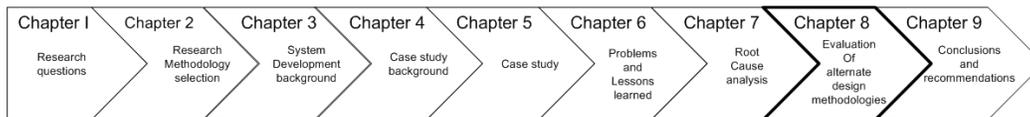


“One reason so many design mistakes are being made today is that design is being done empirically on a trial-and-error basis.”

Nam Pyo Suh 2001



*Investigation into structured design methodologies
Theory of inventive problem solving (TRIZ)
Axiomatic design
Structured design example*

Chapter 8 EVALUATION OF STRUCTURED DESIGN

In the previous chapter, it was shown that the **“effect-to-cause”** design iterations and design changes during the system integration process have a detrimental effect on development project performance in terms of cost and schedule. The reason for this is primarily due to the ripple effect of change as a result of the functional couplings throughout the system hierarchy, in a concurrent engineering development environment. For better development project performance, the Systems Engineering and Project Management processes must work harmoniously together. Alternatives must be investigated to achieve a better SE and PM process interaction by following a more structured design process.

In this research RCA of the case-study program led to an improved design influencing model as well as the development and quantification of a design change impact in terms cost and schedule. The case-study program is a historical fact. The case-study showed that design change risk was a major contributor to the project cost and schedule problems. Critical aspects of the case-study are revisited to illustrate and quantify the benefits of a structured design approach.

The Narrative Inquiry came to the surprising paradoxical finding that PM was the main contributor to the development project cost and schedule overruns. The research question whether a structured design approach can mitigate system development project risk is investigated.

The Narrative Inquiry is revisited and re-evaluated as if the development project was run along a structured design approach.

To achieve reduced design change risk, structured design methodology and in particular an alternative to the **“effect-to-cause”** design influencing is investigated. An important part of a subsystem of the case-study is redesigned using axiomatic design.

The research finding is then validated by the triangulation of the revisited Narrative Inquiry findings, the hypothetical case-study

findings in appendix C and the sub-system re-designed using axiomatic design methodologies.

8.1 Introduction to Structured Design

In the previous chapter the effectiveness of the IPS development model recommended by Roos (2001) on the case-study project was discussed. It was found that this model resulted in a very effective system from a technical performance point of view. However from a project management point of view, the project suffered a substantial cost and schedule overrun despite having very experienced development and project management teams with the full support from the company's top management.

The root cause for the cost and schedule overruns was identified to be the design iterations inherent in the design process. Also these iterations do not stop at CI level but continue right through during integration of all the subsequent system hierarchy levels, affecting all the other functionally coupled CIs of the system. It was shown that in the concurrent engineering environment of the IPS model, the functional couplings throughout the system hierarchy, further exacerbated the project cost and schedule impact due to forced design changes of all affected items.

The problem of cost and schedule overruns on design projects appears to be universal. NASA in their latest systems engineering handbook, (2007), accepts that cost and schedule slips may be unavoidable. All 32 NASA programs in the survey, figure 26, exhibit overruns. Figure 26 also shows the importance of defining the project before starting the detail work. The trend line shows that about 15% project definition effort appears to be optimum.

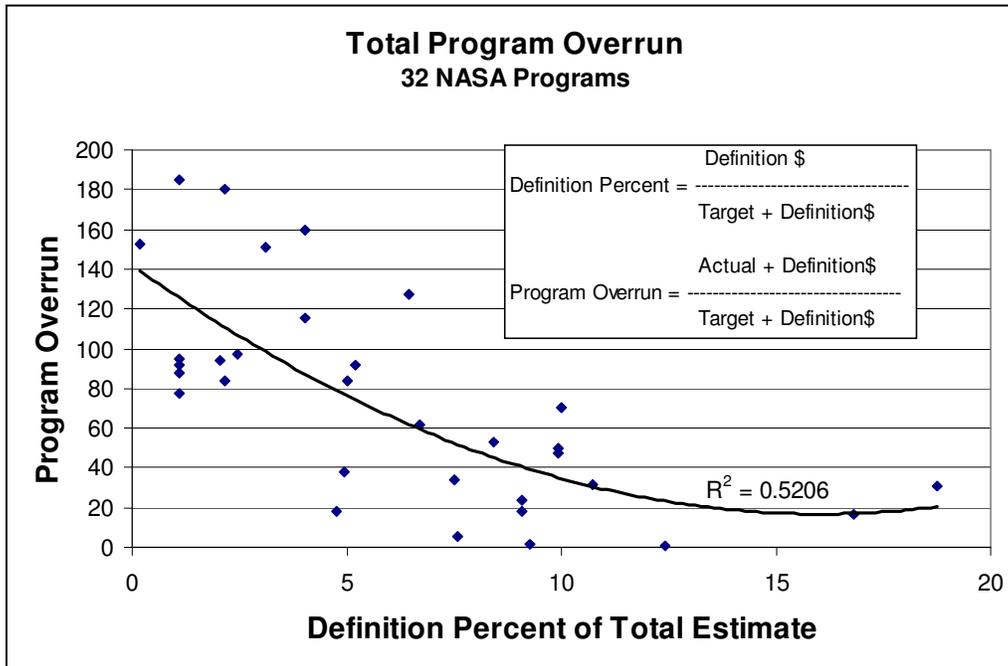


Figure 26: Value of Systems Engineering; Summary Report 1/04

Source: Werner Gruhl, NASA Comptroller's Office, (1992)

General project cost and schedule overruns were also the findings of de Beer (2009) and Steyn (2009). It appears that the more the edge of technology is pushed, the higher the risk of cost and schedule overruns. In today's competitive market, no system-house can afford to develop a new system from older proven technology alone. Some of the systems building blocks should be leading edge technology to provide the competitive edge. This can increase the risk of cost and schedule overruns on the development project. This was also the situation for the case-study project of this research.

