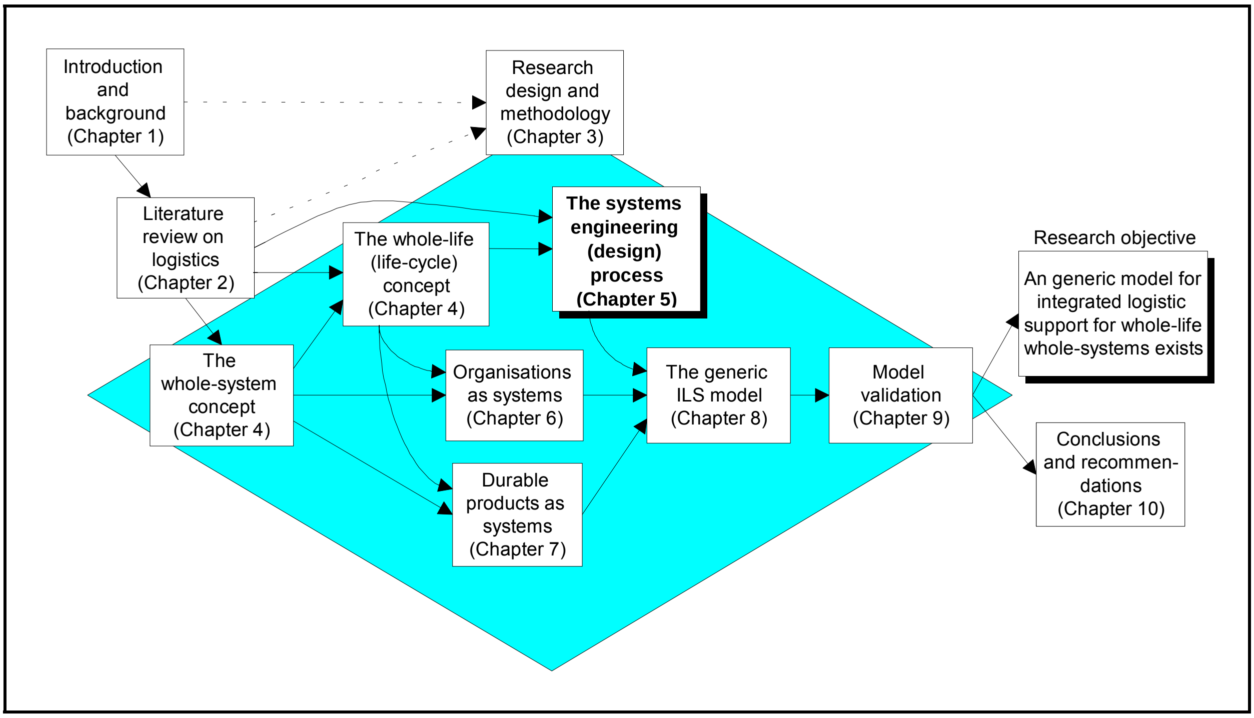
Chapter 5

Bringing systems into being

"Solving problems always use the simplest solution, but no simpler."

Albert Einstein



5.1 Purpose and outline of the chapter

The purpose of this chapter is to provide an outline of the process employed to bring systems into being. This process of bringing systems into being is generally referred to as the systems engineering process. In the previous chapter, an outline of the nature of systems was introduced, providing an insight into the ability, availability and affordability measurements of system success and these measurements' relation to the whole-life whole-system concept. This chapter serves as necessary background for the development of a generic approach to integrated logistic support for whole-life whole-systems.

The outline of the chapter is as follows:

- A brief motivation will be provided for the continuous need for new systems.
- The need for a formal process to bring systems into being is explained.
- Before systems engineering as a formal process can be introduced, the multiple dimensions of such a formal process have to be defined and explained.
- Systems engineering is presented as a formal process that will allow the realisation system to bring the system of interest into being.
- After a formal definition for systems engineering is given, an organisational perspective of systems engineering is provided to allow for deeper understanding of systems engineering.
- An extensive discussion on the role and impact that systems engineering has on ability, availability and affordability follows. Aspects covered are cost-effectiveness, life-cycle cost and design influence considerations.
- The chapter is concluded with the basic requirements for the formal process to bring systems into being.

5.2 The quest for new systems

Man’s involvement in and fascination with systems go back far in history. The first recognition man has had of systems must surely have been the awareness of him being part of the natural systems of the earth and the universe. Irrespective of man’s belief how the universe came into being, the underlying commonality of all religions is that some realisation system in some form or another must have existed (or still exists) that was responsible for creation. As this research is concerned with man-made systems, no attempt will be made to explain creation’s realisation system. It is best left to the cosmologists, theologians, philosophers or the faith one has.

The human-made world is made up of everything that was brought into being by man for use, consumption or appreciation by man. No doubt, many man-made creations are to the detriment of man and his natural environment. The ecological impact of the Aswan Dam on the Nile river is a good example of the negative effects of not following a systems approach [Blanchard and Fabrycky, 1998:5]. Still, there is a continuous pursuit to improve the standard of living of the human race which drives the never-ending quest for new and better products and systems. This never-ending quest requires from realisation systems (those systems that bring other systems into being) to also improve the way they go about determining needs, defining concepts, do preliminary and detailed design, do physical creation (production and/or construction), insertion into the host system, operations and support and finally phasing out.

5.3 The need for a formal process

Already in the early civilisations, the human race has demonstrated the capability to undertake the creation of complex man-made systems. Good examples of this ingenuity are the Egyptian Pyramids and the Great Wall of China. Bringing about creations of this magnitude and complexity, must unquestionably have had a formal process controlling the life-cycle of those systems. In more recent times, during the 1950's and 1960's, the Americans undertook major developments with their space and military programs. These developments required the integration of many distinct disciplines and technologies into an overall complicated but optimised result. This is regarded as the birth of systems engineering in its modern form and has since found application outside of the military and space programs in a wide variety of industries. [Shenhar in Sage and Rouse, 1999:113].

The need for a formal process to bring systems into being is stated unequivocally by Blanchard and Fabrycky [1998:xiii]: *"Experience in recent decades indicates that properly coordinated and functioning human-made systems, with a minimum of undesirable side effects, require the application of an integrated, life-cycle oriented 'systems' approach. The consequences of not applying systems engineering in the design and development and/or reengineering of systems have been disruptive and costly."*

5.4 The multiple dimensions of systems engineering

Before a definition of systems engineering can be provided, it is necessary to clarify the multiple dimensions of systems engineering. Having gone through the agricultural and industrial revolutions, and being within the information age based on the intricacies of electronics as well as the complex interfaces between technologies and humans, it is obvious that the world is becoming more and more complex. New and emerging technologies reveals unexpected opportunities for bringing into being new and improved systems and products [Blanchard and Fabrycky, 1998:14, 17]. In a world where needs can be satisfied in various ways - which means that there can be no single solution to complex problems, or a single way to take a system through its life-cycle - it must be obvious that no single person can be responsible for the creation of a complex system. Complex systems can only be created through the combined efforts of many people and teamwork [Blanchard and Fabrycky: 1998,14-15] . It can be stated further that systems engineering is not only about solving problems ***for*** customers, but ***with*** customers [Shenhar in Sage and Rouse, 1999:114-115 and M'Pherson, 1980:549]. Another implication of the existence of multiple solutions to a problem is that the problem solving will be of an iterative nature to arrive at the 'best' solution.

It can be concluded that there are two distinct dimensions to systems engineering, i.e.:

- The technical component, the dimension that is concerned with the technical requirements and interdisciplinary integration of technologies and humans within the system.
- The managerial component, the dimension that is concerned with the planning, organising, directing, controlling and provisioning of resources to take the system through its life-cycle from need identification to phasing out.

