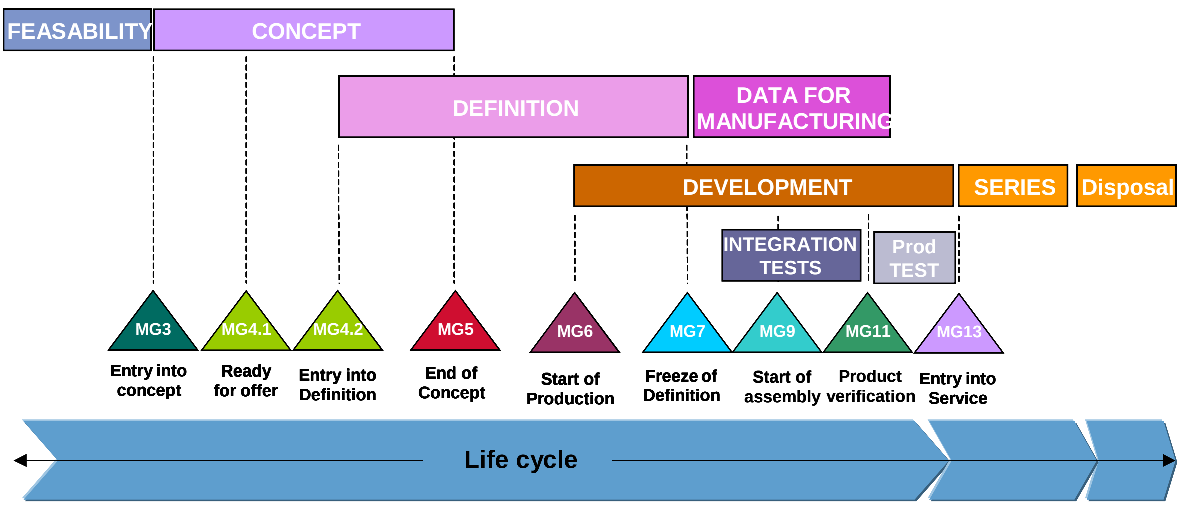
Figure 1: Contributors to the configuration completeness of the product

Product or system related data from those many different domains and disciplines will be essential to the SE process and within this process subject to CM activities.

For instance, customer services, engineering, manufacturing, quality, finance, airworthiness authorities, sales, and procurement are likely to be all involved in CM activities in the context of SE.
Equally, within the context of the engineering function alone, there are many different disciplines or domains that are essential for the overall SE approach and have to be involved in (and also will benefit from) CM processes. Some examples are RE, V&V (Validation and Verification) and Design. Evidence of requirements validation, design verification and product verification activities, as well as design trade-off analyses and decisions made based on those, all need to be properly recorded and traced in order to allow for controlled re-use of validated requirements together with corresponding design solutions and their V&V evidence.

Figure 2 provides an overview of an example product development life cycle from the aerospace context to clearly make the point that CM really has to be applied all over this life cycle to support and enable SE activities for highly complex products and systems. The example development life cycle is sub-divided into milestones that are called ‘maturity gates’ (MG0 - MG13) in order to emphasise the importance of achieving different levels of increasing maturity during the development.

Figure 2: Program Maturity Gate (MG) lifecycle



CM processes need to control product and its associated data all along the product lifecycle.

The following sub-chapter will focus on the early phases of the development life cycle, specifically the feasibility and concept phases, as these seem to offer the greatest potential to extend the use of CM for early development data with the greatest benefit in terms of re-usability and avoidance of rework.

Contributions of Configuration Management
During early phases of the development life cycle

The potential contributions of CM during early phases of the development life cycle can be expected to be significant. However, the right degree of formality needs to be found in order to address two main needs: first, traceability and history of early work need to be documented in some way, e.g. covering customer needs and high-level marketing strategies even before the first requirements are established, so that a company can benefit from that knowledge and experience in the future and validation of requirements later on in the process will be facilitated; second, the CM process during these early phases must not prevent the actors in the development process from being creative and flexible. Rather the process must be sufficiently 'light' to allow for frequent, sometimes revolutionary changes or updates of the development items (user needs, strategies, high-level objectives, top level requirements etc.) while at the same time allowing to record the right level of development information in order to establish traceability, which enables re-use of those items. For instance, if strategy decisions or high-level requirements are changed following a user focus group workshop, those changes and their justification have to be recorded. In the following, we will look in more detail at the feasibility and concept phases of the development life cycle.