Most systems engineering literature are unanimous for the need for design iterations to achieve design maturity and view it as entrenched in the systems engineering process. In the previous chapter it was shown that the **"effect-to-cause"** design influencing by a FD-SD design influencing approach, although very effective, still resulted in design iterations and increased system integration risks partly as a result of project management constraints.

It was discussed in the previous chapter that project management cannot accommodate indeterminate iteration processes. The iterative GERT technique developed by Pritsker, (1966) as work-around to allow iterations in the PM process, was abandoned by the PMI due to incompatibility with other established PM processes. Also discussed previously, was that Systems Engineering cannot function in isolation. Project management is required to manage the consumption of resources and schedules of the systems engineering project. Therefore it is essential that the two processes must work

harmoniously together. Successful systems engineering projects will only be possible if design iterations can be eliminated to facilitate seamless interfacing with project management processes. It falls beyond the scope of this research to resolve the compatibility problems between systems engineering and project management. In this chapter a literature survey will be done of structured design processes with the objective of reducing design iterations.

Systems Engineering is to a certain extent an unpredictable process in that a need for change can occur at any stage in the process. This change can be minor and not impact on the project plan. If however the change is major, serious repercussion on the project plan and cost budget can occur.

The factors that affect the risk of a change are:

- System complexity
- Number of system hierarchy levels
- Technology maturity
- Unexpected environmental factors
- Human factors
- Logistic considerations
- Obsolescence and procurement factors
- Statutory factors.

8.2 Investigation into Structured Design methodologies

A literature search shows that a number of researchers have been investigating other design methodologies with the objective of reducing design iterations.

8.2.1 Theory of Inventive Problem Solving (TRIZ)

The "*Theory of Inventive Problem Solving*", originated by Genrikn Altshuller in (1946,) is known by its Russian acronym TRIZ, (Hu et al, 2002). The goal of TRIZ analysis is to achieve a better solution than a mere trade-off between two elements. According to Hu et al, (2002), Altshuller discovered that when an engineering system was reduced to reveal the essential system contradictions, inventive solutions eliminated the contradictions completely. From this finding, Altshuller identified 76 standard solutions, (Hu et al, 2002).

He also developed the Substance-field and Analysis and Modelling method. The Substance-field model is a model of minimal functioning and controllable technical system. There are four steps to follow in making the Substance-field model, (Hu et al, 2002):

- Identify the elements.
- Construct the model.
- Consider solutions from the 76 standard solutions
- Develop a concept to support the solution.

The process for inventive problem solving (TRIZ) is illustrated in figure 27.

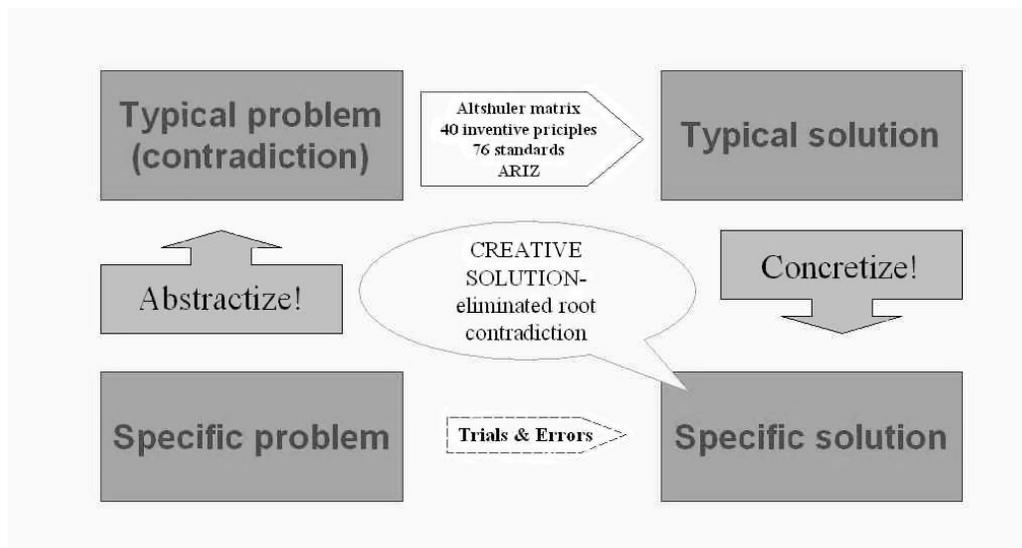


Figure 27: TRIZ process for creative problem solving

Source: Wikipedia, (2010).

TRIZ is an integral module in Acclaro DFSS^{®14}

8.2.2 Axiomatic Design

In this paragraph the different domains of the Axiomatic Design (AD) process will be discussed. In particular the tight linking of the Functional Requirements to the Design Parameters will be discussed. The two fundamental AD axioms will be discussed and the concepts of coupled design, de-coupled design and uncoupled design will be discussed.

¹⁴ Acclaro DFSS[®] is a registered trademark of Axiomatic Design Solutions, Inc. Copyright[®] 1998-2006 Axiomatic Design Solutions, Inc.

Sahlin (2000), states that successful product development is not only about developing products, hardware, software and services, that best satisfies the market wants and needs, but also about doing the job faster and more effectively than the competition. He states that Axiomatic Design provides the principles that can help to take design decisions based upon actual facts related to many parameters (Sahlin 2000).

Research by de Beer, (2009), found that the lack of proper tools to model the information flows, activity iterations and interfacing within a design, led to the development of the Design Structure Matrix (DSM) by Steward in the 1960s. Yassine et al (2003), discuss the DSM method for complex concurrent engineering.

Axiomatic design (AD) appears to have evolved from DSM and in 1990 Suh proposed a framework for axiomatic design, which utilizes four different domains that reflect mapping between the identified needs and the methodologies used to achieve them (Suh, 1990):

- Customer requirements - customer needs or desired attributes
- Functional domain - functional requirements and constraints
- Physical domain - physical design parameters
- Processes domain - processes and resources

Gumus (2005), describes and illustrates the four axiomatic design domains shown in figure 28.

