UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

THE DEVELOPMENT OF COMPLEX SYSTEMS: AN INTEGRATED APPROACH TO DESIGN INFLUENCING

ARIE WESSELS

A thesis submitted in partial fulfilment of the requirements for the degree

PHILOSOPHIAE DOCTOR

In the

FACULTY OF ENGINEERING, BUILT ENVIRONMENT
AND INFORMATION TECHNOLOGY

UNIVERSITY OF PRETORIA

SUPERVISOR: PROFESSOR L. PRETORIUS

2012

ABSTRACT

The aim of this research is to identify and analyze the impact of design changes to a system in a concurrent engineering environment and the development project, and to make proposals how to minimize the impact on the development project performance. A further objective is also to determine the effect of design changes as a result of design influencing. In a concurrent engineering environment system components are being developed in parallel. Any change to one component of the system may impact on other system components under development.

Design as part of the systems engineering process is an iterative and dynamic process. Although the systems engineering process has been very well structured and refined over the years, it still remains to a certain extent an unpredictable process. A consequence of this is that changes to a design of a subsystem or component comprising the system can occur at any stage of the process.

The systems engineering process is a "***static***" process since there are no time constraints or management of consumption of resources on the different systems engineering processes and steps. As such system engineering cannot function in isolation. To bring a system into being, systems engineering must function within a project management environment to provide the management of schedule and the consumption of resources. The interaction between project management and system engineering processes can have a distinct influence on the systems engineering process and must be taken into account when studying the performance of system development projects. This research investigates the project management/systems engineering interface with specific focus on cost and schedule.

Since project management is the encompassing process wherein a system is being developed, its influence on the system engineering process will also be investigated. This research has the following research objectives:

- Optimization of design influencing by dividing the design teams into two different complementary but opposing mindset groups.

- Evaluate the impact of design changes in terms of cost and schedule overruns in a concurrent engineering development environment.

A comprehensive development project was used as a case-study. A Narrative Inquiry comprising the main system development project players investigated the problems experienced on the project and found that management was the major cause for the project cost and schedule overruns. The principal finding of this research showed, that unplanned, unexpected and forced design changes was the primary area of conflict between systems engineering and project management, leading to development project cost and schedule overruns. The Narrative Inquiry findings were actually the symptoms of a deeper underlying problem. Root Cause analysis identified the fundamental mechanisms of design change and the influence of management on the process.

This research identifies the fundamental mechanisms that result in design iterations and the influence that management has on this process. An improved "***Effect-to-Cause***" design influencing model is proposed to reduce the risk of design changes during system integration. A mathematical model has been developed to quantify the impact of a design change on a multi-layer, multi-component system. This model confirms that the system hierarchy design is very important to minimize the impact and consequential development project risk should a design change be required for one of the system components. By means of the mathematical model, a proposed system's architecture can be modelled. The model quantifies the impact of a system component design change on the rest of the system development project. This model will facilitate the optimization of system architecture to reduce development project cost and schedule risks. The system architecture model will also enable design review boards to make informed decisions when considering options for a system component design change.

This research also found that the Systems Engineering process must function harmoniously within the larger Project Management environment for the optimum performance of a development project. 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 so far as the number of design iterations is concerned. 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. A structured "***Cause-to-Effect***" design influencing methodology has been investigated. Indications are that this may be the road forward for systems engineering process development to even further reduce the risk of a design change during system integration and consequential detrimental impact on the development project performance.

Table of Contents

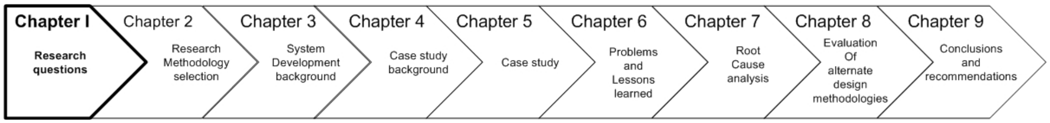
Table of figures

Abbreviations

AD	Axiomatic Design
ADM	Advanced Development Model
ADM	Arrow Diagramming Method
ARRL	American Radio Relay League
ATGM	Anti-tank guided missile
BIT	Built-in Test
BITE	Built-in Test Equipment
BOM	Bill of Materials
CDR	Critical Design Review
CI	Configuration Item
CFE	Customer Furnished Equipment
CM	Configuration Management
DoD	Department of Defence (USA)
DRB	Design Review Board
DSM	Design Structure Matrix
DSR	Design Science Research
ECP	Engineering Change Proposal
EDM	Engineering Development Model
ERA	Explosive Reactive Armour
ESEE	Early Systems Engineering Effort
ET&E	Engineering Test and Evaluation
FBS	Functional Breakdown Structure
FD	Failure Domain
FFF	Form, Fit and Function
FMECA	Failure Mode, Effects and Criticality Analysis
FRB	Failure Review Board
FTA	Fault Tree Analysis
GERT	Graphical Evaluation and Review Technique
Hdbk	Handbook
HEAT	High Explosive Anti-Tank
IEEE	International Electronics and Electrical Engineering
ILSP	Integrated Logistics Support Plan
INCOSE	International Council on Systems Engineering
IPC	Illustrated Parts Catalogue
IPS	Integrated Product Support
IPT	Integrated Project team
ISP	Integrated Support Plan
ITAR	International Traffic in Arms Regulations
LAN	Local Area Network
LCC	Life Cycle Cost
LORA	Level of Repair Analysis
LSA	Logistic Support Analysis
LSAP	Logistics Support Analysis Plan
LSAR	Logistic Support Analysis Record
MCLOS	Manual Command to Line of Sight
Mil	Military
MIS	Management Information System
MIT	Massachusetts Institute of Technology

MRV	Maintenance Recovery Vehicle
NASA	National Aeronautics and Space Administration (USA
NAVSO	Navy Standard Order (USA)
OT&E	Operational Test and Evaluation
PBS	Product Breakdown Structure
PC	Personal Computer
PCMB	Project Configuration Management Board
PDM	Precedence Diagramming Method
PDR	Preliminary Design Review
PM	Project Management
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
PPM	Pre-Production Development Model
PRACAS	Problem Reporting and Corrective Action system
PSP	Product Support Plan
QA	Quality Assurance
RBD	Reliability Block Diagram
RBDO	Reliability Based Design Optimization
RCA	Root Cause Analysis
RCM	Reliability Centered Maintenance
SACLOS	Semi-Automatic Command to Line of Sight
SANDF	South African National Defence Force (SANDF
SD	Success Domain
SDD	Software Design Document
SE	Systems Engineering
SEMP	Systems Engineering Management Plan
SRD	Software Requirements Document
Std	Standard
TAAF	Test Analyse and Fix
TEMP	Test Engineering Management Plan
TRAMP	Testability, Reliability, Affordability, Maintainability and Produceability
URS	User Requirements Statement
XDM	Experimental Development Model

"Design is directed toward human beings. To design is to solve human problems by identifying them and executing the best solution."
Ivan Chermayeff



System concepts and definitions
Research objectives
Research contributions
Problems statement
Research questions
Research roadmap

In this chapter the researcher discusses the rationale of this research. A synopsis of system concepts and definitions are provided followed by a discussion on research objectives, research contributions and research questions. A brief introduction on the preferred development model and structured design is provided. These will be discussed in detail later in this dissertation.

Chapter 1 INTRODUCTION

Modern systems comprise many subsystems and components. INCOSE, (2010), coins the term "system-of-systems". System-of-systems are systems whose subsystems are themselves self contained systems. These subsystems in turn comprise lower level subsystems down to components. The self contained systems or subsystems can be multi-disciplinary and may be of varied technologies (INCOSE, 2010).

A system is more than a collection subsystems and components. The interactions or functional couplings between the different subsystems and their components provide the emergent properties of the system, (Sparrius, 2008), (Johnson, 2006). Modern systems therefore can entail a multi-dimensional hierarchy consisting of many levels with a myriad of functional couplings between the system components. A system therefore does not live in isolation but is always part of a larger system. These characteristics will be taken into account when studying the behaviour of a system.

One of the research objectives is to determine and assess the factors that often result in the poor performance of development projects of integrated complex[1] multi-component systems. The particular focus will be to provide better insight into the design process. The constraints placed on the development process by other processes such as project management will be investigated. The mechanism of the design influencing process, in particular the influence of project management, on this process will also be looked into. This will

---

[1] The meaning of complex in the context of this research is defined in par 1.2. below.

facilitate the identification of factors that may increase the risk of a design change later in the development process.

The main aim of this research is to identify and analyze the impact of a design change on the rest of the system hierarchy and to make proposals on how to minimize the impact on the development project's performance.

1.1 Development Projects Problem areas

The literature contains many references to the fact that system development projects, particularly in the defence industry, very often suffer from cost and schedule overruns, (Christensen 1998), (Smirnoff 2006). To date, not all the causes for these overruns have been identified. It is commonly presumed that a "***rubber***" baseline is the main cause for project cost and schedule overruns. However, studies by Christensen et al, (1998), on the performance of development projects for military systems, found that other factors must also be influencing development project cost and schedule overruns.

Design as part of systems engineering is an iterative and dynamic process. The history of the iterative and incremental development process is discussed by Larman et al, (2003). Although the systems engineering process has been very well structured and refined over the years, it still remains to a certain extent an unpredictable process. A consequence of this is that changes to a subsystem or component of the system can occur at any stage of the process, often during the system integration stages. During the system integration phases, very often a latent design defect of a system component surfaces. This may force a corrective design change to the affected component to overcome the problem.

In order to reduce development time, integrated complex multi-component systems are developed in a concurrent engineering environment. This entails components and subsystems being developed in parallel, and subsequently integrated into higher level subsystems until the final system integration. The development of a multi-component, multi-disciplinary system generally entails the development of individual system components by different development teams. The development teams may be in-house or outside companies depending on the skills and facilities required.

The accepted process for the development of new systems is the documented Systems Engineering process by INCOSE, (2010) and NASA (2007). Both state that a design is successively refined until it is mature and acceptable for further integration into the system.

The impact of an unexpected design change is exacerbated in a concurrent engineering environment where system components are being developed concurrently during the development of multi-component systems.

Browning et al, (2000), developed models that modelled some of the important characteristics of the development process. They found that design iteration is a fundamental but an often under estimated characteristic of the product development process. The design change impact is mediated by the activity structure or architecture of the process. Models were developed that allow simulation of the process architecture to minimise design change impact thereby reducing development project risk. To demonstrate their model, they used a case-study for the development of an uninhabited aerial vehicle (UAV). Their research, however, was aimed at the managerial system development process and it does not address the influence of the detail design process on development project risk.

In this research a case-study for the development of an anti-tank weapons system project is investigated and then subjected to Root Cause Analysis (RCA), with the objective of determining and analysing the actual factors resulting in project cost and schedule overruns.

The main research objective is to determine and assess the factors that often result in the poor performance of development projects of integrated complex multi-component systems. The particular focus will be to provide better insight into the design process. The constraints placed on the development process by other processes such as project management will be investigated. In particular, the influence of project management on the design influencing process will also be investigated. This will facilitate the identification of factors that may increase the risk of a design change later in the development process.

A generic mathematical model will be developed to quantify the impact of design change on the rest of the system hierarchy. The model will enable optimisation of the system hierarchy to reduce design change impact on other components of the system. The combined effect of these cascaded design changes may have a profound effect on the development project performance. The findings of this research will enable the proposal of improved systems engineering, and design processes to mitigate development project cost and schedule overruns.

The benefits of structured design methodologies such as Axiomatic Design will also be investigated, with the objective of reducing design iterations and risk of later design changes, (Melvin et Al, 2002). Reduced design iterations and reduced unexpected design change risk will also reduce development project risk.

1.2 Concepts and Definitions

The following concepts and definitions are used in this research:

- **Complex systems**

  According to Joslyn et al, (2000) a complex system is a system composed of interconnected parts that as a whole exhibit one or more properties, (behaviour among the possible properties), not obvious from the properties of the individual parts.

  Complex systems in the context of this research will have the meaning of a multi-disciplinary, multi-component system in a multi-hierarchical system level structure.

- **Design**

  Design of a product is an activity to satisfy a consumer need, (Alexander, 2009). Design of a system composed of interconnected parts imposes interface requirements on the subsystems of that system.

  Garner, (1991) supported by Smith et al, (1997), claim that it is fundamentally incorrect to state that design is a problem solving activity, because, very rarely can a definitive answer to a design problem be provided. They also state that design problems do not lend themselves to being “***solved***”. The result is that there is no universal definition for design but in general, design can be perceived as a process of compromise involving conflicting factors. The best a designer or design team can hope for is to resolve the conflicts by trading the conflicting factors or constraints off, against a value system. All these factors are, to a greater or lesser extent, in a state of flux and are resolved via a process of trade-off and optimization cycles of development and evaluation. As a result, design is not a one-time activity but a development process to grow the product to maturity.

  From the above arguments, it can be concluded that it is probably more correct to state that design is the art of compromise, since there is no unique design solution to satisfy a specific consumer need. Different designers may very well come up with different solutions, (Garner, 1991), (Smith et al, 1997).

- **System**

  Ludwig von Bertalanffy, (1968), describes a system as a set of mutually dependent variables, and states that a set of variables comprising a system is to hypothesize that each variable in the set is a function of every variable in the set, and uses the following definition: "*A system is a set of variables that maintain functional relations through time, where the present state of a given variable is dependent on its own past state as well as the other variables*". In other words a system can be described by a set of differential equations with time as the independent variable.

Another definition of a system is provided by Andrews et al, (1997) during a workshop on systems thinking. They state that "*A system is an arrangement (pattern, design) of parts which interact with each other within the system's boundaries (form, structure, organization) to function as a whole. The nature (purpose, operation) of the whole is always different from, and more than, the sum of its unassembled collection of parts*". From Andrew's definition it can be concluded that a system is not just a collection of building blocks, there must be some form of functional coupling between the different elements to comprise a system.

- **Systems Engineering**

Systems engineering (SE) is defined by INCOSE, (2010) as follows: "*Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs*".

From this definition it can be deduced, that the systems engineering process, includes not only the system but also the logistical system to operate and support the developed system.

- **Development model**

The preferred product development process in the author's defence industry, is the Integrated Product Support (IPS) model, (Roos, 2001). The IPS development model is a concurrent engineering development model.

According to Roos (2001, P196), the IPS model is divided into 6 phases shown in figure 16:

- Management aspects of the development phase.
- Concept, exploration and definition phase.
- Demonstration and validation phase.

- Full scale engineering development phase.
- Production phase
- Commissioning and support phase.

The IPS model is discussed in more detail in paragraph 5.1 below. The IPS model ensures a structured concurrent systems engineering approach, at the macro system development project level. The detail design described by Roos (2001) is limited to high level design objectives. His discussions of the IPS model do not address the detail design processes, (Roos, 2001).

- **Project Management**

Project management is the discipline of planning, organizing, and managing resources to bring about the successful completion of specific project goals and objectives, (PMBOK, 2008). PMBOK defines a project as follows: - "*A project is a* ***temporary*** *endeavour undertaken to create a unique product or service.* ***Temporary*** *means that every project has a definite* ***beginning*** *and a definite* ***end****. Unique means that the product or service is different in some distinguishing way from all similar products or services* ", (PMBOK 1996).

Project management is a structured "*milestone-by-milestone"* process, PMBOK, (2008).

- **Development team member behaviour**

The behaviour of people in a project environment is aptly described by Goldratt, (2006), as follows: "*Tell me how you measure me, and I will tell you how I will behave. If you measure me in an illogical way … do not complain about illogical behaviour*".

Thus, according to Goldratt, (2006), workers in the work place, behave in accordance with how they are measured. This is logical since an employee's salary increases and career movements are dependent on their performance against the measurement metric.

Generally, the primary performance metric for systems and design engineers is technical performance and compliance to user requirements, of the systems and designs they have developed. As a consequence project performance is considered of secondary importance.

Project managers on the other hand are primarily measured on project performance, particularly in terms of cost and schedule, whilst system technical performance is of secondary importance to them (PMBOK, 2008).

The systems engineers and project managers are however, dependent on one another, for overall project success and as a team they need to take ownership for the overall success of the project, (INCOSE, 2010).

Team member interaction plays a pivotal role in the success of the system development project. According to Roach (2010) lack of vision, failure to take personal responsibility, personality conflict, power struggles, lack of clear identity of team member roles and lack of coaching, are the main reasons why teams fail. These pitfalls must be avoided when selecting and managing a design team.

The above concept discussions provides the reader with a clearer view of the limitations of design, the concept of a system, the systems engineering process to bring a system into being and the team members involved on the development project of a system.

1.3 Systems Engineering and Project Management Articles

A Google Scholar survey shows that since the early 1950s, 272,000 articles have been published discussing systems engineering topics. Over the same period, 886,700 articles have been published discussing project management topics, (refer to figure 1).

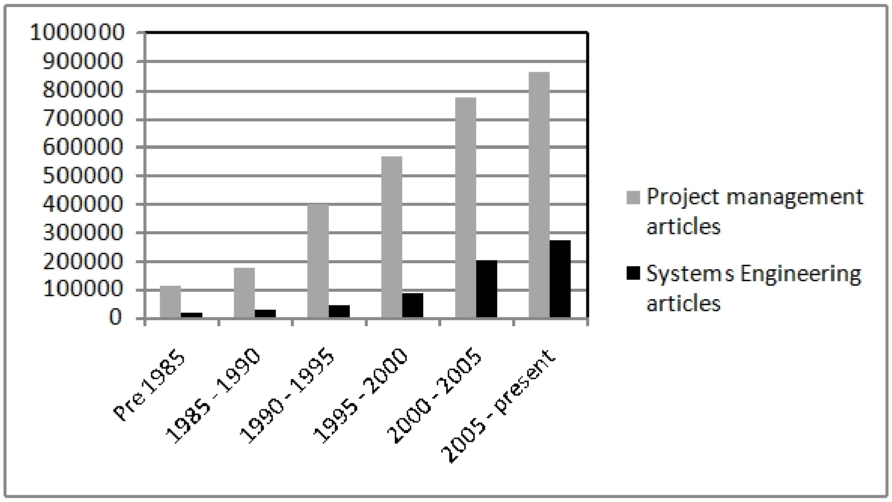


Figure 1: Published Articles

A further literature search indicates that there is a gap in the knowledge of the effects of design changes, during the development of complex systems, in a concurrent engineering environment.

In addition the consequences of design changes on the performance of a development project appear not to have been extensively researched.

This research will delve into the interactions between systems engineering and project management, in a development project environment. A multi-disciplinary development project for a 3rd generation anti-tank weapons system is used as a case-study. The objective being the identification of the root causes of project problems.

The analysis of the root causes of project problems will provide a better understanding of the fundamental mechanisms of the phenomena observed. This knowledge will allow the development and proposal of remedial and mitigating actions.

1.4 Problem Statement

This research is focussed on the concurrent development and integration of system products and the impact of a design change in a concurrent engineering environment. This research will also specifically focus on the influence and impact of project management on the design process.

According to Institute for Defence research Report R-338 (1988), "*Concurrent engineering is a systematic approach to the integrated, concurrent design of products and related processes, including manufacture and support. This approach is intended to cause developers, from the outset, to consider all elements of the product life cycle from conception through disposal, including quality, cost, schedule, and user requirements*".

System development in a concurrent engineering environment introduces new challenges for the system engineer. Sometimes, during integration, a problem is encountered with one component or subsystem, which requires a modification to overcome or correct the problem. If the correction affects an item's Form-Fit-and-Function (FFF), it can also force design changes to other components or related processes of the system, which are in various stages of their own development process.

A single design change of one component may result in design changes of a number of other components in the system. The problem is exacerbated for more complex systems with multi-layers of subsystems and components. The impact of these changes may ultimately have a detrimental effect on the system development project's cost and schedule performance.

Studies by Christensen et al, (1998), into the performance of 400 Defence acquisition projects, found that these often exhibit schedule and cost overrun problems. They also found that this phenomenon is not affected by baseline stability and contract type. They propose that other possible causal factors should be more closely examined, specifically management practices relative to change management.

Sanjay et al, (2000), studied 418 plants mainly in the motor industry and found that management has a profound effect on the quality of design.

Steyn (2009) proposes the compilation of a list of project cost and schedule overruns, and the identification of their causes. The interrelationships between them can then be identified.

Thunnissen, (2004), in his dissertation for mitigating uncertainty in the design of complex multidisciplinary systems, developed project sampling and modelling techniques in order to classify project uncertainty. He showed that the quantitative methods had benefits compared to the current heuristic-based methodology.

This research will focus on the interaction between systems engineering and project management and its influence on project quality. Problems have been identified, inter alia, by Christensen et al, (1998), and Sanjay et al, (2000), in the fundamental understanding of the mechanisms of design iterations and the influence of project management on design maturity.

1.5 Research Objectives

The objective of this research is to determine and assess the impact of design changes on the system, as well as on the development project, in a concurrent engineering environment, and to propose ways to mitigate the problems encountered.

The influence of project management as the encompassing process for system development will also be evaluated.

In this research the following research objectives will be addressed:

- Optimisation of design influencing by dividing the design teams into two complementary but opposing mind-set groups.

- Evaluate the impact of design changes in terms of cost and schedule overruns in a concurrent engineering development environment.

1.6 Research Contributions

The academic contribution of this research is the identification and detail analysis of the mechanisms and effects of design change, in a concurrent engineering environment, and their impact on the overall development project.

The primary body of knowledge for systems engineering, INCOSE (2010), and NASA (2007), focus on the systems engineering process. They, however, do not discuss the impact of design changes on the rest of the development project. The impact and influence of the project management process on system development is also not discussed.

The interaction between the systems engineering and the project management processes, as far as cost and schedule impact is concerned, have been investigated and a design influencing approach will be proposed to facilitate better overall project performance.

1.7 Research Questions

The following research questions are posed:

- Can design influencing models be established to depict the success/failure domain interactions in a dynamic project management environment?
- What is the impact of a design change, in terms of the functionally coupled items, on the development project?
- Can structured design methodologies reduce the number of design iterations, thereby reducing the project’s cost and schedule? The Design Structure Matrix (DSM) approach ensures a direct link between the functional breakdown structure (FBS) and product breakdown structure (PBS), (Yassine et al, 2003).

1.8 Research Roadmap

The focus of this research is to determine and evaluate the effects of design changes on the performance of a complex, multi-disciplinary, system development project. Data from a case-study will be analysed to study the impact that design changes have on such development project. The case-study used for the research, is a fully fledged multi-disciplinary and multi-component system, developed in a concurrent engineering environment.

Design iterations are fundamental to the systems engineering process, primarily due to design influencing to drive the design to maturity, (INCOSE 2010), (NASA 2007). For this research it is important to first determine the fundamental mechanisms that give rise to design changes, before being able to determine the effects of these changes on a concurrent engineering system development project. Once these effects are fully understood, will it be possible to:

- Answer the research questions posed.
- Investigate alternative design methodologies to reduce development project risk, in terms of cost and schedule.

The roadmap followed by this research is illustrated in figure 2.

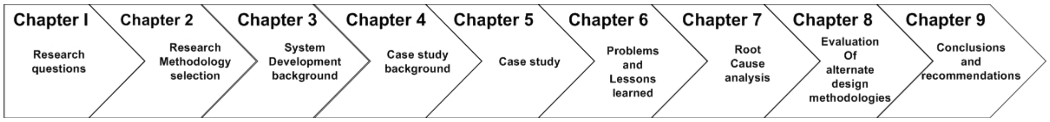


Figure 2: Research roadmap

Particular focus is given to the design influencing process and its impact on the development project as a whole.

System concepts and definitions will be discussed in chapter 1, this will provide the relevant background to systems engineering and project management. This chapter will present the research objectives, the research questions and the research contributions.

Research methodologies and their selection will be discussed in chapter 2. An overview of Root Cause Analysis will also be presented.

Systems, system engineering, project management and facility management, with specific focus on the research's objectives will be discussed in chapter 3. The design team structure, as well as the influence of project management will also be discussed.

The background of armour and anti-tank weapons and the evolution of anti-tank missile systems will be discussed in chapter 4. This will provide a better understanding of the case-study. This chapter will provide an overview of the contract, the top-level user requirements and the project management model.

The case-study for the development of a third generation anti-tank weapons system will be discussed in Chapter 5.

Chapter 6 will evaluate the case-study and present the problems experienced. The project Problem Reporting and Corrective Action System (PRACAS) data will be quantified and analysed.

The identification of the problems' root causes and their mitigating solutions will be discussed in chapter 7. The design influencing model and how the constraints, imposed by project management, can increase the system integration risk, will be discussed. A mathematical model will be developed to quantify the development project risk, and facilitate the optimisation of the system hierarchy.

Alternative design methodologies will be discussed in chapter 8, with the objective of improving project management process compatibility.

The research findings and the answers to the research questions will be discussed in chapter 9. This chapter will identify and make recommendations for further research.

1.9 Chapter Summary

The aim of this research is to identify and analyze the impact of a design change on the rest of the system, and to make proposals on how to minimize the impact of the change on the performance of the development project.

INCOSE, (2010), confirms that design iterations are fundamental to the systems engineering process, it does not, however, elaborate on the reasons for the iterations, or the impact of a design change on other components of the system under development. It also makes no mention of the influence that project management may have on the systems engineering process.

The novelty of this research lies in the identification of design influencing mechanisms, the impact of design changes on the concurrent development of other of the system components and the subsequent impact on the project.

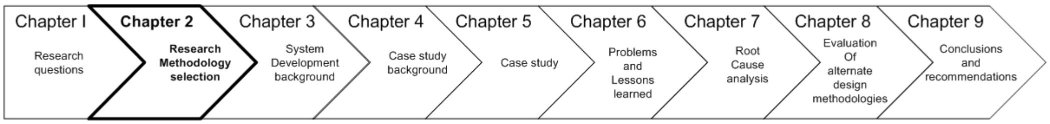
The systems engineering process does not place any constraints on either the activity time, or a resource requirement on the individual process steps. According to Kossiakoff et al, (2003), Systems Engineering must operate within the Project Management environment. This provides for the coordination and management of the schedule and the consumption of resources.

The project management process on the other hand, must function within the constraints of the organization's business process, which amongst others is cash flow and profit focused. It is therefore evident that the systems engineering process does not function in isolation, but functions within other processes. These other processes place constraints on the systems engineering process that may influence the system, under development, as well as the project's performance.

This research will determine the influence of project management on design influencing, and the effect on the overall project. The detrimental effects management has on the design quality, as highlighted by Sanjay et al, (2000) will also be explored.

In the next chapter, the selection of the research methodologies to identify the fundamental design process mechanisms, in a multi-level system hierarchy, will be discussed. As will the data collection and analysis approach to determining the root causes of problems experienced on the project.

"Dissertation Research Methodology is no doubt an important part of a dissertation project. Even it can make or break your project."
Lori Blake, 2008



Discussion of Research and Analysis Method
Selection of Research Methods
Root Cause Analysis (RCA)

In this chapter the researcher discusses the appropriate research and analysis methods, and the selection of the research methods. Also a background to root cause analysis is provided in preparation for the analysis of the case-study data.

Chapter 2 RESEARCH METHODOLOGY

2.1 Discussion of Research and Analysis Method

In the previous chapter the research objectives and research questions, in other words the "***what"*** for this research have been discussed. In this chapter the "***how"***, or the research methodologies and subsequent selection of the optimal or appropriate research method will be discussed.

First the research method to categorise and classify the problem, reporting and corrective action system data from the case-study will be discussed. This research method is generally limited to the identification of the symptoms of the problems observed. To provide a deeper understanding and subsequent finding of solutions to the problems observed, other research methods will also be investigated.

Research is in essence the search for knowledge by systematic investigation to establish the facts. The objective of research is to develop new or additional knowledge on a given subject. Generally research can broadly be classified into the following categories, Trochim, (2006):

- Exploratory research which identifies structures and quantifies new problems.
- Empirical research which tests the feasibility of a solution using empirical evidence.
- Constructive research which develops solutions to a problem.

Research may also build on previous research or knowledge which can be applied to a better understanding of a specific situation, (Trochim, 2006).

2.1.1 Exploratory Research

Exploratory research as the name implies, is a category of research also used if the problem has not been clearly defined. Its objective is to provide better insight and understanding of an observed phenomenon. Exploratory research will provide the "***why***", "***how***" and "***when***" of a specific observed phenomenon, but generally does not quantify the results. The outcome of exploratory research generally requires further research, Kotler et al, (2006).

2.1.2 Empirical Research

Empirical research in essence tests the feasibility of a solution using empirical evidence, that can be used to test a hypothesis. Empirical research according to **de Groot,** (1992), follows a cycle of Observation; Induction; Deduction; Testing and Evaluation as indicated in figure 3:

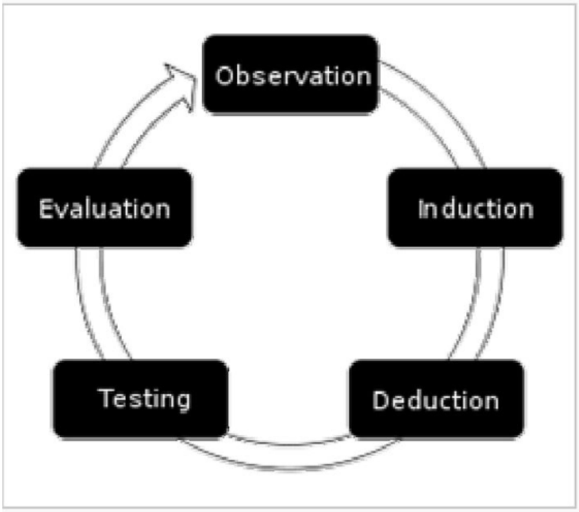


Figure 3: Empirical research cycle

Source: de Groot, (1992)

2.1.3 Constructive Research

The constructive research method evolved for finding and developing solutions to problems. The advantage apparently being that the approach demands a form of validation that does not need to be quite as empirically based as in other types of research, (Crnkovic, 2010).

Ryabov, (2009), states that constructive research aims at producing novel solutions to practically and theoretically relevant problems. According Ryabov, (2009), constructive research is widely used in software engineering and computer science. Constructive research method is building an artefact that solves a domain problem in order to create knowledge about how the problem can be solved, and if previous solutions exist, how the solution is new or better than previous ones.

A further development of Constructive Research is Action Research (AR). The literature is not clear as to the origins of AR but the concept started to appear in publications from the early 1990s with a primary focus in the computer science field. In 2004 van Aken defined the concept of Design Science Research with a more general focus, (van Aken, 2004). Livari et al, (2009) studied the similarities and differences between Action Research, and Design Science Research from several perspectives. They found that often AR does not share the paradigmatic assumptions and the research interests of DSR.

2.1.3.1 Design Science Research

The fundamental purpose of Design Science Research (DSR) is to develop general knowledge, which can be used by professionals in the field in question to design solutions to their specific problems, (van Aken, 2004). DSR is the observation, analysis, understanding and finding of a solution to a phenomenon observed in industry, (Venable et al, 1999) and (Gero et al, 2006). From the above discussions it can be concluded that DSR has evolved from Action Research which in turn has evolved from Constructive research.

2.1.3.2 Narrative Inquiry Research

Clandinin et al, (2000), describe the Narrative Inquiry as a method that uses field texts as data sources, such as stories, autobiography, journals, field notes, letters, conversations, interviews, family stories, photos (and other artefacts), and life experience. According to Clandinin et al, (2000), the narrative threads coalesce out of a past and emerge in the specific three-dimensional space called the inquiry field: "*Living, Telling, Retelling, And Reliving Stories"*. The Narrative Inquiry emerged from the field of Knowledge Management under the sphere of Information Management. The Narrative Inquiry research method is a qualitative approach to understanding the behaviour of a process. According to Clandinin et al (2000), a Narrative Inquiry is an understanding of "*narrative as both phenomena under study and method of study.*"

Generally the Narrative Inquiry is a difficult and time consuming process. Since the Narrative Inquiry was limited to the registered

PRACA reports available on the company Local Area Network (LAN) it could be effectively employed by the expert group, discussed in 2.2 and 6.1 below.

2.2 Selection of Research Methods

There are relatively few large budget completed complex multi-component system development projects, particularly in a small country such as South Africa. Also most of the detail data required for in-depth research will generally be propriety company confidential information. Companies would be very reluctant to provide intimate large budget project performance data in an open survey. This eliminates the more generally adopted empirical research methodology due to inadequate detail data to provide an acceptable level of confidence on the findings.

According to Feagin et al, (1991), a case-study is an ideal methodology when a holistic, in-depth investigation is needed. In a case-study, the researcher collects extensive data on projects and events on which the investigation is focused. Leedy at al, (2001), states that the data often includes observations, interviews, documents, and past records. In many instances, the researcher may spend extended period of time on-site and interact regularly with people working on the systems that are being studied. Stake, (1995), states that the researcher must also record the details about the context in which the case is found, including information about the physical environment and any historical, economic, and social factors that have bearing on the situation. By identifying the context of the case, the researcher helps the reader to draw conclusions about the extent to which findings may be generalised for other applications, (Stake, 1995). Stake, (1995), also further suggests that the data generated by the case-study would often agree with the experience of a broad cross section of readers, thereby facilitating a greater understanding of the phenomenon. Yin, (1994), lists four applications for a case-study model:

- To explain complex causal links in real-life interventions.
- To describe the real-life context in which the intervention has occurred.
- To describe the intervention itself.
- To explore the situation in which the intervention, being evaluated has no clear set of outcomes.

A case-study research of a full scale multi-disciplinary complex weapons system development project using the concurrent engineering IPS development model, (Roos, 2001), within a project management environment will be used. A single case-study in depth research has also been successfully applied for doctoral studies by Gumus (2005), Thunnissen, (2004), and Melvin (2003).

Using the Problem Reporting and Corrective Action system (PRACAS) database as input, a Narrative Inquiry research method is used by a Design Review Board (DRB) comprising the key development team members. The team member responsible for the PRACA item under review provides the team with the detail and background to the problem. The team discusses all aspects of the problem before categorising and allocating a Likert scale value (Likert, 1932).

This will be a precursor in preparation for a subsequent root cause analysis of the project problems experienced. The project problems experienced data will be grouped, classified and quantified after completion of the project.

For further deeper analysis, understanding as well as finding of a solution to phenomena observed, the DSR methodology developed by van Aken, (2004), was selected.

Applying a Narrative Inquiry followed by a DSR approach to the case-study; it will be possible to find answers to the research questions by:

- Developing models to depict the success/failure domain interactions in a dynamic project management environment.

- Quantify the project impact as a result of a design change of one item, in terms of the total functionally coupled items in the system hierarchy to the affected item.

- Determine whether structured design can alleviate some of the project problems.

2.3 Root Cause Analysis (RCA)

The Narrative Inquiry into the case-study problems experienced will provide the basis for further detail RCA in preparation for subsequent DSR.

Literatures abound on RCA, (Wilson 1993), (Ammerman 1998), (Mobley 1999), Leszak et al, 2000), Latino 2010). NASA-HDBK, (2008), provides detailed guidelines for the management and resolution of problems on NASA projects most of which are also very applicable in the defence industry. The handbook by Mobley, (1999), provides in three parts detailed processes for failure and root cause analysis primarily with the focus on plant performance.

RCA is described as a thought process by Latino, (2010). NASA-HDBK, (2008), states that RCA aims at attempting to correct or eliminate root causes, as opposed to merely addressing the immediately obvious symptoms. Literature agrees that corrective action is best aimed at root causes rather than symptoms to minimize the likelihood of problem recurrence, (Spolsky 2007), (Leszak et al, 2000).

In essence RCA is the in depth analysis of an undesirable phenomenon to allow a fundamental understanding of the mechanism causing the effect. The literature describes numerous ways and techniques to systematically arrive at the root cause of a problem, (Spolsky, 2007). The appropriate or best technique depends on the type of problem under investigation and the circumstances. The different techniques all boil down to making a series of measurements or observations under controlled conditions, followed by analysis of the results with the objective of isolating the undesirable effect or phenomenon. Once the phenomenon can be repeated, it can be studied and analysed until it is fully understood. Once the mechanism is fully understood, a solution to the problem can be developed to prevent recurrence of the problem. This is generally referred to in the literature as Corrective Action, (NASA, 2008). In practice RCA is generally an iterative process, and can be used as a tool for continuous improvement of a process or design. For instance design maturity and reliability growth, can most effectively be achieved by means of Test-Analyse-and-Fix (TAAF) testing, (DoD Hdbk-189, 1981). It is beyond the scope of this research to go into all aspects of RCA. For the purpose of this research, RCA will be focused on the system development environment.

If a process is not controlled, in other words the loop is not closed; no control or corrective improvement to the process can occur. For better understanding, the analogy of a closed loop process is similar to negative feedback in an analogue electronic signal amplifier to improve the output signal quality of the amplifier, (Langford-Smith 1960), (Terman, 1955).

By implementing a closed-loop Problem (or Failure) Reporting and Corrective Action System (PRACAS or FRACAS), feedback can be obtained about a development project performance, as well as details of any problems encountered, (NAVSO 1998). This is a system for recording problems and failures, and providing a management framework to analyse manage and eliminate the root causes of the problems and failures. The PRACAS term is used to widen the scope of addressing system and development process issues. During system development problems rather than failures are more likely to occur.

Since process problems can occur at any stage in a process and corrective action development takes time, it is generally impractical or impossible to stop the process until a solution is found, (Polsky, 2007). The exception is if the problem results in a safety hazard. In this case statutory regulations normally demand an immediate shut down of the process e.g. grounding of aircraft. However if the problem is not safety related, then generally the process is allowed to continue by implementing a temporary fix, or as is often called a band aid until a permanent solution to the problem can be found to prevent recurrence. Spolsky, (2007, calls this process improvement method "*Fix it twice or fix it two ways*".

Central to the success of PRACAS processes is the principle of a double feedback system as indicated in figure 4. An initial action is required to get the problem under control. A subsequent action is also required to ensure that the root cause of the problem is identified and eliminated. This will ensure that the problem will not recur, (Wessels, (1997).