In view of these assertions and using the simple model provided by Shenhar in Sage and Rouse [1999:115], a revised model is proposed in Figure 5.1. *"In its simplest terms, systems engineering is both a technical process and a management process. To successfully complete the development of a system, both aspects must be applied throughout the system's life- cycle."* [Defence Systems Management College, 1983]. To many it may be a contentious point whether systems engineering really covers the total life-cycle of the system of interest. Even though this contention will not be discussed further at this point, it assumed that both components of systems engineering are required throughout the total life-cycle of the system of interest. The premise for this is that systems

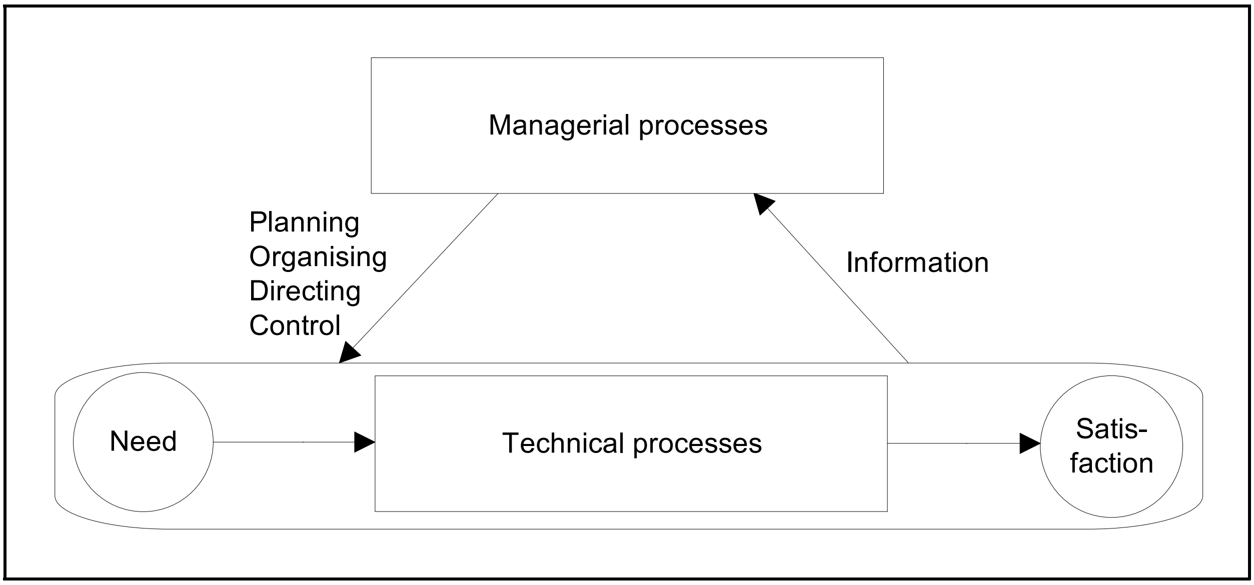


Figure 5.1: The multiple dimensions of systems engineering
[Adapted from Shenhar in Sage and Rouse:1999,115]

engineering is concerned with whole-life whole-systems which means the management component of the system of interest does not cease once the system goes into operation, but takes on a different meaning. This different meaning will be put into perspective later in the study .

5.5 Systems engineering defined

Shenhar in Sage and Rouse [1999:114-115] provides an extensive discussion of the different definitions for systems engineering. As this study is not primarily focussed on systems engineering, a simple definition for systems engineering will be stated and supported by a more comprehensive definition that is found in recent literature.

> **In its most basic form, systems engineering is the formal iterative process of management and technical activities required by the realisation system to bring the system of interest into being and to aid in ensuring a useful life as part of the host system until retirement.**

A more formal definition is provided by Shenhar in Sage and Rouse [1999:115]. This definition is revised to demonstrate the managerial component of systems engineering more specifically, which is lacking in Shenhar's original definition. The more formal definition is as follows [Adapted from Shenhar in Sage and Rouse, 1999:115]:

Systems engineering is the application of scientific, engineering and managerial efforts to:

- **Identify an operational need (in a commercial or military area, in the public or private sector), together with a marketing and technological opportunity that leads to the creation of a system that will address this need.**
- **Transform an operational need into a description of system performance parameters and a preferred system configuration through the iterative use of analysis, definition, synthesis, design, test and evaluation.**
- **Ensure integration of all related (physical and functional) parameters and interfaces within the system of interest, with the host system as well as with the wider system in a way that best meets the operational need.**
- **Ensure the management of system ability, availability and affordability throughout the life-cycle of the system of interest, through evaluation and improvement of both technical and managerial components of the system of interest.**

The formal definition for systems engineering implies that:

- Systems engineering covers the entire life-cycle, from need identification up to the point where the system of interest does not exist anymore.
- Systems engineering requires a whole-system design approach (design of all technical ***and*** managerial components of the system of interest) without which successful integration on the different system levels will not be possible.
- Systems engineering seeks to find the best combination of technologies, processes, interfaces and relationships within a system and its environment to satisfy the need, implying that no single 'optimal' solution for a problem exists.

5.6 M'Pherson's perspective on systems engineering

M'Pherson [1980:553-557] provides three different perspectives on systems engineering. The foundation for his perspectives is the assertion that one has to build a framework based on the combination of the whole-life and the whole-system concepts. This framework is then used to *"tackle the problems of planning, designing and managing the whole-system realisation problem"* [M'Pherson, 1980:553]. This statement again supports the view that there are two distinct components within systems engineering, namely the technical activities and the management activities.

M'Pherson's first perspective (the organisational perspective) deals with the realisation processes within the realisation system along with a time line (the whole-life concept), leading to insertion of the system of interest into the host system. Using this perspective, systems engineering is a *"dynamic multivariable, multiloop system which will have an interesting dynamic behaviour that is likely to require careful tuning, co-ordinating and control if it is not to display signs of instability"* [M'Pherson, 1980:554]. The organisational perspective is discussed later in this chapter using Figure 5.2.

M'Pherson's second perspective is what he calls the system design perspective, which deals with creating the right system. M'Pherson [1980:554] argues that *"no matter how well the project is planned and organised, the effort will be largely wasted if the operational system that results is the wrong system, even though it may be excellently designed in itself".* Systems can be wrong by not meeting the customer's requirement (ability, availability, affordability and safety), not being matched to the actual operational environment, being badly timed, or not being matched to the social or natural environment. Systems engineering strives to guard against such errors by integrating as many considerations and criteria as necessary into the design process resulting in whole-system design [M'Pherson, 1980,554-555].

The third perspective provided by M'Pherson [1980:556] is called the system planning perspective. This perspective shows the system of interest as the end result of an integrated design process consisting of six dimensions that all have to exist in an integrated way. These six dimensions are:

- The operational structure of the system of interest.
- The associated equipment of the system of interest.
- The logistics system to provide support to the operations and maintenance of system of interest.
- The maintenance system to restore availability of the system of interest after deterioration of ability has taken place.
- The manpower system to provide management, operational and support skills to the system of interest.
- The management system to take operational, maintenance and financial responsibility for the system of interest.

No single dimension can be designed in isolation nor can it exist on its own. This perspective of M'Pherson places the emphasis on the management of the systems engineering process throughout the realisation phase, and it also emphasises the realisation of a management system for the system of interest.

M'Pherson's organisational perspective [1980:554] is combined with his system planning perspective [1980:556] in Figure 5.2 to emphasise important characteristics of systems engineering. Figure 5.2 highlights the management of the systems engineering process and the insertion of the management system of the system of interest into the management system of the host system.