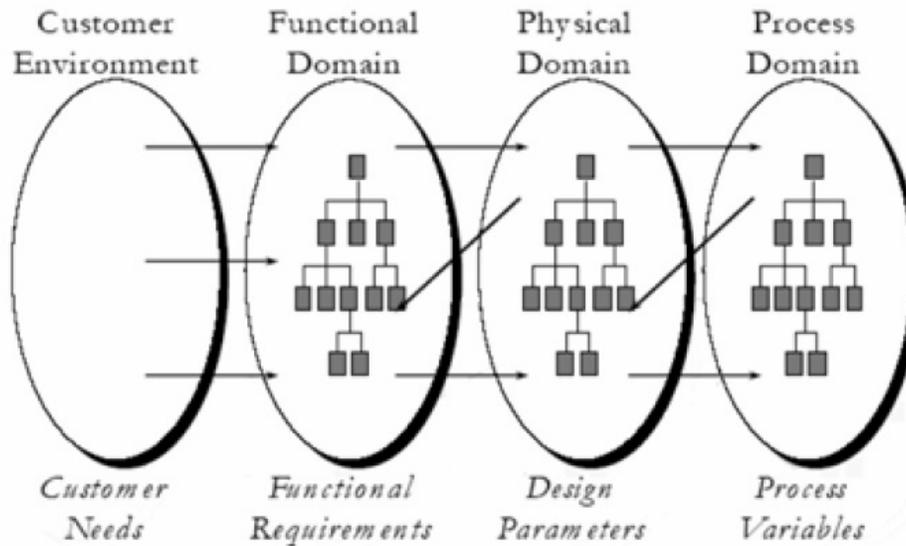


Figure 28: Axiomatic design domains

Source: Gumus, (2005)

Melvin et al, (2002), studied the rearrangements of the system hierarchy as a tool to eliminate design iterations. He found that when a system is designed that results in some unintended interactions between design elements, it is possible to achieve a non-iterative design process by rearranging certain elements in the system hierarchy.

The research by Gumus et al, (2008), focussed on different design methodologies and system/product development lifecycle models. They introduced an Axiomatic Product Development Lifecycle (APDL) model based on the AD method developed by Suh, (1991), with the aim to cover the whole product development lifecycle including the test domain. The objective according to Bullent (2008), of the APDL model is to improve the quality of the design, requirements management, change management, project management, and communication between stakeholders as well as to shorten the development time and reduce the cost.

Gumus et al, (2008), proposes the Trans-disciplinary Product Development Lifecycle (TPDL) model for new product development. In this model, AD method is extended to cover the whole product development lifecycle, including the test domain. New domain characteristic vectors are introduced to systematically capture and manage the input constraints and system components. The objective of the TDPL model according to Gumus et al, (2008), is to improve the quality of the design, the management of requirements, design change and project management. TPDL improves communication between stakeholders as well as to

shorten the development time and reduce the development cost, (Gumus et al, 2008).

According to Hu et al, (2002), to achieve a robust design, it is important to select the appropriate system output response. Hu et al, (2002) asserts that currently, this selection process is more like art than a science. The consequence being that, depending on the experience of the design engineer, the appropriate system output response, can lead to trial-and-error design iterations. Hu et al, (2002) claims that by applying TRIZ and Axiomatic Design principles robust design can be enhanced. They substantiate their statement with a case-study in a large automotive company, (Hu et al, 2002).

The above literature confirms that the general aim of the AD is to reduce design iterations and thereby improve development project management. Also since design iterations was found to be a fundamental cause for the case-study project cost and schedule overrun, AD will be investigated and discussed in more detail in the next paragraphs.

Suh, (1990), formulated two axioms for axiomatic design:

- **Axiom 1 - Independence Axiom**

(Maintain the independence of the functional requirements)

In an acceptable design, the Design Parameters (DPs) and the Functional Requirements (FR) are related in such a way that a specific DP can be adjusted to satisfy its corresponding FR without affecting other FRs. This is a very intuitive and logical axiom. Should a FR be split amongst more than one DP for example, the system becomes amongst others less maintainable. Diagnostic testing tests a function and cannot uniquely point to a failed component particularly if there is more than one component providing that specific function. Restoration of the failed function can only be performed by replacing a faulty component.

- **Axiom 2 (Information Axiom)**

(Minimize the information content of the design)

The best design has the minimum information content (functional couplings) which means the maximum probability of success. This agrees with this author's root cause analysis and the derived equation to quantify the ripple effect of a change in a multi-hierarchical system, discussed in chapter 7.

AD reduces iterations and improves skilled resource efficiency since it is in essence a "**cause-to-effect**" design influencing

process as a result of the FR and DP linking. The analysis of multi-hierarchical systems in chapter 7 showed that functional couplings and iterations can never be completely eliminated in real systems. Structured design methodologies however can reduce the risk of unplanned iterations.

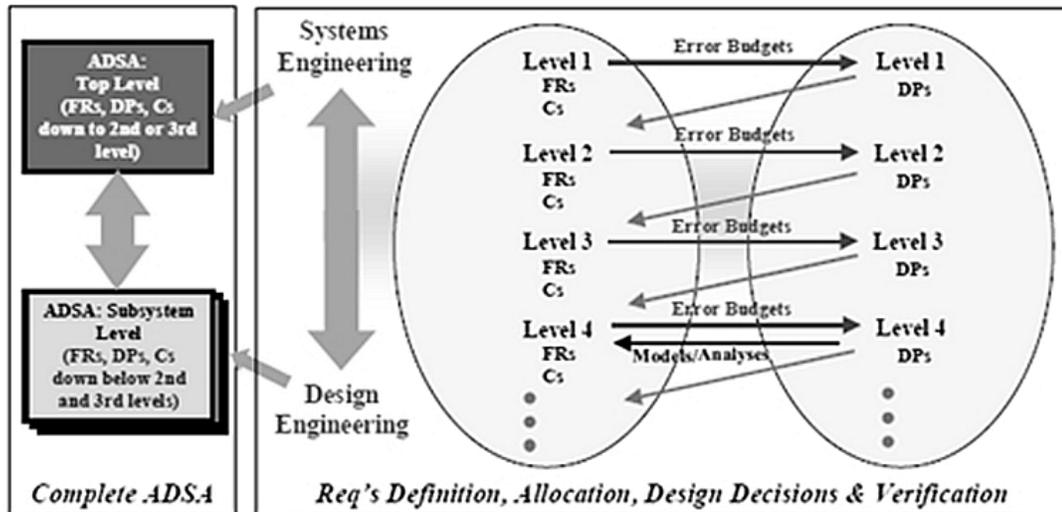


Figure 29: Distributed organisation of the AD system architecture

Source: Hintersteiner et al, (2000)