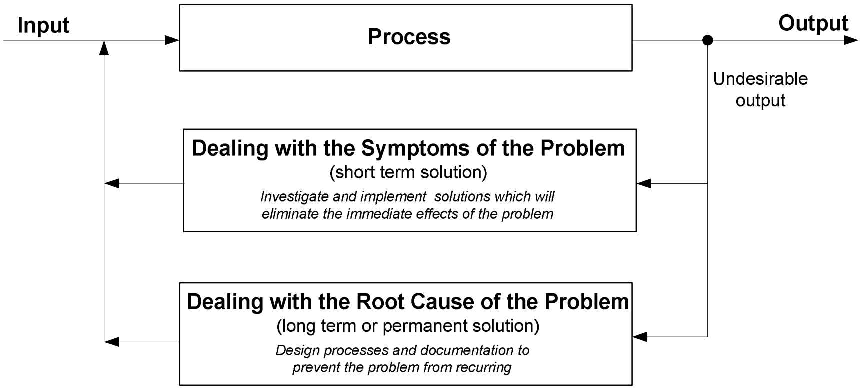


Figure 4: Double loop corrective action process

Source: Wessels, (1997).

NASA (2004), uses the closed loop PRACAS shown in figure 5.

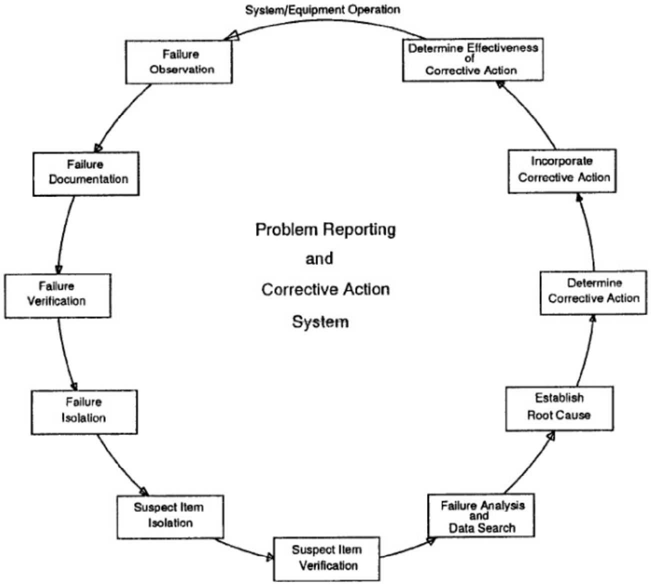


Figure 5: Closed-loop PRACAS

Source: NASA PD-ED-1255, (2004).

The information provided by PRACAS allows areas in possible need of improvement to be highlighted to engineering for development of a corrective action, if deemed necessary. With this system in place in the early phases of a project, means are provided for early identification and attention to eliminate the causes of problems/failures, (NAVSO, 1998). This contributes significantly to reliability growth and customer satisfaction. The system also allows trending data to be collected for systems that are in place. Trend analysis may show areas in need of design or operational changes, (NASA PD-ED-1255, 2004).

PRACAS addresses all problems experienced on a project and is not limited to only the system technical aspects, it includes external factors also, (NASA 2004), (NAVSO, 1998), (Wessels 1997).

In the next paragraph the research's primary focus is to find answers for the research questions and research road map will be discussed.

2.4 Chapter Summary

A literature investigation into research method and approaches has been made to enable selection of the most appropriate research method.

The PRACAS and closing the loop process were discussed to prevent recurrence of problems and to determine the root causes of these.

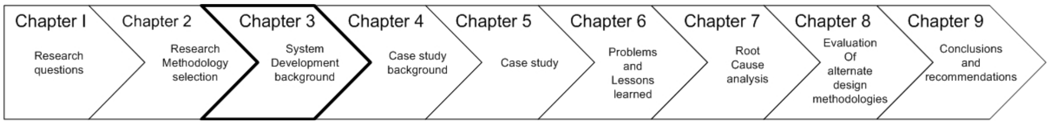
The Narrative Inquiry research method was selected for categorizing and grouping of the project PRACAS data. The Narrative Inquiry will provide a better background and insight of the context in which the development project problems occurred.

The Design Science Research (DSR) methodology was selected as the most appropriate methodology, for finding answers to the research objectives and research questions discussed in Chapter 1.

It is anticipated that by means of a better fundamental understanding of problems and their root causes, development of solutions to resolve or mitigate the problems can be found. To outline the approach, a research roadmap has been provided in chapter 1, figure 2.

In the next chapter the properties of a system, the systems engineering process, and the project management process to manage the system development project will be discussed. A design influencing approach and model will be proposed with the objective to effectively optimise a design.

"A complex system that works is invariably found to have evolved from a simple system that works."
John Gall, 1978



System – characteristics and properties
Systems Engineering
Systems engineering process
Project Management
Matrix organisational structure
Design influencing

In this chapter the researcher discusses the system characteristics and properties; an overview of systems engineering and systems engineering process, and an overview of the management structures for the management of the project resources. Design influencing is discussed and a design influencing model is developed.

Chapter 3 SYSTEM DEVELOPMENT BACKGROUND

In the previous chapter, the research approach and analysis methods have been discussed. In this chapter exploratory research will be used to provide the background needed to facilitate the achievement of the research objectives; optimising design influencing and evaluation of impact of a design change. By delving deeper into the design process, the research objective of optimisation of design influencing by dividing the design teams into two different mindset groups will be investigated. Introducing the dynamic effect of project management, prepares for the research question: "*Can models be established to depict the success/failure domain interactions in a dynamic project management environment?"*

To better understand why design influencing gives rise to iterations as well as the impact of design changes in a concurrent engineering environment, a brief overview, properties and main characteristics of the following will be discussed:

- **A system**
  A system's main properties and characteristics
- **Systems engineering**
  The systems engineering process and outcomes
- **Project Management**
  The project management process characteristics and constraints

- **Matrix organizational structure**
  The matrix organizational structure, characteristics and constraints

- **Design influencing**
  The objectives of design influencing and effects on the system under development

During the system development process, the above interact continuously and impact on the development project performance.

These properties and characteristics will be playing a pivotal role in the case-study for the development of an anti-tank weapons system.

3.1 System

In chapter 1 an overall view and definitions of a system was given. The fundamental reason for developing a system is to perform a function or number of functions in a specific environment, (Sparrius, 2006), (Booch et al, 1998). According to Booch et al, (1998), a real system must have some dynamic dimension to them, and these dynamics are triggered by things that happen externally or internally. From this it can be deduced that a real system is dynamic and that there must be functional couplings between the different elements of the system for the system to be able to function as a whole entity.

3.1.1 Characteristics and Properties of a System

All systems exist in a multi-layer hierarchy, each top layer more complex than the one below. Each layer typically forms a system in its own right. Entities from the next lower layer form its constituent's components whereas an entity from the next higher layer forms its environment. Therefore we can conclude that each entity at any layer is both a system, a component of a system and part of an environment, (Sparrius, 2008). The hierarchical structure of a system is not its only characteristic. A system also exhibits emergent properties. The performance of a system is determined not only by the performance of its subsystems or components but also by their interaction. The emergence and hierarchy principles are fundamental to systems engineering and can be defined as follows (Sparrius 2008):

- **Emergence Principle and emergent properties**

  In philosophy, systems theory, science, and art, emergence is the way complex systems and patterns arise out of a multiplicity of relatively simple interactions. Emergence is central to the theories of integrative levels and of complex systems.

Sparrius, (2008), states that: *"Every system exhibit emergent properties that derive from its components and their interaction but cannot be reduced to them. These emergent properties are meaningful only when attributed to the system, not its components"*.

Emergent properties are the principle that whole entities exhibit properties that are meaningful only when attributed to the whole not to its parts. In other words an integrated system's worth is more than the sum of its components. None of the individual subsystems, components, operating system and software of a PC can perform a word processing function, yet the integrated PC is imminently suitable for word processing (Sparrius, 2008). Sommerville, (1996), states that emergent properties are properties of the system as a whole, rather than properties that can be derived from the properties of components of a system. Emergent properties are a consequence of the relationships between system components, they can therefore only be assessed and measured once the components have been integrated into a system. In this research, these relationships between system components will be referred to as functional couplings.

- **Hierarchy Principle**

Sparrius (2008), also states that: *"All systems exist in a hierarchy. Each layer in a hierarchy is a system in its own right. The next higher layer is its environment and the next lower layer its components. The principles governing one hierarchical layer are also governing the other layers. Emergent properties distinguish layers"*. In figure 6, Hitchins (1992), illustrates a hypothetical k-level hierarchy system, and the principle that whole entities exhibit properties which are meaningful only when attributed to the whole and not its parts. He states that every system exhibits emergent properties which derive from its component activities and structure but cannot be reduced to them. He also describes the hierarchy principle according to which entities are meaningfully viewed as wholes and further states that in a hierarchy, emergent properties denotes levels. Hitchins (1992) defines the primary task of systems engineering as: "*To identify, realize and maintain the requisite emergent properties of a system to meet customer's and end-user's needs*". From the above, it can be deduced that there are always functional couplings between system components, its parents and their parents until the highest system level in the hierarchy.

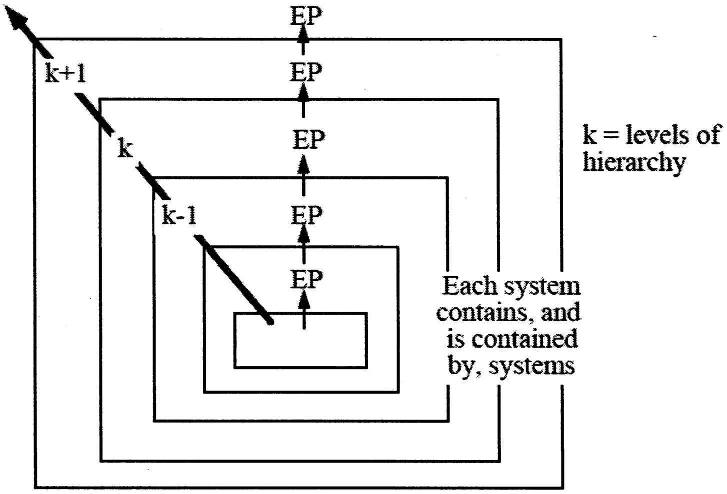


Figure 6: System emergent and hierarchy properties

Source: Hitchins, (1992).

From the above it can also be deduced that the consequences of design influencing or design change of one component in a system has a direct impact on the system as a whole as a result of the functional couplings. This will be further discussed and analysed in chapter 7.

3.1.2 System dynamics

"A perfectly static system would be intensely uninteresting because nothing ever happens."
Booch et al, (1998).

Design influencing has the connotation of design change or design amendment. Since a system is more than the sum of its components to reveal the emergent properties, design influencing must not only focus on the design item itself but also study the influence of any design change on the system's behaviour and dynamic stability

From the previous paragraphs, discussions and literature reviews, it can be concluded that real systems must be dynamic. Bertalanffy, (1968), describes a mathematical model of a system as a set of variables that maintain functional relations through time.

Viljoen, (2007) describes Bertalanffy's equations for a dynamic system by means of a set of differential equations for each state as a function of all the system elements. On analysis of the equations for a system in equilibrium, he found that the roots (real or imaginary), determine the dynamic response and stability of the system. A complete analysis of the equations by Viljoen, (2007), is provided in appendix A.

Viljoen (2007), further shows that for a system in equilibrium, the time derivative in the equations are equal to zero and that the equations can then be solved algebraically. He then proceeds to introduce a new variable, representing a modification or addition to an already stable system and finds a general solution for the modified system. He finds that a number of conclusions can be drawn from inspection of the roots of the equation.

- If all the real parts are negative, the system is stable.
- If the roots are imaginary with negative real parts, the system is asymptotically stable.
- If there are any real roots that are positive, the system is unstable.

Detail mathematical interpretation of Viljoen's (2007) findings, falls outside the scope of this research.

A consequence of the findings by Viljoen (2007), is that a change or modification to a component in a system must be approached with care, since a change or modification to one component in a system may result in the system or the affected portion of the system becoming unstable.

However from a design influencing point of view during system development, a change of the dynamic characteristics of one component of a system, can have an impact on the integrated system's dynamic performance as a result of the emergent properties, (Hitchins, 1992).

Also from Viljoen's (2007) analysis, it can be concluded that a modification or addition to an already optimally working system, is not trivial and extreme care must be taken since the delicate balance of all the elements comprising the system, can be disturbed affecting the performance of the revised system. Also a change to one element in a system very often has an impact on other elements in the system, due to the functional couplings increasing the risk of an unstable system. A change to a system element can mathematically be described as a change to the transfer function of the element in a control system. This can result in a sufficient shift to the roots of the equation, to cause the system to become unstable emphasising the caution before implementing a change.

In general debugging an unstable system is extremely difficult because of the closed control loop, and normally one has to resort to mathematical analysis and computer simulations to identify the root cause of the instability. The relevant extract of Viljoen's (2007) presentation is reproduced in appendix A.

Sparrius (2006), states that the behavioural view of a system describes its behaviour over time including which functions are active, their control and timing behaviour, their states and the conditions and events that trigger transitions between states. To facilitate design synthesis, system functions are broken down into states. A "state is a collection of descriptive variables that contain all information about the system." (Sparrius, 2006). A system stays in a state either when the function is active or waiting for an event.

The above discussions provided an overview of the characteristics and properties of a system. From this it can be deduced that design influencing must not only focus on the design item itself but also study the influence of any design change on the system's dynamic behaviour. Even small changes can influence the system's dynamic behaviour and in extreme cases may cause the system to become unstable.

In the next paragraph the salient engineering characteristics for the development of a system (systems engineering) will be discussed.

3.2 Systems Engineering

The National Council on Systems Engineering (INCOSE) web page states that*: "The term systems engineering dates back to Bell Telephone Laboratories in the early 1940s [Schlager, 1956; Hall, 1962; Fagen, 1978]. Fagen [1978] traces the concepts of systems engineering within Bell Labs back to early 1900s and describes major applications of systems engineering during World War II. Hall [1962] asserts that the first attempt to teach systems engineering as we know it today came in 1950 at MIT by Mr. Gilman, Director of Systems Engineering at Bell"* (INCOSE, 2010).

According to INCOSE (2010), the need for a more formal process of system development arose when it was no longer possible to rely on design evolution, using previous designs, to improve and expand upon a system. The existing methods were not sufficient to meet growing demands, and the problem of complex system development is further complicated by the need for the use of different technologies for the various subsystems. To reduce the system development project risks, new methodologies began to develop to address the modern problems of system development. It can be seen that the concept of systems engineering is not new, what is new however, is the formalisation and development of a disciplined process for system development. The development in better more streamlined methodologies is ongoing and is actively pursued by

INCOSE with the following mission statement: "*Our mission is to advance the state of the art and practice of systems engineering in industry, academia, and government by promoting interdisciplinary, scaleable approaches to produce technologically appropriate solutions that meet societal needs.*" (INCOSE, 2010). The evolution of Systems Engineering as it continues to this day comprises development and identification of new methods, and modelling techniques and methods that can aid in better comprehension of engineering systems as they grow more complex (INCOSE, 2010). There are a number of definitions for systems engineering in the literature. The definition by Blanchard (1997), is the most applicable to this research: *"An interdisciplinary collaborative approach to derive, evolve and verify a life-cycle balanced system solution which satisfies customer expectations and meets public acceptability".* INCOSE (2010) states that:"*Systems engineering is a multi-disciplinary effort that involves both the technical effort and technical project management aspects of a project*" (INCOSE 2010).

In the next paragraph the systems engineering process that brings a system into being, will be discussed.

3.3 Systems Engineering Process

The NASA System Engineering Manual (2004), states that the most important reason to apply Systems Engineering is that systems engineering provides the context, discipline, and tools to adequately identify, define, and manage all system requirements in a balanced and orderly manner. Systems engineering provides the disciplines required to produce complete solution concept and system architecture. Systems engineering also provides the discipline and tools to ensure that the resulting system meets all of the requirements that are feasible within specified constraints. In other words following the disciplined systems engineering approach will result in a first-time-right design resulting in reduced project risks.

Currently there are no other engineering or management disciplines that provides for the comprehensive context, or results that can be achieved with the systems engineering process. The need for effective systems engineering becomes more apparent with large, complex system developments, such as weapons and transportation systems. Systems engineering is also important in developing, producing, deploying and supporting of much smaller systems and even consumer products since the discipline lends itself admirably to design optimization within given sets of constraints.

Systems Engineering (SE) is an iterative development and design process described by INCOSE Handbook (2006), INCOSE Handbook (2010), NASA Systems Engineering handbook (NASA 2007), Blanchard (1997) and others. The iterative development and design processes can run concurrently according to IDA Report R-338 (1988) and Hill (1997). They call this concurrent development process or Concurrent Engineering (CE), (IDA Report R-338, 1988), (Hill, 1997).

3.3.1 Systems Engineering Outputs and Summary

The objective of systems engineering is to produce data packs, (Mil-Std-499B, 1994) with the aim for production and through life support of the system. According to Mil-Std-499B, (1994), a data pack "*is a technical description of an item (product and process) adequate for supporting an acquisition strategy, production, engineering, and logistics support. The description defines the required design configuration and procedures to ensure adequacy of item performance. It consists of all applicable technical data such as drawings, associated lists, specifications, standards, performance requirements, quality assurance provisions, and packaging details*"

The Systems Engineering process has the following outputs:

- **Product data pack.**
- **Production data pack.**
- **Support and Operating data pack.**

The systems engineering process output is ***data*** and not ***hardware***. All hardware models build during the SE process is solely for the verification and qualification of the data packs, (Mil-Std-499B, 1994). An important component of the qualification is product reliability management (Murthy et al, 2008) and reliability growth tests (Mil-Hdk-189, 1981). Design influencing during reliability growth is to test the item until failure, analyse the failure, find the root cause and develop and implement a fix. This process is called Test-Analyse-And-Fix (TAAF) testing, (Mil-Hdk-189, 1981).

The hardware models built during development must be kept under configuration control to enable test and evaluation of modifications as part of through-life engineering support (Mil-Std-1521, 1995), (Saaksvuori et al, 2005).

The systems engineering process described in the literature (INCOSE, 2010), (NASA, 2007), is a good practices sequence of different activities to be performed, in order to develop an effective system according to NAVSO Best Practices (1986). The systems engineering process literature does not stipulate a time or resource criteria on an activity in the systems engineering process. The deduction that can be made is that the systems engineering process on its own is a static process or guideline. In order to bring a system into being, the systems engineering process also needs a resource and time management process described in the next paragraph.

The salient characteristics and properties of a system and the process to bring a system into being have been discussed. The process so far has been static and in order to develop a system and manage the resources, the relevant characteristics of project management will be discussed in the next paragraph.

3.4 Project Management

Project Management was developed from different fields of application which include construction, engineering and defence. Henry Gantt is credited as a “father” of planning and control techniques by various internet publications in the United States of America (Gantt, 2010). The Project Management Institute (PMI) plays an important role in the project management profession. It has an active global community of more than half a million members distributed in more than 170 countries, including South Africa. It is a leading membership association for the project management profession (PMI website, www.pmi.org, July 2010). Therefore in the project management sphere, PMI and its technical developments play an important role. The PMI has developed the Project Management Body of Knowledge (PMBOK®), which is an internationally recognized standard (IEEE Std 1490, 2003) for fundamentals of project management irrespective of the type of project. The risk management methodology of the PMBOK® is one of the mostly used methods to control risks in projects (INCOSE, 2004). The fourth edition of PMBOK was published in 2008, (PMBOK, 2008).

The PMI states that a project is “*a unique* ***temporary*** *endeavour, with a set* ***beginning*** *and* ***end****.*” PMBOK (2008), states that project management is “*the application of knowledge, skills, tools and techniques to a broad range of activities in order to meet the requirements of a particular project*”. PMBOK identifies five process groups:

- Initiating
- Planning
- Executing
- Monitoring and Controlling
- Closing.

Processes are described in terms of

- Inputs (documents, plans, etc)
- Tools & Techniques (mechanisms applied to inputs)
- Outputs (documents, products, etc)

There are nine knowledge areas that are applicable across nearly every industry worldwide,(Kerzner, 2009):

- Integration
- Scope
- Time
- Cost
- Quality
- Human resources
- Communications
- Risk management (the subject of this study)
- Procurement

The elements of project success are illustrated in figure 7.

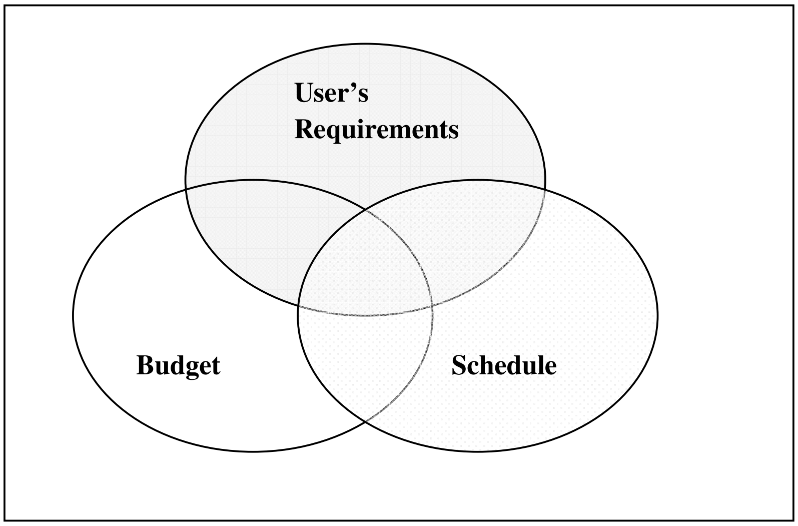


Figure 7: Elements of project success

Source: Coblands Consulting, (1995)

Referring to figure 8, a system cannot be developed using the systems engineering process by itself. It requires project management to coordinate and manage the schedule as well as the consumption of resources to ensure ultimate project success, (Kossiakoff, 2003).

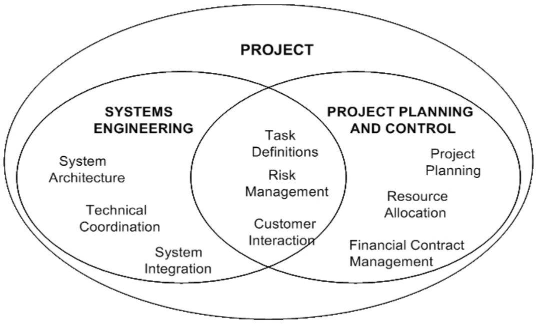


Figure 8: Systems Engineering environment

Source: Kossiakoff, (2003)

From the above discussion, it can therefore be construed that the hierarchy principle is not only applicable to systems but also to the processes that bring the systems into being. No process can run in isolation it is always encompassed inside other processes. As such the process interfaces can have a distinct influence on the process under investigation.

As discussed above, PM manages the resources for a project. To ensure availability of human resources at the right time on the project plan, the matrix organisational structure of human resource management will be discussed in the next paragraph.

3.5 Matrix Organisational Structure

The project for the development of multi-component, multi-disciplinary systems can be efficiently managed in a matrix organisational structure.

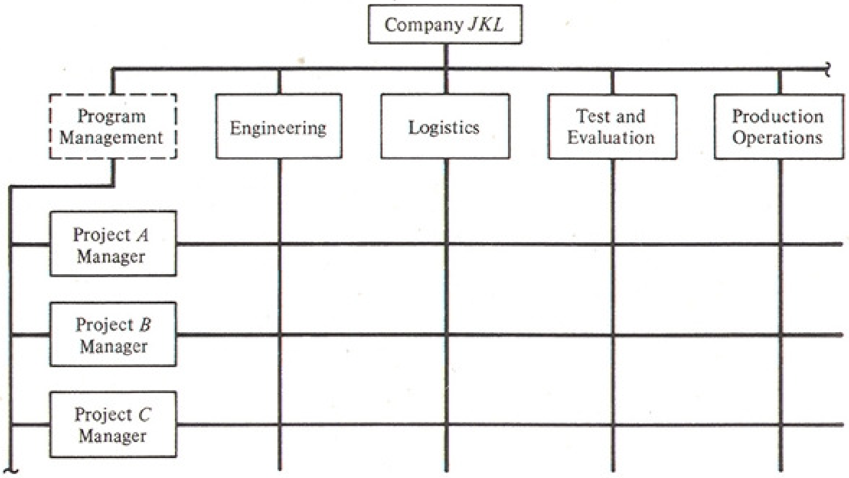


Figure 9: Matrix Organisational Structure

Source: Blanchard, (1998)

The Matrix organisational structure indicated in figure 9, is a two dimensional structure with project lines and functional capability lines generally referred to as facilities. This structure is the preferred organisational structure for the smaller system development companies since it allows efficient sharing of scarce skilled resources between different projects, (Blanchard, 1998).

According to Guzman, (2011), the matrix organizational structure divides management authority both by functional area and by project. In a matrix structure, each employee answers to two immediate supervisors - a functional supervisor and a project supervisor. The functional supervisor focuses on hiring, training and managing employees in their field, while project supervisors can focus on achieving the goals of their specific projects or products. He also states that by placing employees in functional areas allows them to specialize in a particular field. Instead of being good at a variety of tasks, specialized employees can excel at tasks in their field of focus, (Guzman, 2011).

According to Bassiouny, (2011), the advantages of the matrix organisational structure are:

- Allows employees from different departments to come together temporarily to work on special project teams.

- Provides flexibility to respond quickly to a customer need by creating a team of people who devote all their time to a project and then return to their department or join a new project team.

The downside of the Matrix organisational structure for system development is that once the specific subsystem design has been completed, the resources are re-allocated to other projects and are not available anymore to the current project. This can lead to project delays should during integration a problem be experienced and the original design resources are required.

Another downside is that team members are often selected on availability at the time of the project (Starbek et al, 2002). It is very difficult to compose an optimal team for a specific design project task in a multi-project facility environment. In the smaller companies specialist skilled resources are scarce and in demand by other projects running in the company. This may impact on the selection of an optimal team not only from a technical perspective but also from a human relations aspect and may result in reduced team cohesion, (Kim et al, 2008). The reduced design team cohesion can only be partially countered by a structured design methodology to be further discussed in chapter 8.

The next paragraph will discuss design influencing and the structuring of a design team.

3.6 Design Influencing

The primary focus of this research is design influencing. Design influencing can be viewed as design improvement proposals evaluated in the context of the project's value system.

From the previous discussions it can be seen that a process always functions within another process, and that the other process can have a distinct influence on the original process's performance. The objective of design influencing is to accelerate design optimisation with the aim to drive the design to maturity. The earlier this is addressed in the design process, the lower the cost impact of a design change, (Wessels et al, 1998). Buede (2000) states that design influencing is a process to improve the future status of the product, and one that culminates in the allocation of resources to affect the chosen change. The most commonly allocated resources are human, material and time (Buede 2000).

Design influencing can be made more objective and repeatable by the application of influence diagrams and decision trees (Buede, (2000). INTELLECT, (2003), further refines design influencing by applying success frame and failure frame considerations to a design.

Design influencing has the connotation of design change or design amendment with the objective to develop a better product. Design influencing cannot be done in isolation as can be seen from previous discussions. A change of the characteristics of a component of a system may also influence the emergent properties and the dynamics of the system in a multi-level hierarchy. The problem is further exacerbated in a concurrent systems engineering development environment. This will be further discussed in chapter 7.

One of the objectives of this research is to optimise design influencing. The design process is a multi-facetted process and is difficult to model, (INCOSE, 2010). Since design is a process for satisfying requirements. Requirements set out what the design must do. This may be functional requirements to describe a service or function or may be non-functional requirements or constraints placed on the design (Sommerville 1996).

Before design influencing can be considered in detail, and a model developed, it is necessary to have a clear perspective of the most basic requirements of a design. A good design must, amongst other factors, function properly within its design parameters and environments and be cost effective. These environments are external influences and are at best predictions that can't be controlled by the designer. To ensure that a design always behaves in a controlled and orderly fashion the designer must also consider the design's behaviour for out of specification conditions. A good example would be a software module processing the data from an external sensor. If the sensor provides data that is erratic and/or out of specification, the software must behave in an orderly manner and must not hang-up, but elevate the condition to the next system level.

Therefore, a good design team must not only focus on the technical requirements for a design but also on the constraints and external conditions which are inherently imposed on the design. This requires two different and almost opposing mindsets which are very difficult to vest in the design team alone.

The studies by Kim et al, (2008) and Kuhn et al, (2006) found that teams that developed integrative conflict management styles made more effective decisions than teams that utilized confrontation and avoidance styles. They also found that teams that never developed a stable style were less effective than teams with integrative styles. Also Kim et al, (2008), found that cross-functional cooperation between teams in new product development had a positive impact on product development performance. These studies also found that work groups comprised of people with opposing mindsets produce better results. A topic under review by this type of work group will be investigated more thoroughly than would be the case for a group with a homogeneous mindset (Kim et al, 2008), (Kuhn et al, 2006).

Optimising a development team's effectiveness can be approached in a DSR setting by dividing the development teams into two groups addressing different aspects of the design process. One group to focus on the functional requirements and another group must focus on the non-functional requirements.

To achieve the functional requirements, the design team must focus on all aspects design success. The mindset of the team focussing on compliance with the functional requirements is therefore in the design success domain.

To address the non-functional requirements, the design team must focus on how the design can fail and how it must behave under those conditions to achieve the requirements. The mindset of the team focussing on the non-functional requirements can therefore be said to be in the failure domain.

Such a division would lead to a Success Domain (SD) and Failure Domain (FD) team. The SD design team would then focus on the functional requirements whilst the FD design team would focus on the non-functional requirements. However, in practice, the SD and FD teams already exist in most Systems Engineering development environments. The SD team is comprised of those team members responsible for the functional and detailed designs of the system, whereas, the FD team is made up of those members of the team responsible for the Reliability and Logistics Engineering aspects of the design.

Applying these principles and improving team interaction and effectiveness, the design teams are divided into two groups with opposing mindsets:

- A system/subsystem development team, referred to as the Success Domain (SD) team.

- A logistics engineering development team referred to as the Failure Domain (FD) team.

A systems engineer heads each team. One systems engineer was responsible for the development and architecture of the system and the other team was responsible for the reliability and logistics system engineering tasks as well as the subsequent development of the logistics products.

This will create a constructive conflict environment and it is now possible to develop a model to study the interaction between the two domains.

3.6.1 Success Domain Team (SD)

The “***Success Domain***” design team must strive for design success. In other words the mindset of the SD team is: “*what is the minimum acceptable success?*” The mind-set of the SD team comprising systems engineering, subsystem development teams and design engineers are therefore set in the “***Success Domain***”.

This team’s objective is to get the system, subsystems and associated software working in compliance with the requirements and development specifications.

3.6.2 Failure Domain Team (FD)

The “***Failure Domain***” team must identify design weaknesses. In other words the mindset of the FD team is: “*what is the maximum tolerable failure and what are the weaknesses in the design?*” The mind-set of the FD team is failure mitigation of the design. The whole objective is to analyse the system, subsystems architecture and designs to determine what makes them fail and the maximum tolerable failure.

The Success/Failure domain concept is shown in the figure 10:

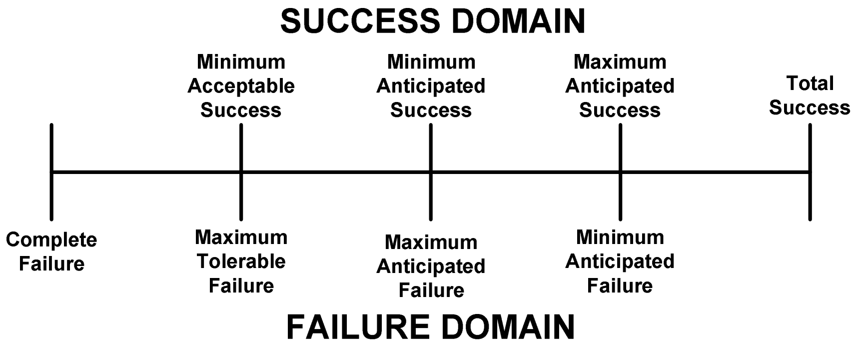


Figure 10: Success/Failure domain concept

Source: Reliability Practitioner’s Guide, (2003)

Part of the FD analysis is also to evaluate system behaviour when external conditions are outside specification, in order to ensure an orderly and safe system performance under failure conditions. This will ensure a more robust and safe design. Logistics systems engineering also evaluate designs for Testability, Reliability, Affordability, Maintainability and Produceability (TRAMP) requirements.

Applying the Success/Failure domain concept to the systems engineering process, (INCOSE, 2010), the design influencing model shown in figure 11 can be developed. This model illustrates the interaction between the systems engineering and logistics engineering processes.

The outcomes of the FD/SD domain analyses are used for design influencing shown in the figure 11:

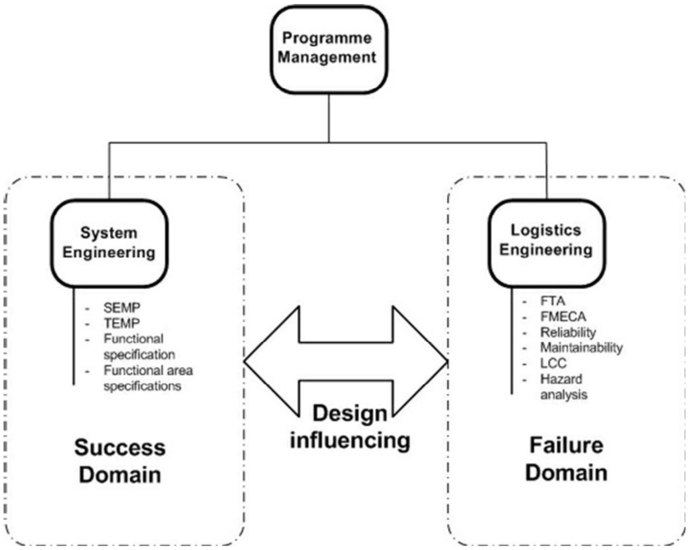


Figure 11: Design influencing model

The design influencing model in figure 11 systems engineering team (SD team) focus on design success by ensuring design functional specification compliance using the established systems engineering processes described in INCOSE (2010).

The logistics engineering team (FD team) analyse the design from a non conformance perspective and the severity and impact on the system using the logistics engineering processes, (Mil-Std-1369A (1988).

Although under the Logistics Engineering umbrella, the FD teams are responsible for design analyses. Logistics Engineers specify the logistic and production products requirements which in turn are developed using a similar SD-FD design influencing process.

The processes discussed above are static in the sense that the processes have no schedule time constraints and have not been considered in a project management environment. Kossiakoff (2003), has shown that a system can only be developed in a project management environment, since project management provides the time function (schedule) to the system development project.

Placing a project management time function on the Success Domain (SD)-Failure Domain (FD) requirements and constraints, a dynamic design influencing model can now be developed shown in figure 12. This model makes the static design influencing processes illustrated in figures 10 and 11 dynamic. The model in figure 12 shows the iterative design influencing process between the success and failure domains.

Figure 12 shows the iterative design influencing between the SD and FD teams. The objective of both teams is a successful compliant design.

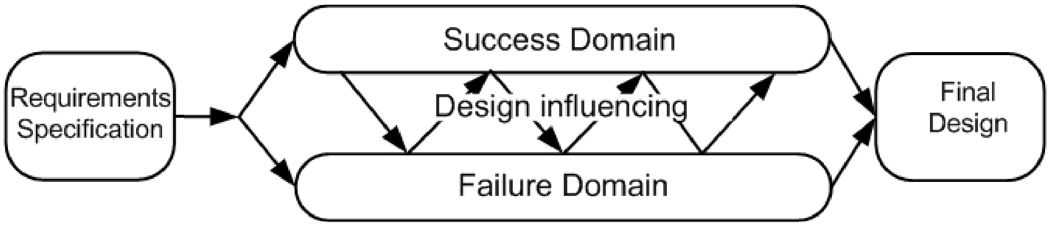


Figure 12: Interaction between the SD and FD teams

3.6.3 Project Management Team (PM)

The project management must satisfy the requirements of the project stakeholders (PMBOK, 2008). Therefore the primary development team objectives others are:

- Successful project
- Satisfied client
- Satisfied company Management

For the development of an anti-tank weapons system, the systems engineering process must function within a project management environment, to manage the consumption of resources to ensure project success.

The PM team is responsible for ensuring that the project is completed according to the contract, within cost and schedule. To achieve this, the project management team must manage the key systems engineering interfaces, (INCOSE, 2010).

Superficially the SD-FD teams and the PM team's objectives appear similar in that both strive for project success; they are in fact distinctly different. SD-FD teams are product focussed and their performance is measured in terms of design success and compliance. The PM team on the other hand is business model focussed and their performance is measured in terms of company business success (PMBOK, 2008).

3.7 Chapter Summary

In this chapter the main characteristics of a system were discussed. An overview was given of the systems engineering process, required to bring the system into being and the project management environment within which the systems engineering process must function. The matrix organisational structure advantages and disadvantages in the context of a complex system development environment were discussed. A model for design influencing was proposed providing a constructive conflict environment for efficient design optimisation.

The **Success Domain (SD) - Failure Domain (FD)** model was introduced as part of exploratory research in a DSR setting, with the objective of optimising design influencing. Applying the project management time resource function to the model, results in the dynamic design influencing model shown in figure 12. This model will be further expanded and discussed in chapter 7 as part of the case-study root cause analysis.

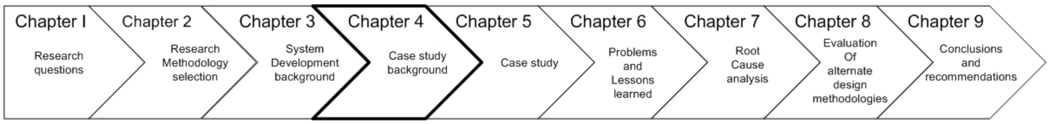
It was also shown that no process can function in isolation, and that the interactions with other processes must be taken into account when analysing a process. In the system development environment, the systems engineering process interfaces with the project management process and the facility management process. These processes in turn function within the company business management processes. All these processes have a distinct influence on the behaviour and performance of the systems engineering process under investigation and must be taken into account.

The research question: "*Can models be established to depict the success/failure domain interactions in a dynamic project management environment?*" has been shown theoretically feasible. A case-study for the development of an anti-tank weapons system will be used to confirm this model in practice. The development of the anti-tank weapons system will provide a good testing ground for the model since this system consists of multiple subsystems and components applying a broad spectrum of engineering disciplines.