This combined perspective clearly shows the management implications for systems engineering are much more profound than commonly believed. The whole-life whole-system concept requires that the whole-system is to be designed, followed by an insertion of the operational, support and monitoring systems. It has been demonstrated earlier in this chapter that systems engineering includes a management component, thus it follows naturally that in order to follow the whole-life whole-system concept, the management system of the system of interest should be realised and acquired in an integrated way with the rest of the system. As acquisition includes insertion, it follows that, not only should the physical system be inserted into the host system, but the management system of the system of interest should also be inserted into the management system of the host system. This adaptation to M'Pherson's organisational perspective of systems engineering, along with the continuous monitoring and evaluation feedback loop, implies a management

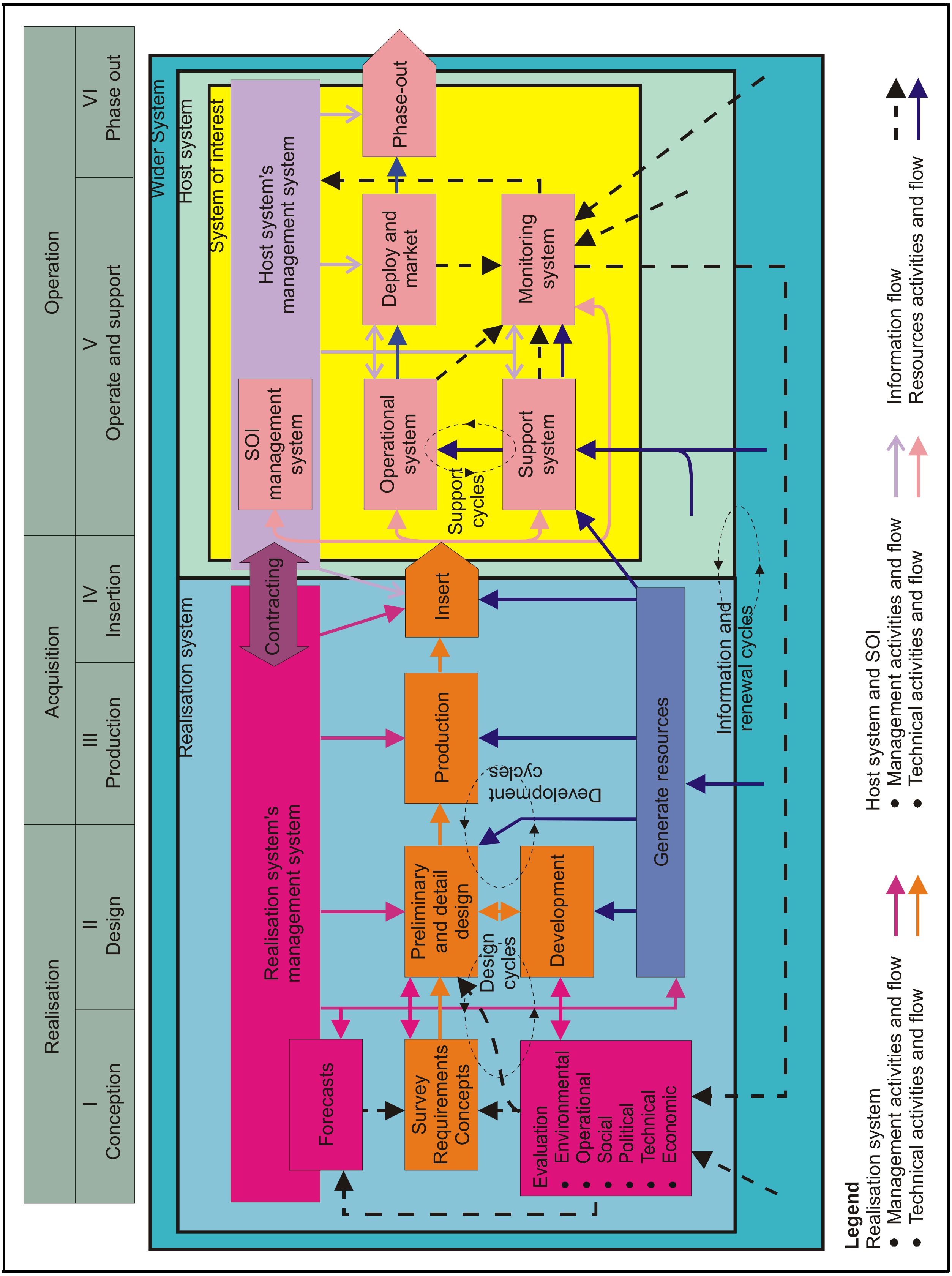


Figure 5.2: An integrated perspective of systems engineering
[Adapted from M'Pherson, 1980: 554, 556]

system within the realisation system to continue to exist throughout the life-cycle of the system of interest, as well as continuously interfacing with the management system of the host system, normally through contracts in some form, placed by the host system on the realisation system.

The continued existence of the management system of the realisation system throughout the life of the system of interest, is required in order to:

- Provide the interface between the host system and the realisation system for continued support of the system of interest.
- Gather data to trigger an improvement program for the system of interest. This improvement program requires the systems engineering process to be followed once more, as the systems engineering process is applicable to reengineering/ improvement programs as well [Blanchard and Fabrycky, 1998:xiii].
- Gather data and information to better satisfy the requirements of the customer when a new system of interest is to be realised within the particular environment.

5.7 The role and impact of systems engineering

Systems engineering is about effectiveness. Effectiveness is doing the right things. The emphasis on doing the right things is required primarily during the acquisition phase in order to make sure that when it comes to the utilisation phase, the system will continuously meet the ability, availability and affordability requirements. The utilisation phase is primarily concerned with efficiency, or doing things right, with the minimum amount of waste. This is based on the premise that the inherent design characteristics of a system is determined very early in its life-cycle, and that the effectiveness of a system is thus determined very early in the systems realisation phase.

Blanchard and Fabrycky [1998:xiii] state that *"the consequences of not applying systems engineering in the design and development and/or reengineering of systems have been disruptive and costly"*. This view is supported by other literature [Defence Management College, 1983:1-i]: *"These large, highly interactive systems that are on the forward edge of technology, have a natural process of evolution, or life-cycle, in which actions taken (or* *not taken) in the very early stages can mean the difference between success and failure. When the outcome is certain only after the expenditure of hundreds of millions of dollars, even wealthy nations cannot afford many failures. The purpose of systems engineering is to prevent these failures through a unified approach that completely defines all requirements on the system and establishes a system configuration which is proven early-on to be capable of meeting those requirements. Systems engineering is often referred to as a 'front-end' process. That is, the majority of systems engineering tasks are completed in the initial phase of the program, when about 5% of the program's funding is expended. This initial effort results in defining the configuration and size of the system and its logistical support. The resulting program commitment of funds typically represents 90% of program life-cycle costs. Accuracy and completeness of the early systems engineering effort is therefore essential in maintaining the program within budget constraints"*. What is not emphasised enough, however, by this view, is that the systems engineering effort does not only have a major impact on affordability (life-cycle cost), but also ability and availability.

Many authors [Blanchard and Fabrycky; 1998:573; Defense Management College, 1983:16-7; M'Pherson, 1981:76] provide frameworks for cost-effectiveness of a system. For the purpose of this chapter, it would suffice to use a simple framework derived form the arguments presented in § 4.8 that emphasises benefit-effectiveness rather than cost-effectiveness. This framework is depicted in Figure 5.3.