According to Hintersteiner, (2000), the real goal of the overall design effort is to optimise the performance of the system which is not necessarily the same as optimising each component. In figure 29 he illustrates the top-down zigzagging process down the system hierarchy. It was shown in Chapter 3 that a system is more than a collection of components. A system development objective is amongst others to optimise the emergent properties which are not necessarily the same as a collection of optimised components. AD is not intended to replace SE but should be integrated into the SE process, (Hintersteiner, 2000).

From the above discussions, it can be deduced that AD, due to the direct linking of FRs to DPs, is a “*cause-to-effect*” process and in essence obviates the iterations of the “*effect-to-cause*” processes. Melvin et al, (2002), showed a technique by rearranging the FR-DP matrix, iterations in the design process may be reduced. From a PM point of view such a process is much more acceptable under tight budget and schedule constraints. With increasing demand for shorter development times and higher quality, design effectiveness has received growing attention from both academia and industry, shown in figure 30. In industry, unsatisfactory design results in a great number of process iterations. Improving the design effectiveness is important in order to shorten product development times and lower costs.

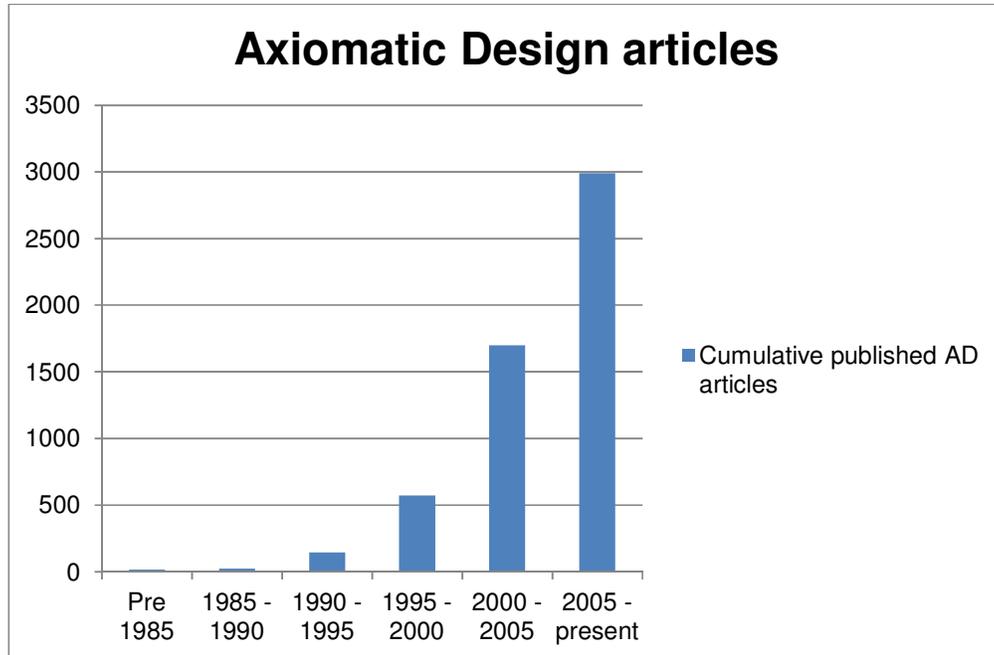


Figure 30 Axiomatic Design articles published

Acclaro DFSS[®] is a structured design tool using axiomatic design (AD) as the underlying methodology with the ultimate aim to reduce design iterations. A designer, by modelling in Acclaro[®] DFSS, can optimise his design using tools such as:

- TRIZ - Theory of Inventive Problem Solving
- VOC - Voice of the Customer - User Requirements
- PUGH - Analysis of System Dynamics (decision matrix)
- DSS - Design For Six Sigma
- QFD - Quality Function Deployment

AD reduces iterations thereby improving skilled resource efficiency discussed in Chapter 7.

Applying Axiomatic Design to the design of complex systems, an optimal system architecture that captures hierarchical structure and interrelationships between the functional requirements and design parameters and constraints of the system can be evolved.

8.3 Case-Study - Problems Experienced and Lessons Learnt

The PRACA data evaluated by the Narrative Inquiry revealed that unexpected design changes were the main contributor to the system development project's cost and schedule problems.

To verify the benefits claimed in the literature for AD, (Hintersteiner, 2000), (Melvin, 2002), the case-study development project was revisited as if the project was run along axiomatic design methodology and the principles of structured design was applied.

8.3.1 Structured Design Example: a subsystem of the case-study

The Sight Guidance Optical Unit (SGOU) is a subsystem of the Anti-Tank missile system of the case-study project. From the PRACAS it was found that a design change to this particular CI had a change impact on a number of other subsystems concurrently under development. The need for a design change came about due to a relatively minor change in requirements and was identified by the FD team. The design correction required an additional design iteration and as a result of the functional couplings, the change affected a number CIs concurrently under development.

The first release of the subsystem was modelled in Acclaro DFSS[®] with objective of evaluating the design from a structured design perspective. Figure 31 shows a Tree Diagram of part of the subsystem. From the Tree diagram, it can be seen that there are functional couplings between the commander and gunner monitors. There are also functional couplings between the selection of day night capability and fields of view.