The defence industry worldwide is a specialised industry and with its own terminology. Specifically for the benefit of the non-defence industry reader, an introduction to case-study has been provided. In the next chapter an introduction to anti-tank armour and the case-study background will be discussed to provide a better perspective of the constraints and processes within which the development project of the anti-tank weapons system had to function.

"The arms industry is a global industry and business which manufactures and sells weapons and military technology and equipment. It comprises government and commercial industry involved in research, development, production, and service of military material, equipment and facilities."

Wikipedia (2011)



Background of Armour and Anti-Tank Weapons Systems
Introduction to Anti-Tank Missile systems
User requirements background
Ingwe missile description
Top-level User Requirements statement
Primary constraints invoked by client
Contract overview
Project management model

In this chapter the researcher provides an introduction and management background to the case-study.

Chapter 4 BACKGROUND TO THE CASE-STUDY

In the previous chapter, the relevant characteristics of a system, the systems engineering process and the management process within which the systems engineering process functions were discussed.

Exploratory and Narrative Inquiry research methodologies will be applied to the data obtained from the case-study. This will enable the observation of the symptoms of underlying problems. DSR will provide the deeper insight and understanding, required for the research question to allow the development of models, for the success/failure domain development teams' interaction with project management. DSR methodology will provide the means for achieving the research objective of optimising design influencing.

DSR will also facilitate obtaining an answer to the research question to determine the impact of a design change in a system hierarchy discussed in chapter 2. In order to address the research objective to optimise design influencing, as well as evaluate the impact of design change in a concurrent engineering development environment, a case-study for the development of a third generation anti-tank weapons system has been selected.

In this chapter the background of the system and development environment that will be used for the case-study, will be discussed in order to provide a better insight into the case-study for the upgrade of the ZT3 Anti-Tank Weapons System. This will provide better understanding and appreciation of the problems experienced on the development project.

4.1 Purpose and Outline of the Chapter

The purpose of this chapter is to provide background to anti-tank weapons systems, in order to illustrate the multi-discipline technology and complexity for the development of a third generation anti-tank system. A third generation anti-tank weapons system is a good example of a multi-disciplinary, multi-component system in a multi-hierarchical system level structure. The case-study will provide the basis for the research objectives discussed in chapter 1.

This chapter then proceeds to provide background to the procurement, user requirements and high-level contractual requirements. An overview of the management and infrastructure of the contractor is also provided.

The contractor has very well established infrastructures for the development of complex multi-discipline systems and is equipped with the latest information systems and tools. This background should enable the reader to objectively evaluate the case-study and problems encountered.

The objectives of the case-study are:

- Evaluate the concurrent engineering IPS development model on a full scale development project for a complex multi-disciplinary system applying the design influencing model.

- Evaluate the effectiveness of the SD-FD design influencing model.

- Provide data to facilitate identification and quantification of any problems experienced on the project.

- Provide detailed data to enable Root Cause analysis for any problems experienced.

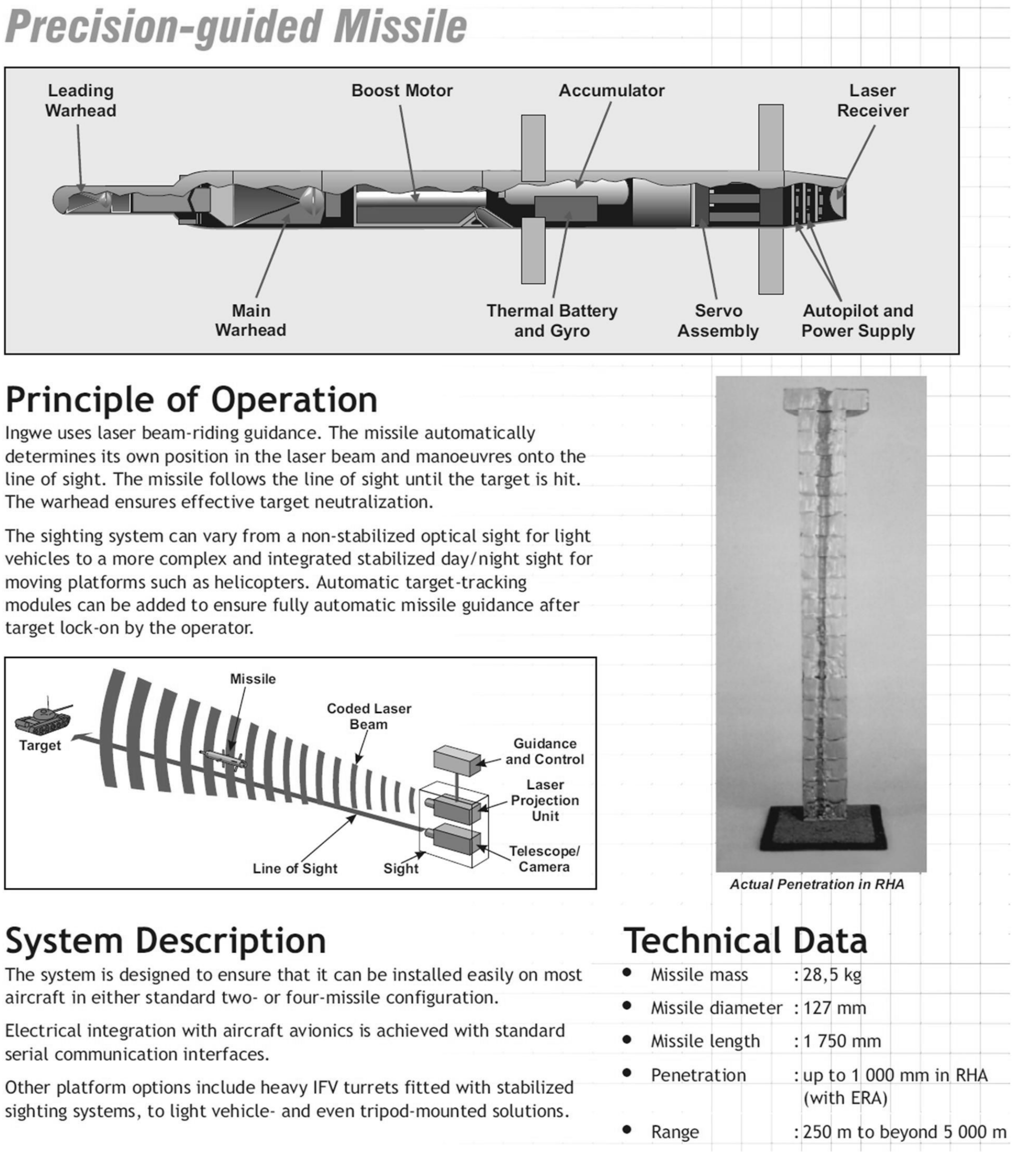
4.2 Background of Armour and Anti-Tank Weapons Systems

This is a case-study of a fully-fledged development project for a complex multi-disciplinary system, inclusive of the associated logistic system, required to provide through-life support for the newly developed system. Detail PRACA (Problem Reporting and Corrective Action) data will be collected for evaluation of the systems engineering process, as well as the development model effectiveness within a project management environment. The author has been involved with this project from day one until the system was finally and successfully put into production and operational use. The project took 3 years to completion.

The PRACA data will be analyzed, classified and quantified using the Narrative Inquiry research method. DSR will then be used to provide a better insight and understanding of the analysis results and finding of a solution to the phenomena observed, (Venable et al, 1999) and (Gero et al, 2006).

The system to be discussed for the case-study represents an extensive upgrade to an already existing system in the client's inventory, with many subsystems and software modules being totally new developments. Generally no system development project starts with a zero baseline. Existing system/s and components are generally re-deployed as building blocks for a new developed system configuration, primarily to reduce the cost, development schedule and technical risks (Tomaiko, 2008). For the weapons system development project used in the case-study, certain existing components of the original system were retained but mostly were extensively modified or upgraded. This was motivated particularly from the lessons learned during the operational deployment of more than 10 years of the existing system.

Although this system is equipped with a target auto tracker, it is not a fully-fledged "*Fire-and-Forget*", (Hogg 1996), system, and requires limited human control until the target is destroyed. The design and ergonomics therefore must be such that the workload and human errors are to a large extent minimised. This is also a primary safety requirement of the system. The system description provided in the company marketing brochure is shown in figure 13. More details can be found in reference: Denel Dynamics Ingwe Missile brochure, (2009).



Precision-guided Missile

Principle of Operation

Ingwe uses laser beam-riding guidance. The missile automatically determines its own position in the laser beam and manoeuvres onto the line of sight. The missile follows the line of sight until the target is hit. The warhead ensures effective target neutralization.

The sighting system can vary from a non-stabilized optical sight for light vehicles to a more complex and integrated stabilized day/night sight for moving platforms such as helicopters. Automatic target-tracking modules can be added to ensure fully automatic missile guidance after target lock-on by the operator.

Actual Penetration in RHA

System Description

The system is designed to ensure that it can be installed easily on most aircraft in either standard two- or four-missile configuration.

Electrical integration with aircraft avionics is achieved with standard serial communication interfaces.

Other platform options include heavy IFV turrets fitted with stabilized sighting systems, to light vehicle- and even tripod-mounted solutions.

Technical Data

- Missile mass : 28,5 kg
- Missile diameter : 127 mm
- Missile length : 1 750 mm
- Penetration : up to 1 000 mm in RHA (with ERA)
- Range : 250 m to beyond 5 000 m

Figure 13: Anti-Tank Weapons System

Source: Denel Dynamics Ingwe Missile brochure, (2009)

This case-study has been compiled from the author’s extensive diary of events during the approximately three-year project. As such this research may be categorized as Design Science Research (DSR) in a case-study setting, (Venable et al, 1999) and (Livari et al, 2009). Due to company confidentiality reasons, technical information and product specific data have been limited to what is available in the public domain such as the Internet or marketing brochures. These sources have been extensively referenced where applicable.

4.3 Scope of the Case-study

The case-study will provide the platform to study the optimisation of design influencing by dividing the design teams into two different mind-set groups and to evaluate the impact of design changes in terms of cost and schedule overruns in a concurrent engineering development environment.

This case-study is extensive as it covers a fully-fledged multi-discipline complex weapon system development complete with the logistics package from day one until the final handover to the client. The prime objective of this research is to evaluate in practice in a DSR research setting, the effectiveness of the proposed design influencing model and the impact of design changes on the project. With this in mind and in order to keep the volume of the research work acceptable and not to lose focus of the primary objective of this research, the project management aspects of the project have been excluded except for areas of conflict between the systems engineering process and project management. The focus for this case-study will therefore primarily be on the systems engineering and design process and be limited to the research objectives.

4.4 Introduction to Anti-Tank Missile Systems

Armour is not new to man. History shows that since the early days, armour has been used in warfare. Soldiers used shields to protect themselves against enemy swords, spears and arrows. As the technology developed, the weapons and armour became more mobile and sophisticated. The Egyptians and Romans used wooden horse drawn chariots (carts) as armoured fighting vehicles. With the arrival of gunpowder, the armour changed from wood to leather to steel. Today we have the modern battle tank in place of the chariot, and high-explosive ammunition in place of the swords, spears, arrows and catapults.

Today, modern armour is highly mobile and has devastating firepower over long distances. Because of its mobility and firepower, it became more and more difficult to destroy modern armour with rockets and conventional guns. Therefore, anti-tank guided rocket (missile) systems were developed. With guided missiles, moving targets can be destroyed over long distances. (Hogg, 1996).

4.5 Evolution of Anti-Tank Weapons Systems

This paragraph discusses the evolution of the anti-tank weapons systems and highlights the complexity and multi-disciplinary characteristics of these systems.

4.5.1 First Generation Anti-Tank Missile Systems

According to Hogg (1996) modern tanks originated during WWI (1914) and immediately after its apparent successful introduction into warfare, anti-tank weapons were developed. Modern anti-tank weapons use a guided missile, since it is highly manoeuvrable and relatively easy to launch and very effective against armour.

In a military situation, the operator identifies a target and launches a missile. Once the missile is launched, two thin wires are dragged out behind the missile and a light source illuminates in the rear end of the missile. The two thin wires form the communications link between the missile and guidance unit whilst the light source indicates the position of the missile in flight to the operator. The operator tracks the target through his optical sight. The operator has to steer the missile using a joystick towards a target that can be stationary or moving. The light source at the rear of the missile indicates the present position of the missile. The operator controls the missile by moving the joystick according to where he wants the missile in relation to the target. When the operator moves the joystick, the joystick generates electrical control signals that are transmitted to the guidance unit. The guidance unit processes the data from the joystick, generates the control signals and transmits the control signals to the missile in flight via the two thin wires. This is referred to as Manual Command to Line of Sight (MCLOS), (Hogg, 1996). The above discussion illustrates system complexity.

Despite being relatively cheap and portable, the main disadvantages of first generation anti-tank missile systems are:

- High operator workload. The operator must track both the target and missile.
- Expensive training. The operators must be well trained and can only train with real missiles.
- Limited range of approximately 2000 m due to the wire link.
- A second missile cannot be launched whilst the first missile is in flight.
- The disadvantage is that the operator must remain stationary and in view of the target during the flight time of the missile.
- This makes the operator vulnerable while guiding the missile.

4.5.2 Second Generation Anti-Tank Missile Systems

In this case the operator again identifies a target and launches a missile. Once the missile is launched, two thin wires are dragged out behind the missile for communication between the missile and the guidance unit. Some systems make use of an optical data link instead of wires for communication between the missile and the guidance unit. Immediately after launch, an infra-red light source illuminates in the rear end of the missile. The infra-red light source indicates the position of the missile in flight. A goniometer that is sensitive to infra-red light determines the position of the missile in flight, relative to the operator's line of sight. The goniometer sends this data to the guidance unit. The guidance unit processes this data and generates control signals for the control of the missile. Control signals are transmitted to the missile in flight via the two wires or via an optical data link. The operator uses a control stick (joystick) to control his line of sight as he tracks a moving target. The goniometer reference is the same as the operator's aiming point. This illustrates the complexity of the man-machine interface and precise interworking of electronics, mechanics, and aero dynamics.

As the line of sight changes, so does the infra-red light source in relation to the window of the goniometer. The guidance unit controls the missile to fly on the line of sight. This is referred to as Semi-Automatic Command to Line of Sight (SACLOS), source: SACLOS, (Hogg, 1996).

The advantages of the second-generation anti-tank systems are:

- Minimum operator interface.
- Training is relatively cheap. Simulators can be used to train operators.
- Long range approximately 4000 m
- Can be carried on a variety of vehicles, for example helicopters and infantry fighting vehicles

The Disadvantages however are:

- Can launch only one missile at a time
- The operator must remain stationary during the missile's flight.
- The operator is vulnerable while the missile is in flight

- Interference to the operator's line of sight due to light, water or terrain.

4.5.3 Third Generation Anti-Tank Missile Systems

Third-generation anti-tank missile systems are primarily automated second-generation anti-tank missile systems of the “fire-and-forget” type. Once the target is identified the missile needs no further guidance during flight. The missiles are “beam riders” and the systems are equipped with auto trackers. These systems also have crossfire capabilities allowing multiple targets to be engaged. The fire-and-forget missiles are more subject to electronic countermeasures than MCLOS and SACLOS missiles. Modern anti-tank guided missiles (ATGMs) have shaped-charge high explosive anti-tank (HEAT) warheads, designed specifically for penetrating armour. A counter measure that is used on tanks is to use explosive reactive armour (ERA). The Tandem-charge missiles attempt to defeat ERA protected armour. The small initial charge sets off the ERA while the follow up main charge attempts to penetrate the main armour source: Anti-tank Guided Missiles, (Hogg, 1996).

The above paragraphs provided a brief overview of the evolution of anti-tank missile systems. Hogg (1996) deals extensively with the subject and is recommended for further reading.

The South African National Defence Force (SANDF) identified a need to upgrade their existing 2nd generation anti-tank weapon system (ZT3A1) to a third generation system.

4.6 User Requirements Background

Anti-tank weapons forms part of the South African National Defence Force (SANDF) army’s armour formation, (source: SA Army Armour Formation website, March 2009).

4.6.1 Existing Anti-Tank Armoured Vehicle

The South African National Defence force (SANDF) has operated the ZT3A1 anti-tank missile system very successfully for a number of years, refer to figure 14.

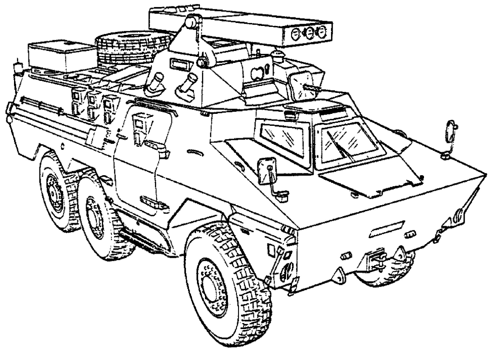


Figure 14: ZT3A1 Anti-tank Missile System

Source SA Army Vehicles website (March 2009)

This system was becoming obsolete and difficult to operate and maintain. The system was primarily an analogue control system with interaction between many critical functions and tasks adding to operator work load and task complexity. The main disadvantage of the system was that it needed well-trained and skilled operators to successfully apply the system. The system did not have automatic target tracking to ensure fully automatic missile guidance after target lock-on by the operator, an essential feature particularly against moving targets. The missile range and penetration, particularly against reactive armour, made it less effective against modern tanks. The system had no night and cross fire capability. Crossfire is the ability of engaging multiple targets by a battery of ZT3s by the coding of laser beams and missiles so that a missile in flight could not accidentally jump from one beam to the other.

The ZT3A1 has an extensive and well-established integrated logistic system consisting of:

- Ratel missile platform support infrastructure
- Ratel Turret support infrastructure
- Missile weapon system support infrastructure
- Maintenance and recovery vehicles (MRVs)
- Training system

This integrated logistic system interfaces with the user's standard support systems and tactical communications network.

The above discussion provides the contractual framework for the project.

4.7 Ingwe Missile Description

A critical component for any successful anti-tank weapons system is the missile. The SADNF already had an effective air-to-ground missile (Ingwe) complying with all the new anti-tank requirements in their inventory. In the drive for standardisation, this missile was selected for the new anti-tank weapon system.

The primary function of any weapons system is to destroy an enemy target. The missile must be designed and controlled by the weapons platform to be able to perform this function. The Ingwe missile is a modern South African developed multi-role laser guided anti-tank guided missile (ATGM) manufactured by Denel Aerospace Systems. The missile was designed to be employed in various roles, either by infantry or as a vehicle or helicopter mounted system for targets at ranges from 250m to 5,000m and is fitted with a tandem shaped charge warhead, able to penetrate ERA protected armour up to 1000mm thick. Ingwe is also fitted with a dual redundant, standoff fuse, optimizing warhead penetration against pre-confirmed targets. The missile's on board software is able to detect the launch platform and download the correct software for the application during launch time. This feature enables the use of a single missile across all launch platforms. The missiles can be fired on crossing flight paths, at different targets on the battlefield without guidance disruptions. The missile can also be used at night by means of thermal imagers, integrated with the sight, source: Denel Dynamics Ingwe Missile brochure, 2009.

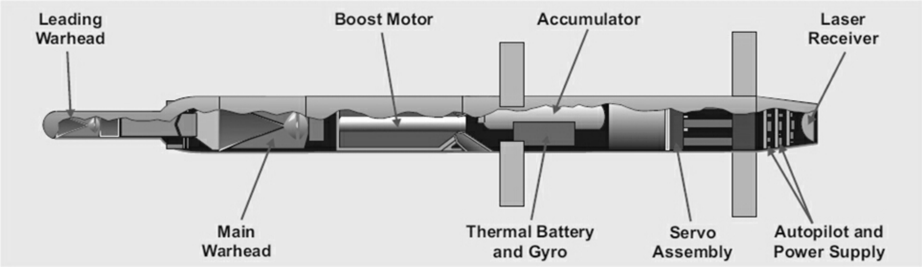


Figure 15: Ingwe Missile cut-away

Source: Denel Dynamics Ingwe Missile brochure, 2009

4.8 User Requirements

Discussed earlier, the client identified a need for a 3rd generation anti-tank weapons system since its existing anti-tank weapons system which was becoming obsolete. To this effect, the client in its aim to modernise their current anti-tank weapons system has prepared a high level User Requirements Statement (URS). The URS specified the requirements for a third generation Missile Products System identified by the acronym NGMPS. To contain costs, the client decided to upgrade their existing ZTA1 anti-tank inventory to NGMPS level of capability and performance. The URS specified both the functional and non-functional (constraints) requirements. The main requirements can be broken down into:

- Mission requirements
- System requirements
- Interface with other system requirements
- Design and construction requirements
- Missile requirements
- Testability, Reliability, Affordability, Maintainability and Produceability (TRAMP) requirements
- Logistic requirements
- Support system requirements

4.9 Primary Constraints Invoked by the Client

The primary constraints for the development of the new 3rd generation anti-tank weapons system invoked by the user were:

- The use of customer furnished items (CFE)[2] such as the Ratel Anti-Tank armoured vehicle, Ingwe missile and Training System.
- The user emphasized that the silhouette of the upgraded weapons system vehicle must be similar to that of the standard Ratel Mk3 or of the ZT3-A1 Ratel Mk3 to ensure difficulty in distinguishing between the old and the upgraded anti-tank missile systems. This implied that no major modifications to the outside of the armoured vehicle and turret were allowed.
- The use of the existing Ingwe missile in an ordnance standardization drive. The Ingwe missile is already in its inventory and is being successfully deployed by the attack helicopters (AH) and other armed forces.
- Statutory requirements such as International Traffic in Arms Regulations (ITAR, 2011), road ordinance compliance and safety requirements.

---

[2] CFE is a non tradable requirement that places a constraint on the system architecture

- Timescale.
- Costs.

4.10 Contract Overview

The contract placed was for the upgrade of the existing:

- Missile system which includes:
  - Upgrade to the existing Ratel missile platform
  - Upgrade to the existing Ratel missile turret
  - Upgrade to the missile weapon system
- Upgrade to the existing ZT3A1 training system
- Upgrade to the existing maintenance recovery vehicle (MRV)

The client included contractual penalty clauses to ensure project performance and schedule. The production contract was excluded from the development contract and would only be placed once the system has been fully qualified and accepted by the client.

4.11 Project Management model

Project management for the weapons system development project consisted of the five project processes (PMBOK 2008) discussed in chapter 3 and applied to the nine knowledge areas identified by Kerzner, (2009).

A project manager with supporting engineering and QA functions headed the customer procurement organisation. A project manager supported by systems engineering, quality assurance, configuration management and procurement headed the contractor organisation.

These two teams formed an Integrated Project team (IPT) for all technical and baseline issues. The IPT met on a regular basis.

Various workgroups with client operational and specialist personnel were formed for detailed client information to facilitate design optimisation e.g. ergonomics, munitions, training, support work groups etc. These groups got together as required.

To manage scope creep, the work groups had to make recommendations to the IPT for final approval.

4.12 Contractor’s Management Model

The contractor organisation was structured on the matrix management organization structure where the different facilities were contracted by the project, Blanchard (1998). The contractor has SAP3®[3] implemented as the management information system to manage labour, finance and material resources as well as for configuration management. For requirements traceability, the contractor used DOORS®[4] for traceability of requirements through all the specifications, (DOORS, 2010). Failure reporting and corrective action system (FRACAS) was implemented more broadly as a problem reporting and corrective action system (PRACAS) since all development/design problems, project problems as well as test failures were recorded on this system. This system is also implemented on the contractor’s SAP3® management information system. Regular project configuration board (PCMB) and failure review board (FRB) meetings were held to approve engineering change proposals and to activate corrective action on the confirmed recorded problems and failures. This PRACAS database was also used for reliability growth evaluation and substantiation during qualification testing. Apart from the comprehensive management information system, the different facilities in the contractor’s organisation were equipped with specialist analytical tools such as Relex®[5], Simulink®[6], Creo®[7].

4.13 Introduction to the Case-study Summary

The aim of this chapter is to provide introduce the case-study project as part of the selected research methodologies discussed above and pave the way for achieving the research objectives and answers to the research questions posed in chapter 1.

This chapter has provided a background and overview of the system to be developed, the high-level customer requirements and the contractor’s organisation management infrastructure. The risk of scope creep has been abated through a dedicated IPT management forum. In the next chapter, the detail development process of the anti-tank missile weapons system will be discussed.

---

[3] SAP is the registered trademark of SAP, Germany, www.sap.com

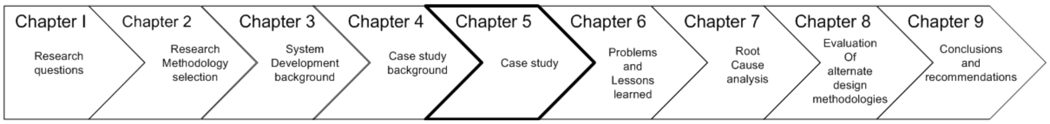
[4] DOORS® is supplied under licence by IBM® Rational® DOORS®; http://www-01.ibm.com/software/awdtools/doors/ (August 2010).

[5] RELEX® is supplied under licence by RELEX Software Corporation, 540 Pellis road, Greensburg, PA 15601, USA.

[6] Simulink is supplied under licence by The MathWorks, Inc; 3 Apple Hill Drive; Natick, Massachusetts 01760 USA.

[7] Creo is supplied under licence by PTC Corporate Headquarters, 140 Kendrick Street, Needham, MA 02494, USA

"A case-study is an intensive analysis of an individual unit (as a person or community) stressing developmental factors in relation to environment."
Merriam-Webster's dictionary (2009)



Development model and development process
Development project objectives
Development strategy
Systems engineering process selection
Development project logistics engineering process
System hand-over

In this chapter the researcher provides the details of the case-study project and systems engineering process.

Chapter 5 CASE-STUDY

The case-study will provide the PRACA data which will be analysed using the Narrative Inquiry research method. The phenomena observed during the case-study project will be analysed using the DSR research method. Using this approach will enable the achievement of the research objectives and provide answers to the research questions posed in chapter 1.

This chapter discusses the case-study for the upgrade of the ZT3 Anti-Tank Weapons System to a third generation anti-tank weapons system.

5.1 Purpose and Outline of the Chapter

The purpose of this chapter is to describe the application of the systems engineering process and the development model used for the case-study anti-tank weapons system project. The IPS development model was the model of choice by the case-study contractor for system development projects. The IPS development model is shown in figure 16 and follows the systems engineering process (Mil-Std-499B, 1994) and technical reviews described in Mil-Std 1621 (1995). In essence it follows a top-down development process until system industrialisation.

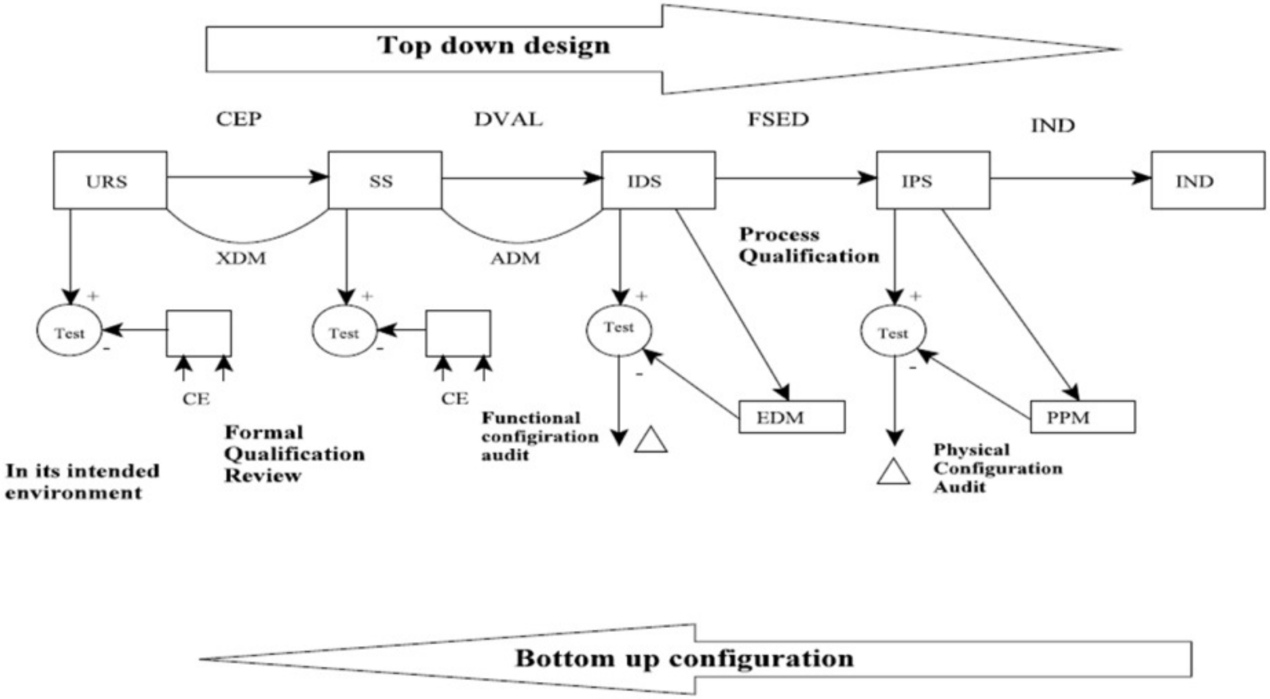


Figure 16: IPS development model

Source: Roos, (2001)

According to Roos (2001), development consists of six phases:

- **Management aspects of development (phase 0).**

  This is the initiation phase of any project and the development model is tailored during this phase.

- **Concept, exploration and definition phase (phase 1).**

  This phase is represented by the transition from the User Requirement Specification (URS) to the System Specification (SS), this phase is called the Concept Evaluation Phase (CEP).

- **A demonstration and validation phase (phase 2).**

  This phase of the development is shown in figure 16 as the Demonstration and Evaluation phase (DVAL). The typical activities during phase 2 are one or more of the following:

  - Detailed simulation and analysis of building blocks of the design.
  - Rapid prototyping of critical parts or high risk circuits where applicable.
  - Building of evaluation test beds and breadboards of part of the design that cannot be simulated.
  - Designing and building of man-machine interfaces that need to be evaluated.

- **Engineering and manufacturing phase (phase 3).**

  This phase is shown in figure 16 as the transition from a System Specification (SS) to an Item Development Specification (IDS), called the Full Scale Engineering Development phase (FSED).During this phase the hardware development starts. The trade-off studies and the simulations should have been completed.

- **A production phase (phase 4).**

  “The primary objective of the production phase is to produce and deliver an effective, fully supported system to the client at an optimal cost” (Roos, 2001).

- **Commissioning and support phase (phase 5).**

  Production items are delivered to the client and support begins with the commissioning of the product and continues throughout the usable life of the product (Roos, 2001).

According to Roos (2001), the IPS model deals with the first four phases of the development since no development should take place during the last two phases. One of the aims of this model was to ensure that no development is necessary during the last two phases (Roos 2001).

This is a convenient model since the client has structured the contract payment milestones according to Mil-Std 1621 (1995) technical review points. The logistics package for the anti-tank weapons system was developed in parallel with the system development process. The logistic system is required to operate and support the new system in the client's environment. This approach necessitated the incorporation into the IPS development model of not only the systems engineering process, but also the logistics engineering process, as well as the interactions between the two processes and finally the development of the logistics infrastructure to support the new system.

The case-study therefore also enables the evaluation of the IPS model and provides enough data to determine whether an adaptation of the IPS model or another model would perhaps have been more successful.

Since the case-study is a large complex multi-discipline system, in order to keep the size within acceptable limits, the focus of the case-study will primarily be on the overall process followed rather than the technical detail. Where applicable, further detail information has been provided in the appendices.

5.2 Development Model and Development Process

The case-study company has very well established infrastructures for the development of complex multi-discipline systems and is equipped with the latest information systems and tools.

Since both the system and the logistic system had to be developed taking full cognisance of the TRAMP criteria, it was decided to follow the IPS model (refer to fig 16) recommended by Roos (2001), as the better model for the development of the 3rd generation anti-tank weapons system. According to Roos, (2001), the IPS development model is a model for development in a multi-disciplinary environment for complex products. It can be tailored for different disciplines. The implementation of small design steps is a contribution towards the development of a new product in a multi-disciplinary development environment. This model underwrites the principles of concurrent engineering but avoids some of the disadvantages, (Roos, 2001).

5.3 Development Project Objectives

Budget and time-scale constraints for the development of the anti-tank weapons system necessitated a development strategy with minimum risk and first time right results. The macro strategic planning by the client identified a requirements window for the anti-tank weapons system. To ensure project performance and schedule any unacceptable cost and schedule overruns could have resulted in the application of contractual penalties and subsequent contract termination, discussed in chapter 4.

The production contract was excluded from the development contract and would only be placed once the system has been fully qualified and accepted by the client, discussed in chapter 4.

5.4 Development Strategy

The system development followed the IPS process for the system and logistic system development. Design influencing was achieved by means of Success Domain (SD) and Failure Domain (FD) teams at all system hierarchy levels discussed in chapter 3. The design teams were divided into Success Domain (SD) and Failure Domain (FD) teams. The SD teams consisted primarily of design engineers and analysts whilst the FD teams consisted primarily of the logistic engineers analysing the ***"-ilities"***[8] of any proposed design from the SD team.

The FD teams specifically also investigated and analysed "*out-of-specification*" behaviour of a design and its influence on the overall system behaviour using fault tree analysis (FTA) and failure mode and criticality analysis (FMECA). The ensured system robustness and safety by orderly system behaviour under abnormal conditions.

5.5 Systems Engineering Process Selection

The system contractual boundaries and hierarchical interface with the client environment for the anti-tank weapons system are shown in fig 17.

---

[8] Any of the engineering "-ilities" (e.g., reliability, testability, producibility, supportability), (NASA, 2010).

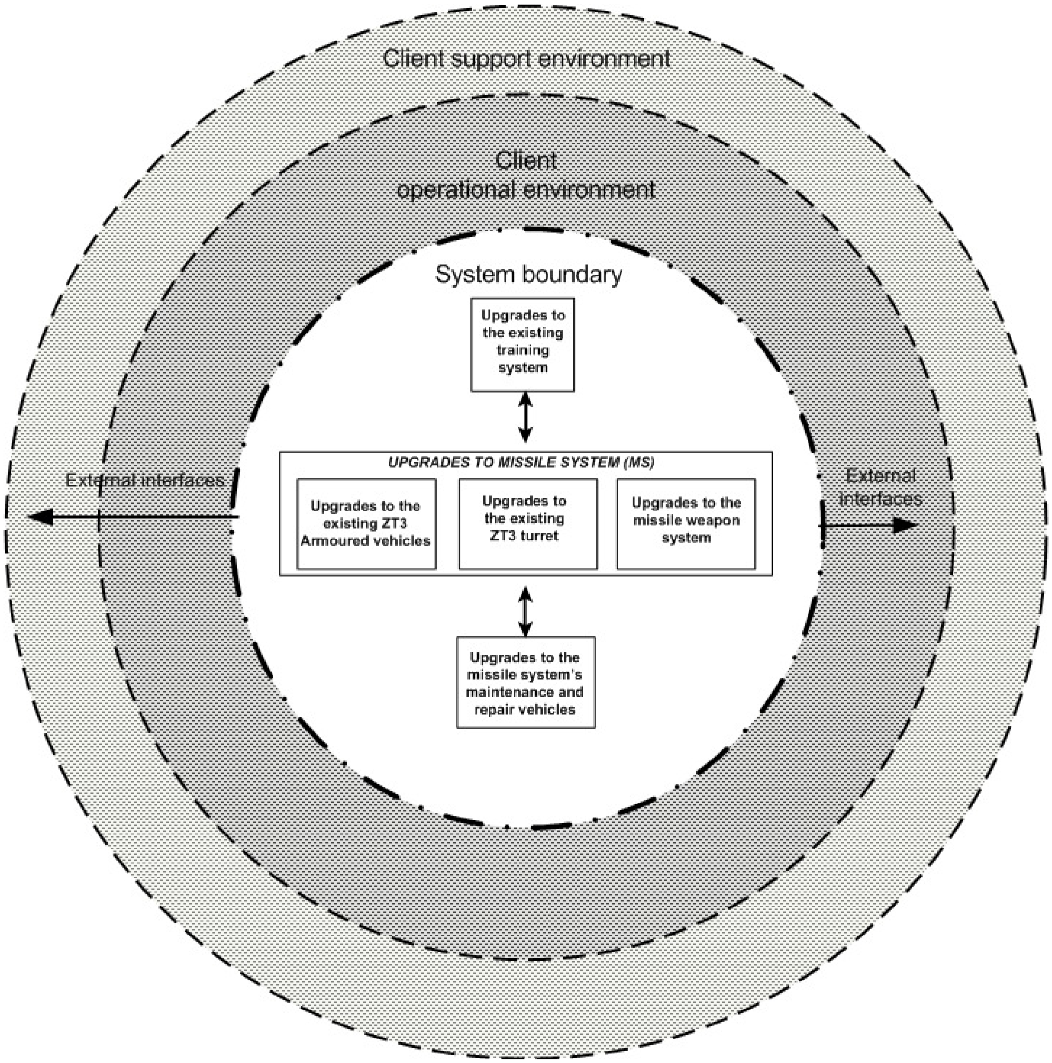


Figure 17: System boundaries and client interface

The upgrade of the existing system and interfaces to the client operational and support environments are shown. The upgrade must also utilise the CFE and comply and constraints invoked by the client as discussed in chapter 4.

The system upgrade consists of:

- Upgrades to the missile system consisting of:
    - Upgrade to the existing ZT3 armoured vehicles

      The Ratel weapons platform must be upgraded to accommodate the new weapons system.

  - Upgrade to existing ZT3 turret

    The Ratel turret must be upgraded to accommodate the new weapons system.

  - Upgrade to the missile weapons system

    The existing anti-tank weapons system must be replaced with the new 3rd generation anti-tank weapons in the existing CFE turret.

- Upgrade to the existing training system

  The existing training and simulator system must be replaced and a new training and simulator for the new anti-tank weapons system must be developed.

- Upgrade to missile system's maintenance and repair vehicles.

  The existing anti-tank missile system maintenance and repair vehicles must be adapted for the new anti-tank weapons system.

5.5.1 Applied Systems Engineering Process

The IPS development model (Roos, 2001) discussed above was used as guide for the systems engineering process for the development of the 3rd generation anti-tank weapons system and associated logistics infrastructure. The specific systems engineering process is shown in figure 18.