Feasibility Phase (MG0-MG3)

The feasibility study can be defined as 'a study to identify and analyse a problem and its potential solutions in order to determine its viability, costs, and benefits' [14]. The feasibility phase (MG0-MG3) covers the SE feasibility studies. It is very important by the end of MG3 to have a relatively complete set of top-level requirements to keep the level of rework to a minimum. Eliciting and analysing the customer needs are essential activities in order to translate those into top-level requirements. The latter have to be validated with the customer(s) and risks should be identified, registered and analysed, as well as mitigation actions defined, before MG3 is passed. Unfortunately, in many development contexts there is a lot of pressure to meet aggressive program targets and very often maturity gates are passed with significant outstanding actions, which sometimes disguises essential, outstanding work that should have been completed to meet the maturity review baseline. In those contexts, it will take quite a cultural change effort for many companies to improve their way of dealing with these situations in an honest manner. CM can help to make this counterproductive use of maturity gates visible.

The four main disciplines of CM are (1) Configuration Identification, (2) Change Control, (3) Configuration Status Accounting, and (4) Configuration Audit.

(1) Configuration Identification shall establish the appropriate levels of product structure to facilitate baseline configurations. For complex products it will be necessary to identify elements such as Configuration Assemblies (CA), Configuration Items (CI) and Design solutions (DS). To maintain the evolution of the product structure it is necessary to establish a convention and method for naming and numbering these items. A release process to distribute this information also needs to be established.

(2) After the initial release of configuration documents, all changes should be controlled through the Change Control process. The impact of the change will decide the degree of formality in processing the change. The change process shall identify the document and justify the change by evaluating the consequences. Approval of

changes will be obtained at formal 'configuration control boards'. Deviations and waivers support the achievement of consistency between the as-built and as-designed configuration state.
(3) Configuration Status Accounting shall collect, record, process and maintain all data necessary to report the status of established configuration documents thorough the considered lifecycle. This includes departures from the configuration baselines.
(4) Configuration Audit shall detail the format of the audit, interested parties and procedures to follow. The audit shall check and validate CM process adherence and performance features, report findings and in cases of non-compliance the actions to be taken to correct the findings. A list of audits to be conducted and occurrences within a project shall be scheduled.

It is relatively straight forward to apply these 4 CM disciplines and obtain CM control for a given project/program during the period from maturity gates MG0 – MG3, provided requirements are managed in a central repository (requirements database) and are dynamically linked to the CM database. One single PLM tool would be the desired solution. Unfortunately, for most companies this is not the case.

Figure 3: The product lifecycle table (MG0–MG14)

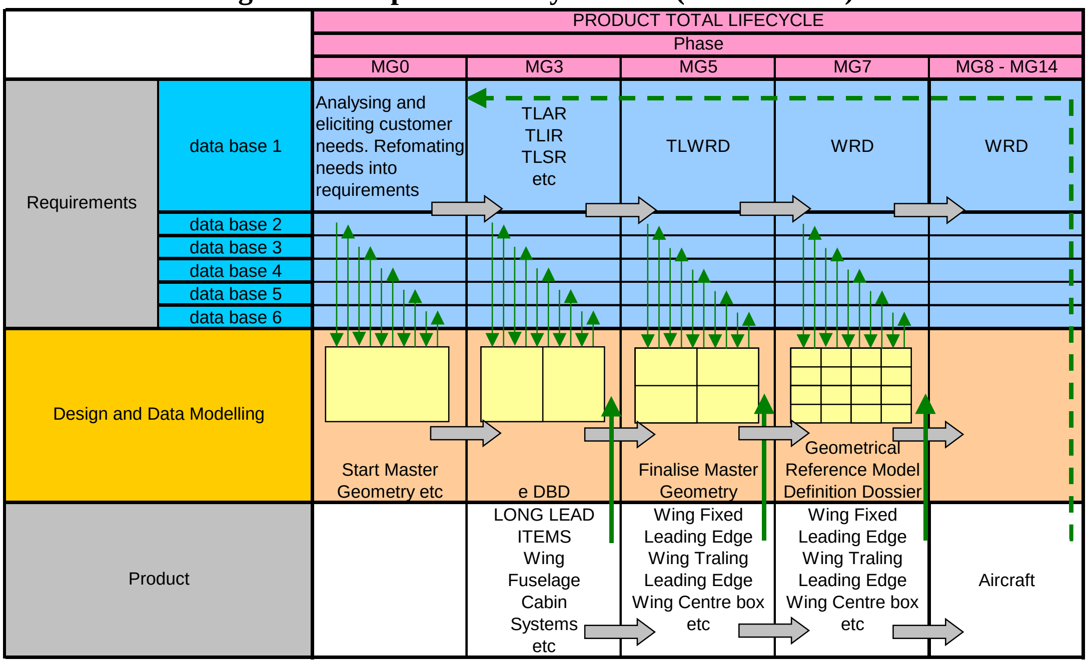
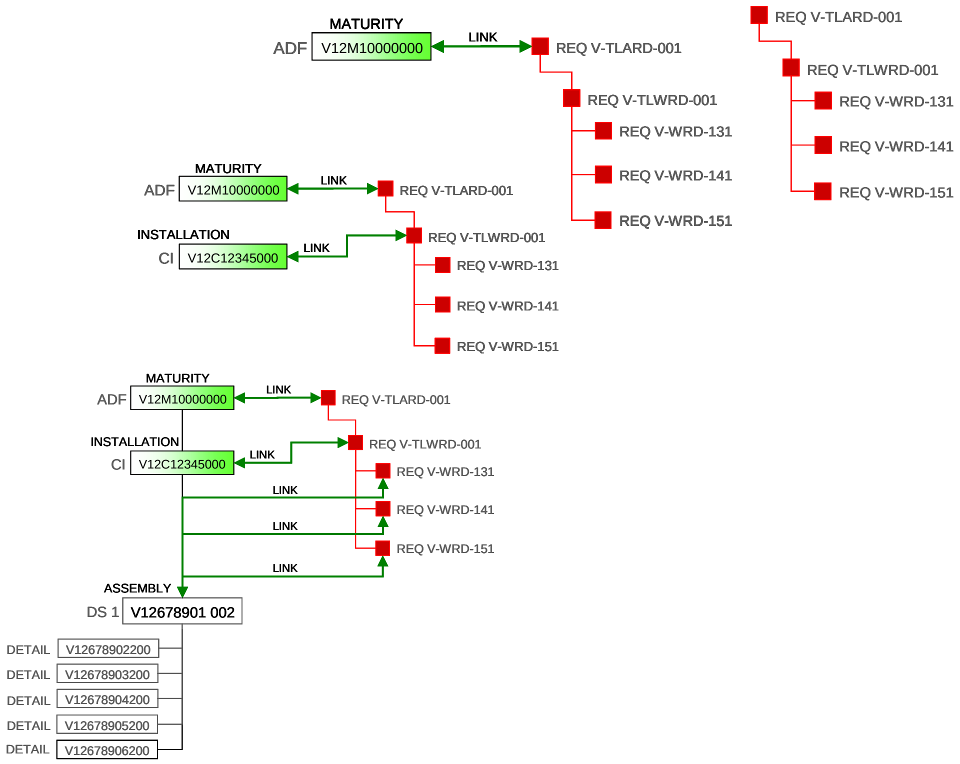