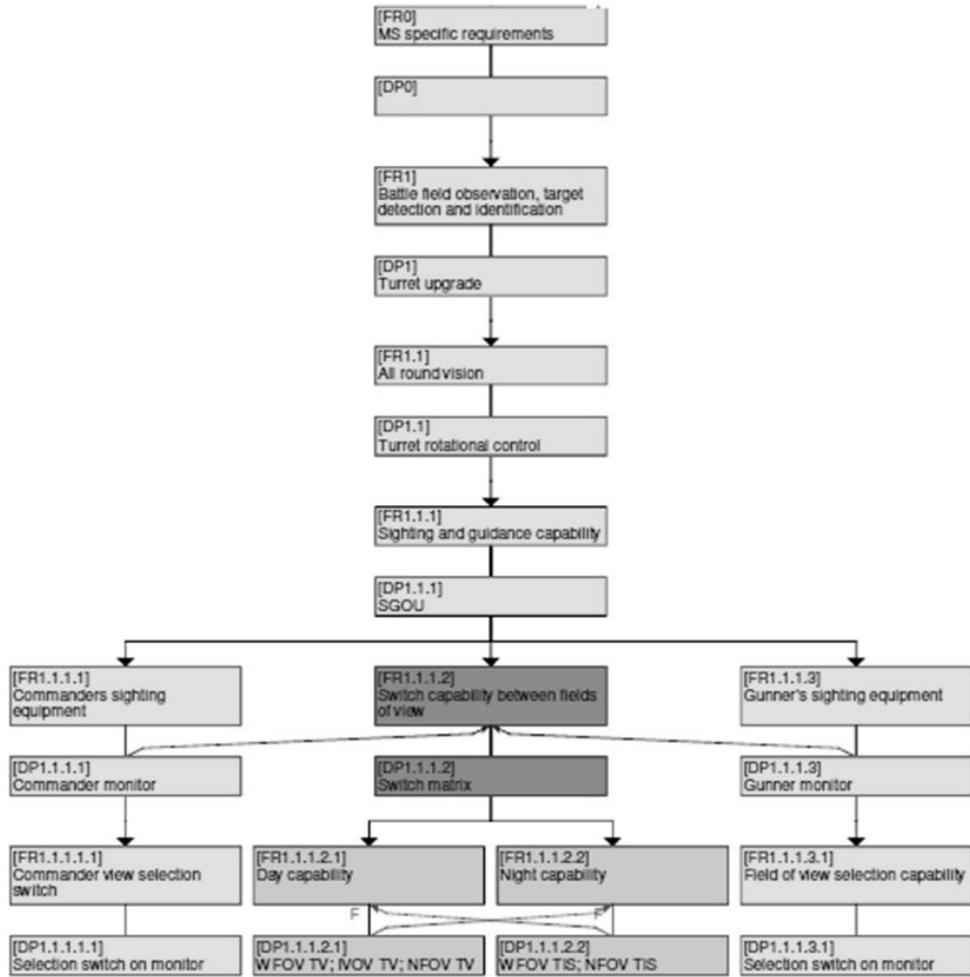


Figure 31: Part of the SGOU Tree Diagram

The Design Matrix diagram in figure 32, highlights the functional couplings even more prominently.

The Acclaro DFSS® model confirms that structured design approach can reduce design iterations resulting in system development cost and schedule savings.

This example illustrates the advantage of the “*cause-to-effect*” design influencing instead of the customary *Effect-Cause* design influencing.

8.3.2 Revisit of the Narrative Inquiry Analysis findings (Grouping and Quantification of the Problems Experienced)

To verify the benefits claimed in the literature for AD, (Hintersteiner, 2000), (Melvin, 2002), the case-study development project’s PRACAS data was reappraised as if the project was run along axiomatic design principles. Although it was not possible to reconvene the original Narrative Inquiry’s focus group, the PRACA data was reviewed by 3 experts who all have extensive experience in this area. The objective of the PRACA data review using AD principles is to determine the benefits that can be obtained by adopting an axiomatic design methodology for the case-study project.

The outcomes of this re-evaluation and moderation have been supplied in appendix D and is summarised in figure 33. For comparative purposes, figure 21, Chapter 6 was extended to include the revised figures.

Problem area	Consolidated incidents	Total impact	Revised Impact	average %	Revised Average
Management related project problems	32	100	57	56.8%	43.8%
System Engineering related project problems	12	35	32	19.9%	24.6%
Quality Assurance (QA) related project problems	7	15	15	8.5%	11.5%
Configuration management (CM) related project problems	10	26	26	14.8%	20.0%
Total	61	176	130		

Figure 33: Revised problems experienced using AD methodology

Figure 33 shows that the impact of the problems experienced have been reduced from 176 to 130. Also significant is that management related problems have been reduced and systems engineering related problems have increased. The QA and CM related problems are not significantly affected.

The validity of this finding is confirmed through triangulation, (Greene, 2007), and the predictions by Hintersteiner, (2000) and Melvin, (2002) when using an AD system development methodology.

There is now a more even spread of problems between management and systems engineering indicating that there are fewer conflicting interactions between the two processes. This is beneficial for development project performance in terms of cost and schedule.

The before and after findings of the Narrative Inquiry findings show a clear benefit by using axiomatic design.

8.4 Summary and Conclusions

This chapter investigated alternatives to design influencing with the objective of reducing development project cost and schedule risk.

It has been shown that the general **“effect-to-cause”** design influencing gives rise to design iterations. The influence of PM constraints increases the risk for later design changes particularly during system integration. A design change of one system CI may result in a ripple effect of forced design changes in other CI’s in the system hierarchy due to the functional couplings.

Alternatives to the **“effect-to-cause”** design influencing has been investigated. It has been illustrated by means of a case-study CI example that **“cause-to-effect”** may have the potential to reduce the development project risk by reducing design iterations. To this effect, structured design methodologies have been further investigated.

The Theory of Inventive Problem Solving (TRIZ) has been available (1946) before systems engineering became formalised, yet it has not been taken up into the systems engineering process. A literature survey, figure 33, shows after an initial acceptance it appears to have stagnated. Axiomatic design on the other hand has slowly increased to a 1% penetration into systems engineering. Where TRIZ uses and adapts an already known solution to a problem, AD starts with a zero baseline in finding the optimum design solution, (Yang et al, 2000). From the experience gained on the case-study and the findings of this research, further research in this field is recommended.