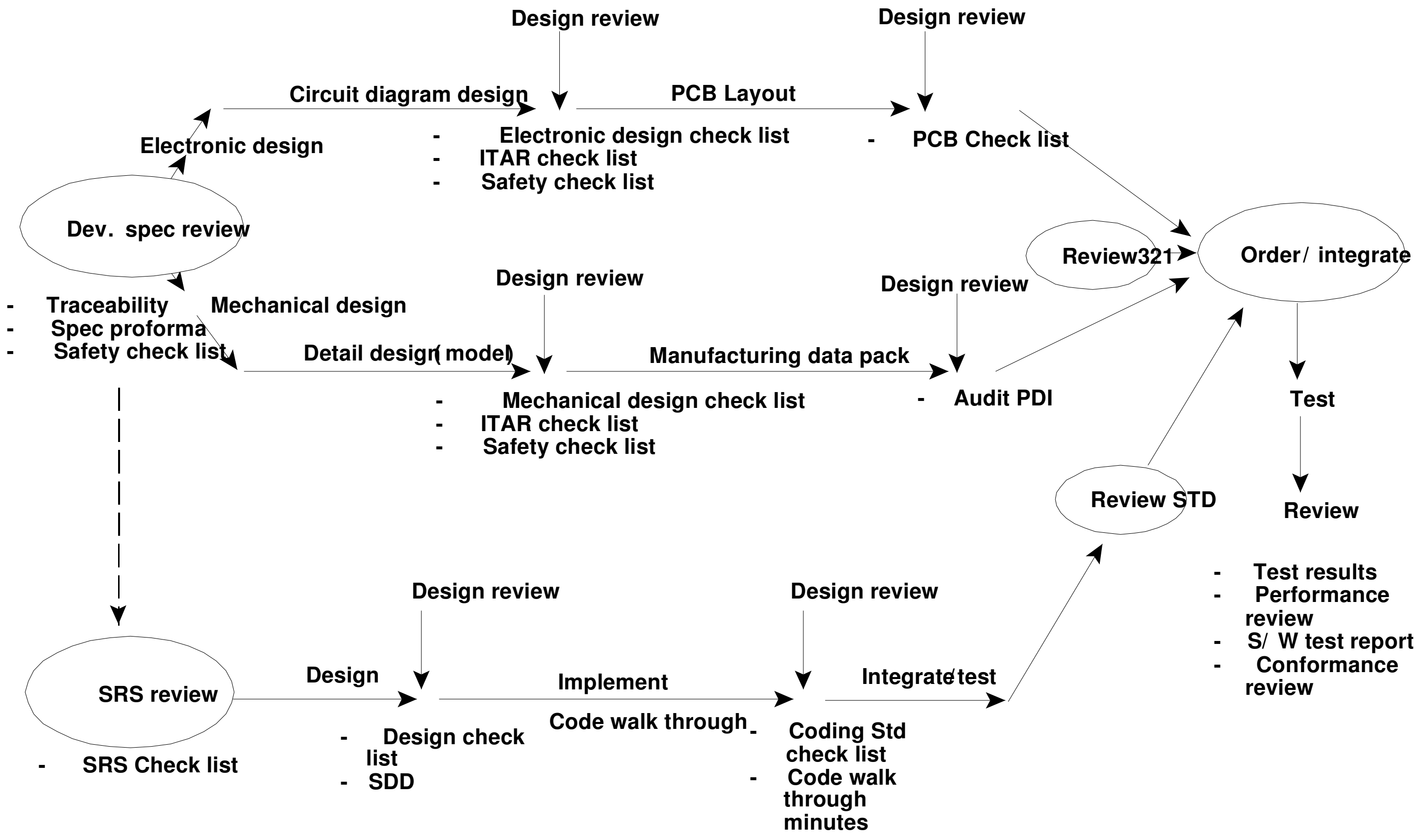


Figure 18: Case-study Systems Engineering process

The fist 4 phases of the IPS development model discussed above applied to the case-study are shown in figure 18. Separate development tracks for the electronic design, mechanical and software design are shown to depict their fundamental differences. Each track starts with a development specification review. The aim of this review is to set the requirements and constraints to the design team. A formal design review follows a completed design before acceptance to the next level of integration. This is followed by the development of a production data pack as part of the systems engineering output discussed in chapter 3. This data pack is also subjected to a formal review before acceptance.

To ensure standardisation and design quality, a project specific set of checklists and design guidelines have been developed to be discussed in the next paragraph.

5.5.2 Design Reviews and Baseline Management

A structured design review format was followed to ensure an orderly and structured method of evaluating subsystem designs against applicable specifications; standards, project value system and that all TRAMP parameters and requirements have been addressed. The aim of TRAMP is to influence the design of the system from as early in the design process as possible to improve the Testability, Reliability, Affordability (unit cost as well as life cycle cost), Maintainability and Producibility. This concurs with the research findings of Maylor et al, (1998) on new product development and the research findings of Lu et al, (2006) with the early inclusion of process design for manufacture.

Design reviews had the following generic format:

- Summary of design requirements and constraints of the applicable system/subsystem or component.
- Applicable standards.
- Trade-off studies and value system used.
- Final design implementation.

To assure quality and uniformity of design standards, the following checklists and guidelines have been developed from past experience and implemented:

- Components identified by the International Traffic in Arms Regulations (ITAR, 2011)
- Mechanical design

- Electronic design
- Digital design
- Printed circuit board design
- Design reliability
- Software requirements
- Safety
- Standardisation guideline
- Built-in Test (BIT) coverage guideline
- Production Readiness review

Once a design has been approved at the critical design review (CDR), all further changes had to be formal by means of Engineering Change proposal (ECP) and approval process by the Project Configuration Management Board (PCMB).

5.5.3 Engineering Change Management

Regular project configuration board meetings were held to evaluate and approve all ECPs. All the stakeholders were represented at the PCMB meetings so that the impact and ripple effect of a change was thoroughly discussed and investigated before final approval.

5.5.4 PRACAS Management

In Chapter 2 the closed loop Problem Reporting and Corrective Action System (PRACAS) was discussed. The case-study development project PRACA was captured on the company's integrated SAP3® management system and provided a database for the subsequent RCA for resolution of the problems encountered on the project. The outcome of the PRACAS is also used to update the design review checklists. This ensures a continuous development process improvement and provides a guide for the new design engineers.

5.5.5 Overview of the Final Evolved System

The anti-tank missile system was completely integrated into the vehicle turret, source: Denel Dynamics Ingwe Missile brochure, (2009). The vehicle platform was kept standard in compliance with the user requirements. Figure 19 shows the basic principle of operation of the beam rider system. The camera system also incorporates a third generation thermal imager camera for night target acquisition.

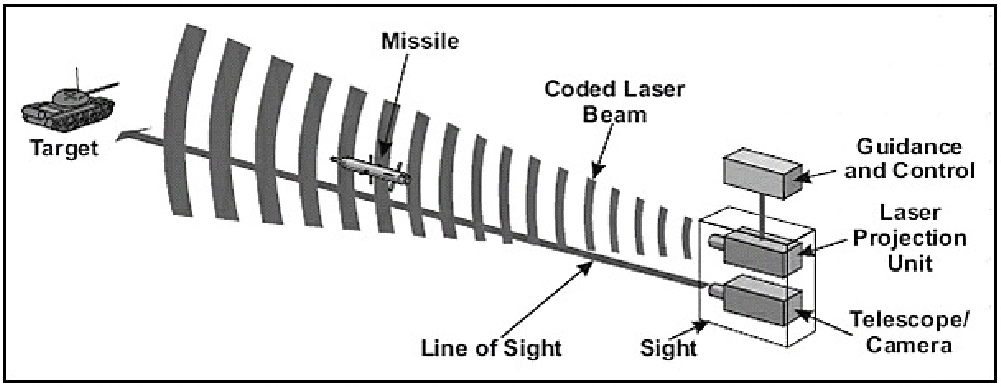


Figure 19: Anti-tank missile system integrated into the ZT3 turret

Source: Denel Dynamics Ingwe Missile brochure, (2009).

5.6 Development Project Logistics Engineering Process

The URS provided the main input driver for the logistics engineering process. The Use study in accordance with Mil-Std-1388 1A, (1990) complemented the URS to provide sufficient resolution of the operational environment for the system. The Integrated Logistic Support Plan (ILSP), provide the strategy (“***WHAT***”) to be followed for the logistics engineering and logistics product development. From the ILSP, the Integrated Support Plan was developed to specify “***HOW***” for the logistics engineering process. From the ILSP, the Logistics Support Analysis Plan using Mil-Std 1388 2B, (1993), as a guide was developed followed by the Product Support Plan detailing how the system has to be supported during the operational phase.

The logistic engineering process used the Mil-Std-1369 (1988) and Mil-Std-1388 (1990) as well as Jones, (1987) as guides. Figure 20 shows the interrelationship between the systems engineering and the logistics systems engineering.

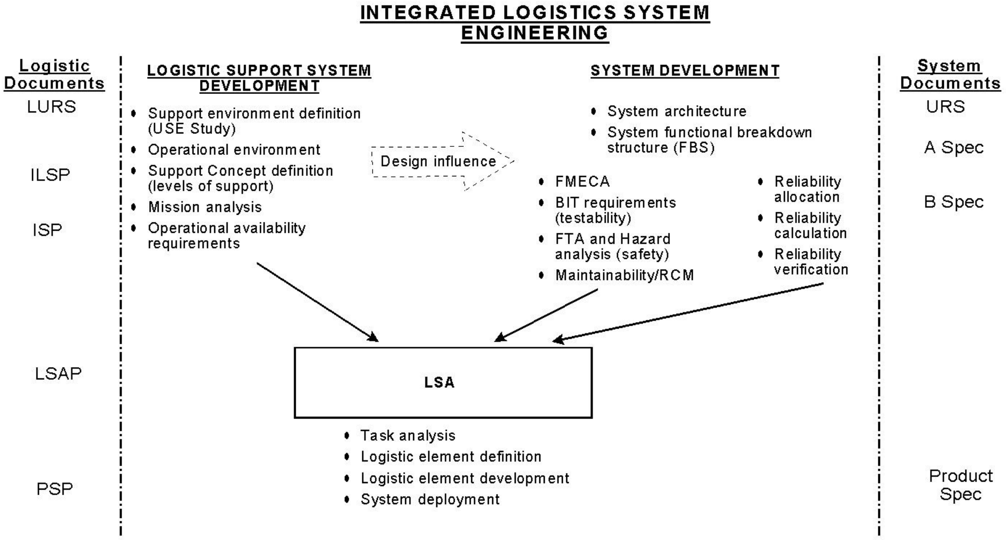


Figure 20: Systems and Logistics engineering interrelationship

The outcomes of the logistics engineering process included:

- Design influencing with specific focus on for all the ***"ilities"***.
- Logistic products development.

The system development team were the SD team whilst the logistics engineering team were the FD design influencing team described above.

As the system architecture and functional breakdown structures (FBS) became available, it was modelled in RELEX®[9]. From the FBS the Reliability Block Diagram (RBD) was developed followed by reliability modelling. The top down Fault Tree analysis was followed by the bottom-up FMECA process, source: Reliability Practitioner's Guide, (2003).

Maintainability was analysed using Mil-Hdbk-472, (1966), as a guide. The relatively low technical skill levels of the operators and first line maintenance personnel required that extensive Built-in Test Equipment (BITE) be developed. The requirements was for a Built-in Test (BIT) coverage >80% with a better than 90% confidence level.

The logistics engineering team established and maintained a standardised terminology list under configuration control to ensure

---

[9] RELEX® is supplied under licence by RELEX Software Corporation, 540 Pellis road, Greensburg, PA 15601, USA.

terminology standardisation amongst all development and logistic products documentation. Any additions to this list also had to be approved by the client. The logistics engineering team also focussed on availability optimisation, level of repair analysis (LORA), Interchangeability and standardisation, Bill of materials (BOM), obsolescence, human engineering and life cycle cost (LCC).

The logistics products development project was responsible for:

- The requirements specifications for the development of support test equipment at the different levels of support in accordance with the LORA.

- Operating and support documentation development.

  The logistic support technical writers were responsible for the development of the support documentation suite.

- Training simulator and training material development.

  The logistics-training specialists were responsible for the requirements specification of the training simulator, the development of the training curricula and training material as well as the initial training of the client's training instructor personnel.

- The logistic support analysis record (LSAR) development and subsequent integration into the client's management information system (MIS).

The administrative support team consisted of:

- Quality assurance (QA)

  The QA manager is responsible for all project QA activities in accordance with the QA plan and liaison with the client QA manager.

- Configuration management (CM)

  The CM is responsible for the administration of all engineering change proposals ECPs and scheduling of project configuration management board (PCMB) meetings in accordance with the project configuration management plan.

- Procurement

  The procurement manager is responsible for parts and material procurement as well as sub contracting of component manufactures, the administrative BOM and procurement specification management.

- Safety

  The safety manager is responsible for corporate and client safety compliance and liaison with the client safety manager.

Summarising, the logistics engineering process follows the systems engineering process described by the IPS model (figure 16), since analysing a design from a logistics engineering perspective can only occur once a coherent design is available. The logistics engineering process to support the system first analysed the designs with two objectives:

- Design influencing
- Establish the requirements for the development of the logistic products.

From the logistic product requirements, the logistic products such as operator- and maintenance manuals as well as training courses were then developed using the IPS model discussed above.

5.7 System Hand-Over

The final system hand over and Integration into the Client's Inventory prior to the closure of the contract was done in phases:

- Engineering Test and Evaluation (ET&E) performed by the development engineering team with the client personnel observing.
- Operational Test and Evaluation (OT&E) performed by contractor personnel with the client personnel observing.
- Qualification Test and Evaluation performed by fully trained client personnel with the contractor observing and providing limited technical support.

Once these phases have been successfully completed and all outstanding items closed will the system be formally incorporated into the client's inventory and the project closed.

5.8 Chapter Summary

This chapter provided a description of the systems engineering process, followed in the development of the ZT3 Anti-Tank weapons system, using the IPS development model recommended from the research by Roos (2001) for the development of a full-scale complex multi-discipline weapons system, (figure 16). The logistic products were also developed using the IPS development model described above.

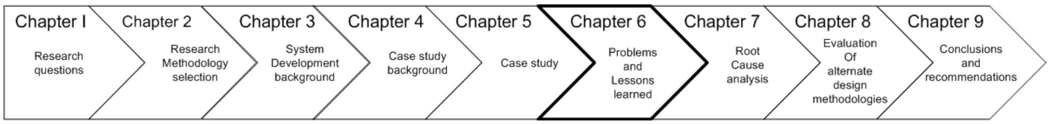
The size and multi-discipline nature of the project necessitated further refinements, to ensure efficient and effective group interaction during design reviews, by dividing the development team into a Success Domain (SD) team with the objective of getting the system working in accordance with the specifications, and a Failure Domain (FD) team with the objective of finding and eliminating weaknesses in the evolved designs, and to ensure that all the "***-ility***" objectives and requirements have been achieved.

Logistic engineering analysis of the designs was used to firstly influence the design and once the design is accepted, as requirement for development of logistic products. The logistic there functioned as the FD team for design influencing. This will be further discussed in chapter 7.

The extensive use of checklists completed by the individual design engineers, and distributed as part of their design review documentation prior to a design review substantially facilitated the design review effectiveness and time management. These checklists were continually updated from PRACAS database to prevent recurrence of a problem and to provide a learning curve for the other design teams. This closed the corrective action loop by preventing recurrence of a problem discussed in chapter 2. The checklists also served as a very effective guide for the lesser-experienced design engineers on the team.

In the next chapter the problems experienced with the development process and lessons learned will be discussed.

"From this paper it would perhaps be easy to conclude that it is hard (and probably impossible) to design the perfect evaluation. In retrospect, there is always something else that could impact the results."
Nava Tintarev UMAP (2009)



Evaluation of the case-study and problems experienced
Grouping and quantifying of the problems experienced
Evaluation of analysis results
Causes of problems encountered

Chapter 6 PROBLEMS EXPERIENCED AND LESSONS LEARNED

The purpose of this chapter is to evaluate the case-study project and identify the fundamental root causes for the problems experienced. The objective being to find answers to the research questions posed in chapter 1. Discussed in Chapter 2, applying the DSR research methodology, the PRACA data collected will first be analysed using the Narrative Inquiry research method to reveal the symptoms of the problems observed. According to Clandinin et al, (2000), the Narrative Inquiry is a powerful tool in the transfer and sharing of knowledge in a work group environment. Since the primary research objective was to get a better understanding of the project failure phenomena as well as the factors at play, the Narrative Inquiry qualitative research approach was performed on the project PRACA data. On completion of the project, the team unanimously agreed to invest time for a project review work session to review the problems experienced applying the Narrative Inquiry research methodology. The results from this work session will then be further analysed using RCA to identify the fundamental root causes of the phenomena observed in accordance with the DSR research methodology.

It is a general fact that projects often over-spend and deliver late. Steyn, (2009), shows by means of a simplistic cause-and-effect diagram some of the factors at play that may result in a project schedule over-run. Steyn (2009) points out that corporate politics and hidden agendas have a detrimental influence on project performance. Corporate political factors, quite often, have a negative influence on the results produced by case studies, disturbing the parameters to be analysed. This makes analysing the results difficult and in a lot of cases makes the isolation of the real root causes of cost and schedule over-runs very difficult.

The long standing trust between team members and management negated the negative influence of corporate politics facilitating more accurate isolation and analysis of the root causes of the problems experienced on the project.

6.1 Review of the Case-study

In chapter 5 the background to the case-study for the development of the ZT3 Anti-Tank weapons system and the systems engineering process using the IPS development model was discussed. This specific case-study was chosen in that the development project ran continuously from start to finish with the same team. The author was a major player on the team from the beginning and familiar with the background issues of the project and could provide invaluable DSR inputs.

Another factor for selecting this specific case-study was that the major team players, the client, the procurement agency and the contractor were very experienced. They have been involved in past projects as a team. They were keen cyclists belonging to the same team and often went as a team to cycling events. There was a strong camaraderie and trust between all players, very seldom found on commercial projects. This was evident at project meetings where honest reporting was done without any attempts to move the blame but rather a strong focus on how to correct and overcome problems. When a team member slipped up there was no witch hunting, rather support was given to assist and prevent a recurrence. Open and honest reporting had a very positive effect on the team resulting in a positive work culture as well as mitigating the negative influence of corporate politics.

The case-study had all the ingredients of being a successful project using the IPS development model recommended by Roos, (2001). There was total commitment and co-operation between all the players. Additionally, the different subsystem development teams were divided into SD and FD teams discussed in chapter 3. The Success Domain and Failure Domain participants at design reviews proved to be very effective and productive. It was particularly at preliminary design reviews (PDR) that system and subsystem behaviour during out-of-specification conditions was thrashed out and agreed upon. The out-of-specification behaviour, also sometimes referred to in the literature as negative scenarios, (Alexander et al, 2009), added to the robustness of the design. These conditions were incorporated into the development-, test- and qualification specifications. Qualification tests also involved TAAF testing for reliability growth in accordance with the guidelines of Mil-Hdbk-189, (1981).

Client specialist operational personnel were actively involved in work groups with the contractor development personnel addressing specialists' areas such as ergonomics, munitions and Command and Control, (Alberts et al, 2006). The client operational personnel valued the new anti-tank weapons system as being very successful, complying with all their requirements and expectations. Yet as a project, the project failed since the project was over cost and schedule.

At the Production Readiness Review, the disappointed team felt that there must be deeper fundamental factors at play resulting in the cost and schedule overruns.

Since all the project resource data as well as the PRACAS were integrated into the contractor's SAP3® management information system and available on the company Intranet, a full day work session was held in a conference room, complete with laptop PC, LAN connection and overhead projector with all the major stakeholders present, enabling instant access to any project and system data during the discussions.

Discussed in chapter 1, the objectives of this research are:

- Optimisation of design influencing by dividing the design teams into two different mindset groups.
- Evaluate the impact of design changes in terms of cost and schedule overruns in a concurrent engineering development environment.

The first step of achieving the research objectives was to hold a narrative inquiry discussed in chapter 2. The participants of this team formed a structured focus group. The outcomes of this work session were then verified using the Delphi Technique, (Hsu, 2007), and triangulation, (Greene, 2007), to confirm the research facts and data until consensus between all the participants were reached.

The group consisted of the following participants:

- System and subsystem engineers (8)
- Logistic and reliability engineers (6)
- Technical authors (4)
- Procurement (1)
- Configuration (1)
- Quality assurance (1)
- Project management (1)

Most of the engineers and project team members had post graduate qualifications in their respective fields. The technical authors were specially trained technologists with years of shop floor and field experience. The author was the facilitator.

All participants were emailed a copy of NAVSO P-6071, (1986), in preparation for the work session to review the traps applicable to their specific area and to what extent these have been mitigated and avoided.

6.2 Grouping and Quantification of the Problems Experienced

The PRACA items were analysed during the structured focus group work session using qualitative research methods, (Morgan, 1997), getting as many points of view, as possible, from all members of the group. The grouping and quantification of the PRACA items were then cross examined using triangulation methodology, (Greene, 2007), to ensure confidence in the findings. The PRACA items relating to specific PM-SE engineering problems were further investigated by the iterative Delphi Technique, (Hsu, 2007). A number of problems impacted in more than one category. In that case the value allocated would be the impact to that specific category. In other words it is possible that the same problem be allocated different values under different categories.

The problems experienced on the project were quantified using a five-point Likert scale (Likert, 1932) to determine the impact on the project:

- 0 = no effect/not applicable
- 1 = not important
- 2 = slightly important
- 3 = important
- 4 = very important
- 5 = crucial for project success

Guided by the research objectives and research questions, the work group categorized problems experienced into the following categories:

- Management related problems
- Systems Engineering related problems
- Quality Assurance (QA) related problems

- Configuration management (CM) related problems

6.3 Evaluation of the Analysis Results

The Narrative Inquiry structured focus group was unanimous that the IPS model was a very effective development model. The recorded PRACA items were then subjected to critical review by the structured focus group applying the Narrative Inquiry methodology. It was found that in total 61 recorded PRACA items relevant to this research were identified and analysed and subsequently quantified. Those PRACA items not relevant to this research were ignored. The total Likert scale value of the 61 analysed PRACA items was 176. A breakdown and summary of the problems experienced and allocated Likert scale values have been provided in Appendix B. The summary of the findings are shown in the table 21:

Problem area	**Consolidated incidents**	**Total impact**	**average %**
Management related project problems	32	100	56.8%
System Engineering design related project problems	12	35	19.9%
Quality Assurance (QA) related project problems	7	15	8.5%
Configuration management (CM) related project problems	10	26	14.8%
Total	**61**	**176**	

Figure 21: Summary of problems experienced in the case-study

In general more problems were reported by team members further downstream in the systems engineering process, such as the logistics product development personnel and in particular the technical authors responsible for the development of the operational manuals, support manuals and training material. This was to be expected since deficiencies upstream in the systems engineering process are generally more likely to manifest themselves further downstream.

A detailed problem summary obtained from this research has been provided in appendix B.

6.3.1 Management Related Project Problems

From the analysis, it was found that 32 PRACA items with a total impact value of 100, equating to 56.8% of the problems experienced on the project, was apportioned to management.

The management related project problems were primarily due to:

- Client requirements - baseline shift.
- Matrix organisational structure – unavailability of resources and fragmented indirect project functions.
- Project management and schedule – unexpected post CDR design changes.

Detail problems encountered and explanations are discussed in the next paragraphs.

6.3.1.1 Client Requirements – baseline shift

The client requirements baseline changed during the project due to miss interpretation of requirements and client requested change.

The procurement agency systems engineer has drawn up a very comprehensive client requirements specification. This was augmented with a number of comprehensive associated documents in specialists' areas such as logistic support environment and training. The contractor at the planning and contracting phase mistakenly assumed that the existing ZT3 operational and support manuals as well as the training material could be adapted for the newly developed system in order to achieve a cost and schedule saving.

This has proven to be wrong since the client has revised his technical manual and training standards in the interim. This oversight resulted in a costly re-development of these documents. This problem was further exacerbated by the lack of internal detail knowledge of the new standards. This resulted in an unusually large number of problems reported by the technical authors.

Also as the system evolved the client subtly exploited opportunities to include additional changes to existing requirements. The ripple effects of these were sometimes underestimated leading to cost and schedule overruns on the affected elements of the project.

In conclusion, the lesson learned is that missing a requirement or a constraint (non-functional requirement, (Sommerville, 1996), invariably leads to re-work and wasted resource effort.

This often impact negatively on the overall project schedule and cost. Also it is important that any uncertainties and assumptions are tested with the client up-front prior to the start of the project.

6.3.1.2 Matrix Organisational Structure Related Problems

(Unavailability of resources and fragmented indirect functions)

In a matrix organisational structure (Blanchard, 1998), development resources are not available on-demand since they are re-allocated to other projects on completion of their specific milestone on the project. Also indirect functions that are not directly coupled to project milestones are considered as part of the company overhead and fall under the relevant facility structures resulting in fragmentation of these functions. The detailed consequences are discussed in the next paragraphs.

- **Unavailability of on-demand resources**

Under the Matrix organisational structure, once a specialist resource has completed his task on the weapons system development project, the resource is allocated to new tasks on other projects. The consequence is that the resource is not immediately available should his expertise be required later to solve latent problems.

Developing a complex weapons system demands a large number of multi-disciplined specialists' skills. These expert skilled resources are only required for relatively short periods of time in the total development project schedule. As discussed in chapter 3, the downside of the matrix organisational structure, apart for the large internal subcontracting workload for the project manager, is the unavailability of on-demand expert specialist resources.

These scarce resources are immediately re-deployed by the facilities to other projects. This places a severe handicap on the schedules of a development project should an expert resource be required for say the identifying and analysis of a root cause failure and the subsequent development of an engineering change to correct the problem.

- **Fragmented indirect project functions**

The internal company rule mandated that only direct personnel could work on project structures since all resource expenditure must be coupled to payment milestones.

The company management information system SAP3® is so structured to enforce this rule. The consequence of this rule is that the project's indirect functions such as configuration management, procurement and quality assurance at subsystem level are performed by facilities and as such becomes fragmented from a project point of view.

The PRACAS identified this problem fairly early in the project resulting in top management amending this rule. To alleviate the problem, configuration, quality assurance and procurement personnel were seconded to the project. Particularly the seconded procurement personnel enabled bypassing the corporate production procurement organisation expediting the procurement and delivery of small quantities of critical engineering components for development models.

Facilities in a matrix organisation generally provide resources for a number of projects, making it impractical to cater for the individual requirements of a specific project. This is a fundament impediment particularly with larger facilities. On the other hand workload variations cannot be handled effectively by smaller facilities.

The matrix organisation resulted in:

- Fragmented indirect functions such as QA, CM and procurement.
- Lack of specialist design expertise availability after the CDR of a specific configuration item.
- Large internal subcontracting workload.

In general very good interpersonal relations between project managers and facility managers can mitigate these deficiencies of the matrix organisational structure. However when it comes to subcontracting to other business units and outside subcontractors this may not always be possible.

6.3.1.3 Project Management and Schedules

(Unexpected post CDR design changes)

Project management decided to implement the critical chain principles and provision of schedule buffers to prevent bottlenecks (Goldratt, (1997). This project management approach was selected due to the large number of internal and external subcontracting on the project. The large number of internal and external subcontracting was primarily as a result of the diverse nature of all the disciplines required for the development of the weapons system.

It was anticipated that the critical chain approach with its inherent buffers would reduce the project bottleneck risks that could result in schedule and cost overruns. Fundamentally this worked very well right up to the CDRs of the individual configuration items for the weapons system. In general at that stage the individual elements of the project were on schedule and within cost and no project schedule and cost risk was foreseen until the integration level of the items.

The project cost and schedule problems started to manifest themselves at the integration level where unforeseen problems forced post CDR modifications on certain configuration items. The unforeseen problems continued at each subsequent level of integration. Although the problems reported were relatively few given the complexity and multi-disciplined nature of the weapons system the effect on cost and schedule proved to be detrimental on the project since they were not planned for.

The contractor organisation configuration management system uses three categories to control the status of configuration items by using Mil-Hdbk 61, (1997) as a guide. To facilitate service-life tracking, these components normally have their own serial and revision numbers and are referred to as Configuration Items (CI) in the military industry.

At subsystem level the classifications are defined as follows:

- **Class I**

  A class I modification is a modification where the Form-Fit-and-Function (FFF) of the configuration item is affected. In other words an interface or performance characteristic of the affected configuration item is changed. This also results in part number and/or revision number changes and interchangeability is affected, (Mil-Hdbk 61, 1997).

- **Class II**

  A class II modification is one that does not affect the FFF and performance and is contained in its entirety internal to the configuration item. Generally this only results in a revision change, (Mil-Hdbk 61, 1997).

- **Rework**

  A rework is not a modification issue but rather a quality issue where an item during test did not conform to specification and had to be reworked to bring it back to specification. The part number or revision of the affected item is not changed. Rework manifests mostly during pre-CDR in the development phase and during production although sometimes items also fail during integration testing.

Class I and Class II changes apply to the data pack of the affected configuration item and rework applies to a physical hardware item. Modifications and rework are unplanned activities and consume resources and as such have a detrimental impact on cost and schedule of a project.

During development particularly at integration level, most of the changes were generally class I where the FFF of an item is affected. This causes a ripple effect throughout the system structure mainly due to functional couplings and emergent properties of the different levels in the system hierarchy. This is further exacerbated in a concurrent engineering environment due to other system components that are concurrently under development. This generally has a negative effect on cost and schedule since not only the affected item but also the functionally related items in the system hierarchy must be modified.

To reduce this ripple effect it was sometimes decided during the Project Configuration Board (PCMB) reviews to overcome a problem that the root cause item itself is not modified but rather a lesser impact item is modified provided the overall system dynamic performance as described by Viljoen, (2007), is not affected. This is sometimes called a "*band-aid*" fix. The technical authors being further downstream in the systems engineering process developing the operating, support and training material were mostly affected by these changes. It is also in this area where most of the schedule and cost overruns occurred.

In conclusion, the integration problems experienced on this development project generally occurred on the interfaces between configuration items and was seldom due to the configuration item itself. This may be attributed to the extensive pre-CDR qualification testing of configuration items making it highly unlikely for latent design defects remaining in the configuration item itself.

The cost and schedule overrun on the project, can to a very large extent, be ascribed in order of priority to:

- Modification ripple effects of functionally coupled configurations items in the system structure.

- Specialist expert resource availability as a result of the matrix organization structure.

6.3.2 Systems Engineering Related Problems

The systems engineering related problems were grouped and collated to 12 events with an average impact value of 2.92 (19.9%).

The systems engineering problems can be summarised into:

- Specialist and subsystem expert resource availability after the CDR of the relevant configuration item.
- System and CFE data integrity and availability.
- Standardized terminology.

6.3.2.1 Specialist Resource Availability

Discussed in chapter 3, the matrix organisation structure resulted in resource unavailability after completion of the design and CDR.

To overcome this problem, a small percentage of specialist time was contracted with facility managements and subcontractors for consultation and technical assistance purposes during the integration and testing phases at system level. On paper this sounded like a good solution but in practice proved to be almost unworkable because of the pressure of work by other projects for the specialists' resources.

6.3.2.2 System Data Availability and Data Integrity

The customer's furnished items and associated data packs were accepted at face value from the client. Subsequent audits further downstream in the project revealed a number of deficiencies that resulted in avoidable engineering changes and rework particularly by the technical authors.

The main problem that stood out however was that the current established systems engineering process described by INCOSE, (2010), NASA System Engineering Manual, (2007) and Mil-Std-499B, (1994) focuses from the Customer Needs through all the specification levels to the final product specifications on the "***WHAT*** ". In other words "***what***" is the system/subsystem required to do to the final confirmation after qualification testing that the envisaged system indeed does "***what***" it is required to do. The total focus for the system data pack development is on the requirements and requirements traceability.

The processes described by Alexander et al, (2009), how to discover and specify requirements have been used on this project. In complex multi-functional systems such as the ZT3 weapons system upgrade, requirements traceability tools such as DOORS® have been used to ensure that all requirements trace down to the lowest level specifications and documents in the systems hierarchy to finally facilitate system qualification and handing over to the client.

Nowhere is there a requirement for the system, subsystem and design engineers to produce formal documents that describes "***HOW***" the particular design works. It can be seen in figures 18 and 20 chapter 5, the formal document structure as part of the systems engineering process followed only specifies the "***WHAT***". Informally the "***HOW***" information was generally provided during the design reviews as part of the particular design engineer's presentations but then at a relatively high level unless forced to provide more details by the questions and answers of the attending participants.

The attending participants at design reviews were stakeholders actively involved at that particular phase of the systems engineering process. Stakeholders further downstream of the systems engineering process such as the technical authors were not represented and this impacted negatively on early design influencing.

The technical authors developing the operational and support manuals and training material had access via the company intranet to the project configuration system and full system data pack. However they found that they still could not perform their tasks since the vitally needed "***HOW***" information for the development of the technical manuals and training material were not available. This problem was further exacerbated by the fact that they generally did not have the detail engineering skills and knowledge to analyse the designs and deduce "***HOW***" the designs work without assistance from the experts. Also the unavailability of engineering expertise for consultation and redlining of the draft manuals aggravated the situation. The Failure Domain team to a large extent assisted the technical authors since they had already analysed the designs and had a good grasp of the "***HOW***" a specific design worked.

In order to provide a more formal work around the lack of "***HOW***" data, project management contracted design engineers to produce formal Storyboard Descriptions describing how their particular designs worked, (Wikipedia 2009). This worked very well but since it was unforeseen and unplanned, it impacted negatively on the project cost and schedule.

6.3.2.3 Standardised Terminology

Lack of early and initial terminology standardisation resulted in a lot of rework particularly by the technical authors. As the project progressed, new items were identified and named initially by the designer. These names were later amended or changed as other stakeholders became involved and felt that their naming of a particular item was more appropriate. This naming evolution was carried through further down the systems engineering process. Finally the shop floor personnel and client operational and support personnel, who have to live with the name for the product lifecycle, decided on more practical names from their point of view. The stores personnel on the other hand had their own unique naming nomenclature and standards prescribed in Standard No. 5F, (1982), and the Cataloguing Handbook, (1988), adding to the problem.

This item name evolution process during the total system design, apart from the confusion in understanding the different level of documents, resulted in a lot of rework by the technical authors particularly in documents such as the illustrated parts catalogue (IPC).

This problem was identified by the PRACAS fairly early in the project and it was decided to initiate a standardised terminology list that also had to be approved by the client.

This reduced the problem to certain extent but did not completely resolved it, since ideas of what a specific item should be called changed as the development process progressed, and different levels of people became involved and more familiar with the specific item. This resulted in a number of ECPs and their associated ripple effects.

From this experience it was agreed by the team that a terminology list, instead of being a dictionary of names, should be a thesaurus of synonym names that also identifies the final name to be used in the bill of materials (BOM) and Illustrated Parts Catalogue (IPC).

6.3.3 QA and CM Related Problems

The Quality Assurance (QA) and Configuration Management (CM) related problems were grouped and collated to 7 events with an average impact value of 2.14 (8.5%) and 10 (14.8%) respectively.

The main problems in these two areas originated from the fragmented functions as a result of the company management rule that QA and CM are considered indirect functions. Discussed above under matrix organisational structure related problems, these functions fall under the different facilities of the matrix organisational structure.

The seconding of QA and CM personnel to the project to a large extent mitigated this problem.

6.3.4 Development Process

The Integrated Product Support (IPS) model was used for the development of the upgrade of the ZT3 anti-tank weapons system. Key goals of the IPS design model according to Roos, (2001), are design maturity at the end of the development phase, design verification, low risk transition from development to production, and high production quality.

From past design review experience in the company, the author introduced a formal design review guideline to ensure keeping focus on the big picture and to prevent the very common side tracking on detailed technical issues. It was found that generally, depending on the designer’s experience, primarily due to their Success Domain mindset, they very often did not fully understand the full extent of the design problems and requirements. To facilitate and ensure consistent standards and quality, the use of checklists, discussed earlier, was introduced. This gave the designers a better idea of what was required.

Design reviews were held in two stages. A preliminary design review (PDR) and a critical design review (CDR). The purpose of the PDR was to ensure that the requirements of the particular development specification were fully understood by the tasked design team whilst the purpose of the CDR is to ensure compliance with all the requirements.

With the focus on design influencing early on in the project, the Failure Domain team’s presentations of the development specifications at the PDR design reviews highlighted and emphasized what was really required from a particular design. The primary objective of the CDR was to ensure that the particular final design complied with all the requirements. The design review guideline can be summarized into the following main categories:

- Specification and associated documents.
- Requirements and constraints.
- Value systems of both the client and company.
- Trade-offs.
- Technical solutions.
- Design review checklists.

The opposing mindsets of the SD and FD design teams proved to be very effective and efficient as predicted by Kuhn et al, (2006). The opposing viewpoints from the SD and FD design review team members led to very constructive and thorough discussion with optimal outcome of the final solution.

The ECP database in SAP3® confirmed not a single design, apart from interface design changes during integration, had to be extensively modified or redesigned. This process also proved to be particularly valuable for software requirements document (SRD) and software design document (SDD).

6.4 The Causes of the Problems Encountered

The observation, analysis, understanding and finding of a solution to a phenomenon observed is the objective of DSR discussed in chapter 2. The Narrative Inquiry research findings show that the two main problem areas are Management and Systems Engineering. Unplanned design changes appear to cause conflict with project management and lack of suitable data in the design data packs appear to cause problems with the logistic system development team. Appendix B provides a detailed breakdown of the problems experienced.

The Project Management and Systems Engineering categories are not independent. It can be simplistically argued that without project management there would not have been systems engineering and system development. Likewise without systems engineering there would not have been a system development project. It can therefore be construed that the findings in table 21 are the symptoms of deeper underlying problems.