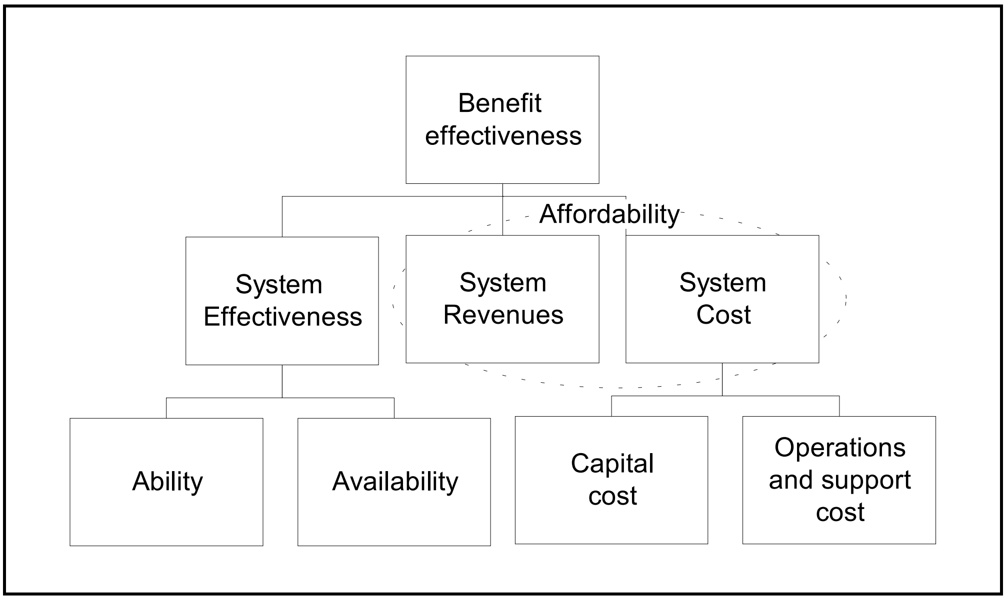


Figure 5.3: Benefit-effectiveness breakdown

It can thus be concluded that the systems engineering process has to emphasise the ability, availability and affordability of the system while it is still under design in order for the system to meet requirements during the operational/utilisation phase.

5.7.1 The ability perspective of systems engineering application

A system's ability is the reason for the existence of the system in the first place. Using the problem or need statement of the customer or market into consideration when designing the system's ability is therefore of the utmost importance. As more and more over-capacity tend to emerge in the USA, similar to what already exist in Europe and the East, there will be increasingly a customer-driven market rather than a market dictated to by the realisation systems. Thus only products (including services) meeting customer requirements will be sold [Wang, Han, Spoerre and Zhang in Wang, 1997:2]. Also, customers have become more sophisticated, knowledgeable and insistent on high quality products and services regardless of price [Watkins in Wang, 1997:21].

In the past, a sequential design process was used primarily because realisation systems were organised along functional lines. This resulted in partial designs being 'thrown over the wall' from the one functional group to the next. The end effects of this approach were long development times, poor integration, severe production problems, outputs not meeting market requirements and excessive change costs.

From the ability perspective, systems engineering has to place the engineering of systems ahead of concern for components of the system and specifically emphasise [Blanchard and Fabrycky, 1998:18]:

- Methods to improve definition of requirements and how they relate to true customer needs. Methods such as quality function deployment (QFD), value engineering and functional analysis can be used.
- Addressing the total system from a whole-life perspective to ensure that ability can be maintained throughout the total life-cycle. This addresses both the production and support functions without which ability is not possible.

- Considering the interfaces of the system of interest with its host system to the extent they are compatible and dependant to allow full system ability.
- Organising the system design and all other related design disciplines into one concurrent mainstream design effort.
- Establishing a disciplined review and evaluation process for appropriate and timely feedback to ensure requirements consistency throughout the life-cycle.

System requirements are to be related to specific design criteria and all design decisions should be traceable to original requirements. This is especially important to ensure that system designs do not include unnecessary parts which are not needed, making for a less reliable and more costly design. Systems designs having unnecessary parts or functions are becoming more common primarily because of the ease with which design features can be added through the use of electronics and software. On the other hand care must be taken that functions are not omitted, something that frequently happens with order qualifiers (see § 6.5 for a definition of order qualifiers) as these are many times not explicitly expressed as a requirement; the customer assumes that the realisation system knows the minimum requirements.

Ability of the system of interest nor only depends on how well it was designed, but also on how well it is produced. A well designed system but produced poorly will not achieve the ability levels expected of it. Design for produceability is therefore also important to consider during the design. Design guidelines for produceability have also been found to generally apply to operations (including support), thus an additional benefit is gained in the sense that it does design for producability not only improves ability, but also availability. Naturally improved availability will also lead to improved affordability. Schonberger an Knod [1997:101] provide a general list of guidelines for design for operations (including production):

- Design to target markets and target costs.
- Minimise the number of parts and number of operations (viz. production, operations and maintenance activities)
- Design to known, complete customer requirements.
- Design to process capabilities of own process and those of suppliers.
- Use standard materials, processes and procedures with known, proven quality.

- Use modular/reusable designs. This includes standardisation and rationalisation.
- Design for ease of production, operations and maintenance.
- Design for foolproof production, operation and maintenance ('*poka-yoke*' is the Japanese word for avoid mistakes).
- Design for robustness.

Design for ability is very much dependent on the nature of the system and the technology employed for its design. Further detail on the overall approach to ability design can be obtained from Blanchard and Fabrycky [1998], Wheelwright and Clark [1992] and Wang [1997].

5.7.2 The availability perspective of systems engineering application

Availability of the system of interest is dependent on two variables, namely reliability and maintainability, discussed in § 4.8.2. These two variables are inherent design characteristics, which means that the values these variables will take on during the system of interest's life, will depend primarily on the design of the system of interest itself. When the interrelationships between design and these inherent characteristics are analysed, understanding can be gained how the application of the system engineering process can influence the future availability of the system of interest.

As has been indicated in § 5.7.1, the first perspective of system engineering application is to specify and create those characteristics of the design that will render the functional ability to meet the requirements stated for the system of interest. In order to carry out design, individual technical design disciplines are required, for example mechanical design, electrical design, electronic design, structural design and information design. By virtue of the fact that design decisions are made, for example size, weight, material composition, tensile strength and interfaces between components, the reliability and maintainability characteristics are established. It is thus important for the functional designer to be aware of the implications of the design decisions on reliability and maintainability, as these characteristics are inherently set by the design itself.

The detail or subsystem designer is not the only one responsible for the inherent characteristics of the system. On the system level a major role is to be played by the systems engineer with regards to the inherent characteristics of the whole-system. On the system level the system engineer is responsible to oversee the operational support system and the maintenance system design. The scope, magnitude and success of these systems largely depend on integrated design on system level and the inherent design characteristics of the functional and physical design itself. As indicated in the previous chapter, availability can be expressed as [Blanchard, 1998:90]:

$$Availability = \frac{MTBF}{MTBF + MTTR} \tag{5.1}$$

where MTBF is the mean time between failure, an indication of reliability, and MTTR is the mean time to repair, an indication of maintainability.

This equation can be analysed further in terms of the MTTR figure. Firstly, the inherent availability of the system can be expressed as [Adapted from Blanchard, 1998:80]:

$$Availability_{Inherent} = \frac{MTBF}{MTBF + MTTR_{CM}} \tag{5.2}$$

where $MTTR_{CM}$ is the mean time to repair, including corrective maintenance time, but excluding preventive maintenance time and maintenance lead time.

Inherent availability is accomplished within an ideal support environment (i.e. all support resources are available when required) [Blanchard, 1998:80-81].

Secondly, availability can be viewed as achieved availability, in a similar fashion to inherent availability, but with preventive maintenance time included and can be expressed as [Adapted from Blanchard, 1998:81]:

$$Availability_{Achieved} = \frac{MTBF}{MTBF + MTTR_{CM+PM}} \tag{5.3}$$

where $MTTR_{CM+PM}$ is the mean time to repair, including corrective maintenance time and preventive maintenance time, but excluding maintenance lead time.

Achieved availability is also accomplished within an ideal support environment (i.e. all support resources are available when required) [Blanchard, 1998:80-81]. An argument can be raised which of the inherent or achieved availability should be the higher of the two. The first impression is that inherent availability should be larger than achieved availability, as the MTTR component consists only of corrective maintenance as opposed to achieved availability, where the MTTR consists of both corrective and preventive maintenance. The purpose of preventive maintenance is to maintain the system in an operable condition, i.e. to perform maintenance activities that prevent failures from occurring, it is thus clear that an increase in preventive maintenance time should lead to a reduction in corrective maintenance time. Thus, if the decrease in corrective maintenance time is more than the increase in preventive maintenance time, then it can well be that the achieved availability be higher than the inherent availability. If the achieved availability is higher than the inherent availability, it is an indication of a well designed preventive maintenance program.