Figure 3 shows a typical situation for many companies involved in the development of complex products. This table indicates that the requirements have been created and stored in different databases (using different tools). The diagram does not make the distinction between different types of requirements or design or product. The table also shows the need to verify design and data models against requirements (vertical arrows), which gives a first impression of the complexity of the CM matrix.

There may also be a need to produce some physical items for proof of concept. This would add another vertical validation and verification level with corresponding links.
The introduction of a requirements product structure at this early stage of the project may seem premature, yet there seem to be significant benefits in terms of traceability. Figure 4 provides an example of how CM can use the product structure to manage compliance of requirements throughout the entire project/programme lifecycle.

Figure 4: Evolution of the Product Structure through the lifecycle (MG0–MG13)



Without requirements we have nothing to do. Without well configured requirements we will have a lot more to do than we think. In other words, as was shown by many studies in recent years [e.g. 8], getting the requirements right and keeping them in line with changing customer needs over time is crucial to any successful completion of a project/programme. However, every attempt to keep changes under control without having traceability in place between requirements, design and product baselines is a tremendous challenge, and most likely to result in a lot of unnecessary extra work, which invariably leads to unplanned budget and schedule impacts.

Concept Phase (MG3 –MG5)

The concept phase represents the first substantial opportunity to influence the design by balancing the requirements with technology, schedule, funding, performance etc. In the concept phase (MG3-MG5) strategic decisions will also be taken, such as assessing the maturity of the content of a proposed concept or concepts. By the end of this phase the business decision for the industrial launch and to engage in major investments and contract negotiations with both customers and suppliers.

Preparing for a successful and timely concept requires a number of preparatory reviews. Some typical examples are the review MG4.1 ‘Ready to offer’ (RTO) or ‘Authority to offer’ (ATO). At this point you should be in a position to publish your intentions to produce a product that demonstrates the main contributing parameters collected from your customers needs. This review will record the data maturity and assessed limit of uncertainty within selected margins e.g. + or – 5% on expected life, + or – 2% on weight, etc.

CM will treat these reviews as milestones with corresponding baselines. This does not mean the baseline is frozen and cannot change. CM baselines will record and store the status of all the information submitted to the review at that point in time. The CM baseline discipline should allow for retrieval of any baselines created at any time. When there is a need to evolve requirements or design concepts the CM change process will control these activities. This ensures that all documentation including any design maturities produced can be associated or linked and validated with appropriate design maturities and intern design solutions. CM will also control the distribution and release of this information.