The question that remains now is "***why***" were these problems experienced on the case-study project? What are the fundamental underlying causes? The approach taken on the case-study project was analysing the problems experienced and then asking the question "***why***" these manifested. This approach is also supported by Seusy, (1988). The causes of the problems experienced can be grouped into the following two categories:

- **Project management and the systems engineering processes have areas of incompatibility**

  Project management is in essence a structured sequential milestone driven process with a beginning and an end, definitions abound. The PMBOK, (2008) definition has been discussed in chapter 3. The most appropriate definition applicable to this research can be found by Rasmussen, (2009) "*Project Management is a formalised and structured method of managing change in a rigorous manner. It focuses on achieving specifically defined outputs that are to be achieved by a certain time, to a defined quality and with a given level of resources so that planned outcomes are achieved*" Project management is in essence a rigidly structured sequential milestone driven process until the end goal is achieved and the project completed.

  Systems engineering on the other hand is defined by Eisner as: "*Systems engineering is an **iterative** process of top-down **synthesis**, development, and **operation** of a real-world system that satisfies, in a **near optimal** manner, the full range of requirements for the system,*" (Eisner, 2002). Eisner (2002), places specific emphasis on "***iterative***", "***synthesis***", "***operation***", "***near optimal***" and "*satisfies the system requirements*". Eisner, (2002), further states that the design and building of a system involves several loops of iteration such as "*synthesis-to-analysis"*, "*concept-to-development"* and "*architecting-to-detailed design"*.

  The project manager is measured against project performance primarily in terms of cost and schedule. The systems engineer on the other hand is measured in terms of system performance and compliance with requirements. The project manager and the systems engineer are measured differently and are subjected to different performance metrics. As such the project manager and the systems engineer have conflicting value systems.

  Indeterminate events such as iterations, re-work etc. are classed as risks in project management and a certain amount of resources are normally allocated to a project risk pool based primarily on the project manager's previous experience.

  The systems engineer and his design team (Success Domain) iterate from concept to design in order to find the optimum solution. Once a coherent design is available can this design be analysed by the logistics engineering team (Failure Domain), (Eisner, 2002). The outcome of this process invariably will result in another iteration loop of design. The number of design loops is in relation to the technological complexity and risk of the design. State-of-the-art designs invariably involve high technological risks. The systems engineer is forced by the client's requirements into these high risk areas particularly if his system solution is to be effective and competitive.

In conclusion, the structured "*milestone-by-milestone"* project management process cannot effectively accommodate the indeterminate iterations of the systems engineering process. Once a milestone has been completed, project management cannot accommodate a revisit to that milestone unless it was anticipated and planned for under a new milestone. The proposal by Grundy (1998) for PM to implement cycles of deliberate and emergent change as opposed to linear strategy development has the potential to alleviate some of the problems experienced.

According to Goldratt, (1997), a manager behaves in accordance with how he is measured. Since the project manager and the systems engineer are measured against different and opposing performance criteria, it can be deduced that areas of project management and systems engineering processes are in conflict. For the smooth and efficient running of a complex systems development project, soft systems methodology must be used as described by Checkland, (2001). Such a process is difficult to quantify and measure and depends to a large extent on the cooperation and team spirit of the individual members of the development team, (Checkland, 2001).

- **Systems Engineering primarily develop "*WHAT*" data and insufficient "*HOW*" data**

The current systems engineering process and data pack development primarily focuses on the "***What***" and generally does not provide sufficient data on "***How***" the system, subsystems and designs work.

The systems engineering process is entirely requirements driven from the customer's needs right through to the final product specification, (INCOSE, 2010), (Mil-Std 499B, 1994) and (Military Standard 1521, 1995). All the formal documentation produced addresses and describes the "***WHAT***" and how to build and test the particular configuration item. No formal system documentation as part of the development model describes "***HOW***" a particular design works.

This qualitative "***HOW***" data is not available from the formal configuration controlled design data pack. The "***HOW***" design information is crucial for the following reasons:

- **Development of operational and maintenance manuals and associated training material**

  Operational and maintenance personnel must have a clear understanding how the system and the particular subsystem work in order to be able to optimally operate, deploy and apply the system. This information is a requirement for the operator and maintenance manuals. This is also a vital input for the development of the training systems. In essence the "***How***" data becomes part of the URS for the training system development.

- **System diagnostics and maintenance**

  In order to be able to efficiently and effectively identify, diagnose and localize system malfunctions, maintenance personnel must have a deeper level of understanding of the system architecture, interfaces and in particular "***How***" a subsystem and component performs their functions.

- **Through-life engineering support**

  Military systems typically have an operational life of 20 years. During this time the system may need modifications and upgrades of subsystems and components for various reasons such as field problems experienced, obsolescence etc. A different design engineer must be able to analyze and develop a modification for the affected subsystem or component. To this effect apart from the "***WHAT***" data, he also needs the "***HOW***" information, in particular the classification of characteristics (CoC) and rationale of the design as described in DoD-Std-2101, (1979).

6.5 Chapter Summary

In this chapter the problems experienced and the identification of the root causes in the IPS development model used in the case-study project have been discussed. The summary of the research findings are that management accounted for 57% of all the problems experienced on the project whilst systems engineering accounted for 20%. The remainder of the problems fell into the specialised categories of QA and CM.

Discussed earlier, most of the engineers and project team members had post graduate qualifications in their respective fields and practical field experience. Therefore the project team focus group was qualified and experienced in research. Applying the Narrative Inquiry, Clandinin et al, (2000), two fundamental causes for the problems experienced have been identified:

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

- Systems Engineering primarily develop "***WHAT***" data and insufficient "***HOW***" data.

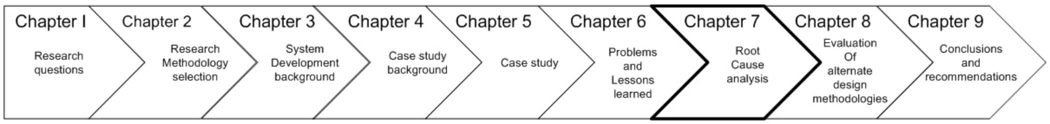
Both these factors have a significant detrimental effect on development project performance and accounted primarily for the cost and schedule overruns on the case-study project.

It is significant that management related problems overshadowed the other problems on the case-study project. The results reflected in table 1, however are misleading in that they reflect the symptoms of the incompatible interactions between the SE and PM processes. As discussed in Chapter 3, the one process cannot function without the other. What is observed is an apparent paradox in that the very team that wants the project to be on schedule and within cost are apparently the cause for the cost and schedule overruns! The competence of management has been ruled out and the fundamental root causes for project failure for being over-schedule and over-cost must now be identified.

In the next chapter the performance of the IPS model and whether there is a theoretical ground for the case-study findings will be discussed.

"We live in a complex world. People and organisations don't believe they have the time to perform the in-depth analysis required to solve problems. Instead they take remedial actions to make the problem less visible and implement a patchwork of ad hoc solutions they hope will prevent recurrence. Then when the problem returns, they get frustrated – the cycle repeats."

Duke Okes, (2009)



Evaluation of development model
Finding the root cause
Determining theoretical background
Development of generalised impact equation
Impact of functional couplings
How can the development model be improved
What other models would be appropriate

Chapter 7 ROOT CAUSE ANALYSIS

7.1 Purpose and Outline of the Chapter

In the previous chapter the problems experienced using the IPS development model and the identification of the root causes for these problems on the case-study project have been identified and quantified. It was found that management related problems overshadowed the other problems for the case-study project. This result is surprising, since according to Roos, (2001), the IPS development model's primary management focus is on:

- Effective organization policy.
- Utilization of design practices for the different development disciplines.
- Development of low risk methodologies for transition from design to production.
- Interrelation between development disciplines in a multi-disciplinary environment.

The result of the case-study problems experienced was surprising, particularly in view of the combined experience of the project team. A number of advocates for the IPS development model used in the survey by Roos, (2001) were prominent players in the ZT3 Weapon System development project used for the case-study. Project cost and schedule overrun is a project quality issue and the negative outcome is confirmed by the research by Sanjay et al, (2000). Using 418 quality practices from multiple industries, Sanjay et al, (2000), concluded that management has a major impact on quality. Pretorius et al, (2007), investigated the role of design and design management in development project failures using two case studies and concluded that a systems view accompanied by “***proper design management”*** will result in project success. The paper however does not detail “***proper design management***”. For the case-study project of this research, the project and systems engineering management was highly experienced suggesting that other causal factors are at play as suggested by Christensen et al, (1998). The two top level causes have been identified and discussed in the previous chapter.

The purpose of this chapter is twofold:

- Evaluate the effectiveness of the IPS development model for a complex weapon system development project inclusive of all the logistic products.
- To determine whether there is a theoretical ground for the findings of the case-study.

7.1.1 Evaluation of the IPS Development Model

Discussed in Chapter 6, technically the development project was a success. The client operational, maintenance and munitions personnel were very pleased with the new weapons system. The client-contractor specialist workgroups to influence the ergonomics, maintainability and munitions safety design really proved their worth. It was announced in the internal company newsletter in October 2009 that the ZT3 Anti-Tank Weapons System project was selected for the Chairman’s award. With small adaptations to the anti-tank weapons systems, the company also markets the system for universal vehicle mount as announced in a media release on 12 September 2008. From the company point of view the cost and schedule overruns were a worthwhile investment for a competitive marketable product range (ALRRT turret missile range, Denel Dynamics’ brochure 0269, (2012)).

The question that must be further researched is what are the fundamental reasons for development projects of complex multi-disciplinary systems cost and schedule overruns?

Christensen et al, (1998) found that often, a contract is not fully defined when it is awarded, and changes to the contract occur as the project progresses. They termed this a “rubber baseline” or baseline instability. In a study using cost performance data from over 400 defence acquisition projects, Christensen et al, (1998), came to the surprising conclusion that there is no relationship between baseline instability and cost overruns. Their research also showed that the results were insensitive to the contract type. They suggested that other possible causal factors should be further researched.

Before a configuration item (CI) design is frozen and baselined[10] anticipated Iterations are planned for upfront and incorporated into the project plan. The cost and timescale is a function of the complexity and maturity of the technology for the design of that specific item. If it were a state-of-the-art design, invariably the planning would include the building and evaluation of various development models such as XDM[11], ADM[12] and other models depending on the complexity and technology maturity of the specific item. Apart for a project technical risk, this generally does not lead to unplanned iterative rework with their associated negative cost and schedule impacts.

Cost and schedule pressures however may force the premature release of a design CI for a system by the design review board. This may increase the technical risks at the subsequent levels of system integration. The real problem arises after the CI has been baselined and a latent design defect is found at the integration stage of the system. Holt, (2009), by using a systematic approach, developed maturity models to reduce system integration risk. A number of researchers propose the Stage Gate model to reduce the risk of design entering the next phase before the objectives of the first phase have been accomplished, (Markeset et al, 2003) and (Kleinsmann et al, 2005). If however during the integration process a problem is identified with the item, the impact from a project management point of view will be the following:

- Under the matrix organisational management structure, (Blanchard, 1998), the original design resources will not be available anymore since they would have been allocated to other projects. The waiting for resource availability will result in an unplanned project activity delay. If the activity lies on the critical path, the project will suffer an overall slippage.

- The additional cost for the unplanned activity is not planned for and must be financed from the project reserve. This reserve due to tender price competitiveness is generally kept to an absolute minimum. Unplanned costs can easily exhaust this reserve and lead to overall project cost overrun.

- In a concurrent engineering environment, all functional coupled items are also affected when one item in the system hierarchy is changed. The consequences of the functional couplings are that one unplanned change can result in a number of unplanned changes and resultant impact on the project.

---

[10] Baselined implies that the item's configuration status moves from draft (revisions a, b, etc.) to revision 1.

[11] XDM Experimental Development Model sometimes called breadboard model.

[12] ADM Advanced Development Model evolved from the XDM

Generally problems with a CI identified at integration level result in a class I change of the item since this is generally the first time that the full interface of the item is comprehensively tested. It is very seldom that a latent design defect that can be fixed with a Class II change will surface at integration level due to the rigorous qualification testing of the CI prior to baselining and releasing the design.

Concurrent engineering is integral to the IPS development model. A class I change in the concurrent engineering environment underlying the IPS development model will result in a ripple effect throughout the system hierarchy. The extent of this ripple effect is a function of the functional couplings of the affected item with other CIs in the system as a result of the emergent properties, discussed in chapter 3. This finding is confirmed by the research of Smith et al, (1997), who state that engineering design involves a very complex set of relationships among a large number of coupled problems in a concurrent engineering environment. Concurrent engineering is the underlying basis of the IPS development model. In a concurrent engineering environment, the affected functionally coupled items that must also be changed can result in a change avalanche effect with very negative consequences for the project cost and schedule performance.

Summarising, the structured focus group of Narrative Inquiry into the case-study problems experienced, found that the IPS development model is a very effective model for the development of complex systems in a multi-disciplinary environment. It resulted in a very successful product for the client. Described in chapter 6, the problems, experienced on the case-study project could not be ascribed to the IPS model but appear to be rather as a result of causal factors in the SE and PM processes. This will now be further investigated.

7.1.2 Finding the Root Cause

Addressing symptoms very often introduce unexpected other problems, particularly in complex multi-discipline real time control systems since the system's dynamic balance could be disturbed as discussed in chapter 3. It is therefore essential that the root cause/s of a problem be thoroughly understood before a corrective solution is developed and implemented.

7.1.3 Determining the theoretical ground

In this paragraph, the theoretical ground for the findings of the case-study will be determined. In chapter 6 the problems experienced on the case-study project have been discussed. This has led to the identification of the following top level root causes:

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

- Systems Engineering primarily develop "***WHAT***" data and insufficient "***HOW***" data. The development specification focuses on "***WHAT***" the design must do; the product specification confirms that the design complies with the "***WHAT***" requirements. There is no requirement in the specifications to describe "***HOW***" the specific design solution works.

The next paragraphs will discuss these causes to provide a better understanding and basis for deeper research into the fundamental causes. Discussed earlier once these causes are fully understood, only then will it be possible to develop mitigating solutions.

7.1.3.1 Project management and systems engineering processes

(PM and the SE processes have areas of incompatibility)

The cornerstone of the systems engineering process is iterations essential for design optimisation and the achievement of design maturity. Iterations are fundamental to the systems engineering process (INCOSE 2010), (NASA 2007). Iterations are an indeterminate process in so far that the number of iterations required cannot always be predicted upfront of the project to enable incorporation into the project planning. Project management cannot accommodate an indeterminate iteration process due to cost and schedule constraints. Under project management rules, a completed milestone cannot be revisited (PMBOK 2008). A new milestone should have been defined at the start of the project which is not possible since the number of iterations required for the design of a CI are not known at the start of the system development project.

This was the primary reason why development projects of complex systems very often suffer cost and schedule overruns and will be further researched in this chapter to find the theoretical reason for the apparent conflict between the project management and the systems engineering processes.

7.1.3.2 Systems Engineering Shortcomings

(Systems Engineering primarily develop "***WHAT***" data and insufficient "***HOW***" data)

A shortcoming in the current SE process is that the formal SE process is entirely focussed on the development of "***WHAT***" product data, (INCOSE, 2010). The process does not formally require to design engineers to develop "***HOW***" product data. The SE process output product data packs contain primarily "***WHAT***" product data against which the product is tested and qualified for compliance to specifications. This view is supported by the DoD Systems Engineering Management College, (2001).

Specification practises standard (Mil-Std-490A) used in the industry provides a guideline for prime item functional specifications. The design engineer must design the item to comply with this specification. Once the design has been qualified to comply with the functional specification, the designer must prepare a product specification describing the performance parameters of the product for its intended use, as well as the necessary interface and interchangeability characteristics. The performance parameters include all essential functional requirements under service environmental conditions or under conditions simulating the service Environment, (Mil-Std-490A). These specifications form part of the formal product data pack. There is no formal requirement for the design engineer to describe precisely *"**HOW**"* his design solution works.

The lack of *"**HOW**"* data leads to problems further downstream in the systems engineering process, particularly when the logistics package for the system is being developed, specifically the operator manuals, maintenance manuals and training system. In order to optimally deploy a system and be able to effectively diagnose any problem, a thorough understanding of "***HOW***" the system and subsystems work and "***HOW***" these different system components interact is essential. The formal system data pack under configuration control does not make provision for the "***HOW***" information to enable the technical authors and training specialists to proceed with the development of these logistic products.

The lack of "***HOW***" system data also affects the effectiveness of through-life engineering support of the system during the operational life cycle. It is highly unlikely that the original design engineers will still be available a number of years later when the system is in the operational phase and in need of a modification or upgrade. The effect at that stage is that the designs must be reverse engineered before changes can be implemented. This may lead to unexpected problems and generally lead to extensive re-qualification testing which could have been avoided. The classification of characteristics in accordance with DoD-Std-2101, (1979) to a certain extent reduces this risk but is generally limited to safety and critical performance characteristics of a specific CI.

This lack of "***HOW***" data is a contributing factor for the failure of development projects of complex systems. A work-around solution has been offered in Chapter 6. This problem area falls outside the scope of this research since it may ultimately result in a major change and adaptation of the current established systems engineering process. It is suggested that further research in this field be undertaken.

This research will focus on the first cause in the context of design influencing with the aim to better understand the fundamental mechanisms of design at the lowest level. Once these mechanisms are fully understood will it be possible to quantify the impact of a design change and find mitigating solutions.

7.1.4 Detailed Analysis of design iterations

Development projects of complex multi-disciplinary systems are an intimate and coordinated process of project management and systems engineering. A system cannot be developed using the systems engineering process by itself. It requires project management to coordinate and manage the schedule as well as the consumption of resources to ensure ultimate project success. For the development of complex systems, the one process cannot function without the other.

The first cause for development project failure discussed above, identified that the systems engineering and project management processes have areas of incompatibility. This is a paradox and poses a serious question on how any systems can be brought into being? It is also inconsistent with real life experience in that many successful systems have been brought into being and are being successfully deployed.

Before an answer can be provided, further analysis is required to fully understand and substantiate this finding. Literature abound discussing project management aspects and systems engineering aspects, refer to figure 1. There is however a dearth of literature discussing the interaction between the two processes. Eisner, (2002), is one of the few authors discussing both processes yet he fails to assess and discuss the individual interactions between the two processes.

7.1.4.1 Established design process

The NASA Systems Engineering Handbook, (2007) is one of the many published sources describing the systems engineering process with particular focus on complex systems. The successive design refinements are illustrated in figure 22.

According to NASA (2007), successive refinement involves a recursive and iterative design loop, driven by the set of stakeholder expectations where a draft architecture/design and derived requirements are developed. Each step also involves an assessment of potential capabilities and pitfalls identified through experience-based review of lessons learned from other projects.

The handbook however fails to state when the iteration process will no longer produce any more meaningful (desirable) changes (improvements) to the system design.

The Systems Engineering Handbook by INCOSE, (2004), view the number of iterations planned as part of the tailoring process but makes no mention when design iterations are complete. In the updated INCOSE Systems Engineering Handbook, V3, (2006), there is also no mention when design iteration ends. Similarly in the latest INCOSE Systems Engineering Handbook, V3.2, (2010), there is also no mention of when iterations ends. The systems engineering standard prepared by Pennell et al, (2005), confirms that iterations are part of the systems engineering process.

According to v/d Merwe, (2002), the spiraling process, such as the planning or any other activity phase must be "***completed***" before the next phase is entered and so on. He identifies the reason for the spiral process as a result of the focus shift from phase to phase during the process. The research by Ashton et al, (1998), found that for the development of an optimization model in the automotive industry, multiple levels of iterations are required in their model. At business level, the research by Asharayri et al, (1998), showed that multiple iterations are essential to re-engineer a business process.

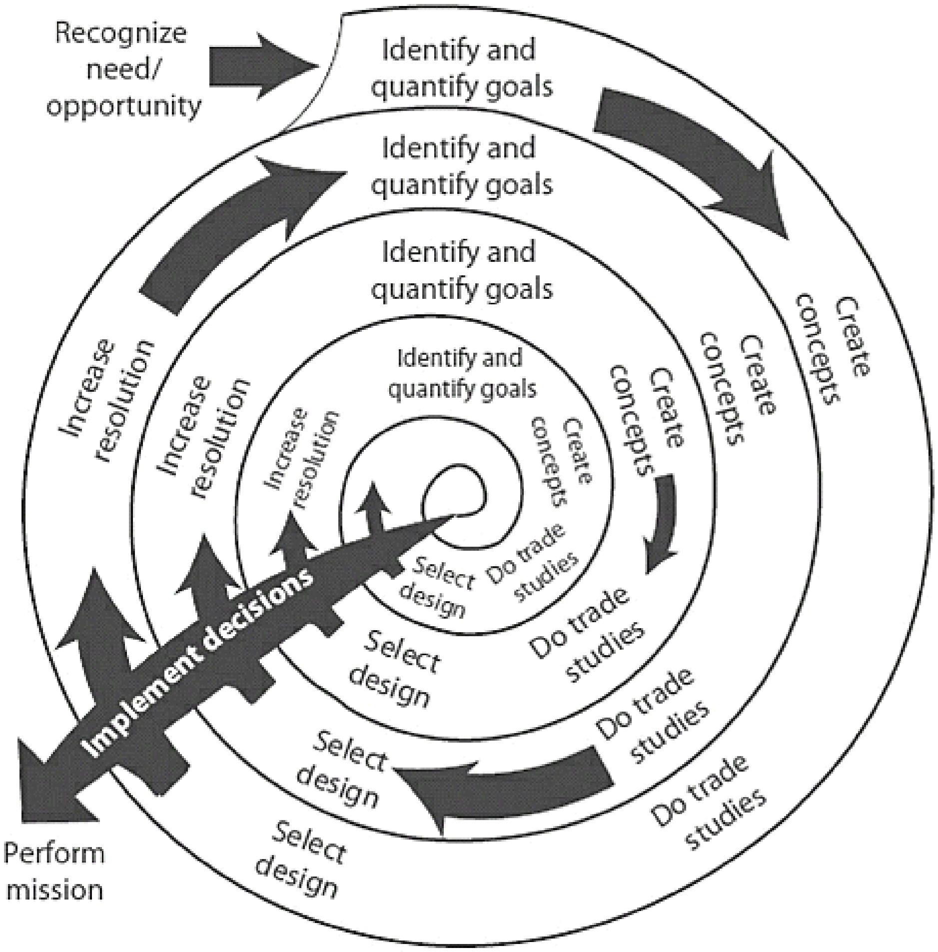


Figure 22: Successive design refinement

Source: NASA Systems Engineering handbook, (2007).

Most literature states that the design iterations ends when the design meets specification. This is a very broad statement since a large number of system requirements are often non-functional requirements or constraints that are difficult to quantify. Alexander, (2009), suggests a "***soft systems***" engineering approach for these requirements. My own view based on experience is that design ends when design maturity has been achieved. The question that now remains to be answered is "***what is design maturity*?**" The literature supplies many definitions and views of design maturity but none of these definitions actually fit into the context of design influencing and design refinement. Healy, (1989), provides the following definition: "*a system is mature when it performs its required function at specified performance levels at an optimum Life Cycle Cost for a stated period of time"*. The field of design maturity in the systems engineering context should be further researched.

For the purpose of this research, it is sufficient to accept that design refinement is an indeterminate process where the number of iterations required cannot be predicted with certainty at the start of a development project. This is in direct conflict with the fundamental project management principle where all activities and resource expenditure must be accurately planned and managed at the start of a project. This finding is supported by the research of Lu et al, (2001) that found that the Project Evaluation and Review technique (PERT) method does not support representation of iterations of the process. In an attempt to reduce design iterations, Torczon, (2007), developed a method using approximations to accelerate engineering design optimization. Li et al, (2008), using Reliability Based Design Optimization (RBDO) by decoupling the nested loops to reduce the computational workload, developed the d-RBDO model with the objective to make design based optimization deterministic.

Pritsker, (1966), developed the Graphical Evaluation and Review Technique (GERT) or conditional diagramming technique and systems dynamics models to allow for non-sequential activities such as loops in project management. This technique was taken up in the Guide to the Project Management Body of Knowledge (PMBOK 1996). However PMBOK (1996), cautions that the Precedence Diagramming Method (PDM) and the Arrow Diagramming Method (ADM) do not allow loops or conditional branches. The Project Management Institute (PMI) due to disuse discarded GERT from its third edition of PMBOK, (2004) onwards.

It can therefore be concluded that project management does not cater for iterative loops that is an essential part of the systems engineering process to enable design optimisation. To find the root cause of the management related case-study project problems, the quantitative interaction between the systems engineering and project management processes must now be determined.

7.1.4.2 Application of the SD-FD design influencing model

In Chapter 3 the structuring of the design teams in a DSR setting into ***Success Domain*** (**SD**) and ***Failure Domain*** (**FD**) teams was proposed. A design influencing model will now be developed to provide better insight of the design process at lowest level.

In preparation to find an answer to the research question: "*Can models be established to depict the success/failure domain interactions in a dynamic project management environment?*"

the proposed success frame and failure frame concepts discussed in chapter 3, were applied to the case-study design teams.

A systems engineer headed each team. One systems engineer was responsible for the development and architecture of the system and the other team was responsible for the logistics system engineering tasks and the subsequent development of the logistics products. The author had the responsibility for the logistics systems engineering and the development of the logistics products.

Systems engineering are also the custodians of the DOORS®[13] tool for requirements traceability and ensuring that all the requirements at each hierarchical level of the system have been addressed.

Eisner, (2002), states that if there is no coherent design, there is nothing to analyse. This implies that the SD team must first provide a concept design before it can be analysed by the FD team. Only when the Success Domain (SD) team makes a draft design available, can it be analysed by the Failure Domain (FD) team and feedback provided to the design review board (DRB). In practice this is an informal iterative process between the SD and FD teams with short iterative cycles.

Expanding figure 12 showing the interaction between the SD and FD teams, discussed in chapter 3, an unconstrained design influencing model can now be developed. Once the SD team has prepared a concept design, can it be analysed by the FD team and submitted to the DRB. The DRB identifies any deviations of the concept design from the specification and order another iteration until all design requirements have been satisfied. Once the design is acceptable, is the design baseline fixed and released for further integration into the system.

The DRB functions as a gate, similar to the Stage Gate model proposed by Markeset et al, (2003). This process effectively results in design iterations until the design is optimised and acceptable as illustrated in figure 23.

---

[13] DOORS® is supplied under licence by IBM® Rational® DOORS®; http://www-01.ibm.com/software/awdtools/doors/ (August 2010).

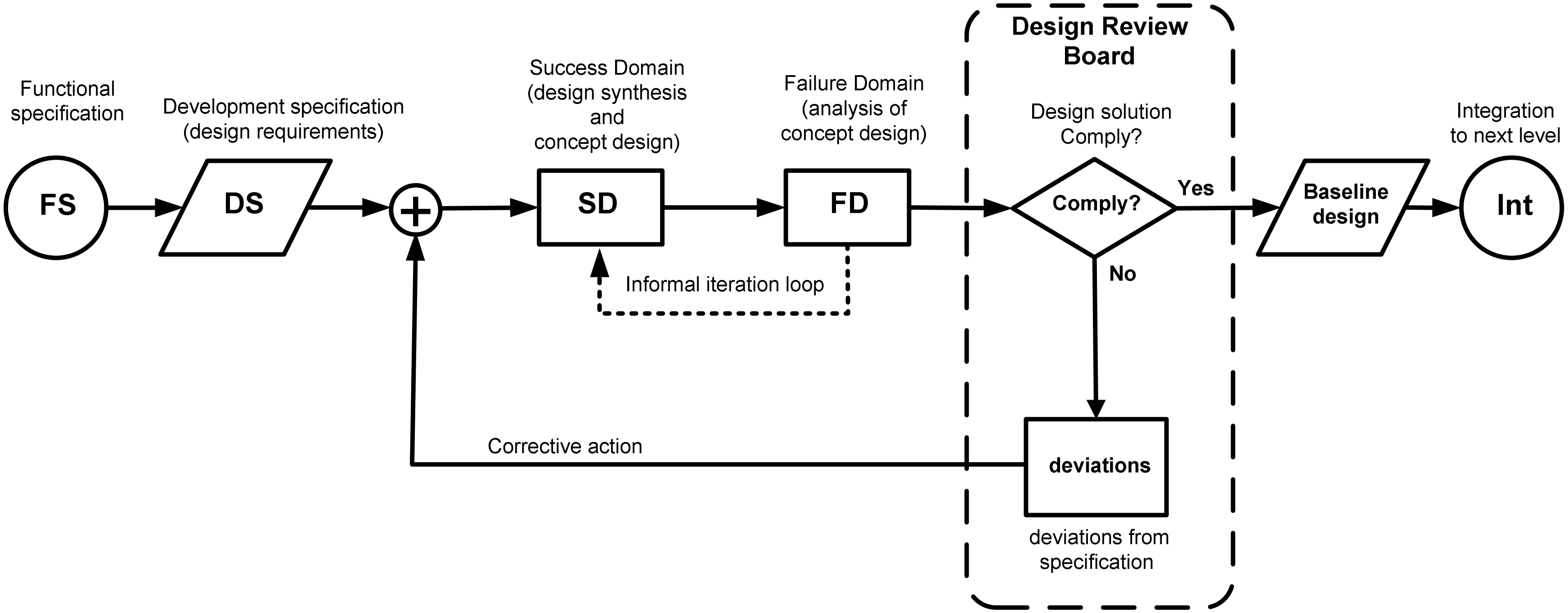


Figure 23: Unconstrained "*effect-to-cause*" design influencing model

Expanding SD block in figure 23, the design engineer as part of the SD team, by means of synthesis of the requirements and constraints, produces a draft design. Expanding the FD block in figure 23, the logistic engineering analysts as part of the FD team analyse this draft design for the "***-ility***" performance against the requirements. The Design Review Board (DRB) refers any shortcomings or deviations from the requirements back to the SD team for another design iteration. This iterative design process continues until the design complies with all the requirements and the design configuration is frozen and placed under configuration control in preparation for the next level of system integration. The number of iterations required is generally determined by the maturity of the technology selected and the technical complexity of the design (Smith et al, 1997). The FD team can only perform the analysis ***after*** a concept design has been provided by the SD team. In other words design influencing is an "***effect-to-cause"*** process.

This process, although at CI level, agrees with the systems level process by NASA illustrated in figure 22. Again the question remains "***when is the design acceptable****?*" This question is not trivial since a number of the design requirements such as reliability can only be verified after extensive qualification (TAAF) testing, Mil-Hdbk-189, (1981). Experienced design review teams normally take a calculated risk based on past experience with similar technologies and designs to expedite the release and baseline of a design.

7.1.4.3 Real world design influencing model

Discussed in Chapter 3, the Systems Engineering process by itself cannot bring a system into being. It requires the Project Management process to structure and manage the systems engineering activities and the consumption of resources, thereby ensuring within budget and on time delivery of the system to the client. The two processes therefore cannot be separated and must function in an integrated harmonious manner.

In a DSR setting, the developed unconstrained design influencing model shown in figure 23 will be expanded to incorporate the influence of project management.

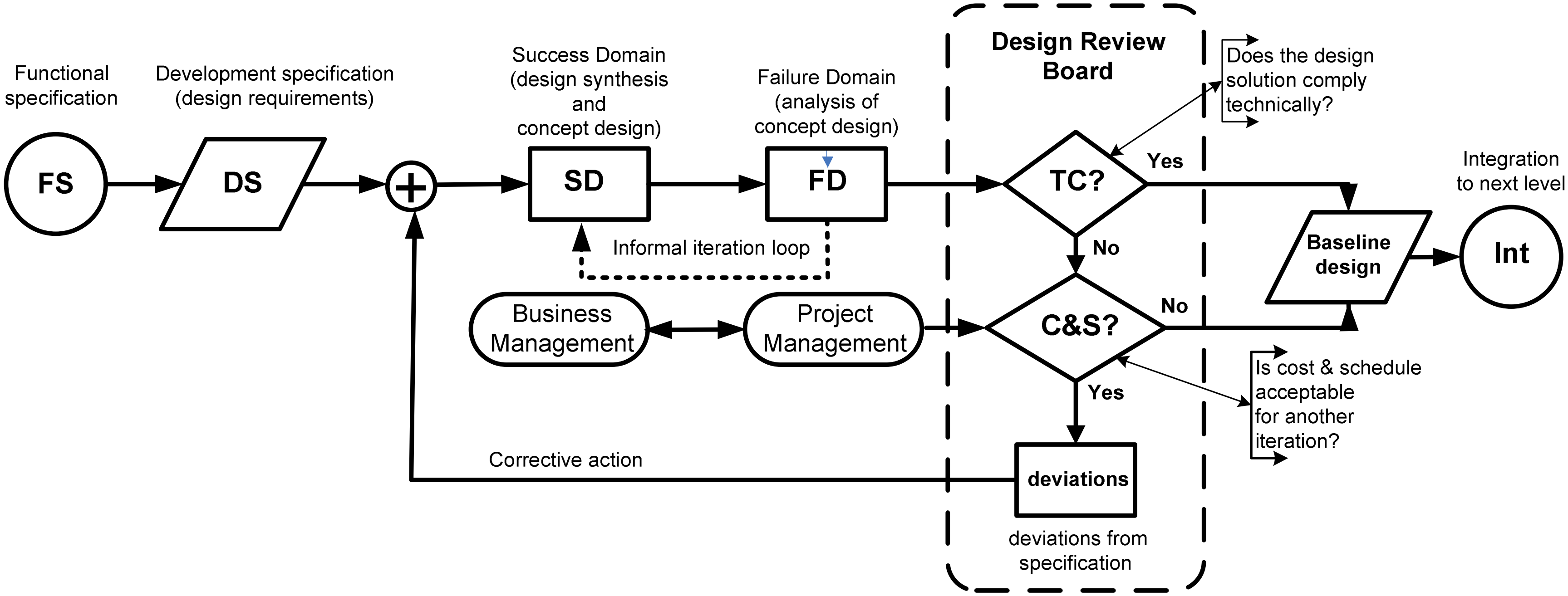


Figure 24: Constrained "effect-to-cause" design influencing model

Expanding figure 23 and introducing another gate, the project management gate, a constrained design influencing model can now be developed. The model adds project management to the DRB. Project management is now formally represented on the DRB and can apply its influence to the design process.

Whereas the systems team review a concept design from a pure requirements and technical perspective, the project management team review a proposed design also from a project cost and schedule perspective.

Again in the constrained design influencing model, the SD team prepares a concept design, to be analysed by the FD team and submitted to the DRB. The DRB identifies any deviations of the concept design from the specification and if acceptable, the design baseline is fixed and released for further integration into the system similar to the unconstraint design influencing model in figure 23.

Discussed in chapter 3, project management constraints are different from those of systems engineering and as such can influence the design process. If the DRB identifies any deviations of the concept design from the specification, project management has the final decision to allow another design iteration or to force a release of the design for the next level of integration.

Design influencing in the real world is constrained by project management as shown in figure 24. The iterative design for constrained “***effect-to-cause”*** process design influencing model is identical to the unconstrained design process with the addition of a gate in the iterative design process by the project manager. The project manager, depending on his constraints, generally cost and schedule, can allow another design iteration or force a premature design release. The design is therefore not fully optimised and mature to the satisfaction of the SD and FD teams. This increases the risk that problems may occur at the next level of integration of the system as a result of the prematurely released design. Design review checklists to a large extent mitigate these risks, (INCOSE 2010).

Design review checklists must be dynamic and must be regularly updated from company MIS sources such as PRACAS. The checklists must be universal and not project specific. The checklists must be developed to incorporate the lessons learned from not only the present system under development but also other systems under developed as well experience from fielding data.

The NASA FTA handbook, (2002), suggests that fault tree analysis (FTA) with the purpose of design influencing should take place as early as possible in the program to avoid costly changes later on. From the research of Markeset et al, (2003), the "**Stage Gate**" model was developed to reduce development program risk. The gates ensure that the next phase of the program is not entered before the objectives of the first one have been achieved, confirming the validity of the developed models shown in figures 23 and 24. According to INCOSE (2010), the gate ensures that the next step is achievable and the risk of proceeding is acceptable. This also agrees with the findings by v/d Merwe, (2002).

Underfunding and applying too stringent and unrealistic schedules to a development project exacerbate the project risk. NAVSO P-6071, (1986), NAVSO P-3686, (1998), NASA System Engineering Manual, (2004) as well as NASA Systems Engineering Handbook, (2007), support the view that apart from minimising the technical risks, ensuring that a project is not under budget and realistic timescales have been set, can reduce the risks of a complex multi-disciplinary development project. They also caution against project underfunding. The rationale for this caution can be deduced from the developed constrained design influencing model shown figure 24 where a project manager under severe pressure may be forced to take very high risks and release an otherwise unacceptable design.

In practice all that happens is that the problem is shifted to the next level of integration, where the resources required for corrective action becomes considerably more primarily, due to the ripple effect of the corrective action throughout the system hierarchy discussed earlier. The NAVSO Best Practices Manual (1986) cautions that underfunding and unrealistic timescales may sometimes lead to the total failure of an otherwise promising project.

Summarising, from a DSR setting, a model has been developed to better understand why design iterations are fundamental to design. This model has been expanded to a constrained design influencing model that provides a better understanding of the influence of project management in the design process.

The model agrees with the discussed literature. This model shows that the project manager, particularly if he is under unrealistic constraints, can force a premature design release for integration to the next system level. The developed model provides a fundamental understanding of the design process.

The question now remains what happens when a premature design is released to the next level of system integration. What would the impact of such a premature design be at system level?

In the next paragraph the impact of a design change at the system integration level will be studied.

7.1.5 The Generalised Design Change Impact Equation

(Development of the generalised design change impact equation)

In this paragraph the research question: "*What is the impact of functional couplings between system components of a concurrent engineering design?*" posed in chapter 1 will be discussed.

Before the ripple effects of design changes can be studied in the context of a concurrent design process, it is necessary to have a clear understanding of a typical complex multi-disciplinary system. Designs can be uncoupled, decoupled or coupled. In an uncoupled design each functional requirement is satisfied by one design parameter. This is considered the best design. The next best design is a decoupled design where functional requirement independence can only be achieved if the design parameters are arranged in a proper sequence. The least ideal design is the coupled design. Here the functional requirements are more than the design parameters selected to satisfy the functional requirements. An everyday example of a coupled design is a bathroom water faucet. The two functional requirements are *control-the-temperature* and *control-the-flow rate*. The two design parameters are the hot- and cold-water tap handles. This design is coupled because it is impossible to adjust either design parameter without affecting the other functional requirement: Each handle affects both temperature and flow rate, (Gumus 2005).