Thirdly, real life availability is expressed as operational availability [Blanchard, 1998:81], where the system is used in an actual operational environment and can be expressed as [Adapted from Blanchard, 1998:81]:

$$Availability_{Operational} = \frac{MTBF}{MTBF + MTTR_{CM + PM + MLT}} \tag{5.4}$$

where $MTTR_{CM + PM + MLT}$ is the mean time to repair, including corrective maintenance time, preventive maintenance time and maintenance lead time.

In the calculation of operational availability, the mean time to repair includes all the time required from the point in time a support requirement has been identified, until the point in time where the functional ability has been restored (corrective maintenance). It also includes the total time required to retain a system in its full operational state (preventive maintenance). Maintenance lead time (as used in Equation 5.4) includes all delays such as ordering, provisioning, transportation, acquiring the required personnel and other resources, administrative delays and others. This figure ($MTTR_{CM + PM + MLT)}$) is also referred to as the maintenance downtime (MDT) [Blanchard, 1998:58].

The application of the availability figure can vary widely, as can be seen with the different definitions. In the final instance, however, the most important availability figure is the operational availability. Operational availability is where reality determines the success of the system of interest. Blanchard [1998:81] makes out an interesting case when each of the availability figures should be used and applied. In the case where the realisation system has no control or influence over the host system it is proposed that inherent and achievable availability figures be used for evaluation of the system. This author, however, is of the opinion that the realisation system can influence the operational availability to a large extent even if the realisation system has no direct control after insertion of the system of interest. This influence can be achieved through proper design of the inherent characteristics (MTBF and $MTTR_{PM + CM}$) as well as good overall logistic support system design. If the system of interest does not perform to its ability when it is not operated according to the instructions and requirements set by the realisation system, the realisation system cannot be held responsible for non-performance. However, if the host system operates the system of interest properly according to the set operating instructions, it will be able to utilise the ability of the system of interest as intended. Likewise, if the system of interest has good inherent characteristics (MTBF and $MTTR_{PM + CM}$) and a proper integrated logistic support system has been designed with associated instructions, requirements and management procedures, ensuring a low logistic delay time ($MTTR_{LDT}$) figure, and the host system adheres to that, then the realisation system does have a major influence on and control over the operational and support arena of the system of interest. Understanding of the above is crucial in establishing a generic approach to integrated logistic support for whole-life whole-systems.

5.7.3 The affordability (life-cycle cost) perspective of systems engineering application

If the expenditure over the life-cycle of a system is analysed, a distinctive cost profile common to most systems exists. The cost profile of a system can be broken down into the cost of development, acquisition, operations and support and retirement/disposal. The distribution of these costs over time can be plotted to provide the system life-cycle cost [Adapted from Blanchard and Fabrycky, 1998:591] in Figure 5.4.

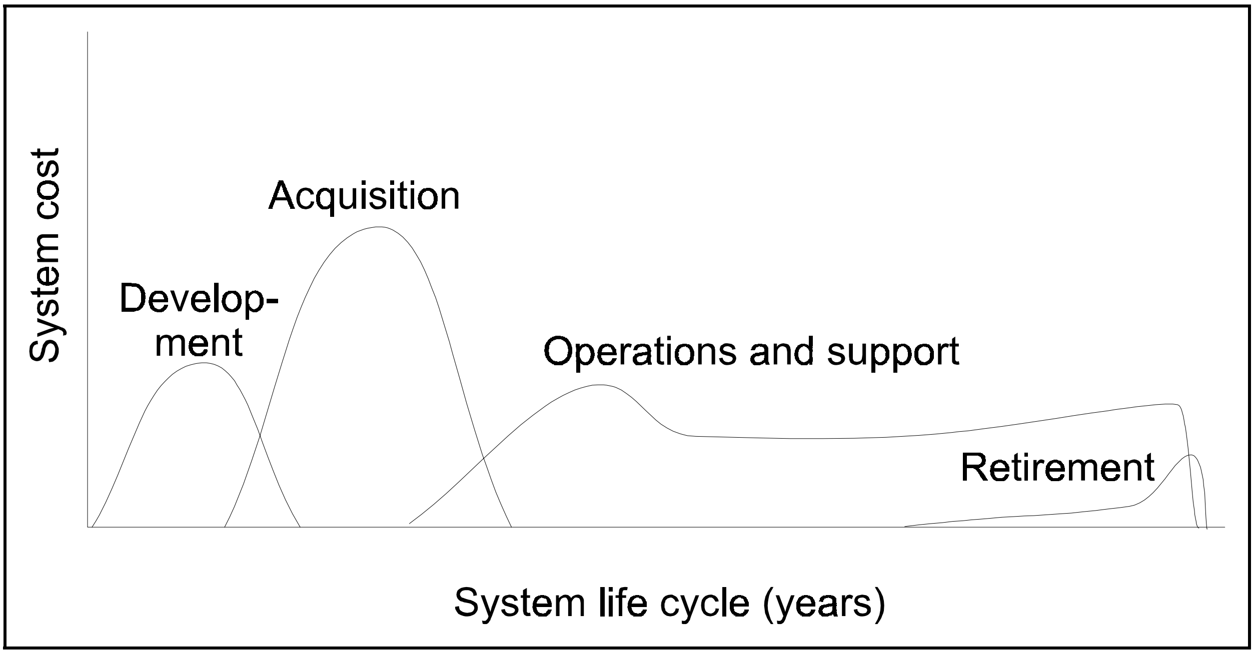


Figure 5.4: The life-cycle cost components
[Adapted from Blanchard and Fabrycky, 1998:591]

The money spent during the development phase is normally less than the money spent during the acquisition phase. The reason is that the physical creation (production or construction) is much more capital intensive than design. This normally does not hold true for software design. Another reason is that changes to a concept (paper or electronic format) during design is much less expensive than changes to something that already exists in its physical form, which is true once production/construction has started. The protrusion on the graph at the start of the operations and support phase (Figure 5.4) is associated with the insertion cost of the system of interest into the host system. Insertion cost is normally spent on establishing infrastructure, initial stocking, and the staffing and training of personnel associated with the new system. Insertion cost was chosen randomly to be shown as part of the operations and support cost. Choosing where to show the insertion cost is irrelevant because the graph only shows all associated costs with the life-cycle phases and does not take into account which system (realisation or host) bears the cost for each life-cycle phase. Where it is known which system is to bear the cost of a specific phase, different life-cycle cost profiles will be needed for the host and realisation systems. For the time being this distinction is not required. Thus, when the costs of all the different phases are added together the resultant life-cycle cost profile can be seen in Figure 5.5. [Blanchard and Fabrycky, 1998:591].