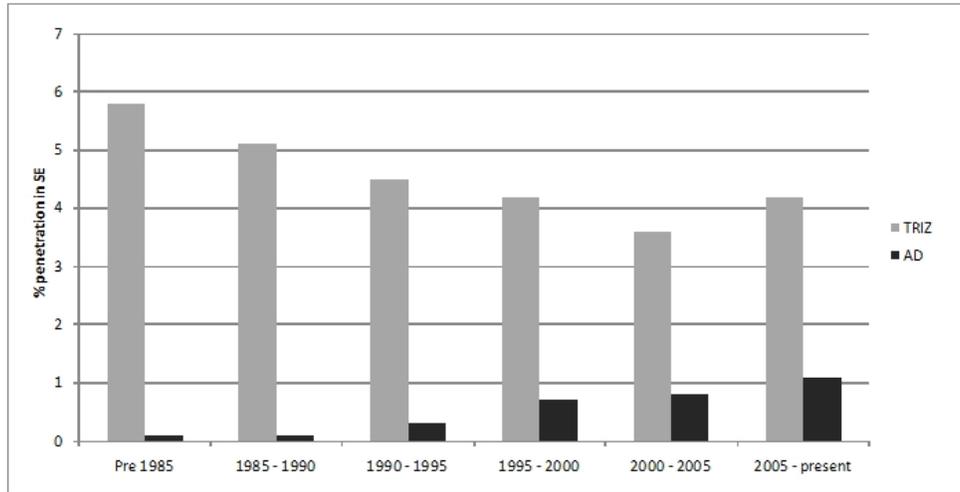


Figure 34: Penetration of TRIZ and AD into Systems Engineering

AD has been investigated and it was found that AD is a “**Success Domain**” “**cause-to-effect**” design influencing process.

No reference to axiomatic design could be found in the two major systems engineering process sources, INCOSE (2010) and NASA (2007). A SE-AD combined published article survey on Google Scholar, May (2010), was performed. Comparing the survey results shown in figure 34 to those shown in figures 1 and 30, it can be concluded that considerable further research is required in this field. This future research should aim amongst others to quantify and establish the boundary conditions of development project performance improvement using AD integrated within the SE process.

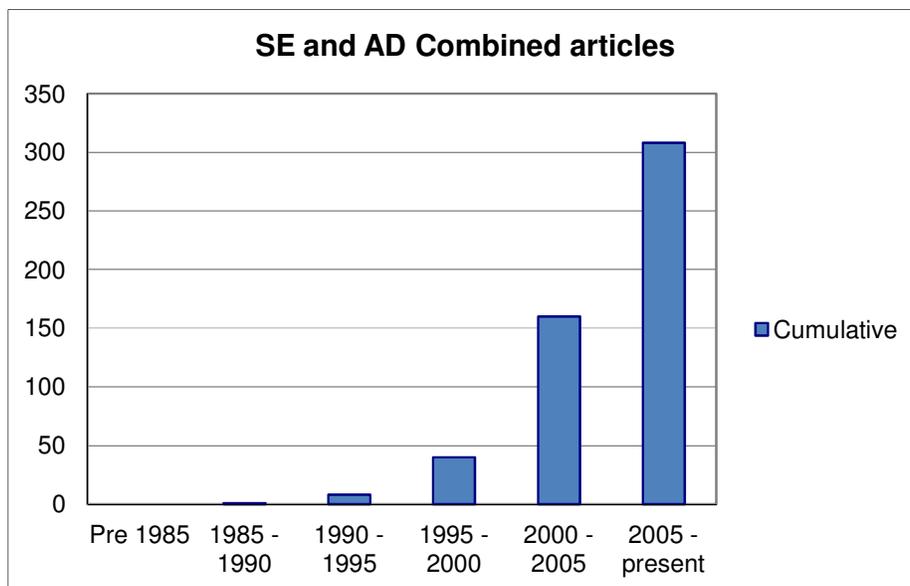


Figure 35: AD articles in context of SE published

This low acceptance of AD methodology in SE was also confirmed by Hintersteiner, (2000), who investigated the integration of AD into the SE process. Hintersteiner defines a 5 level AD maturity model to provide a clear roadmap implementing AD within the SE process that will have both engineering and management support. This author agrees with Hintersteiner since the systems engineering knowledge base is maturing fast. It would be much more practical and less risky if structured design methodologies are to be integrated into established proven SE processes and become part of the SE discipline. The major advantage of such integration would be a “**Cause-to-Effect**” design influencing, reducing the risk of indeterminate iterations and unplanned design changes. The benefit will be reduced risk of development projects being overspent/over schedule and better compatibility with the project management discipline without which SE cannot function in the real world. The SE process must develop towards a structured design process with the objectives of reducing iterations and minimising functional couplings in a complex multi-disciplinary system. INCOSE already has a work group addressing the development of a SEBOK for SE to standardise and structure the SE processes in a similar vein as PMBOK for PM (INCOSE, 2010). This research falls outside the scope of this dissertation and must be further researched.

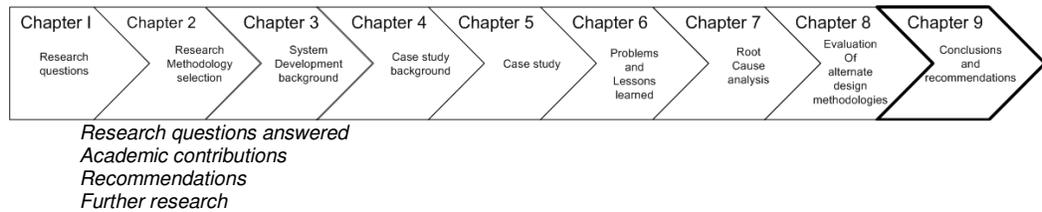
Structured design process specifically axiomatic design has been discussed. Indications are that structured design can have a beneficial effect on system development projects. This was illustrated with one example of a design item of the case-study project.