Real physical complex multi-disciplinary systems, typically have a multi-dimensional hierarchal structure, of which the individual system functional elements may be coupled or uncoupled as discussed above. Real systems generally have numerous functional couplings between the different system structural elements (Smith et al, 1997), e.g. the logistic system PBS lies actually behind the operational system PBS with functional couplings between them. This presents a practical problem of presenting the complete system on a two-dimensional sheet of paper. For convenience and simplicity, as part of the system decomposition process, it is customary for complex multi-disciplinary systems to present each discipline on its own PBS such as the logistic system, software system, hydraulic system, pneumatic system, optical system, (NASA, 2007).

This creates the misconception that these product breakdown structures are separate and independent when in actual fact they are not since they are part of the system functional breakdown structure (FBS), (NASA, 2007). Generally there are numerous functional couplings between the different system PBS elements as confirmed by Smith et al, (1997). This implies that for the analysis of the ripple effect of a change, the system must be viewed as a whole multi-dimensional entity. The hierarchical system in figure 6 (Hitchins 1992), can be redrawn to also show the functional couplings between system elements that result in the emergent properties of the system.

Figure 25 illustrates a simplified hypothetical multi-level system showing possible functional couplings between elements.

To avoid obscuring the illustration, figure 25 reflects a very simplified multi-level system of i Levels, each level consisting of CIs pertaining to that specific level. The numbering of each CI identifies it to its level and to its position in the hierarchy. Possible functional coupling are illustrated by the double ended arrows.

The ripple effect of a change on one CI in say the hydraulic system may manifest in the electrical, software, logistical system elements depending on the individual functional couplings. The functional couplings between parents and children are as a result of the emergent properties discussed in Chapter 3.

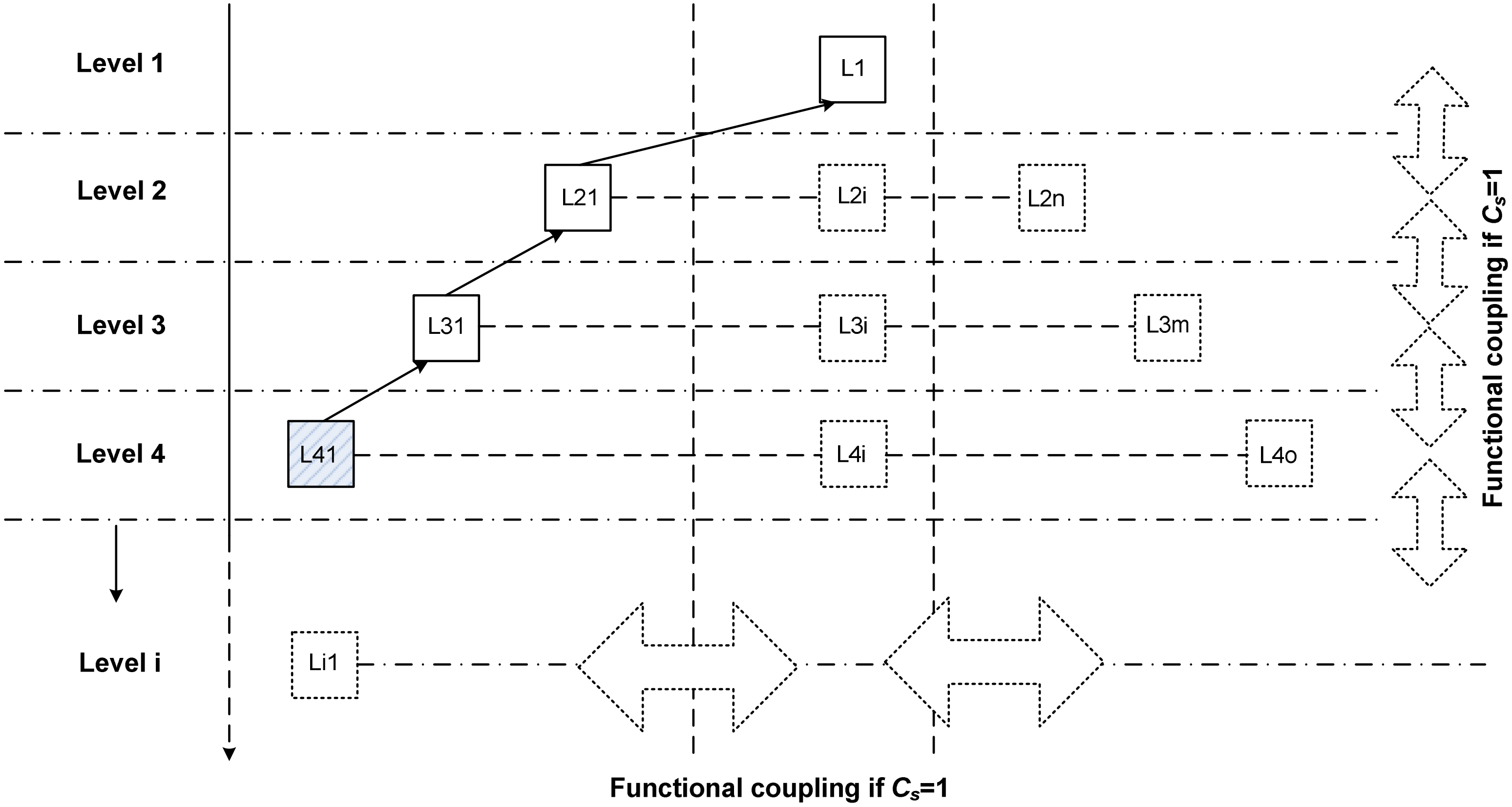


Figure 25: Multi-level system showing possible functional couplings

7.1.5.1 Impact of functional couplings

A mathematical model will be developed to quantify the impact of a design change in a multi-hierarchical system in a concurrent engineering environment as a result of functional couplings between the system functional elements, (Wessels et al, 2011). In Chapter 6 the following categories of configuration control have been identified:

- Class I modification
- Class II modification
- Rework

It is only the Class I modification that will result in a ripple effect to other functionally coupled system elements in the system hierarchy since the interfaces to the outside world or FFF is affected.

The Class II modifications and rework categories are contained within the CI and do not affect the FFF of the item. The interfaces to the outside world are not affected and therefore these categories of modification will not cause a ripple effect. These two categories however will result in unplanned expenditure of resources to correct the affected CI. Apart of unplanned expenditure of resources to correct the design, a Class II change or Rework causes no ripple effect to other system elements.

The ripple effect of a Class I change of a CI occurs by forcing changes to other CI's under development, are as a result of the functional couplings between functional elements in the system hierarchy.

In practice, it is often only during integration of the system that a latent design defect of a CI is identified that may result in a Class I change. At this stage of the system development project other CI designs are already completed or nearing completion. Apart from the corrective design change to the affected CI, the forced design changes to all other functionally coupled CIs, generally impact adversely on the development project cost and schedule.

It can thus be deduced that the root cause for the ripple effect is due to a class I change of a CI as a result of the functional couplings between functional elements in the system hierarchy. Class I changes of a CI only impact other functionally coupled system element at the subsequent levels of integration of the system.

7.1.5.2 Development of the mathematical model

A mathematical model of the hypothetical multi-level system in figure 25 will be developed. From this mathematical model it will be more convenient to study the implications of different system hierarchies in particular the effects of the functional couplings between system elements.

a) Functional coupling rules

Functional coupling rules must be defined as a prerequisite to the development of the impact equation. The dotted lines between system elements shown in figure 25, illustrate possible functional couplings. A coupling constant C_s is defined as follows:

- If there is functional coupling between affected CIs, the coupling constant $C_s=1$.
- If there is no functional coupling between affected CIs, the coupling constant $C_s=0$.
- There is ***always*** a functional coupling between an affected CI and its own parent as a result of the emergent properties, in that case $C_s=1$.
- There is ***always*** a functional coupling between an affected CI and its own children as a result of the emergent properties, in that case $C_s=1$.
- There may be functional coupling between the affected CI and its peers, other parents and children, in all those cases $C_s=1$.
- If there is no functional coupling between the affected CI and any other CI in the system hierarchy then, $C_s=0$.

b) Impact of CI design change

(Impact of a design change of an affected CI on its parents and children)

Using the system hierarchy in figure 25, assume that the affected CI is L41.

Then L31, L21 and L1 are the parents of CI L41. System emergent properties dictate that functional couplings must exist between child and parent. Therefore $C_s = 1$ for these instances (Hitchins 1992).

Similarly, L51 to Li1 are the children of the affected CI, L41 where i is the i_{th} level in the system hierarchy.

System emergent properties dictate that functional couplings must exist between parent and children. Therefore $C_s = 1$ for these instances.

Let R_p be the total impact of a CI as a result of the parent and children functional couplings.
Then

$$R_p = C_sL_1 + C_sL_{21} + C_sl_{31} + C_sL_{41} + - + - + C_sL_{i1}$$

$$R_p = \sum_{i-1}^{l} C_sL_{i1} \qquad (1)$$

Where *l* is the total parent and children CIs and *i* is a real integer reflecting the parent or child CI.

> **Note**
>
> As a result of the system emergent properties, equation (1) ≥ 1

c) General impact of CI change in the system hierarchy

Equation (1) can be generalized to incorporate the whole system hierarchy.

Let R_c be the impact of a specific CI change due to other system functional couplings in the system structure:

Assuming

- *m* represents the total configuration items in the system structure not related to the affected CI structure.
- *j* is an integer reflecting the j_{th} configuration item in the system structure.
- C_s is the functional coupling (C_s=0 if functionally decoupled or C_s=1 if functionally coupled).
- Where *n* is the total number of configuration items in the system.
- C_s=1 for all the configuration items where a functional coupling exist and the affected configuration item.
- C_s=0 for those configuration items where no functional coupling exist with the affected configuration item.

Then $$R_c = R_p + \sum_{j=1}^{m} C_s L_j$$

From equation (1) $$R_c = \sum_{i=1}^{l} C_s L_{i1} + \sum_{j=1}^{m} C_s L_j$$

Since $l+m = n$

$$R_c = \sum_{k=1}^{n} C_s L_k \qquad (2)$$

Equation (2) has been derived from figure 25.

As a result of the system emergent properties, there will always be functional couplings between the affected CI, its parent and children in the system hierarchy. Therefore equation (1) is always) ≥ 1 and can never be zero in a real system. Since equation (1) ≥ 1, equation (2) is also always ≥ 1 in real systems.

Inspection of equation (2) shows that the impact R_c increases substantially with each functional element coupling in the system hierarchy. A complete derivation and relative illustrated examples are provided in appendix C.

- **Implications of equation (2)**

  Equation (2) states that the impact of a change to the design of a CI is a function of the sum of all functional coupled items to that CI.

  Equation (2) for a specific system hierarchy will have a different value for each system CI.

d) Mathematical implications of the model

Equation (2) shows that the impact or ripple effect of a design change of one CI in a concurrent engineering environment escalates as a result of functional couplings between CIs and the size of the system hierarchy. The following conclusion can be drawn from the mathematical model:

- From equation (2), it can be deduced that in order to reduce the ripple effect of a design change to a configuration item, a design objective should be to minimize equation (2).

- To minimize the risk of a design change of one CI on the rest of the system, the system hierarchy must be optimized in such a way that the functional couplings between system elements (CIs) are minimum.

- Avoiding unnecessary functional couplings between configuration items.

- Ensuring that each functional requirement links to only one design parameter.

- Since equation (1) is always) ≥ 1 it precludes equation (2) from ever becoming zero.

- Totally decoupled designs can only be found in very simple single hierarchical level systems such as components or simple products.

- The value of R_c is an indication of a development project's cost and schedule risk should a design change be required.

A simple 3 level system hierarchy structure with 9 CIs at the lowest level was considered as a case-study, refer to appendix C. The summarized findings for the case-study examples in appendix C are:

- A simple 3 level system hierarchy structure with 9 CIs at the lowest level and with minimum functional couplings. This is considered a best-case system design of only functional couplings between parent and children and no peer functional couplings. If these remaining functional couplings for example were to be removed there would be no system but only a collection of CIs without any emergent properties.

- Using the same simple 3 level system hierarchy as above, but this time all the CIs are functionally coupled to one another providing a worst-case system hierarchy design.

- The impact for a design change in this case-study example was a cost increase of 213% and a schedule penalty of 300%.

The case-study examples are hypothetical for illustrative purposes only. In reality, real systems are much more intricate with multiple level system hierarchies and numerous different discipline CIs. Computer simulation is required to analyze these systems. Such a model would provide quantified CI design change impact information to enable design review boards to make informed decisions. Computer modeling development falls outside the scope of this research.

From the analysis it can be concluded that a design change of a CI in a complex multi-component, multi-hierarchical system during system integration in a concurrent engineering environment invariably has a detrimental effect on project cost and schedule. In practice, design modifications/changes during system integration of a complex multi-hierarchical system are virtually unavoidable. The impact of forced changes can only be improved by optimizing the system architecture to keep the system data content or functional couplings to a minimum.

e) Conclusions of the case-study examples

The hypothetical case-study examples provided in the appendices above clearly demonstrate the escalating cost and schedule impact of a design change on a concurrent systems engineering development project. This impact is a function of the number CIs and functional couplings in system hierarchy. Design changes in a concurrent engineering development project have the following consequences:

- Design changes in coupled designs are generally not feasible due to the detrimental project cost and schedule impact. Design changes, discussed above are invariably Class I changes and result in a ripple effect due to the functional couplings throughout the system hierarchy.

- Limited design changes for uncoupled and decoupled designs may be possible for simpler systems.

- The adverse project impact in terms of cost and schedule is generally too high to implement any design changes of a CI for complex multi-level systems.

- ***Effect-to-Cause*** design changes place a severe system optimization constraint on the system designer.

- Design changes are a major development project constraint in a concurrent engineering environment. Very often a band aid fix is the only practical non project intrusive way of solving the problem, which can lead to a non optimal design. This will be discussed further below.

- Further research into techniques and design processes should be performed to reduce design changes for optimal system design.

From the case-study results, it can be concluded that unplanned design changes of even a single CI in a concurrent engineering environment can be a major contributor to development project cost and schedule overrun.

From equation 2, the actual impact of a design change in a real system under development, can be calculated in order to assess the feasibility of allowing the design change, or to rather look for alternative lesser development project intrusive solutions to the problem at hand.

From the analysis it can also be concluded that the system architecture plays a very important role in reducing the system development risks. To achieve this, early system engineering is mandatory to optimize the system architecture with the objective of reducing the system information content amongst others.

Early systems engineering efforts can substantially reduce system development project risks (Honour 1994). With increased demand for "*systems of systems*", systems integration practices have steadily become more formalised and more specialised in recognition of the improved technical, cost, and schedule outcomes that can be achieved by the control and incremental validation of system interfaces throughout the developmental programme.

7.2 Summary of the impact of change

From the above model and analysis, it can be construed that due to the functional couplings, design changes in a complex multi-disciplinary system in a concurrent engineering development environment generally impact negatively on development project cost and schedule. Also design iterations should be curtailed due to project cost and schedule constraints.

In practice the impact of a class I change and associated ripple effect as a result of functional couplings in the system hierarchy is very often curtailed by doing a class I change on another lesser impact CI. This is risky unless the root cause mechanism for the failure is fully understood since a "*Band-Aid"* fix can easily result in a host of other unexpected problems. The preferred candidate by failure review and design review boards for this "*surrogate*" modification to mask the problem is software provided Human-Machine Interface (HMI) is not affected. This is generally very effective but leads to distortion, fragmentation and logical flow of the software. Also unless meticulously documented and kept under configuration control, can severely hamper through-life engineering support of the system. A simple example of such a surrogate fix could be the masking of contact bounce and resulting glitches caused by a relay or switch by means of software. This route may be far cheaper and less disruptive than to source another component but such a fix introduces an extra time delay that might affect system stability.

Equation (2) can be used as the mathematical basis to model a system under development. Such a model will assist design review boards and project management by accurately and quickly determining the impact of a proposed change on the project prior to approval of the change.

Real life reality is that cost and schedule are the primary constraints of any development project. The IPS development model and likewise all the other development models allow only "***effect-to-cause***" design influencing by the FD team. The FD team can only start the design analysis once a coherent design has been made available by the SD team. This leads to design iterations that are in conflict with the constraints of project management due to the severe impact on cost and schedules. It appears that in practice systems engineers are very seldom allowed to fully optimise a system under development.

This conclusion is in conflict with real life experience in that there have been many successful complex systems developed and deployed. The case-study of the upgrade of the ZT3 anti-tank weapons system is a case in point. The cost and schedule overruns were there but not catastrophic to the project. From the cold theoretical analysis it is surprising that a functional system at all was developed and deployed. The logical deduction is that there must be other factors at play. This will be further discussed in the next paragraph.

7.2.1 What other factors are at play?

The analysis discussed so far covered the "***hard***" systems engineering, (Alexander et al, 2009). The "***hard***" systems engineering can be quantified as shown in equation (2) and Appendix C. Checkland, (2001), discusses the "***soft***" system engineering that involves not only technical but also social, political and emotional issues and the relationships between them. According to Alexander et al, (2009), hard systems engineering addresses well defined problems whilst the soft systems engineering addresses vague ill structured problem situations that are difficult if not impossible to quantify.

It is this author's view supported by the case-study that the primary "***soft***" factor at play is the development team's interpersonal skills. An autocratic domineering project manager can very quickly sink an otherwise promising development project. It suppresses team members' creativity and initiative. The author has experienced promising development projects being prematurely terminated primarily because of poor leadership style. This view agrees with the findings by Roos, (2001), for the IPS development model and his recommendation for the team approach.

It can be concluded qualitatively that the success of the anti-tank weapons system project was due primarily to the good leadership and interpersonal relations of the management team as well as the full support from the company's top management.

7.3 How can the IPS model be improved?

Once a design's baseline is frozen, any class I change can have severe ripple effects throughout the system hierarchy. To some extent there is a natural tendency for SD and FD teams to work in isolation, therefore, any improvements to the IPS development model can only be achieved by enabling continuous interaction between these teams. The downside of any continuous SD and FD team interactions are that both teams will be continuously interrupted leading to wasted man-hour resources whilst one team is waiting for the output of the other team. Also in a concurrent engineering environment, a number of these teams will be active at any stage of the development project. This creates a project management difficulty, since such a model will result in joint accountability between the two teams. Therefore PM must be aware of, and be ready to react to any possible overrun of cost or schedule which usually cannot be easily pinpointed.

The concurrent engineering development environment of the IPS model, Roos, (2001), finds that the biggest problem with design teams is that none of them were taught at home or in school to solve problems in a group environment and states that one of the most important challenges for concurrent development is to get engineers to work well in teams. Teamwork is the success recipe to concurrent engineering.

The word "*team*" must be construed in the broader sense to be the development team, the project management team as well as company's top management. If all participants work as a team, the "***hard***" systems engineering system limitations can be mitigated through "***soft***" systems engineering as advocated by Checkland, (2001).

7.4 What other models would be appropriate?

The case-study showed that the IPS development model is a good model for the development of complex multi-disciplinary systems as it combines the best of the other development models discussed by Roos, (2001). The limitation comes in that for the development of a complex system in a concurrent engineering environment, the model must interface seamlessly with project management to manage the resources and schedules. From the case-study and subsequent theoretical analysis this has proved not to be possible, in that project management cannot accommodate indeterminate iteration processes needed to fully mature the developed CIs of the system prior to integration at the next level. Design iteration is an established design optimisation tool and is primarily as a result of the ***"effect-to-cause"*** design influencing process discussed earlier. By forcing a premature design release, the likelihood of a class I change of a system CI at integration level and the associated ripple effect to all other functionally coupled CIs in the system hierarchy is increased.

A more structured systems engineering process may have the potential to reduce the number of design iterations thereby mitigating the risk of a premature design release. This will be further discussed in chapter 8.

7.5 Chapter Summary

Root cause analysis provides a better understanding of the fundamental mechanisms. Once these are fully understood, one is in a better position to develop mitigating solutions.

The purpose of this chapter was to assess the effectiveness for the case-study project of the IPS development model, for a complex weapon system development project inclusive of all the logistic products. The structured focus group during the Narrative Inquiry work sessions was unanimous that the IPS model is very effective.

The development of the anti-tank weapons system using the IPS development model combined with the **SD-FD** design influencing approach has resulted in a superior technical product. However the ***"effect-to-cause"*** design influencing process also resulted in design iterations in order to optimize and mature the design.

From a management perspective system and design engineers are measured according to design excellence in terms of the overall customer requirements, whilst project managers are measured according to project schedules and resource expenditure performance. The two performance measuring metrics are different resulting in conflict within the development team. Goldratt, (1992), Goldratt, (1997) and Goldratt, (2006), confirm this with the following statement: "***tell me how you measure me, and I will tell you how I will behave. If you measure me in an illogical way … do not complain about illogical behaviour.***"

It is now possible to confirm the validity of the high level root cause of the problems experience on the case-study project:

> "***Project management and the systems engineering processes have areas of incompatibility***"

Delving deeper, it was found that the root cause of this incompatibility is in fact the unexpected and unplanned design iterations by the systems engineering process. Design iterations are fundamental to the systems engineering process. Very often design iterations cannot be foreseen and planned for at the start of a project, and are therefore from a project point of view indeterminate. Indeterminate design iterations cannot be accommodated under the project management discipline as confirmed in PMBOK, (2008).

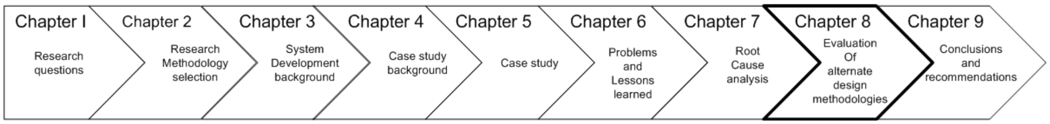
The purpose of this chapter also was to determine whether there is a theoretical ground for the findings of the case-study. A model illustrating the mechanism of ***"effect-to-cause"*** design influencing in an unconstrained and project management constrained environment has been developed. It has been shown that the project management constrained environment may lead to premature design release that can lead to later modifications during system integration. These modifications result in forced modifications to all functionally coupled CIs in the system hierarchy that impact negatively on the project performance management parameters of cost and schedule

Root cause analysis confirmed that the systems engineering and project management methodologies conflict with one another, in that the project management process cannot accommodate unplanned iterations, the corner stone of the systems engineering process. Planned design iterations within a project plan activity generally presents no problem to PM provided the activity is within cost and schedule constraints. Normally, however, once the activity has been completed, PM cannot accommodate a revisit of the activity for design optimisation later in the project.

The above discussion identified design iterations as a consequence of the "***effect-to-cause"*** design influencing process. The fundamental systems engineering process and the project management process are in conflict as a result of the unpredictability of the number of design iterations required. The consequences of this conflict may result in premature design release, increasing the risk of subsequent forced design changes during system integration. Any design change at this stage invariably result in a ripple effect of forced design changes to other functionally coupled CIs that are concurrently being developed.

One approach to mitigate this conflict is to investigate ways to reduce design iterations. In the next chapter ***"cause-to-effect"*** design influencing will be investigated, with the objective of reducing design iterations that are the natural outcome of ***"effect-to-cause"*** design influencing. Employing such a methodology may have benefits by reducing or eliminating iterations. This will make the systems engineering process for the development of complex multi-disciplinary systems more structured and therefore more compatible with the structured project management process.

"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.

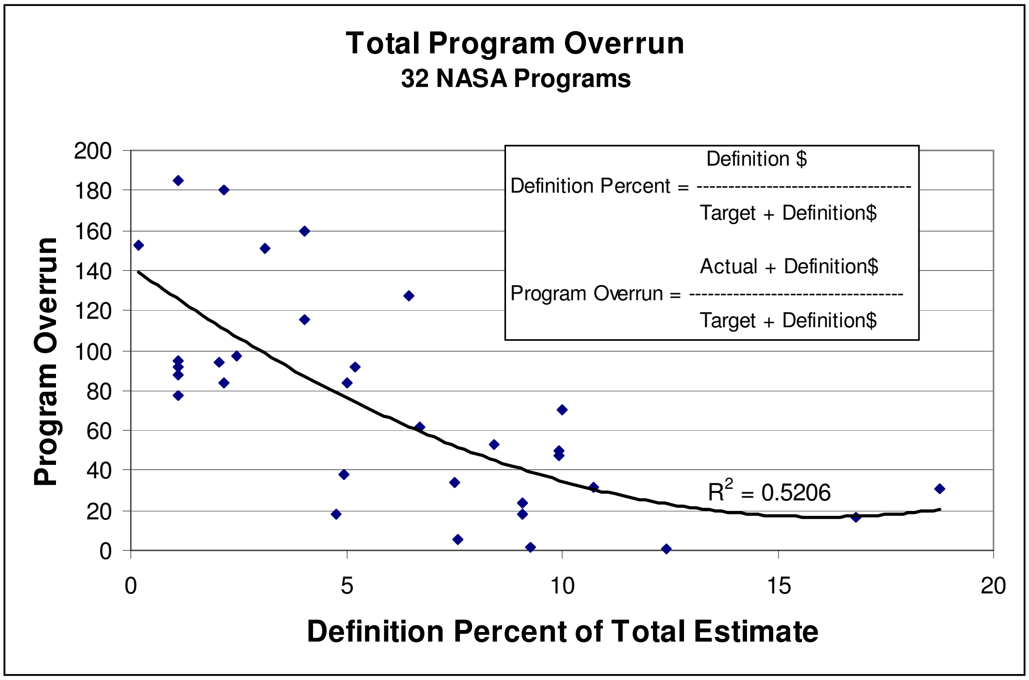


Figure 26: Value of Systems Engineering; Summary Report 1/04

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.

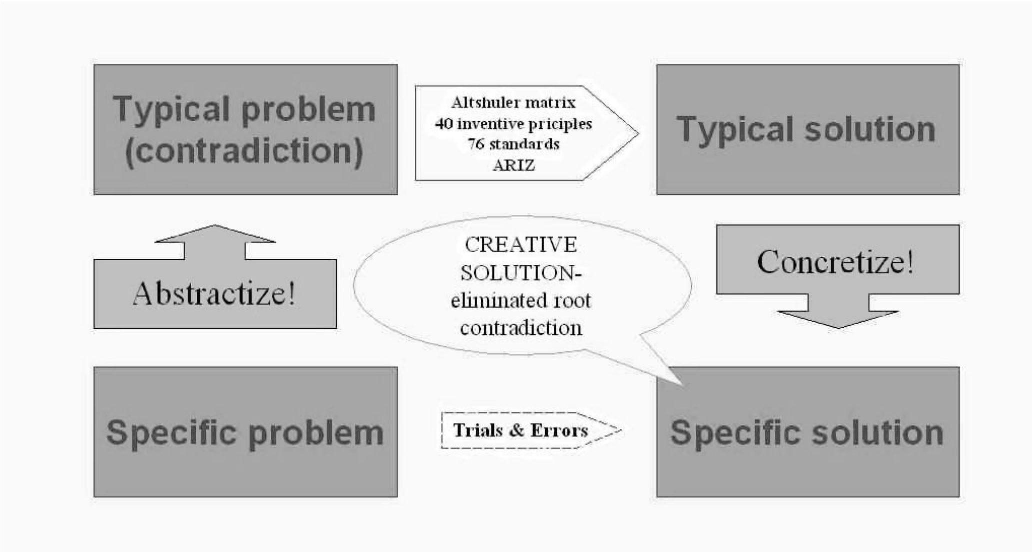


Figure 27: TRIZ process for creative problem solving

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.

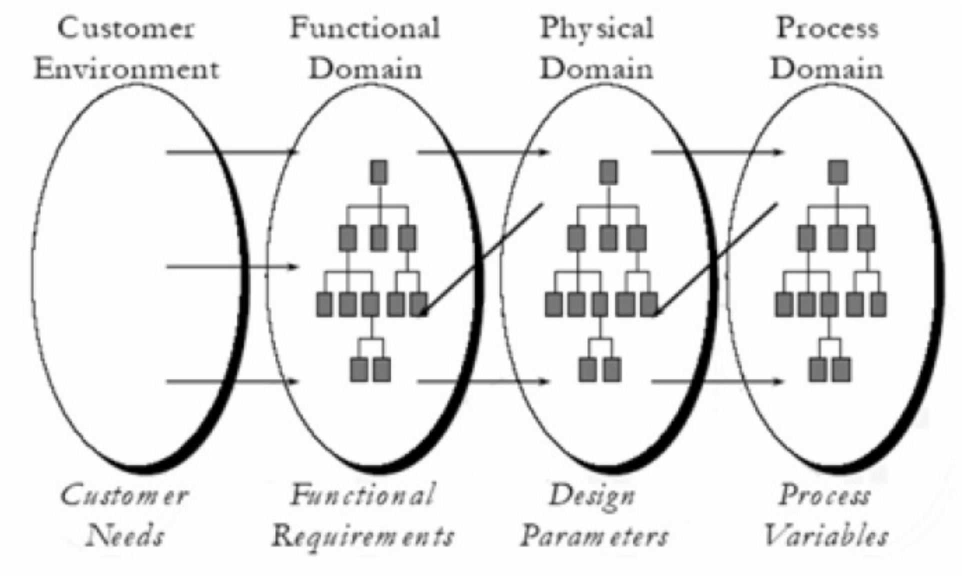


Figure 28: Axiomatic design domains

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.

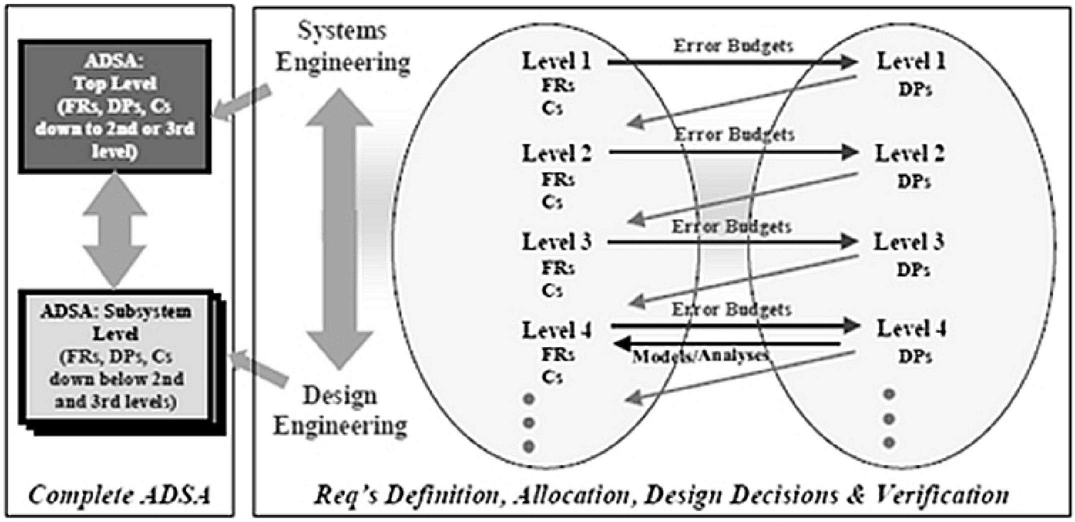


Figure 29: Distributed organisation of the AD system architecture

Source: Hintersteiner at 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.

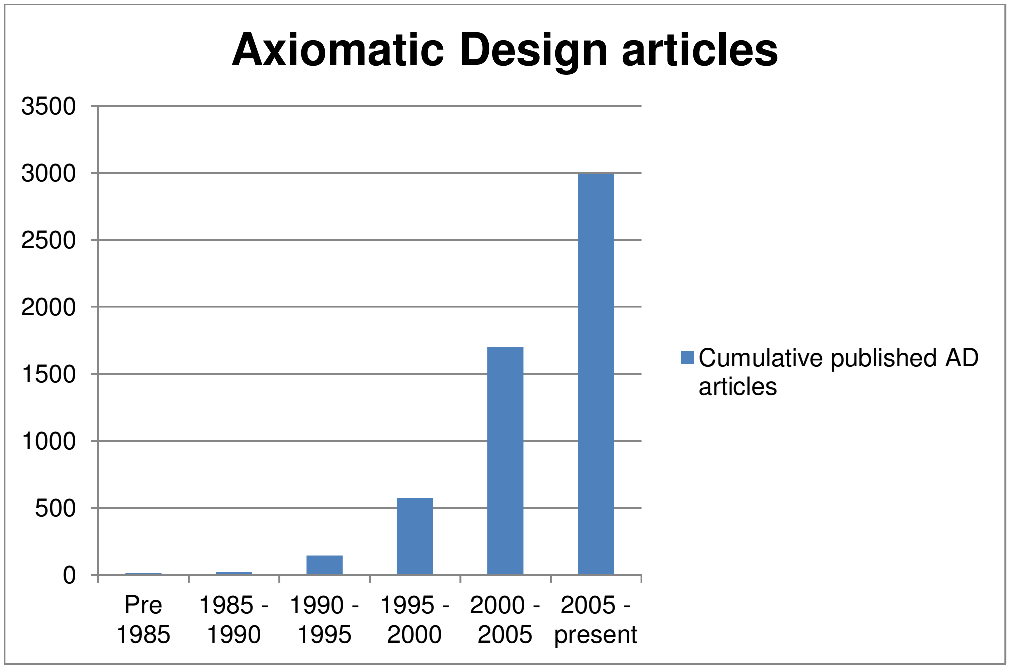


Figure 30 Axiomatic Design articles published

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.

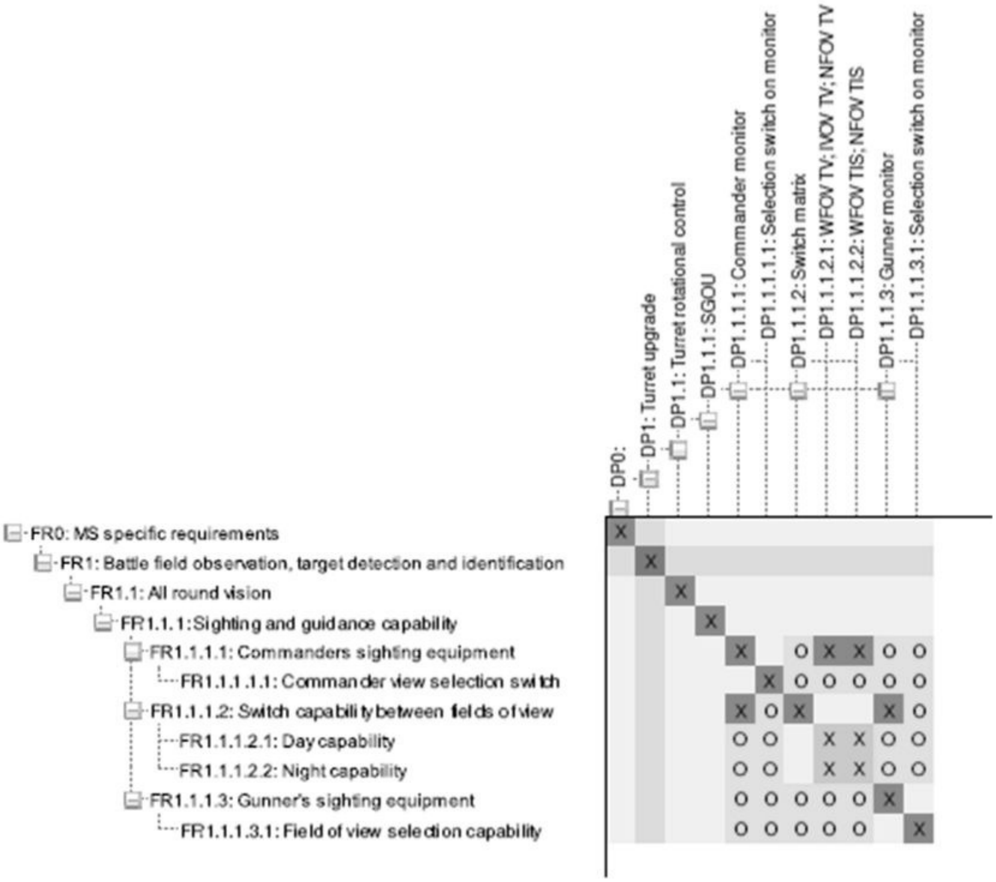
8.3.1 Structured Design Example: a subsystem of the case-study

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.



	DP0:	DP1: Turret upgrade	DP1.1: Turret rotational control	DP1.1.1: SGOU	DP1.1.1.1: Commander monitor	DP1.1.1.1.1: Selection switch on monitor	DP1.1.1.2: Switch matrix	DP1.1.1.2.1: WFOV TV; IVOV TV; NFOV TV	DP1.1.1.2.2: WFOV TIS; NFOV TIS	DP1.1.1.3: Gunner monitor	DP1.1.1.3.1: Selection switch on monitor
FR0: MS specific requirements	X										
FR1: Battle field observation, target detection and identification		X									
FR1.1: All round vision			X								
FR1.1.1: Sighting and guidance capability				X							
FR1.1.1.1: Commanders sighting equipment					X		O	X	X	O	O
FR1.1.1.1.1: Commander view selection switch						X	O	O	O	O	O
FR1.1.1.2: Switch capability between fields of view					X	O	X			X	O
FR1.1.1.2.1: Day capability					O	O		X	X	O	O
FR1.1.1.2.2: Night capability					O	O		X	X	O	O
FR1.1.1.3: Gunner's sighting equipment					O	O	O	O	O	X	
FR1.1.1.3.1: Field of view selection capability					O	O	O	O	O		X

Figure 32: Part of the SGOU Design Matrix

Had the design initially been modelled on Acclaro DFSS®, the extra design iteration and accompanied detrimental ripple effect throughout the system hierarchy, could have been avoided.

Although tree diagrams show the functional couplings, they tend to become cluttered and obscure functional coupling problems in the system hierarchy, particularly with larger systems.

The DSM showing the FBS on the y-axis and the PBS on the x-axis is a very effective way of achieving optimised design architectures. DSM does not become cluttered with larger systems and make any functional coupling problem clearly visible. Functional couplings are clearly illustrated in Figure 32 where the Functional Requirements: FR 1.1.1.1 and FR 1.1.1.2 are each coupled to more than one design parameter.

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.