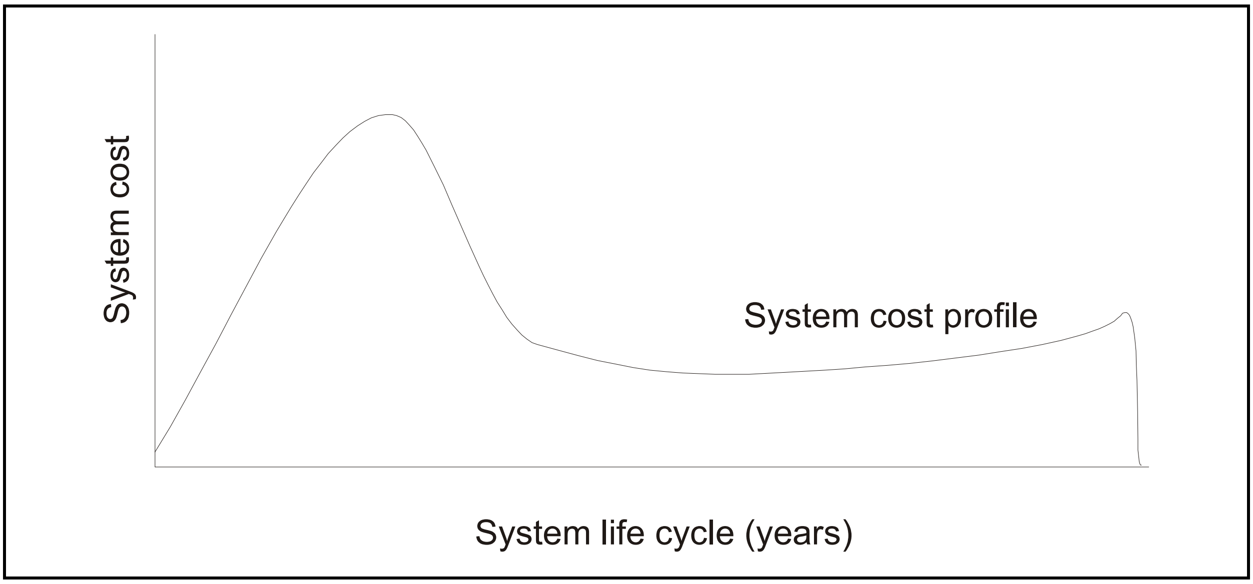


Figure 5.5: Total life-cycle cost
[Blanchard and Fabrycky, 1998:591]

Both the previous two graphs provide the cost per time unit, it therefore follows that the total life-cycle cost of the system is represented by the area under the curve. As systems engineering has as one of its aims to reduce total life-cycle cost - Blanchard and Fabrycky [1998:557] call it design for affordability - it is clear that decisions early on in the system design (and having a major impact on the life-cycle cost or affordability of the system), must be taken with utmost care. Conceptually the design for affordability is the conscious design effort to change the shape of the life-cycle cost profile in order to reduce the area under the curve shown in Figure 5.5, while at the same time meeting and/or exceeding the ability and availability requirements of the system. This extra design effort to reduce the total life-cycle cost is assumed by many to increase the development cost.

The question that might be raised is whether it really costs more to make an affordable design. That is, will the life-cycle cost of the more affordable system (Figure 5.6 (b)) be higher during the initial phases as opposed to the life-cycle cost of the less affordable system (Figure 5.6 (a)) during the initial phases? Without going into an extensive debate, this author is of the opinion that it should not necessarily be more expensive during the initial phases to design the more affordable system as opposed to the less affordable system. It can be motivated by the fact that within the design phase, many design changes cause higher cost. If good engineering is employed - designing well the first time round -

the many design changes will not be needed, reducing the overall design cost. Designing well the first time round is the same principle as the quality principle defined by Philip Crosby [Schonberger and Knod 1997:81], the principle that quality is to be ensured the first time. Schonberger and Knod explain Crosby's notion that all the things which prevent jobs from being done right the first time is costing money and thus constitute a waste, but to be able to get quality for free, one must work hard at it. Thus by working hard at employing good engineering, making new designs will cost less than doing it over and over until the customer is satisfied. It can thus well be that the life-cycle cost of a system, well designed the first time round, be lower for all time periods of the life-cycle, as denoted by line (c) in Figure 5.6.

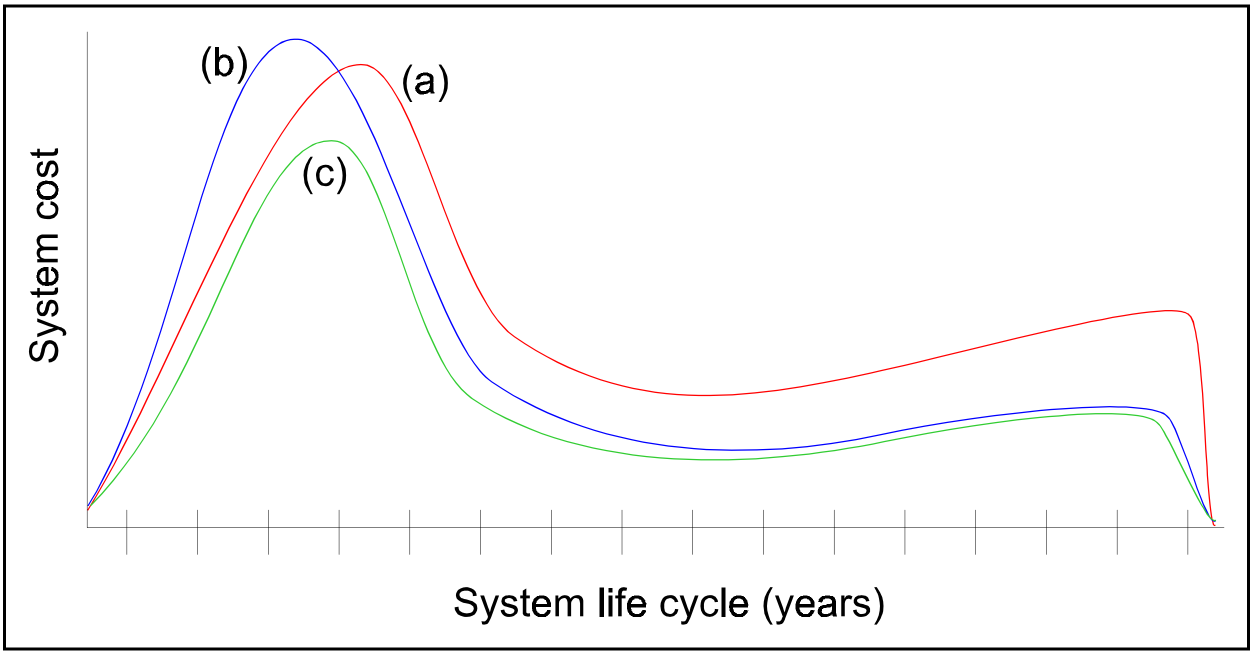


Figure 5.6: Designing for affordability - Shifting the life-cycle cost (LCC) profile

According to Blanchard and Fabrycky [1998:38] it is estimated that anything between 50% and 75% of the projected total life-cycle cost can be committed as a result of the decisions made during the conceptual and preliminary design phases. Another source [Defense Management College, 1983:17-1] estimates that up to 95% of the system life-cycle cost will be committed by the time the detail design has been completed, while at the same time only about 5% -10% of the actual life-cycle cost will have been spent. Whatever the figures are is not important, but it is important to realise what effect early design decisions have on life-cycle cost that will be spent later in the system's life. The relationships between the opportunity to influence the design, life-cycle cost committed, cost of change and cumulative life-cycle cost spent are shown in Figure 5.7 [Adapted from Blanchard and Fabrycky, 1998:37, 561 and Blanchard, 1998:82].