The research objectives have been confirmed by the triangulation of the findings of the Narrative Inquiry focus group, the Acclaro DFSS[®] redesign of the SGOU and the hypothetical case-study best-case and worst-case analyses. Refer to Appendix C, paragraph D.

In the next chapter, the findings and conclusions of this research will be summarised.

“A conclusion is like the final chord in a song. It makes the listener feel that the piece is complete and well done.”

CRLS Research Guide (2011)



Chapter 9 CONCLUSIONS

This research investigated the development of complex systems with particular focus on design influencing and the impact thereof on the rest of the system under development in a concurrent engineering environment. The development model used was the IPS concurrent engineering development model recommended by Roos (2001).

A development project for the development of a third generation anti-tank system complete with its associated logistic system was used as a case-study. The development project was a technical success but the project suffered cost and schedule overruns.

The PRACA data collected during the case-study development project was first analysed by means of a Narrative Inquiry research methodology and subsequently by means of a DSR methodology.

The Narrative Inquiry found that the IPS development model was an effective model for the development of complex, multi-disciplinary systems with multi-layers of subsystems and components. However, the Narrative Inquiry came to the surprising paradoxical finding that PM was the main contributor to the development project cost and schedule overruns.

The Narrative Inquiry findings were actually symptoms of deeper underlying problems, which were subsequently further, researched by means of DSR in an Action Research setting using root cause analysis to identify the root causes, and to provide a better understanding of the fundamental underlying mechanisms in the design process. The root cause analysis led to the modelling of the design process at coal-face level.

The design teams were divided into a SD team focussing on the design requirements, and a FD team focussing on the design constraints. The opposing mindsets of the two teams resulted in very effective design solutions. When the interaction of PM was introduced into the design process model, it was found that the PM interaction with the design process increased the risk of the integration of the

design item at the next system hierarchical level. This increase in risk was primarily due to PM constraints which were different from the SE constraints.

A design change during integration of the design item at the next system level was found to have a negative impact on project cost and schedule performance primarily due to:

- Functional couplings between CIs resulting in forced design changes to functional coupled CIs under development in the IPS concurrent engineering development process.
- Unavailability of resources under the Matrix organizational structure.

The principal finding of this research showed, that unplanned, unexpected and forced design changes were the primary area of conflict between SE and PM, leading to development project cost and schedule overruns.

A mathematical model was developed to quantify the impact of a design change of one CI on the rest of the system under development. This impact was as a result of the functional couplings between CIs in the system hierarchy. Inspection of the mathematical model showed that in order to reduce development project risk in a concurrent engineering development environment, the system hierarchy should be optimised to achieve the minimal functional couplings between system elements.

Design iterations are fundamental to SE primarily as a result of the “**effect-to-cause**” design influencing process. During system integration there is a risk of an unexpected design change of a CI which can result in a ripple effect of design changes to other CIs in the system hierarchy. This is primarily as a result of functional coupling between system elements.

SE must work harmoniously with PM to bring a system into being since PM must manage the schedule and expenditure of resources. PM on the other hand is a very structured systematic process and does not allow revisiting completed milestones. This can create conflict between SE and PM if for instance an unexpected design change requirement is identified during system integration.

An investigation was done into structured design methodology in particular AD with the objective to determine if AD can reduce system development project risk. AD ensures independence of the functional requirements and design parameters as well as the optimisation of the system hierarchy for minimum system information content. AD is in essence a “**cause-to-effect**” (*a priori*) process.

The mathematical model developed in this research not only confirms the minimum system information content requirement of AD, but also provides a means to quantify the impact of a design change on a specific system hierarchy. The developed impact equation in this research, allows quantification of design change impact for a specific system hierarchy. This can be used for trade-off studies between different possible system hierarchies for a specific system function.

It appears that the risk of design changes which can impact negatively on development project performance can be reduced if the SE process migrates towards a “*cause-to-effect*” (*a priori*) structured design methodology.

9.1 Research Questions Answered

From the outcome of this research, the research questions posed In Chapter 1 can now be answered:

- ***How does design influencing give rise to iterations and what are the effects of these on a development program?***

RCA of the problems experienced on the case-study development project of a complex multi-disciplinary system in a concurrent engineering environment, show that the fundamental mechanism of the “*effect-to-cause*” design influencing process gives rise to iterations. Also the “*effect-to-cause*” design influencing result in a ***stop-start*** process between the SD and FD design teams that affects resource efficiency discussed in Chapter 7. This effect can be mitigated by judicious project planning and multi-tasking of scarce resources.

- ***How effective is the IPS development model for the development of a complex weapons system in practice?***
 - Analysis of the case-study problems experienced as well as subsequent system field tests confirms that at the technical level, the IPS concurrent engineering system development model is an efficient and effective model for the development of integrated complex multi-component systems. The case-study RCA showed that the majority of the project problems experienced, were at the management level and were not due to the IPS development model.
 - The development methodology using Success Domain and Failure Domain (SD-FD) design teams proved to be very effective. Generally quality designs were submitted to the CDR for acceptance, reducing system integration risk.

- ***What is the influence of project management on design influencing using the IPS concurrent engineering development model?***
 - Analysis of the fundamental mechanism that result in design iterations, reveal that project management due to time and cost constraints, may force a premature release of a design for integration into the next system hierarchy level thereby increasing the system integration risk.
 - A design change to one functionally coupled CI at a particular level in the system hierarchy, can result in forced changes to all the other CIs which are functionally coupled to it at that level of the system hierarchy. In fact this could ripple further down the system hierarchy of each of the coupled CIs and their sub-CIs.
 - Unexpected design changes can occur at any stage in the SE processes particularly during integration. This can negatively impact on the project schedule and cost.
 - Under the Matrix organizational structure, skilled resources are allocated to other projects after acceptance of the design, and will generally not be immediately available to address any design change of the affected CIs. This can impact negatively on the project.
 - Detail RCA of the case-study problems show that there is an incompatibility between the PM and SE processes. Iterations that are the cornerstone of the SE processes clash with the PM processes that can only accommodate pre-planned and structured iterations.
 - The incompatibility between the PM and SE processes is further aggravated in a concurrent engineering development, environment due to the ripple effect of a change to the other functionally coupled CIs under development.
 - The different performance measurement criteria used to evaluate the SD_FD and Project Management teams did not have a positive influence on the quality of the systems design process.
- ***What is the root cause of each of the problems experienced and how can these be alleviated?***

Interaction by PM in the fundamental design process can increase the risk of a latent design defect and can only be detected later

during system integration. This latent design defect may result in unplanned design changes which may have a detrimental effect on development project performance.