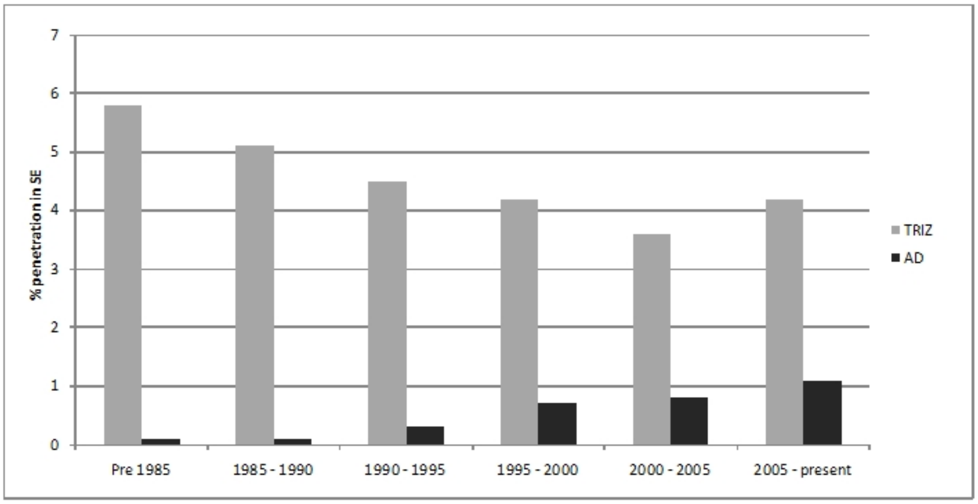


Figure 34: Penetration of TRIZ and AD into Systems Engineering

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.

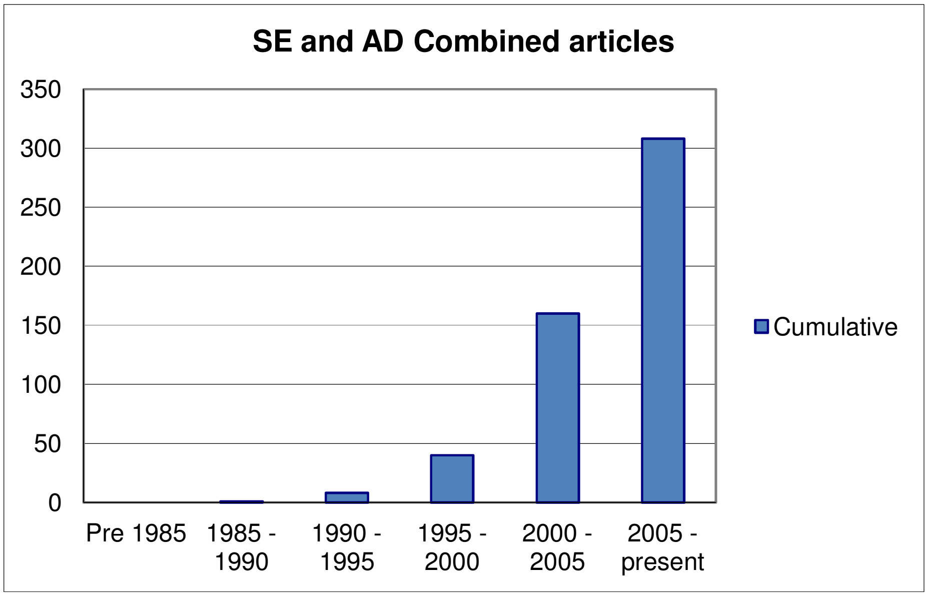


Figure 35: AD articles in context of SE published

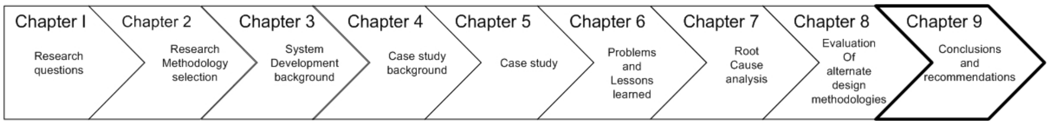
This low acceptance of AD methodology in SE was also confirmed by Hintersteiner, (2000), who investigated the integration of AD into the SE process. Hintersteiner defines a 5 level AD maturity model to provide a clear roadmap implementing AD within the SE process that will have both engineering and management 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)



Research questions answered
Academic contributions
Recommendations
Further research

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.

References

Alberts D.S. and Hayes R.E. (2006); ***Understanding Command and Control***; DoD CCRP Publications series; ISBN 1-893723-17-8.

Alexander I. and Beus-Dukic L. (2009); ***Discovering Requirements, How to Specify Products and Services***; John Wiley & Sons Ltd; ISBN 978-0-470-71240-5.

Ammerman M. (1998); **The Root Cause Analysis Handbook**: A Simplified Approach to Identifying, Correcting and Reporting Work Place Errors; ISBN 0-527-76326-8; Productivity Press NY USA.

Andrews C. and Henn C. (1997); ***Workshop Report on Systems Thinking, Academic Standards, and Teacher Preparation***, Princeton University and Rutgers University; 5 April 1997.

Ashton P.T. and Ranky P.G. (1998); ***Automotive Design and Assembly System modelling research Toolset at Rolls-Royce Motor Cars Limited***; Assembly Automation Vol 18 number 2; 1998.

Bassiouny A. (2011); ***Organisational Structure presentation***; http://www.slideshare.net/ahmad1957/organizational-structure-1340467
2011.

Blanchard B.S (1998); ***Logistics Engineering and Management***, fifth Edition, Prentice Hall.

Blanchard B.S and Fabrycky W.J. (1997); ***Systems Engineering and Analysis***, 3rd ed, Prentice-Hall, New Jersey.

Blanchard B.S. (2004); ***Logistics Engineering and Management***; 6th Edition, ISBN 0-13-124699-2; Prentice Hall, 2004.

Booch G., Rumbauch J. and Jacobson I. (1998); ***The Unified Modelling Language User Guide***; Addison Wesley; 1st edition, ISBN 0201571684; October 1998.

Browning T.R. and Eppinger S.D. (2000); ***Modelling the Impact of Process Architecture on Cost and Schedule Risk in Product Development;*** Working Paper Number 4050, Revised April 2000. Massachusetts Institute of Technology; Sloan School of Management.

Buede D.M. (2000); ***The Engineering Design of Systems; Models and Methods***; ISBN 0-471-28225-1; John Wiley; 2000.

Cataloguing Handbook, (1988); ***Cataloguing Handbook***, Department of the Army Supply Bulletin, General Services Administration H6, Section B; Defence Logistics Agency; Defence Logistics Service

Centre, Battle Creek, Michigan 49017-3084.

Checkland P. and Scholes J. (2001); ***Soft Systems in Action***; ISBN 0-471-986054; Chichester: John Wiley & Sons.

Christensen D. and Gordon J.A. (1998); ***Does a Rubber Baseline Guarantee Cost overruns on Defence Acquisition Contracts?***; Project Management Journal 29; September 1998.

Clandinin D.J. and Connelly F.M. (2000); ***Narrative Inquiry: Experience and Story in Qualitative Research***; San Francisco: Jossey-Bass Publishers, 2000.

Crnkovic G.D. (2010); ***Constructive Research and Info-Computational Knowledge Generation***; Studies in Computational Intelligence, 2010, Volume 314, Model-Based Reasoning in Science and Technology, Pages 359-380.

de Beer O. (2009); Process Coordination within Engineering Procurement Construction Management Companies.; Master's thesis University of Johannesburg; November 2009.

de Groot W.T. (1992); ***Environmental science theory: concepts and methods in a one-world, problem-oriented paradigm***; Elsevier, 1992.

DENEL (2009); ***Guided Missile Brochure***; www.wikipedia.org/wiki/Anti-tank_guided_missiles; 30 March 2009.

Denel Dynamics (2009); ***Ingwe Missile brochure***; http://www.deneldynamics.co.za/ .

Denel Dynamics (2012), ***ALRRT-4M Ingwe; Armed, Long-range, Reconnaissance Turret (Alert) missile system***; Denel Dynamics brochure 0269; 2012.

DoD Systems Management College (2001), ***Systems Engineering Fundamentals***; Defense Acquisition University Press; Fort Belvoir, Virginia 22060-5565; 2001.

DoD-Std-2101, (1979); ***Classification of Characteristics***; DoD-Std-2101; US Department of Defence.

DOORS® is supplied under licence by IBM® Rational® DOORS®; http://www-01.ibm.com/software/awdtools/doors/, August 2010.

Eisner H. (2002); ***Essentials of Project and Systems Engineering Management***; 2nd Edition; ISBN 0-471-03195-X; John Wiley & Son, NY.

Feagin, J, Orum, A, and Sjoberg, G (1991); ***A Case for Case study***; Chapel Hill, NC: University of North Carolina Press.

Gantt H.L. (2010); ***Organizing for Work***; EXACT reproduction of a book published before 1923; Nabu Press; ISBN-10: 1147734984; March 21, 2010.

Garner S.W. (1991); ***Design Topics – Human Factors***; Oxford University press; 1991.

Gero J.S. (2006); ***Research Methods for Design Science Research: Computational and Cognitive Approaches1***; Key Centre of Design Computing and Cognition; Department of Architectural and Design Science; University of Sydney, NSW, Australia; 2006.

Goldratt E.M. (1992); ***The Goal***; Creda press Cape Town; ISBN 0-620-21254-3.

Goldratt E.M. (1997); ***Critical Chain***; Creda press Cape Town; ISBN 0-620-21256-X.

Goldratt E.M. (2006); ***The Haystack Syndrome: Sifting Information Out of the Data Ocean***; ISBN9780884271840; North River press, USA; June 2006.

Goldratt E.M. (2006); ***Theory of Constraints and how it should be implemented***; North River Press; ISBN 0-88427-166-8.

Greene J.C. (2007); ***Mixed methods in Social Inquiry***; ISBN 13-0-978-7879-8382-6; Wiley, 2007.

Grundy T. (1998); ***Strategy Implementation and Project Management***; International Journal of Project Management, Vol 6, No 1, 1998.

Gumus B, Ertas A, Tate D and Cicek I (2008); ***The Transdisciplinary Product Development Lifecycle model***; USA; Journal of Engineering Design; Vol. 19, No. 3, June 2008.

GUMUS B. (2005); ***Axiomatic Product Development Lifecycle***; PhD dissertation; Texas Technical University; 2005.

Guzman O. (2011); ***The Advantages of Matrix Organizational Structure***; Demand Media; http://smallbusiness.chron.com/advantages-matrix-organizational-structure-286.html; 2011.

Healy J. (1989); ***Glossary of Reliability Growth Terms***; ***Reliability Growth Processes and Management;*** Institute of Environmental

Sciences; Mount Prospect, Illinois 60056.

Hill R.R. (1997); ***Engineering Research Technology For Concurrent Engineering From Unified Life Cycle Engineering***; Armstrong Laboratory; Logistics and Human Factors Division; Wright-Patterson Air Force Base Dayton, Ohio; 45433-6503, 1997.

Hintersteiner J.D. and Zimmerman R.C. (2000); ***Implementing Axiomatic Design in Systems Engineering process: An Axiomatic Design Capability Maturity Model***; International Conference on Axiomatic Design, 2000.

Hitchins D.K. (1992); ***Putting Systems to Work***; Wiley, Chichester, ISBN 0-471-93426-7.

Hogg I. (1996); ***Tank Killing; Anti-tank Warfare by Men and Machines***; ISBN 1-885119-40-2; Sarpedon.

Holt A. (2009); ***Developmental Maturity Models to Reduce Systems Integration Risk***; SYPAQ Engineering Discipline Lead; 2009.

Honour E.C. (2004). ***Understanding the Value of Systems Engineering***; INCOSE Symposium 06-2004.

Hsu C.C. and Sandford B.A. (2007); ***The Delphi Technique: Making Sense of Consensus***; Practical Assessment Research & Evaluation; Volume 12, Number 10, ISSN 1531-7714; August 2007.

Hu M., Yang K. and Taguchi S. (2002); ***Enhancing Robust Design with the Aid of TRIZ and Axiomatic Design (Part I)***; International Conference on Axiomatic Design; ICAD 2002.

IDA Report R-338 (1998); ***The Role of Concurrent Engineering in Weapons System Acquisition***; Institute of Defence Analysis, Alexandria, VA, 1988.

IEEE-Std-1490 (2003); ***Guide to the Project Management Body of Knowledge***; IEEE Computer Society; 2003.

Iivari J. and Venable J. (2009); ***Action Research And Design Science Research Seemingly Similar But Decisively Dissimilar***.; 17th European Conference on Information Systems 2009, Manuscript ID: ECIS2009-0424.R1.

INCOSE (2006); ***Systems Engineering Handbook; A Guide for System Life Cycle Processes and Activities***, INCOSE-TP-2003-002-03, Version 3; June 2006.

INCOSE (2010); ***Systems Engineering Handbook; A Guide for***

System Life Cycle Processes and Activities; INCOSE-TP-2003-002-03.2, Version 3.2, INCOSE , January 2010.

INCOSE web, (2010); ***INCOSE website***: http://www.incose.org/about/index.aspx; June 2010.

INCOSE web, (2010); ***INCOSE website***; http://www.incose.org/mediarelations/briefhistory.aspx; June 2010.

INCOSE, (2004); ***Systems Engineering Handbook; A "what to" guide for all SE practitioners***; INCOSE-TP-2003-016-02, Version2a, June 2004.

INTELLECT (2003); ***Reliability: A Practitioner's Guide***; The Information Technology Telecommunications and Electronics Association; Russell Square House; 10-12 Russell Square; London WCIE SEE,UK; www.intellekuk.org/relc.

ITAR (2011); ***International Traffic in Arms Regulations***, (22CFR Parts 120-130 the US Office of Defence Trade Controls (DTC).

Johnson C.W. (2006); ***What are Emergent Properties and How Do They Affect the Engineering of Complex Systems?***; Department of Computing Science, University of Glasgow,
Glasgow, G12 9QQ. Scotland, UK.

Jones J.V. (1987); ***Integrated Logistic Support handbook***; ISBN 0-8306-2921-1; McGraw-Hill Inc.

Joslyn C. and Rocha L. (2000); ***Towards semiotic agent-based models of socio-technical organizations***; Proceedings Artificial Intelligence, Simulation and Planning in High Autonomy Systems (AIS 2000).

Kerzner H. (2009); ***Project Management: A Systems Approach to Planning, Scheduling & Controlling***; 10th Edition; John Wiley & Sons Inc, 2009.

Kim B. and Cross B.(2008); ***Functional Cooperation with Design Teams in New Product Development***; International Journal of Design; Volume 2 no 3.

Kleinsmann M. and Valkenburg R. (2005); ***Learning from Collaborative New Product Development Projects***; Journal of Workplace Learning; Vol 17 No3; 2005.

Kossiakoff A. and Sweet N. (2003); ***Systems Engineering Principles and Practice***; ISBN 0-471-23443-5; Wiley Interscience 2003.

Kotler P, Adam S, Brown L and Armstrong G (2006); ***Principles of Marketing***; 3rd edition, Prentice Hall, 2006.

Kuhn T. and Poole S. (2006); ***Do conflict management styles affect group decision-making? Evidence from a longitudinal field study***; Human Communication Research, Volume 26, Issue 4, 10 January 2006.

Langford-Smith E. (1960); ***Radio Designer's Handbook***; Fourth Edition; Iliffe & Son Ltd, London; 1960.

Larman C. and Basili V.R. (2003); ***Iterative and Incremental Development: A Brief History***; IEEE Computer Society, June 2003.

Latino B. (2010); ***Root Cause Analysis (RCA) – Death of an Acronym?***, Reliability Center, Inc. http://www.reliabilityweb.com – July 2010.

Leedy, P.D. and Ormrod, J.E. (2001); ***Practical Research: Planning and Design***; 7th Edition. Merrill Prentice Hall, New Jersey.

Leszak M., Perry D.E. and Stoll D. (2000); ***A Case Study in Root Cause Defect Analysis***; Proceedings of the 22nd international conference on Software engineering ACM; New York, NY, USA ©2000
ISBN:1-58113-206-9.

Li F. and Wu T. (2008); ***d-RBDO: A Deterministic Approach for Reliability Based Design Optimization***; Department of Industrial Engineering; Arizona State University; November 2008.

Likert R. (1932); ***A technique for the measurement of attitudes***; Archives of Psychology, 140, (eds) Woodworth, R.S., New York University; 1932.

Lu Q. and Wood L. (2006); ***The refinement of design for manufacture: inclusion of process design***; Department of information Systems and Operations Management; The University of Auckland, Auckland, New Zealand, 2006.

Lu S.C. Y. Cai J. (2001); ***A Collaborative Design Process model in the Sociotechnical Engineering Design Framework***; Artificial Intelligence for Engineering Design, Analysis and Manufacturing; Vol 15; 2001.

Markeset T. and Kumar U. (2003); ***Integration of RAMS and Risk Analysis in Product Design and Development Work Processes – a Case Study***; Journal of Quality in Maintenance Engineering; Vol 9 No4; 2003.

Maylor H. and Gosling R. (1998); ***The reality of concurrent new product development***; Cardiff Business School, University of Wales, Cardiff, UK, 1998.

Melvin J. and Suh N.P. (2002); ***Beyond the Hierarchy: System Wide rearrangement as a Tool to Eliminate Iteration***; International Conference on Axiomatic Design, 2002.

Melvin J.W. (2003); ***Axiomatic System Design: Chemical Mechanical Polishing Machine Case Study***; PhD thesis; Massachusetts Institute Of Technology; February 2003.

Mil-Hdbk-61 (1997); ***Configuration Management Guidance***; US DoD; 30 September 1997.

Military Handbook – 189 (1981); ***Reliability growth management;*** USA Army Communications Research and Development; Fort Monmouth, NJ, Department of Defence, Washington DC. 13 February 1981.

Military Handbook-472 (1966); ***Maintainability prediction***; Department of Defence; 24 May 1966.

Military Standard 1369A (1988); ***Integrated Logistic Support programme Requirements***; Department of Defence; October 1988.

Military Standard 1388 2B (1993); ***DoD Requirements for a Logistic Support Analysis Record***; Department of Defence; 21 January 1993.

Military Standard 13881A (1990); ***Logistic Support Analysis***; Department of Defence; 31 July 1990.

Mil-Std 490A (1985); ***Specification Practices***; US DoD Mil-Std 490A; 4 June 1985.

Mil-Std-1521, (1995); ***Technical reviews and Audits for systems, equipment and computer software***; Military Standard 1521, Notice 3; April 10, 1995.

Mil-Std-499B (1994); ***Systems Engineering***; Joint OSD Services Industry, 6 May 94.

Mobley K. (1999); ***Root Cause Failure Analysis***; Butterworth-Heinemann; ISBN 0750671580; 1999.

Morgan D.L. (1997); ***Focus Group Guide Book***; ISBN 0-7619-0760-2; Sage Publications Inc; 1997.

NASA (2004); ***NASA System Engineering Manual***; Version 3.0; 30

September 2004.

NASA (2007); ***NASA Systems Engineering Handbook***; NSA/SP-2007-6105 Rev 1; National Aeronautics and Space Administration; NSA Headquarters, Washington D.C. 20546; December 2007.

NASA FTA Handbook (2002); ***Fault Tree Handbook with Aerospace Applications***; NASA Office of Safety and Mission Assurance, NASA Headquarters, Washington DC, 20546; August 2002.

NASA FTA handbook, (2002); ***Fault Tree Analysis Handbook with Aerospace Applications***; NASA Office of Safety and Mission Assurance; NASA Headquarters; Washington DC 20546.; August, 2002.

NASA PD-ED-1255, (2004); ***Preferred Reliability Practices: Problem Reporting and Corrective Action System (PRACAS)***," practice NO. PD-ED-1255, National Aeronautics and Space Administration (NASA), February 2004.

NASA-HDBK (2008); ***Procedural Handbook for NASA Program and Project Management of Problems, Non-conformances and Anomalies***; NASA-HDBK-8939.18; 2008.

NAVSO P-3686, (1998); ***Top Ten ways to Manage Technical Risk***; Department of the US Navy; NAVSO P-3686; October 1998.

NAVSO P-6071 (1986); ***Best Practices: How to avoid Surprises in the World's Most Complex Technical Process***; Department of the US Navy;; March 1986.

Pennell L.W. and Knight F.L. (2005); ***System Engineering***; Aerospace Report No TOR-2005(8583)-3; Space and Missile System Center; 15 April 2005.

PMBOK (1996); ***A Guide to the Project Management Body of Knowledge PMBOK)***; PMI Standards Committee, PMI, Newtown Square, PA 19073-3299, USA, 1996.

PMBOK (2004); ***A Guide to the Project Management Body of Knowledge PMBOK)***; Third Edition; PMI Standards Committee, PMI, 2004.

PMBOK (2008); ***A Guide to the Project Management Body of Knowledge PMBOK)***; Fourth Edition; PMI Standards Committee, PMI, ISBN 9781033890517.

Prabhakar D.N., Rausand M.M. and Osteras T. (2008); ***Product Reliability – Specification and Performance***; ISBN 978-1-84800-

271-5Springer-Verlag London, 2008.

Pretorius L, Wessels A and Rooney A.C. (2007); ***Design Management for Project Success***; Proceedings of the IEEE; 2007.
Pritsker, A.A.B. (1966); ***Graphical Evaluation and Review Technique (GERT)***; RM-4973-NASA., April 1966.

Rasmussen W. (2009); ***Definition of Project Management***; www.wrasmussen.gov.ck/index.php; 30 September 2009.

RELEX® is supplied under licence by RELEX Software Corporation, 540 Pellis road, Greensburg, PA 15601, USA.
www.relexsoftware.com , June 2009.

Roach D. (2010); ***6 Reasons Teams Fail***; http://ezinearticles.com/?6-Reasons-Teams-Fail&id=4287701; 5 August 2010.

Roos S.D. (2001); ***A model for complex product development using integrated product and support development criteria*****;** Doctoral thesis, Rand Afrikaans University, 2001.

Ryabov V. (2009); ***Research methods, Part II***; Power point presentation; Principal Lecturer in Information Technology Kemi-Tornio University of Applied Sciences; 2009.

Saaksvuori A. and Immonen A. (2005); ***Product Lifecycle Management***; ISBN 978-3-540-25731-8; Springer Heidelberg, 2005.

Sahlin M. (2000); ***A Systematic Approach For Decision Making In A Concurrent Engineering Environment***; Proceedings of ICAD2000 First International Conference on Axiomatic Design, ICAD048 Cambridge, MA – June 21-23, 2000.

SANDF (2009)**;** ***SANDF Armour Formation*****;** http://www.army.mil.za/hq_units/armour_fmn/structure.htm; 30 March 2009.

SANDF (2009); ***Ratel Armoured Vehicle***; http://www.sa-transport.co.za/military/army/ratel_zt3_01_jd03.JPG.

Sanjay L.A. and Dreyfus P. (2000); ***The impact of Design Management and Process Management on Quality: an Empirical Investigation***; Journal of Operations Management volume 18, 2000, p549-575.

Seusy C.J. (1988); ***Reliability Growth Management in Non Military Industry***; Reliability Growth Processes and Management Institute of Environmental Sciences; Mount Prospect; Illinois 60056, P37 – 45. Reliability Growth conference.

Smirnoff J.P. (2006); ***The Impact Of Economic Factors And acquisition Reforms On The Cost Of Defense Weapon Systems***; Thesis, Air Force Institute of Technology; March 2006.

Smith R.P. and Eppinger S.D. (1997); ***Identifying controlling features of Engineering Design Iterations***; Management volume 43, No 3, March 1997.

Sommerville (1996); ***Software Engineering***;, Fifth Edition, Lancaster University, Addison Wesley, England. ISBN 0-201-42765-6.

Sparrius A. (2006); ***The Dynamic Behaviour of a System; Advanced Systems Engineering course note***; ASE_3 rev4-0.pub; December 2006.

Sparrius A. (2008); ***Emergent Properties***; The South African Mechanical Engineer Volume 58; June 2008.

Spolsky J. (2007); ***Seven steps to remarkable customer service***; www.joelonsoftware.com/articles/customerservice.html; 2007.

Stake, R. (1995); ***The Art of Case Research***; Newbury Park, CA: Sage Publications.

Standard No. 5F, (1982); ***Standard Guides for Preparation of Proposed Item Logistics Data Records***; Federal Standard No. 5F.

Starbek M. and, Grum J. (2002); ***Concurrent engineering in small companies***; University of Ljubljana, Faculty of Mechanical Engineering, Ljubljana, Slovenia; International Journal of Machine Tools and Manufacture; Volume 42, Issue 3, February 2002, Pages 417-426.

Steyn H. (2009); ***Why are capital projects often late and over-spent? Putting the puzzle together***; Innovate No 3, 2009; University of Pretoria.
Suh N. P. (1990); ***The Principles of Design***; Oxford University Press, New York; 1990.

Terman F.E. (1955); ***Electronic and Radio Engineering***; McGraw-Hill; 1955.

Thunnissen D.P. (2005); ***Propagating and Mitigating Uncertainty in the Design of Complex Multidisciplinary Systems***; PhD thesis; California Institute of Technology Pasadena, California, 2005.

Tomaiko T.A. (2008); ***Improving The U.S. Navy's Execution Of Technical Authority Through A Common Risk Management And Technical Assessment Process***; Thesis; Naval Postgraduate School Monterey, California; 2008.

Torczon V. and Trosset M.W. (2007); ***Using Approximations to Accelerate Engineering Design Optimization***; American Institute of Aeronautics and Austronautics; AIAA – 98-4800.

Trochim W.M.K. and Donnolly J. (2006); ***Research Methods Knowledge Base***; 3rd Edition; ISBN 1592602916; 2006.

v/d Merwe, (2002); ***A Product Development Process for a Photovoltaic Water Pump Installation in Small to Medium Enterprise***; PhD thesis; Rand Afrikaans University, July 2002.

van Aken J.E. (2004); ***Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-Tested and Grounded Technological Rules***; Eindhoven University of Technology; Journal of Management Studies 41:2 March 2004; 0022-2380.

Venable J.R. and Travis J. (1999); ***Using a Group Support System for the Distributed Application of Soft Systems Methodology***; Proceedings of the 10th Australasian Conference in Information Systems, Wellington, New Zealand, December, 1999.

Viljoen G. (2007); ***System Design***, INCOSE International conference, Pretoria, Aug 2007.

von Bertalanffy Ludwig (1968); ***General System Theory Foundations, Development, Applications***; Publisher: George Braziller Inc, New York.

Wessels A (1997); ***The management of reliability in a multi-level support environment***, Masters thesis RAU; 1997.

Wessels A. and Pretorius L. (1998); ***Reliability Management in a multi-level support environment***; Proceedings of the 14th International Logistics Congress, 1998.

Wessels A. and Pretorius L. (2011); "***Impact Of Design Changes In a Concurrent Engineering Development Environment***"; ISEM Proceedings, September 2011, P33 1-17.

Wikipedia (2009); ***Storyboard***; http://en.wikipedia.org/wiki/Storyboard; September 2009.

Wikipedia (2010) ***Fire power*** ; www.wikipedia.org , March 2010.

Wikipedia (2010); ***Anti-tank warfare*** ; www.wikipedia.org , March 2010.

Wilson P.F. , Dell L.D. and Anderson G.F. (1993); ***Root Cause***

Analysis: A Tool for Total Quality Management; ISBN 0-87389-163-5, 1993; American Society for Quality (ASQ).

Wooley M. and Choi J. (2003); ***Analyzing Multiple Component Failures in a System***; Circuits Assembly; December 2003.

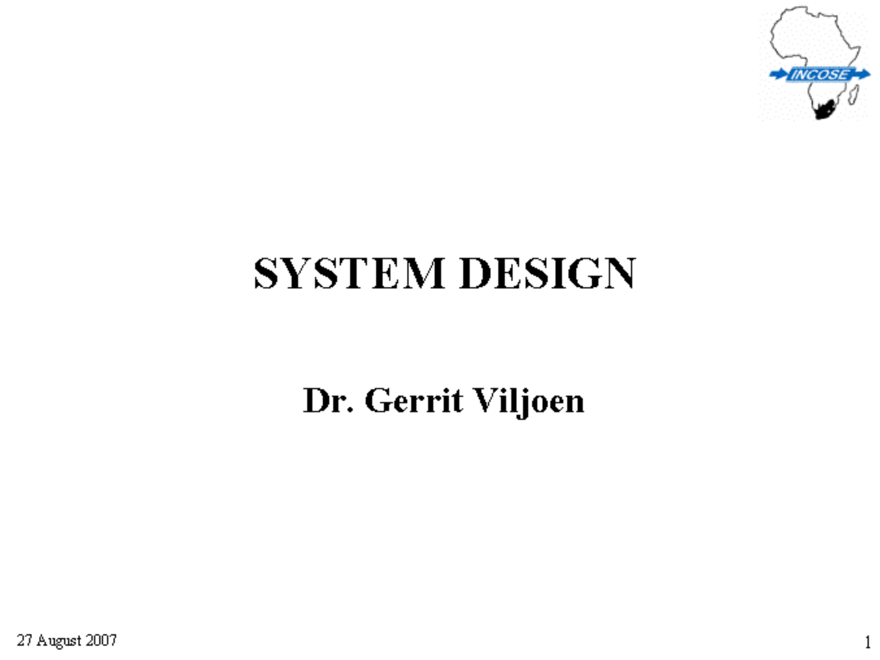
Yang K. and Zhang H. (2000); ***A comparison of TRIZ and Axiomatic Design***; Proceedings of ICAD2000 ICAD56; June 2000.

Yassine A and Braha D. (2003); ***Complex Concurrent Engineering and the Design Structure Matrix Method***; Concurrent Engineering 2003; 11; 165.

Yin, R (1994); ***Case Study Research: Design Methods***; 2nd Edition. Thousand Oaks, CA: Sage Publications.

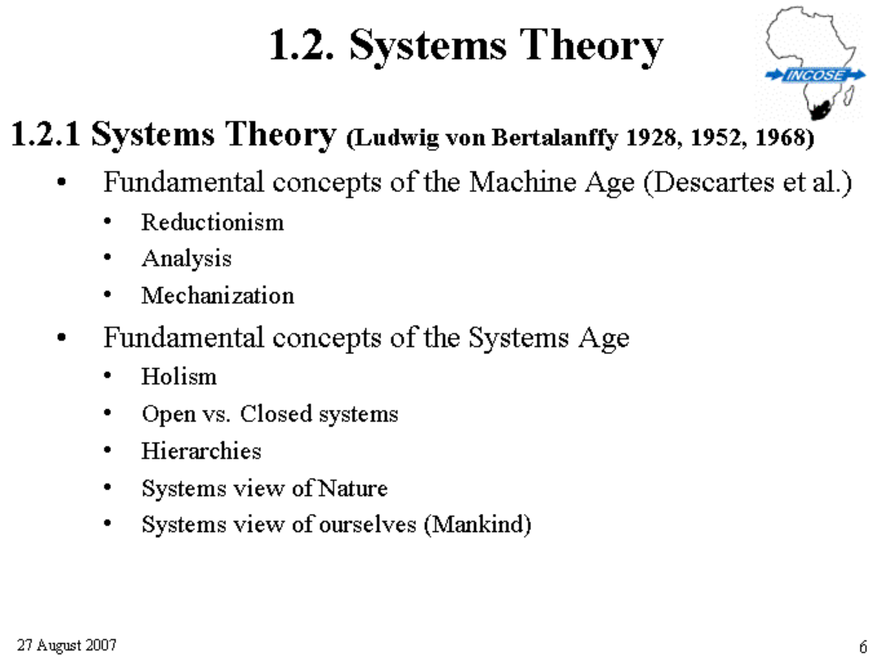
Appendix A System Dynamics

Extract of the presentation on System \Design analysing system dynamics by Dr G Viljoen presented at the INCOSE conference during August 2007, [22].



SYSTEM DESIGN

Dr. Gerrit Viljoen



1.2. Systems Theory

1.2.1 Systems Theory (Ludwig von Bertalanffy 1928, 1952, 1968)

- Fundamental concepts of the Machine Age (Descartes et al.)
  - Reductionism
  - Analysis
  - Mechanization
- Fundamental concepts of the Systems Age
  - Holism
  - Open vs. Closed systems
  - Hierarchies
  - Systems view of Nature
  - Systems view of ourselves (Mankind)

1.2. Systems theory

1.2.2 Systems Theory (Ludwig von Bertalanffy 1928, 1952, 1968)

- A dynamic system can often be described mathematically as follows:
  - If Q_i is the ith state that describes the p elements of the system we have:

$$
\begin{aligned}
\frac{dQ_1}{dt} &= f_1(Q_1, Q_2, \ldots, Q_n) \\
\frac{dQ_2}{dt} &= f_2(Q_1, Q_2, \ldots, Q_n) \\
&\ldots \\
\frac{dQ_n}{dt} &= f_n(Q_1, Q_2, \ldots, Q_n)
\end{aligned}
\tag{1}
$$

1.2. Systems theory

1.2.3 Systems Theory (Cont.)

- This system has equilibrium points which can be stable, or unstable. At the equilibrium point there is no change in the system states, so we have:

$$f_1 = f_2 = \ldots = f_n = 0$$

- We then have n equations for n variables that can be solved:

$$Q_1 = Q_1^*, \quad Q_2 = Q_2^*, \quad \ldots, \; Q_n = Q_n^*$$

- If we introduce a new variable which represents a perturbation around the equilibrium $Q_i = Q_i^* - Q_i'$, we can reformulate the system in (1) with respect to Q_i' and then do a Taylor expansion:

1.2. Systems theory

1.2.4 Systems Theory (Cont.)

$$\frac{dQ_1^{'}}{dt} = a_{11}Q_1^{'} + a_{12}Q_2^{'} + \ldots + a_{1n}Q_n^{'} + a_{111}Q_1^{'2} + a_{112}Q_1^{'}Q_2^{'} + a_{122}Q_2^{'2} + \ldots$$

$$\frac{dQ_2^{'}}{dt} = a_{21}Q_1^{'} + a_{22}Q_2^{'} + \ldots + a_{2n}Q_n^{'} + a_{211}Q_1^{'2} + a_{212}Q_1^{'}Q_2^{'} + a_{222}Q_2^{'2} + \ldots$$

$$\ldots$$

$$\frac{dQ_n^{'}}{dt} = a_{n1}Q_1^{'} + a_{n2}Q_2^{'} + \ldots + a_{nn}Q_n^{'} + a_{n11}Q_1^{'2} + a_{n12}Q_1^{'}Q_2^{'} + a_{n22}Q_2^{'2} + \ldots$$

- A general solution of this system of equations is:

$$Q_1^{'} = G_{11}e^{\lambda_1 t} + G_{12}e^{\lambda_2 t} + \ldots G_{1n}e^{\lambda_n t} + G_{111}e^{2\lambda_1 t} + \ldots$$

$$Q_2^{'} = G_{21}e^{\lambda_1 t} + G_{22}e^{\lambda_2 t} + \ldots G_{2n}e^{\lambda_n t} + G_{211}e^{2\lambda_1 t} + \ldots$$

$$\ldots.$$

$$Q_n^{'} = G_{n1}e^{\lambda_1 t} + G_{n2}e^{\lambda_2 t} + \ldots G_{nn}e^{\lambda_n t} + G_{n11}e^{2\lambda_1 t} + \ldots$$

1.2. Systems theory

1.2.5 Systems Theory (Cont.)

- Where G are constants and λ the roots of the characteristic equation:

$$\begin{vmatrix} a_{11}-\lambda & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22}-\lambda & \ldots & a_{2n} \\ \ldots & & & \\ a_{n1} & a_{n2} & \ldots & a_{nn}-\lambda \end{vmatrix} = 0$$

- Inspection of the roots allow a number of conclusions to be drawn about the system. If all the real parts are negative, the system is stable. If the roots are imaginary with negative real parts, the system is asymptotically stable. If there are any real roots that are positive, the system is unstable.
- These effects can be graphically described on the phase plane

Apendix B Problems experienced

1. Management related project problems

Problem Area	Problem Statement	Cause	Impact
Project Meetings	Project meetings particularly at subsystem level not always regularly held.	Time constraints and major role players' availability	5
Facility meetings	Facilities meetings not structured to cater for the specific needs of individual aspects of the projects	Facilities in a matrix organisation generally provide resources for a number of projects, making it impractical to cater for the individual aspects of a specific project.	4
Guidance Meeting	Development personnel in general not familiar with the lower level client requirements and the user environment.	No initial guidance meetings were held.	5
Lack of knowledge of the user operational doctrines and support environment	Personnel involved with the development of the logistic products did not always understand how the user operates during a typical deployment exercise.	Initial project focus was technical requirements and no study of the operational and support environment was done.	4
Lack of knowledge of the user training doctrines and training environment	Personnel involved with the development of training do not always understood the details on how the user training project functions.	Initial project focus was technical requirements and no study of the operational and support environment was done.	4
Scope creep	The requirements baseline at the lower system hierarchies was not fixed.	Particularly the lower system hierarchy did not always receive the full PM and SE attention and sometimes leading to unplanned baseline shifts.	4
Resources	The extent and scope of the logistic product development was under estimated due to a wrong assumption that the existing pre-upgrade ZT3 products could be adapted.	The project team did not pickup that the client changed their documentation and training material standards since the original ZT3 system, resulting in the logistic product development team being under resourced and over worked, contributing to expensive mistakes and quality problems. There was no spare capacity or contingency planning	4
Increase level of effort (LOE) for source info	Wrong information was sourced due to sub-contractor inputs.	Unfamiliarity with the new client technical manuals and training standards and wrong selection of expert consultant	3
Client Guidance	Availability of expert client personnel	The client has changed its standards since the original ZT3 system	2
Clarity of logistic requirements	The internal technical authors were not familiar with the new client documentation and training material standards.	The client has changed its standards since the original ZT3 system	3