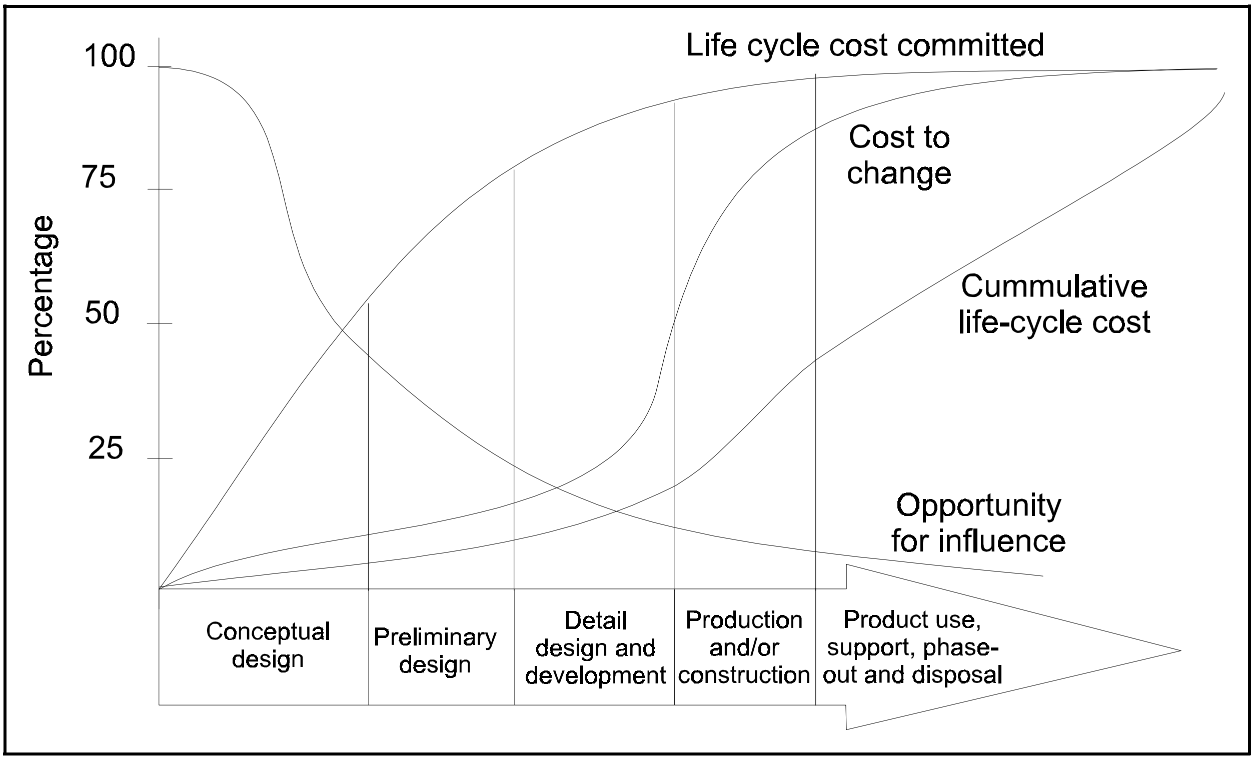


Figure 5.7: Systems engineering influence on design and cost
[Adapted from Blanchard and Fabrycky, 1998:37, 561 and Blanchard, 1998:82]

It must also be realised that with every decision made during the concept phase, not only is the total life-cycle cost determined, but a commitment is made towards the following:

- Technology employed for the system and sub-systems.
- Technology employed for production/construction.
- Technology employed for operations, support and maintenance.
- Operational, support and maintenance resources.

A major problem often experienced with the development of complex systems is the people responsible for taking the system concept through realisation into production, are not the same people responsible for its operation and support, even if the realisation system is within the same organisation as the host system. This functional division within organisations and having a budget allocated to the realisation system to bring the system of interest into being, many times lead to local optimisation of the overall system. As long as the realisation system meet the ability requirements of the system of interest within their allocated budget, it is many times perceived that the realisation system has performed well. After inserting the system of interest into the host system it is handed over to the host system's operational and support function. High life-cycle cost and low availability of the system of interest is now the problem of the operational and support function of the host system. Thus the non-performance of the realisation system in terms of the system's low availability and poor life-cycle cost becomes visible much later in the life-cycle when it is far too late. This non-performance is often wrongly blamed on inadequate management and technical competencies of the operations and support function of the host system, instead of blaming it on the realisation system failing to follow a whole-life, whole-system approach. It is therefore critical that the realisation system is contracted to develop and demonstrate, not only the ability requirements of the system of interest, but also the availability and life-cycle cost as part of the performance parameters of the system of interest.

5.8 The realisation system's approach to bringing systems into being

Having discussed the fundamentals of bringing a system into being, it leaves the discussion of an approach to doing so. As the goal of this research is not to provide extensive guidance to the overall systems engineering process, this section only provides a summary of the necessities of systems engineering.

Many factors influence the actual implementation of systems engineering and will vary from program to program. However, there is general agreement regarding the fundamental principles underlying systems engineering as well as the objectives that are to be achieved when following the systems engineering approach [Blanchard and Fabrycky, 1998:24]. These authors introduce different process models of systems engineering [Blanchard and Fabrycky, 1998:30-31)] which will not be discussed further.

The fundamental commonalities between all systems engineering processes are the following [Adapted from Blanchard and Fabrycky, 1998:23-24]:

- The realisation system should follow a holistic approach and top-down approach to system development in order to make sure that all interdependencies of the system characteristics are understood and consciously designed with the system's end goal, ability, availability, affordability and constraints in mind.
- The realisation system must ensure that the whole-life or life-cycle of the system of interest is considered to ensure proper achievement of system goals and objectives within the life-cycle constraints, and not only the development constraints. The following life-cycle phases should be considered:
    - Need exploration and requirements definition.
    - Concept design.
    - Detail design.
    - Production and/or construction.
    - Insertion.
    - Operations and support.
    - Retirement and phasing out.

  The phases do not always follow one after the other, they may overlap. The purpose of the overlapping of phases is two-fold namely:
    - To reduce the development time, especially where the risk is low to continue to the next phase whilst completing the previous one.
    - To ensure simultaneous consideration of system requirements such as safety, operability, supportability, reliability, maintainability, manufacturibility and affordability. This is sometimes referred to as concurrent engineering, which in any case is just good systems engineering.
- Underlying the systems engineering process is the basic approach to problem solving within each realisation phase namely:
    - Proper identification of needs.
    - Definition of the phase requirements.
    - Generation of alternatives through analysis and synthesis.
    - Selection of the best alternative.
    - Design of the selected alternative.
    - Integration of lower levels with the higher levels of the system.
    - Testing on all levels and comparing test data with the requirements and objectives.
    - Setting objectives for the next phase.

This process consist of many feedback loops and is the basic approach followed in each phase of acquisition. This process is shown in Figure 5.8 [Adapted from Blanchard and Fabrycky, 1998:27] and combines the top-down and bottom-up approaches to ensure system performance through analysis, design, synthesis, integration and test and evaluation.