Functional couplings in a system must be minimised. This requirement is supported by the literature and case-study RCA discussed in Chapter 8. One possible way to achieve this objective is for SE to migrate to towards a “**cause-to-effect**” structured design methodology. This will ensure independence of the functional requirements and design parameters and at the same time minimise the system information content or functional couplings.

9.2 Academic Contributions

A systems engineer has a once in a career opportunity to participate in a complex system development project from its beginning to its finalisation. It is also rare that all the members of the development team are highly experienced and have a strong trust amongst each other, which would have eliminated corporate politics (noise) so prevalent on development projects that run into trouble.

The case-study of the Anti-Tank Weapons System development project presented an ideal opportunity to perform a detailed RCA of the design influencing process in a concurrent engineering environment. This resulted in an in depth fundamental knowledge and understanding of the mechanisms at play on a system development project at grass root level in the design process. The knowledge such gained facilitated the search for alternative and better solutions.

The specific contributions of this research are:

- Structuring of design teams into two opposite groups with opposing objectives to effectively optimize a design - **Success Domain team** and **Failure Domain team**.
- Modeling the fundamental cause and mechanism of design iterations and the inclusion of the effect of management into the model.
- Development of a mathematical model for the effect of design changes in a multi-level multi-dimensional system hierarchy. This model provides a means to quantify the impact of a design change on a specific system hierarchy.

- The developed impact equation enables quantification of design change impact for a specific system hierarchy that can be used for trade-off studies between different possible system hierarchies for a specific system function.
- Identifying the lack of “**How**” data in the formal systems engineering process. Substantiating the requirement of system and design “**How**” data during the development of a formal system data pack.
- Identifying further research fields in systems engineering and structured design, for more optimal development of new systems with the objective of reducing development project cost and schedule risks.

9.3 Recommendations

The research highlighted that the Systems Engineering process must function harmoniously within the larger Project Management environment for optimum development project performance. The road forward to achieve this goal is for the systems engineering and design processes to become more structured and the removal of the unpredictability in the processes. This will enable the systems engineering processes to be more easily accommodated within the structured project management processes to the benefit of the overall development project performance.

9.4 Further Research

This research identified the SE/PM deficiencies areas that require further research:

1. Project management and the systems engineering processes have areas of incompatibility

Further research is required to resolve the apparent conflict between SE and PM discussed in Chapter 6. SE must function harmoniously within the project management environment for development project success.

More research is required to find answer amongst others to the question whether the indeterminate design iterations and unpredictable design changes inherent in the systems engineering process, (INCOSE, 2010) and (NASA 2007), result in compatibility

problems in the project management processes to the detriment of the overall development project performance?

Further research into design maturity is required. Indeterminate design iterations and subsequent forced design changes during system integration are primarily due to the uncertainty of when design maturity has been achieved. Further research in this field will provide a firmer basis of when a design is ready to be released for further integration into the system.

Further research is required into structured design methodologies, with the primary objective of reducing design iterations, and subsequent risks of forced design changes during system integration.

It is envisaged that the above research will resolve some of the SE and PE apparent compatibility problems to ensure better development project cost and schedule performance.

2. Systems Engineering primarily develop “WHAT” data and insufficient “HOW” data.

Any man-made system will at some stage or another fail and may require support. No system can be successfully deployed without an appropriate logistics support infrastructure, (Blanchard, 2004). Operating personnel must know what the system does and how it functions to be able to optimally utilise and deploy the system. Support personnel must know how the system functions in order to be able to diagnose and restore a failed function (Wooley, 2003). For through-life engineering support, support engineers must know and understand the design and its critical functions and parameters in order to develop modifications and upgrades during the life of the system. This requires that the original system data pack developed under the system development project must contain not only the “**WHAT**” data but also the “**HOW**” data. Further research is required to find effective systems engineering methods and processes to achieve this objective.

3. Development of a design change impact tool

The design impact equation must be developed into a tool to enable the quantification of design change impact for a specific system hierarchy to assist DRBs. Such a tool can also be used for trade-off studies when evaluating different system architectural solutions for a specific system functionality.

4. Improved quantification of early systems engineering effort

There are several systems engineering studies, (NASA 1992, Honour 2004) showing that systems development program risk can be reduced considerably, by Early Systems Engineering (ESEE) effort. Despite agreement with these findings, by the systems engineering fraternity, early systems engineering effort is often not sufficient, in practice, to ensure optimal system design. The reason may be that this effort cannot be adequately quantified and coupled to a payment milestone. The impact equation developed during this research will enable the quantification of hierarchy optimisation to allow it to be taken up in a payment milestone. This must be researched further to facilitate the accommodation of adequate ESEE.

9.5 Further Systems Engineering Development

From the case-study and also discussed in Chapter 8 it was found that Systems Engineering is to a certain extent an unpredictable process and a design change requirement may surface at any stage of the process. It has been shown, and a mathematical model was developed that such a design change in a concurrent systems engineering development environment, is a function of the functional couplings between the different system elements. The objective of structured design methodologies is to minimise the functional couplings and thereby reducing development risk. The current Systems Engineering process, (INCOSE 2010), makes it very difficult to optimise a system design without impacting negatively on the development project cost and schedule. This research showed possible advantages during system development that can be achieved using structured design methodologies. The literature surveys performed as part of this research revealed voids of knowledge in the systems engineering and structured design methodologies. Further research is required to quantify the advantages of such integration and find optimal ways of achieving this. Also to reduce the unpredictability in the Systems Engineering process, further research into the integration of structured design methods into systems engineering would be advantageous.