Problem Area	Problem Statement	Cause	Impact
Outsourcing technical manuals and training definition phase	Wrong information was sourced due to sub-contractor inputs.	The scope of work for the consultant tasked with the development of the technical manuals and training definition work was inadequate resulting in the wrong sub-contractor being selected	2
Client management	Sound client management was sometimes lacking leading to unplanned baseline shifts.	The impact of ostensibly small change requests was sometimes under estimated at PM level and allowed without detailed impact investigation.	3
Tasking (Task Structuring)	Not enough detail task structuring for logistic product development because task managers themselves did not always fully understood the task. Also no single clearly demarcated responsibility areas for logistic product development team members leading to inefficiency and quality problems.	Unfamiliarity with the new client technical manuals and training standards.	2
Logistics engineering availability	Logistics Engineering was not contracted to perform assessments of the technical integrity of the logistic products development team outputs.	PM did not make provision for engineering assistance during technical manuals and training material development.	2
System Engineering and subsystem expertise availability	The development team availability particularly during the final system qualification phase was very limited.	PM wrongly assumed that the technical information in the product data pack would be adequate for the technical authors.	3
System Engineering and subsystem expertise availability	System Engineering not sensitive regarding their responsibility towards the Logistic Development Process.	PM wrongly assumed that the technical information in the product data pack would be adequate for the technical authors.	2
PRACA form completion not enforced	Problems were not made visible to enable effective management and co-ordination resulting in persistence of a problem.	PM due to pressure of work did not always schedule regular FRB meetings to timeously address and resolve problems.	3
Contracting / Projects	Inadequate facility contracting	Contracting and in particular outputs were sometimes vague.	3
CFE Data integrity	Source information in particular Customer Furnished Items (CFE) data was not verified up front leading to expensive time consuming re-work later in the project.	PM wrongly assumed that the CFE data supplied by the client was adequate and complete.	3
Planning	Detail planning lacking resulting in inability to measure progress and timeous identified problem areas.	PM due to pressure of work did not always integrate the detail task planning into their project plan.	3
Planning	Task managers not always involved.	PM time constraints.	3
Cost Management	General task overspending.	Task overspending was not always immediately followed-up by PM.	2
Cost Management	Full scope of CFE impact not always visible.	The full scope of CFE generally only became visible late in the program	2

Problem Area	Problem Statement	Cause	Impact
Risk Management	Risks were not identified and managed early enough in the program.	Risk management started too late.	2
Procurement delays	The company rule required that all procurement including engineering proto type components be handled by the company procurement section.	Procurement priorities did not make special provision for small quantity highly specialised development contract procurement items.	3
Technical manuals and training development	Technical authors were not always au fait with the prescribed templates and standards.	Lack of editorial assistance and training.	3
Process	No clear technical manual and training material development process that all team members could follow.	The new client manual and training standard aggravated the problem.	3
Change Control	ECPs were generally not reviewed and counter signed by all the stake holders.	Stakeholder availability.	4
Discrepancies between hardware and data	Redlining of drawings sometimes lagged behind.	Tight schedules and pressure for hardware availability.	5
Fragmented Quality Assurance function	Each facility provided its own QA function.	Company rules did allow indirect personnel on direct programs.	3
Fragmented configuration function	Each facility provided its own configuration management function.	Company rules did allow indirect personnel on direct programs. Project QA manager primarily focussed on top level system QA.	3
Fragmented procurement function	Procurement was a centralised corporate function geared for production procurement.	Company rules did allow indirect personnel on direct programs.	3
Incidents	32	**Sub Total**	**100**

2. Systems Engineering related project problems

Problem Area	Problem Statement	Cause	Impact
Data availability	The system development data pack only addresses the “**WHAT**” the system must do, not the “**HOW**” the system works. .	The systems engineering process is requirements driven. No provision is made for documenting **HOW** the designs work and interact within the rest of the system.	5
System Engineering and subsystem expertise availability	The development team availability particularly during the final system qualification phase is very limited.	Expert resources a scarce resource and once a task is complete, the resource is allocated to another project under the matrix organisational structure.	4
System Functional Block Diagram (FBS) not inline with product Breakdown Structure (PBS)	There was virtually no link between the system FBS and final PBS. The PBS also kept changing during the program. These uncoordinated changes resulted in much fruitless work and reworks of documentation.	Design iterations resulted in changes being accepted and implemented but retrospective data pack updates were not always done due to pressure of work.	4
Maturity of System	Log Products were required before the acceptance and freezing of the system production baseline.	Design iterations (ECPs) resulted in subsystem baseline changes that were not incorporated into the log products development schedules.	4

Problem Area	Problem Statement	Cause	Impact
Hardware availability	Tight schedules and technical development program slips resulted in lack of verification of log products by the log product development team.	Primarily due to cost and time constraints only one demonstration model of each hardware item was built. This resulted in a conflict between qualification testing and log product verification.	4
Hardware availability	The hardware was not available for the documentation development team	Hardware was continuously modified during qualification testing and not released for other use until late in the project. Also the documentation redlining process often lagged behind the hardware status.	4
Terminology List updates	A large number of log product changes late in the project were as a result of terminology changes. The impact of these on cost and schedules were generally under estimated by all parties involved.	The ripple effect of ostensibly minor terminology changes was under estimated.	1
Logistics engineering availability	Logistic engineering expert availability particularly during the final system qualification phase is very limited.	Expert resources a scarce resource and once a task is complete, the resource is allocated to another project under the matrix organisational structure.	2
PRACA form completion not enforced	Problems were not made visible to enable effective management and co-ordination resulting in persistence of a problem.	SE due to pressure of work did not always schedule regular FRB meetings to timeously address and resolve problems.	1
Change Control	ECPs were generally not reviewed and counter signed by all the stake holders.	Stakeholder availability, time constraints.	3
CFE Data integrity	Source information in particular Customer Furnished Items (CFE) data was not verified up front leading to expensive time consuming re-work later in the project.	The CFE data information was accepted on face value without verification.	2
Electronic data control	Drawing office sometimes placed data on the server that was not always verified and approved.	Schedule pressure and fragmented configuration management.	1
Incidents	12	**Sub Total**	**35**

3. Quality Assurance related project problems

Problem Area	Problem Statement	Cause	Impact
Client documentation and training standards	QA not familiar with logistic product requirements and the relevant RSA-MIL-STDs.	Fragmented Quality Assurance function as a result of the matrix organisation structure and company rule of no indirect personnel on direct programs.	3
Technical manuals template problems	QA not familiar with logistic product requirements and the relevant RSA-MIL-STDs.	This is a specialist field outside the scope of general QA.	2

Problem Area	Problem Statement	Cause	Impact
Change Control	ECPs were generally not reviewed and counter signed by all the stake holders.	Stakeholder availability and fragmented QA function.	1
Continuous Evaluation	QA not contracted for continuous evaluation of the logistic products throughout the development phase.	Facility workloads prevented continuous evaluation of one project output.	2
Continuous Evaluation	QA was generally passive and only involved with the project hardware instead of all deliverables (including the log deliverables).	Facility workload.	2
Continuous Evaluation	QA has not been contracted for the evaluation of logistic products.	QA was considered an indirect function as part of a facility.	2
Continuous Evaluation	QA is not competent to evaluate the technical integrity of the Technical Manuals and Training Material.	Limited specialist availability as part of the matrix organisation structure.	3
Incidents	7	**Sub Total**	**15**

4. Configuration Management related project problems

Problem Area	Problem Statement	Cause	Impact
Baseline/ Moving	The baseline kept changing due to enforced ECPs creating difficulty with the log development. (Moving target).	Problems at integration levels very often led to urgent ECPs to resolve the problem. This had a ripple effect throughout the system hierarchy.	4
Data availability	The system development data pack only addresses the "**WHAT**" the system must do, not the "**HOW**" the system works.	The systems engineering process is requirements driven. No provision is made for documenting **HOW** the designs work and interact within the rest of the system.	2
System Block Diagram not in line with PBS	There was virtually no link between the system FBS and final PBS. The PBS also kept changing during the program. These uncoordinated changes resulted in much fruitless work and reworks of documentation.	Design iterations resulted in changes being accepted and implemented but retrospective data pack updates were not always done due to pressure of work.	3
Source info	**T**he technical authors had difficulty in getting data on how the system/subsystem worked.	The systems engineering process is requirements driven. No provision is made for documenting **HOW** the designs work and interact within the rest of the system.	2
Source info	**T**he technical authors had difficulty to get access to the design experts.	After the CDR, facilities immediately reallocate expert resources to other programs.	2
Fragmented CM	Each facility had its own CM. Only at program level was the program CM who could not go into detail of all the subsystem configuration aspects.	Documentation configuration control was part of the company management information system and available on the intranet. Project CM was then reduced to a coordination and audit function.	3

Problem Area	Problem Statement	Cause	Impact
Templates	Technical manual templates were not user-friendly leading to time wasting mistakes by technical authors.	Unfamiliarity with the new client technical manual led to outsourcing of templates.	2
Templates	Templates were not approved prior to population of data.	Templates were accepted at face value and not put through an approval process with the client in order to save time.	3
Templates	Tolerances of the templates must be realistic and within the capabilities of the printers and PCs used in the facility.	The page units of measure in the client technical documentation standards were metric rather than point units for printer drivers. Client QA personnel were rejecting pages due to printer tolerances. Despite the templates being correct.	3
Change Control	ECPs were generally not reviewed and counter signed by all the stake holders.	Stakeholder availability, time constraints.	2
Incidents	**10**	**Sub Total**	**26**
Total	**61**	**Total**	**176**

Summary:

Problem area	Consolidated incidents	Total impact	average %
Management related project problems	32	100	56.8
System Engineering related project problems	12	35	19.9
Quality Assurance (QA) related project problems	7	15	8.5
Configuration management (CM) related project problems	10	26	14.8
Total	**61**	**176**	**100**

Apendix C Design Iteration Impact Study

1. Introduction

Design influencing during the optimization of a design for a configuration item (CI) of a complex system results in design iterations of the item. These iterations are not always contained to the configuration item itself. Project pressures may result in the premature release of a design for that particular CI. Latent design defects in the CI may only surface during the integration process of the system under development. Under concurrent engineering methodology a number of other CIs are being developed concurrently. Any change of the Form-Fit and Function (FFF) of a CI after its baseline has been frozen generally results in a ripple effect throughout the system hierarchy by affecting the FFF of all the other CIs under development. The impact of these system design changes is related to the complexity of the system and the functional couplings between the different system elements. In this study an attempt is made to quantify the impact or ripple effect of a CI change to the whole system.

2. Unconstrained design iterations – ideal design environment

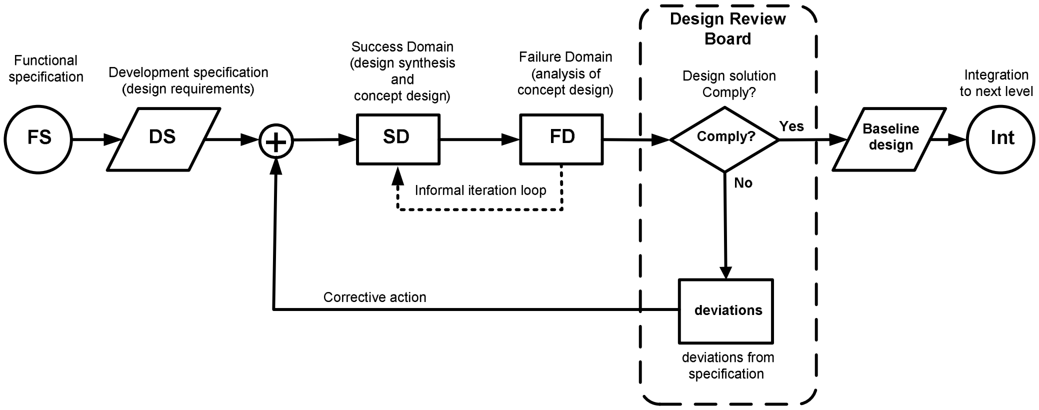


Figure 36: Unconstrained *effect-to-cause* design influencing model

The design engineer as part of the Success Domain team by synthesis of the requirements and constraints produces a draft design. The logistic engineering analysts as part of the Failure Domain team analyse the draft design for the "***-ility***"[10] performance against the requirements. The Design Review Board (DRB) refers any shortcomings or deviations from the requirements back to the Success Domain team for another design iteration. This iterative design process continues until the design complies with all the requirements and the design is base-lined in preparation for the next level of integration. The number of iterations required is generally determined by the maturity of the technology selected and the technical complexity of the design. The Failure Domain team can only perform the analysis ***after*** a concept design has been provided by the Success Domain team. In other words design influencing is an after the fact or an ***"effect-to-cause"*** process.

3. Constrained design iterations – real life environment

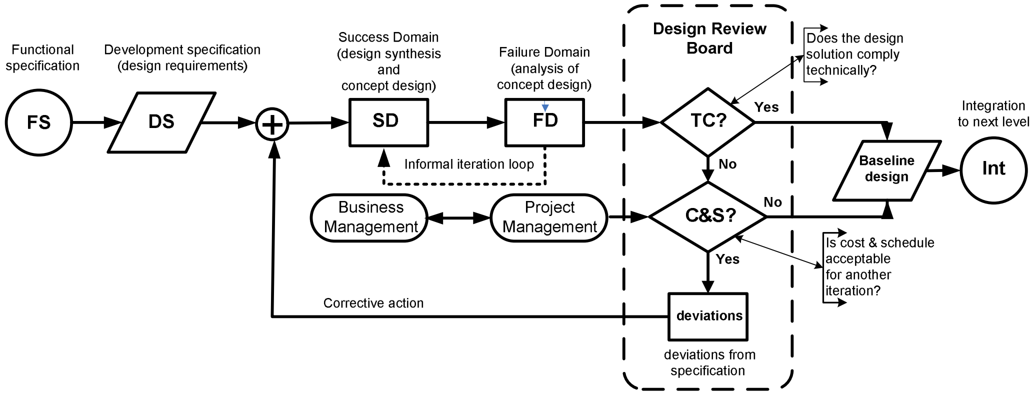


Figure 37: Constrained *effect-to-cause* design influencing model

The iterative design process for a constrained ***"effect-to-cause"*** design influencing model is identical to the unconstrained design process with the addition of a gate in the iterative design process by the project manager. The project manager, depending on his constraints, generally cost and schedule, can allow another design iteration or force a ***premature*** design release. The design is therefore not fully mature to the satisfaction of the Success Domain and Failure Domain teams and may increase the technical risk at the next level of integration.

4. Design change impact in a complex hierarchical system

A complex system generally has several layers in its hierarchical structure. Individual configuration items (CI) in the system structure may have functional coupling to other system CIs. Also real systems are a multi-dimensional hierarchy of functions e.g. the logistic system hierarchy actually lies behind the operational system hierarchy with functional couplings between them. For convenience and simplicity, it is customary to present the logistic system, software system, hydraulic system, pneumatic system, and optical system, etc. hierarchies on separate hierarchy structures leading to the misconception that these hierarchies are separate and independent when in actual fact they are not. In practice most systems are a mix of functional coupled and decoupled functions between the CIs in the different hierarchies of a system.

A hypothetical system presented by means of a simple two-dimensional hierarchy is shown in figure 38.

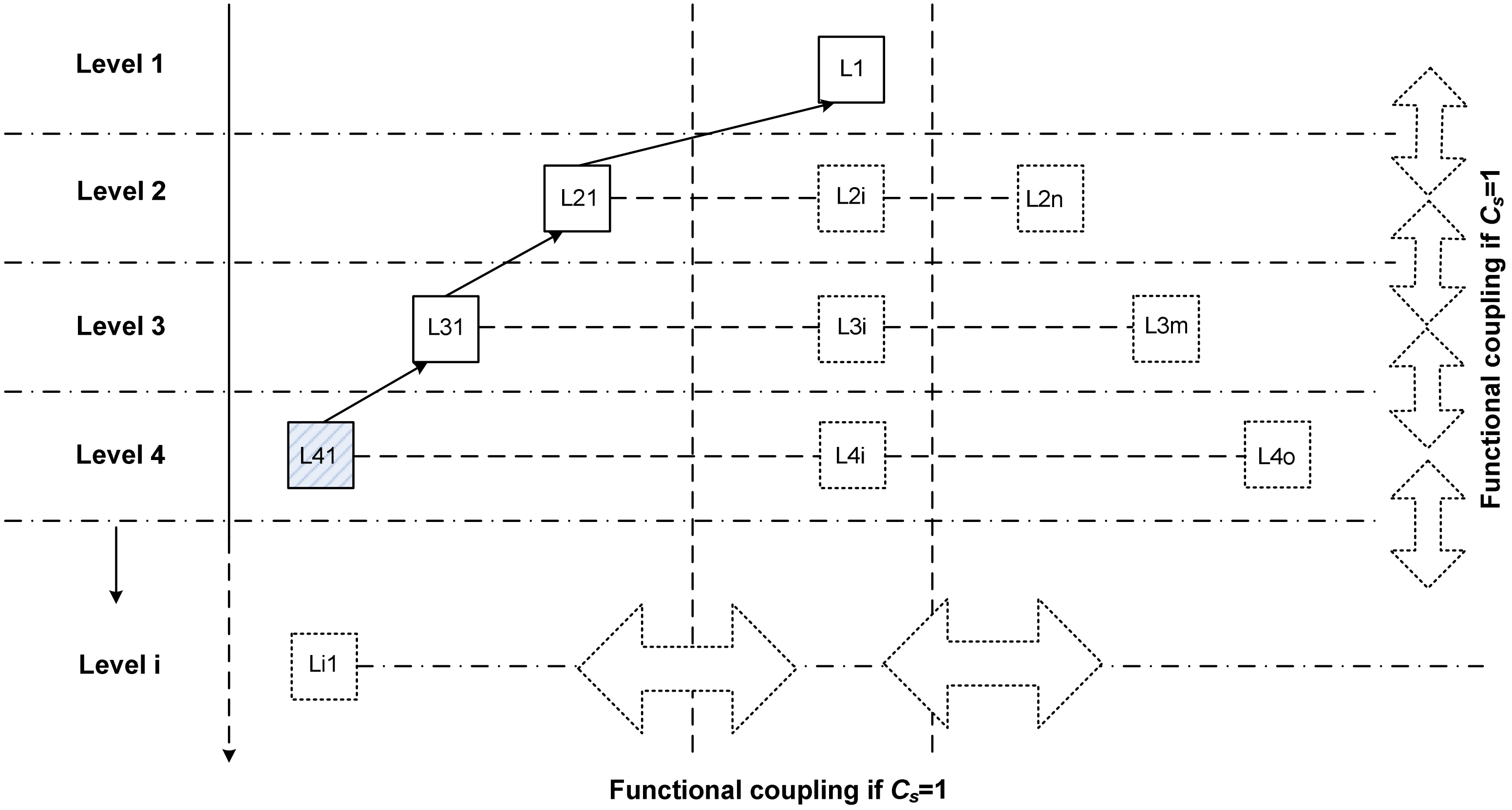


Figure 38: Hypothetical system hierarchy

Functional Coupling rules

- If there is functional coupling between affected CIs, the coupling constant C_s=1.

- If there is no functional coupling between affected CIs, the coupling constant C_s=0.

- There is ***always*** functional coupling between an affected CI and its own parent and C_s=1 in that case (emergent properties).

- There is ***always*** functional coupling between an affected CI and its own children and C_s=1 in all those cases (emergent properties).

- There may be functional coupling between the affected CI and its peers, other parents and children, in all those cases C_s=1.

- If there is no functional coupling between the affected CI and any other CI in the system hierarchy then, C_s=0.

A. Impact of CI change of an affected CI on its parents and children:

Using the system hierarchy in figure 38, assume that the affected CI is L41.

Then L31, L21 and L1 are the parents and functional coupling exists and therefore C_s = 1.

Similarly, L51 to Li1 are the children of the affected CI L41 and C_s=1.

Let R_p be the impact of an affected configuration.

Then

$$R_p = C_s L_1 + C_s L_{21} + C_s l_{31} + C_s L_{41} + - + - + C_s L_{i1}$$

$$R_p = \sum_{i-1}^{l} C_s L_{i1} \qquad (1)$$

Where *l* is the total parent and children CIs and *i* is a real integer reflecting the parent or child CI.

Note

Equation (1) ≥ 1

B. General Impact of CI change in the system hierarchy

Let R_c be the impact due to functional couplings

Then $$R_c = R_p + \sum_{j=1}^{m} C_s L_j$$

Where

- m is the total configuration items in the system structure not related to the affected CI structure.

- j is an integer reflecting the j_{th} configuration item in the system structure.

- C_s is the functional coupling (C_s=0 if functionally decoupled or C_s=1 if functionally coupled).

Generalising $$R_c = R_p + \sum_{j=1}^{m} C_s L_j$$

$$R_c = \sum_{i=1}^{l} C_s L_{i1} + \sum_{j=1}^{m} C_s L_j$$

Since $l+m = n$

$$R_c = \sum_{k=1}^{n} C_s \, L_k \qquad (2)$$

- Where n is the total number of configuration items in the system.

- C_s=1 for all the configuration items where functional coupling exist between the affected configuration item.

- C_s=0 for those configuration items where no functional coupling exist with the affected configuration item.

C. Summary

- From equation (2), it can be deduced that in order to reduce the ripple effect of a design change to a configuration item, a design objective should be to minimise equation (2) by avoiding functional couplings between configuration items.

- Since equation (1) is always) ≥ 1 it precludes equation (2) from ever becoming zero.

- Totally decoupled designs can only be found in simple single hierarchical level systems such as components or simple products.

D. Some Case-study Examples

The following case-study calculations are based on hypothetical figures to illustrate the cost and schedule impact of design changes in a concurrent engineering environment. The two examples also illustrate the advantage of de- or uncoupled designs.

A simple 3 level system hierarchy structure with 9 CIs at the lowest level was considered as a case-study, refer to figure 39. The summarized findings for the case-study examples in appendix C are:

- A simple 3 level system hierarchy structure with 9 CIs at the lowest level and with minimum functional couplings. This is considered a best-case system design of only functional couplings between parent and children and no peer functional couplings. If these remaining functional couplings for example were to be removed there would be no system but only a collection of CIs without any emergent properties.

- Using the same simple 3 level system hierarchy as above, but this time all the CIs are functionally coupled to one another providing a worst-case system hierarchy design, refer to figure 40.

General assumptions:

- Sufficient design iterations to achieve design optimisation and maturity

- Level 1 (L1) Design iteration: cost=ZAR1000k/iteration
  Schedule=18 weeks/iteration

- Level 2 (L2x) Design iterations cost=ZAR500k/iteration
  Schedule=12 weeks/iteration

- Level 3 (L2xy) Design iterations cost=ZAR100k/iteration
  Schedule=6 weeks/iteration

- No concurrent iterations are possible

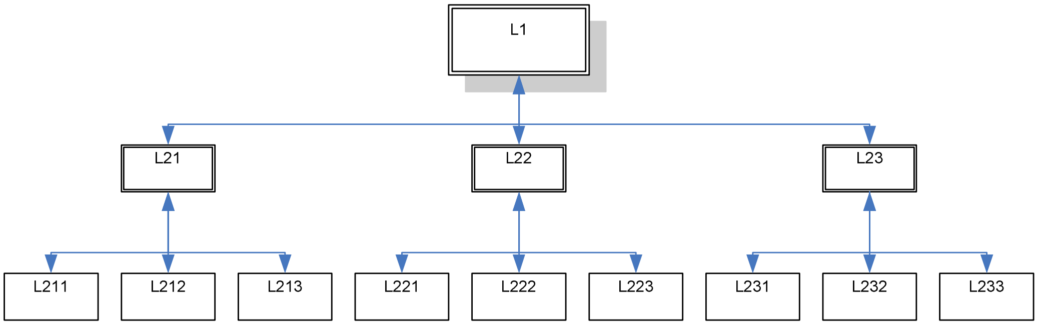


Figure 39: System structure with maximum functional decoupling

Assuming the affected CI is L211, using equation (2), C_s = 1 for the functional coupling to the parent L21 and its parent L1.

Cost impact: R100k + R500k + R1000k = **ZAR1600k**
Schedule impact: 6 + 12 + 18 = **36 weeks**

Taking the same hierarchical system structure but with functional coupling between all the configuration items shown in figure 40:

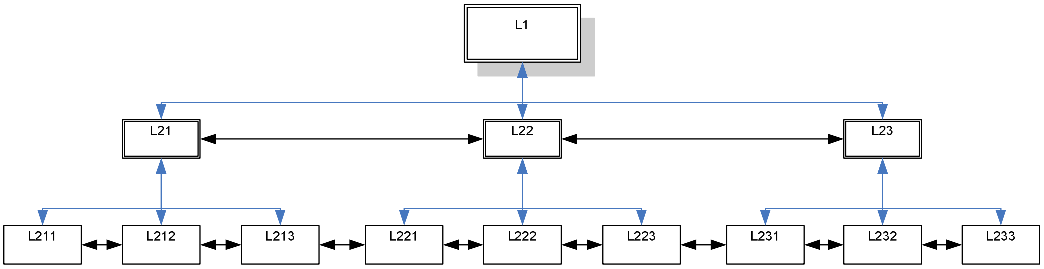


Figure 40: System structure with maximum functional coupling

Again assuming the affected CI is L211, using equation (2), C_s = 1 for the functional coupling to the all the other configuration items in the system hierarchy:

Cost impact: 9(R100k) + 3(R500k) + R(1000k) = **ZAR3400k**
Schedule impact: 9(6) + 3(12) + 18 = **108 weeks**

5. CONCLUSIONS

The hypothetical case-study examples above clearly demonstrate the escalating cost and schedule impact of a design change on a concurrent systems engineering development project. This impact is a function of the number CIs and functional couplings in system hierarchy. Design changes in a concurrent engineering development project have the following consequences:

- Design changes in coupled designs are generally not feasible due to the detrimental project cost and schedule impact. Design changes, discussed above are invariably Class I changes and result in a ripple effect due to the functional couplings throughout the system hierarchy.

- Limited design changes for uncoupled and decoupled designs may be possible for simpler systems.

- The project impact in terms of cost and schedule is generally too high to implement any design changes of a CI for complex multi-level systems without impacting on the project cost and schedule.

- ***"effect-to-cause"*** design iterations essential in the Integrated Product Support (IPS) development model places a severe system optimisation constraint on the system design.

- Design changes are major development project constraints in a concurrent engineering environment. Very often a band aid fix is the only practical non project intrusive way of solving the problem, which can lead to a non optimal design. This will be discussed further below.

- Research into techniques and design processes should be performed to reduce design iterations for optimal system design.

- The impact for a design change in this case-study example was a cost increase of 213% and a schedule penalty of 300%.

The case-study examples are hypothetical for illustrative purposes only. In reality, real systems are much more intricate with multiple level system hierarchies and numerous different discipline CIs. Computer simulation is required to analyze these systems. Such a model would provide quantified CI design change impact information to enable design review boards to make informed decisions. Computer modeling development falls outside the scope of this research.

From the analysis it can be concluded that a design change of a CI in a complex multi-component, multi-hierarchical system during system integration in a concurrent engineering environment invariably has a detrimental effect on project cost and schedule. In practice, design modifications/changes during system integration of a complex multi-hierarchical system are virtually unavoidable. The impact of forced changes can only be improved by optimizing the system architecture to keep the system data content or functional couplings to a minimum.

Apendix D Revised Problems experienced using AD

1. Management related program problems

Problem Area	Problem Statement	Cause	Impact	Revised impact
PCMB meeting	Project meetings particularly at subsystem level not always regularly held.	Time constraints and major role players' availability	5	1
DSRB meetings	Facilities meetings not structured to cater for the specific needs of individual aspects of the projects	Facilities in a matrix organisation generally provide resources for a number of projects, making it impractical to cater for the individual aspects of a specific project.	4	1
FRB meetings	Development personnel in general not familiar with the lower level client requirements and the user environment.	No initial guidance meetings were held.	5	1
Lack of knowledge of the user operational doctrines and support environment	Personnel involved with the development of the logistic products did not always understand how the user operates during a typical deployment exercise.	Initial programme focus was technical requirements and no study of the operational and support environment was done.	4	3
Lack of knowledge of the user training doctrines and training environment	Personnel involved with the development of training do not always understood the details on how the user training programme functions.	Initial programme focus was technical requirements and no study of the operational and support environment was done.	4	3
Scope creep	The requirements baseline at the lower system hierarchies was not fixed.	Particularly the lower system hierarchy did not always receive the full PM and SE attention and sometimes leading to unplanned baseline shifts.	4	3
Resources	The extent and scope of the logistic product development was under estimated due to a wrong assumption that the existing pre-upgrade ZT3 products could be adapted.	The project team did not pickup that the client changed their documentation and training material standards since the original ZT3 system, resulting in the logistic product development team being under resourced and over worked, contributing to expensive mistakes and quality problems. There was no spare capacity or contingency planning	4	3

Increase level of effort (LOE) for source info	Wrong information was sourced due to sub-contractor inputs.	Unfamiliarity with the new client technical manuals and training standards and wrong selection of expert consultant	3	2
Client Guidance	Availability of expert client personnel	The client has changed its standards since the original ZT3 system	2	1
Clarity of logistic requirements	The internal technical authors were not familiar with the new client documentation and training material standards.	The client has changed its standards since the original ZT3 system	3	2
Outsourcing technical manuals and training definition phase	Wrong information was sourced due to sub-contractor inputs.	The scope of work for the consultant tasked with the development of the technical manuals and training definition work was inadequate resulting in the wrong sub-contractor being selected	2	1
Client management	Sound client management was sometimes lacking leading to unplanned baseline shifts.	The impact of ostensibly small change requests was sometimes under estimated at PM level and allowed without detailed impact investigation.	3	2
Tasking (Task Structuring)	Not enough detail task structuring for logistic product development because task managers themselves did not always fully understood the task. Also no single clearly demarcated responsibility areas for logistic product development team members leading to inefficiency and quality problems.	Unfamiliarity with the new client technical manuals and training standards.	2	1
Logistics engineering availability	Logistics Engineering was not contracted to perform assessments of the technical integrity of the logistic products development team outputs.	PM did not make provision for engineering assistance during technical manuals and training material development.	2	1
System Engineering and subsystem expertise availability	The development team availability particularly during the final system qualification phase was very limited.	PM wrongly assumed that the technical information in the product data pack would be adequate for the technical authors.	3	2
System Engineering and subsystem expertise availability	System Engineering not sensitive regarding their responsibility towards the Logistic Development Process.	PM wrongly assumed that the technical information in the product data pack would be adequate for the technical authors.	2	1
PRACA form completion not enforced	Problems were not made visible to enable effective management and co-ordination resulting in persistence of a problem.	PM due to pressure of work did not always schedule regular FRB meetings to timeously address and resolve problems.	3	2
Contracting / Programmes	Inadequate facility contracting	Contracting and in particular outputs was sometimes vague.	3	2

CFE Data integrity	Source information in particular Customer Furnished Items (CFE) data was not verified up front leading to expensive time consuming re-work later in the project.	PM wrongly assumed that the CFE data supplied by the client was adequate and complete.	3	3
Planning	Detail planning lacking resulting in inability to measure progress and timeous identified problem areas.	PM due to pressure of work did not always integrate the detail task planning into their programme plan.	3	2
Planning	Task managers not always involved.	PM time constraints.	3	2
Cost Management	General task overspending.	Task overspending was not always immediately followed-up by PM.	2	1
Cost Management	Full scope of CFE impact not always visible.	The full scope of CFE generally only became visible late in the program	2	1
Risk Management	Risks were not identified and managed early enough in the program.	Risk management started too late.	2	1
Procurement delays	The company rule required that all procurement including engineering proto type components be handled by the company procurement section	Procurement priorities did not make special provision for small quantity highly specialised development contract procurement items.	3	2
Technical manuals and training development	Technical authors were not always au fait with the prescribed templates and standards.	Lack of editorial assistance and training	3	2
Process	No clear technical manual and training material development process that all team members could follow.	The new client manual and training standard aggravated the problem.	3	3
Change Control	ECPs were generally not reviewed and counter signed by all the stake holders	Stakeholder availability	4	1
Discrepancies between hardware and data	Redlining of drawings sometimes lagged behind	Tight schedules and pressure for hardware availability.	5	1
Fragmented Quality Assurance function	Each facility provided its own QA function.	Company rules did allow indirect personnel on direct programs.	3	2
Fragmented configuration function	Each facility provided its own configuration management function.	Company rules did allow indirect personnel on direct programs. Programme QA manager primarily focussed on top level system QA.	3	2
Fragmented procurement function	Procurement was a centralised corporate function geared for production procurement	Company rules did allow indirect personnel on direct programs.	3	2
Incidents	32	**Sub Total**	**100**	**57**

2. Systems Engineering related programme problems

Problem Area	Problem Statement	Cause	Impact	Revised impact
Data availability	The system development data pack only addresses the "**WHAT**" the system must do, not the "**HOW**" the system works. .	The systems engineering process is requirements driven. No provision is made for documenting **HOW** the designs work and interact within the rest of the system..	5	5
System Engineering and subsystem expertise availability	The development team availability particularly during the final system qualification phase is very limited	Expert resources a scarce resource and once a task is complete, the resource is allocated to another programme under the matrix organisational structure.	4	5
System Functional Block Diagram (FBS) not inline with product Breakdown Structure (PBS)	There was virtually no link between the system FBS and final PBS. The PBS also kept changing during the program. These uncoordinated changes resulted in much fruitless work and reworks of documentation	Design iterations resulted in changes being accepted and implemented but retrospective data pack updates were not always done due to pressure of work..	4	4
Maturity of System	Log Products were required before the acceptance and freezing of the system production baseline	Design iterations (ECPs) resulted in subsystem baseline changes that were not incorporated into the log products development schedules	4	2
Hardware availability	Tight schedules and technical development program slips resulted in lack of verification of log products by the log product development team.	Primarily due to cost and time constraints only one demonstration model of each hardware item was built. This resulted in a conflict between qualification testing and log product verification.	4	2
Hardware availability	The hardware was not available for the documentation development team	Hardware was continuously modified during qualification testing and not released for other use until late in the programme. Also the documentation redlining process often lagged behind the hardware status.	4	2

Terminology List updates	A large number of log product changes late in the programme were as a result of terminology changes. The impact of these on cost and schedules were generally under estimated by all parties involved.	The ripple effect of ostensibly minor terminology changes was under estimated.	1	1
Logistics engineering availability	Logistic engineering expert availability particularly during the final system qualification phase is very limited	Expert resources a scarce resource and once a task is complete, the resource is allocated to another programme under the matrix organisational structure.	2	2
PRACA form completion not enforced	Problems were not made visible to enable effective management and co-ordination resulting in persistence of a problem.	SE due to pressure of work did not always schedule regular FRB meetings to timeously address and resolve problems.	1	3
Change Control	ECPs were generally not reviewed and counter signed by all the stake holders	Stakeholder availability, time constraints	3	3
CFE Data integrity	Source information in particular Customer Furnished Items (CFE) data was not verified up front leading to expensive time consuming re-work later in the project.	The CFE data information was accepted on face value without verification.	2	2
Electronic data control:	Drawing office sometimes placed data on the server that was not always verified and approved.	Schedule pressure and fragmented configuration management.	1	1
Incidents	12	**Sub Total**	**35**	**32**

5. Quality Assurance related programme problems

Problem Area	**Problem Statement**	**Cause**	**Impact**	**Revised impact**
Client documentation and training standards.	QA not familiar with logistic product requirements and the relevant RSA-MIL-STDs.	Fragmented Quality Assurance function as a result of the matrix organisation structure and company rule of no indirect personnel on direct programs.	3	3
Technical manuals template problems	QA not familiar with logistic product requirements and the relevant RSA-MIL-STDs.	This is a specialist field outside the scope of general QA	2	2
Change Control	ECPs were generally not reviewed and counter signed by all the stake holders	Stakeholder availability and fragmented QA function	1	1
Continuous Evaluation	QA not contracted for continuous evaluation of the logistic products throughout the development phase	Facility workloads prevented continuous evaluation of one programme output.	2	2

Continuous Evaluation	QA was generally passive and only involved with the programme hardware instead of all deliverables (including the log deliverables)	Facility workload	2	2
Continuous Evaluation	QA has not been contracted for the evaluation of logistic products	QA was considered an indirect function as part of a facility.	2	2
Continuous Evaluation	QA is not competent to evaluate the technical integrity of the Technical Manuals and Training Material	Limited specialist availability as part of the matrix organisation structure.	3	3
Incidents	7	**Sub Total**	**15**	**15**

4. Configuration Management related programme problems

Problem Area	**Problem Statement**	**Cause**	**Impact**	**Revised impact**
Baseline/ Moving	The baseline kept changing due to enforced ECPs creating difficulty with the log development. (Moving target).	Problems at integration levels very often led to urgent ECPs to resolve the problem. This had a ripple effect throughout the system hierarchy.	4	4
Data availability	The system development data pack only addresses the "**WHAT**" the system must do, not the "**HOW**" the system works. .	The systems engineering process is requirements driven. No provision is made for documenting **HOW** the designs work and interact within the rest of the system..	2	2
System Block Diagram not inline with PBS	There was virtually no link between the system FBS and final PBS. The PBS also kept changing during the program. These uncoordinated changes resulted in much fruitless work and reworks of documentation	Design iterations resulted in changes being accepted and implemented but retrospective data pack updates were not always done due to pressure of work..	3	3
Source info	**T**he technical authors had difficulty in getting data on how the system/subsystem worked.	The systems engineering process is requirements driven. No provision is made for documenting **HOW** the designs work and interact within the rest of the system..	2	2
Source info	**T**he technical authors had difficulty to get access to the design experts.	After the CDR, facilities immediately reallocate expert resources to other programs	2	2

Fragmented CM	Each facility had its own CM. Only at program level was the program CM who could not go into detail of all the subsystem configuration aspects.	Documentation configuration control was part of the company management information system and available on the intranet. Programme CM was then reduced to a coordination and audit function	3	3
Templates	Technical manual templates were not user-friendly leading to time wasting mistakes by technical authors	Unfamiliarity with the new client technical manual led to outsourcing of templates.	2	2
Templates	Templates were not approved prior to population of data	Templates were accepted at face value and not put through an approval process with the client in order to save time.	3	3
Templates	Tolerances of the templates must be realistic and within the capabilities of the printers and PCs used in the facility	The page units of measure in the client technical documentation standards were metric rather than point units for printer drivers. Client QA personnel were rejecting pages due to printer tolerances. Despite the templates being correct.	3	3
Change Control	ECPs were generally not reviewed and counter signed by all the stake holders	Stakeholder availability, time constraints	2	2
Incidents	**10**	**Sub Total**	**26**	**26**
Total	**61**	**Total**	**176**	**130**

Summary:

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