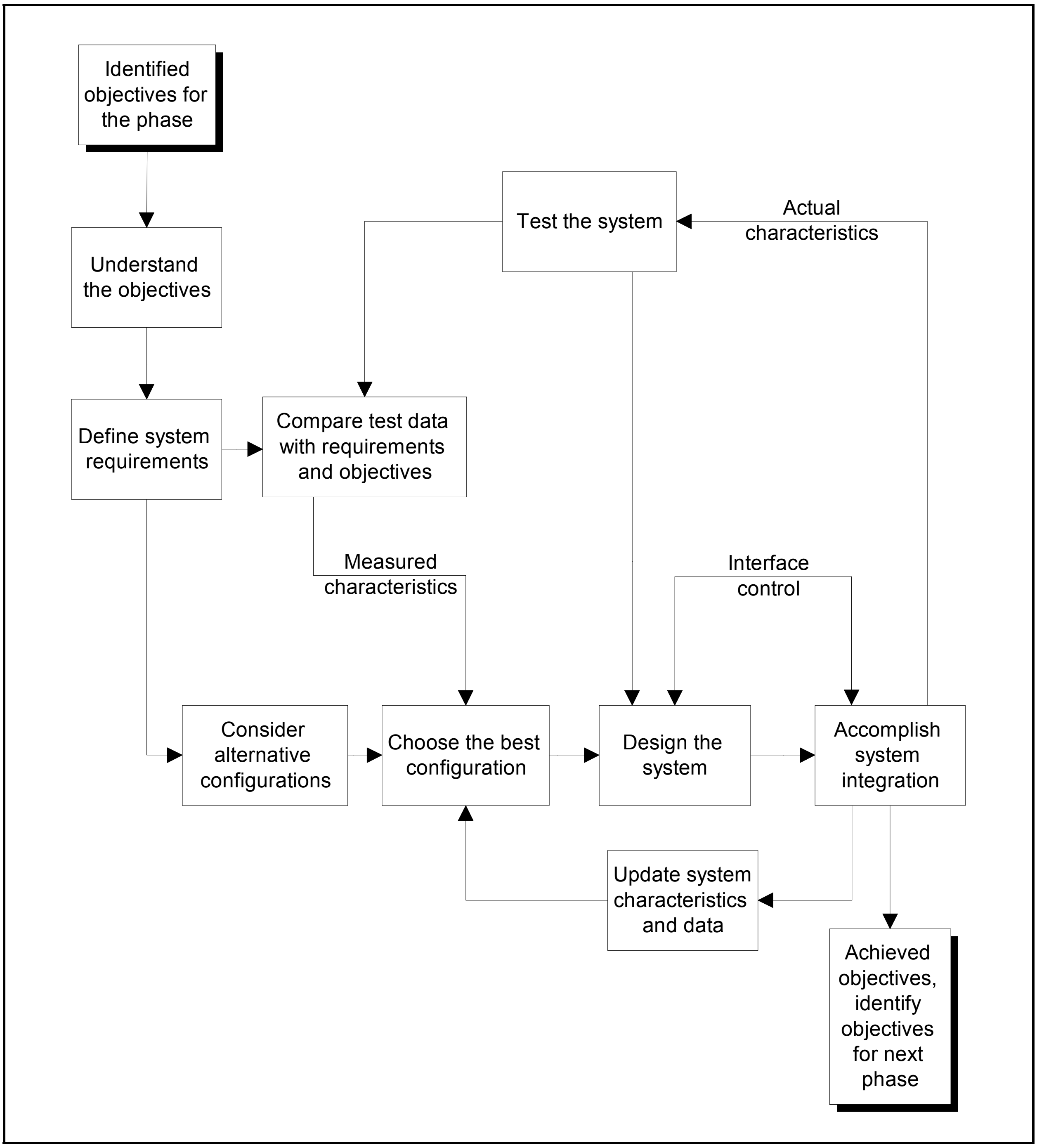


Figure 5.8: Generic steps and feedback loops within acquisition phases
[Adapted from Blanchard and Fabrycky, 1998:27]

- The systems engineering process draws from a wide variety of disciplines and technologies that needs to be integrated into a final system configuration. This requires an excellent understanding of the many interdisciplinary design and operational disciplines on the part of the systems engineer to ensure that the original customer requirements are met effectively and efficiently. In order to do this an extensive knowledge base is required by the systems engineer. The systems engineering process and resources required need to be managed from start to finish, i.e. the complete life-cycle of the system and is illustrated in Figure 5.9, adapted from Rottier [1999:39].

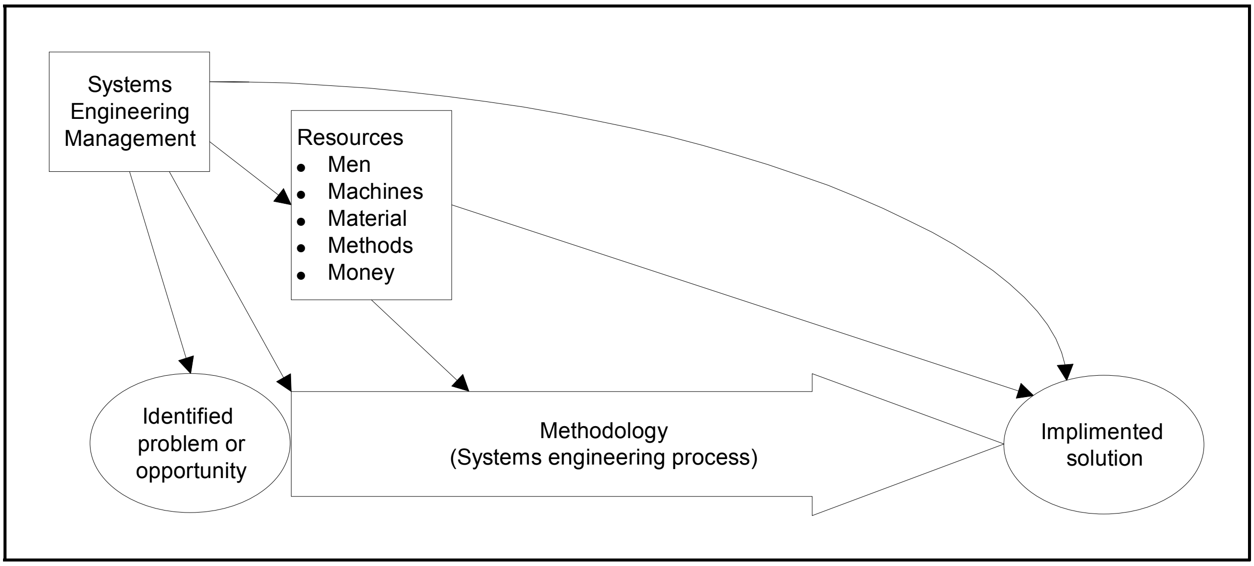


Figure 5.9: The systems engineering process and its interfaces with systems engineering management and required resources
[Adapted from Rottier, 1999:39]

- The resources required for executing of the systems engineering process consist of:
  - Men, those people performing the systems engineering.
  - Machines, consisting of technologies, equipment, facilities, tools, models, hardware and software required for the systems engineering activities.
  - Material and utilities to perform the systems engineering activities.
  - Methods, the know-how to perform the systems engineering activities as well as the knowledge base (data) required to do so.
  - Money, to pay for the resources used in the systems engineering process.

5.9 Chapter summary

New systems are required all the time due to the changing needs of man and also due to new technologies that are emerging which can improve the standard of living of man. Because of the ever increasing complexity of systems it is not possible anymore to have a single person responsible for the creation of a system and taking it through its life-cycle. Therefore team-work is of the utmost importance. Considering the aforementioned, the complexity and team-work requirements of bringing systems into being and operating them throughout their life-cycles require a multi-dimensional approach to bringing systems into being. These two dimensions are:

- The technical dimension that ensure technical compatibility of all parts of the system as well as human interfacing using:
  - analysis,
  - synthesis,
  - integration, and
  - test and evaluation.
- The management dimension that ensures the application of resources (men, machines, material, money and methods) to achieve the system objectives through:
  - planning,
  - organising,
  - directing, and
  - control.

Systems engineering is thus the formal process that ensures both the managerial and technical requirements of the system are met throughout the system's life-cycle, from conception to phasing out. Even though many feel that systems engineering has no role to play after the system of interest has been inserted, commissioned and successfully handed over to the host system, it is not true. The logistics engineering component of systems engineering, which will be discussed in much more detail later in this thesis, has an active and most important role to play during the operational and eventual phase out of the system of interest.

The implications of systems engineering for the realisation system is to view it from an organisational perspective. This means that the realisation system, whether part of the host system or not, must have the following capabilities:

- The capability to take the system of interest from conception to insertion into the host system.
- The capability to develop the management system of the system of interest alongside its technical capability and insert the management system into the host system's management system.
- The capability to provide support (technical and managerial) for the system of interest throughout its purposeful life.
- The capability to manage the realisation system's resources to accomplish the above mentioned.

Systems engineering is concerned with benefit-effectiveness consisting of ability, availability and affordability. When the interrelationships and dependencies between the design activity (whether conceptual or detail design) which provides for ability, and the other two key measurements of a system (namely availability and affordability) are considered, one can clearly see the importance of making the right design decisions early in the design process. Failing to take the availability and affordability into consideration when designing the system of interest, will make the system a total burden to operate from both an availability and cost point of view.

Irrespective of the type of system that is to be brought into being, a generic approach to formally go through the phases of creating the system can be followed. The approach and each phase can be tailored to suit the specific application. Understanding how to bring systems into being is essential for understanding the role and process of integrated logistic support in achieving the systems engineering objectives of ability, availability and affordability.