As the concept continues to develop towards the definition and development phases, maturity continues to increase along with complexity and the number of people involved. This higher level of maturity also enables the development of procedures, processes, and methods specifically applicable for this product, and will be cascaded to lower layers of development (e.g. requirements). As the product matures so CM must mature. The design and data modelling information produced must align to the latest sets of requirements. This alignment can be achieved by good CM change and status accounting methods. MG5 indicates the end of concept and the final concept review or baseline.

Conclusion

The paper briefly looked at the SE context and emphasised the importance of requirements related issues for successful development projects or programmes. Yet RE frequently does not seem to be sufficiently interfaced or integrated with other engineering disciplines such as design, manufacturing, finance, procurement, customer services etc. CM is widely accepted as a key discipline to establish this traceability and enable controlled baseline evolutions or iterations within the development process, and as such CM supports and enables SE. However, the traditional scope of CM more often than not excludes requirements in the early phases of the development process. Therefore, the present paper put the focus on the early phases of the development life cycle, i.e. the feasibility and concept phases.

The interface between RE and other disciplines needs to be under CM although the degree of formality has to be adequate considering the current stage of the development life cycle. In other words, during early phases CM processes have to be much lighter than in later phases of the development life cycle, e.g. when requirements are considered to be ‘frozen’.

By applying good CM practices at the earliest opportunity any requirements related issues (corrective issues v product improvement) can be identified and managed. The CM change process can provide a quick indicator for this by inquiring and recording as part of the change process whether a required change is of corrective nature or represents an improvement. In many companies 40% - 60% of changes raised are requirements related corrective action [15]. Many of these above issues can be reduced by deploying a CM process that covers well the traceability between the design and the product baselines and also providing traceability between the requirements and the design especially in the early phases of the development life cycle. The creation of a dedicated INCOSE working group ‘Configuration Management’ should be considered in order to further promote and integrate this engineering discipline into the SE context.

References

[1] I. Alexander, R. Stevens, Writing better requirements, Pearson, Harlow, 2002.

[2] T. Bahill, S. Henderson, Requirements Development, Verification and Validation Exhibited in Famous Failures, Systems Engineering, Volume 8, No.1, Wiley, 2005.

[3] INCOSE, Systems Engineering Handbook – A Guide for System Life Cycle Processes and Activities, INCOSE SE Handbook Version 3, 2006.

[4] S. Jackson, Systems Engineering for Commercial Aircraft, Ashgate, Aldershot, 1997.

[5] E. Hull, K. Jackson, J. Dick, Requirements Engineering, Springer, London, 2002.

[6] S. and J. Robertson, Mastering the Requirements Process, Pearson, Harlow, 1999.

[7] G. Kotonya, I. Sommerville, Requirements Engineering: Processes and Techniques, Wiley, Chichester, 1998.

[8] http://www.standishgroup.com, 2006.

[9] I.F. Hooks, Why Johnny can't write requirements, AIAA conference, 1990.

[10] P.C. Chatzoglou, L. Macaulay, Requirements Capture and Analysis: A Survey of Current Practice, Communications of the ACM, Volume 38, Issue 5, 1995.

[11] G. Lucchetta, M. Baroni, L. Delaire, P.F. Bariani, Aircraft Requirements Elicitation and Development during Conceptual Design, University of Padova/Airbus, not published, 2008.

[12] E. Fricke, A. Schulz, Design for Changeability (DfC): Principles To Enable Changes in Systems Throughout Their Entire Lifecycle, Systems Engineering Journal, Wiley, 2005.

[13] Blanchard, Fabrycky, Systems Engineering and Analysis, Fourth Edition, Prentice Hall, p. 19, 2006.

[14] ISO/IEC JTC1/SC1: 2382-20, 1987.

[15] V.C. Guess, CMII for Business Process Infrastructure, Holly Publishing, 2002.

Biography

Stephen Watts is an aircraft configuration management specialist and qualified CMII Professional with a post-graduate Certificate in Management from the University of Lancaster (UK). He has almost 40 years of industrial experience (many years of which in the aerospace and defence sectors) in a range of engineering fields such as conceptual design (electronics, electro-mechanical), work-package management and configuration management, both in specialist and leadership roles.

Mario Kossmann is a Systems Engineer and Capability Integrator for Airbus, having previously worked for Blohm & Voss as a Systems Engineer, Technical Manager and also Consultant in Services Marketing. He has served as an officer with the German and French navies, and was awarded a MEng in Aerospace Engineering from the University of the Federal Armed Forces in Munich (Germany), and an MBA from the University of Warwick (UK). Mario is the author of the book 'Delivering Excellent Service Quality in Aviation' (Ashgate 2006). Currently, Mario is the Requirements Management process, methods and tools leader within Airbus. He is also the Airbus lead investigator on the OntoREM project.