16

Creating Synergies for Systems Engineering: Bridging Cross-Disciplinary Standards

Oroitz Elgezabal and Holger Schumann
Institute of Flight Systems, German Aerospace Center (DLR)
Germany

1. Introduction

The increasing complexity of technical systems can only be managed by a multi-disciplinary and holistic approach. Besides technical disciplines like aerodynamics, kinematics, etc. cross-disciplines like safety and project management play an immanent role in the Systems Engineering approach. In this chapter, standards from different cross-disciplines are discussed and merged together to elaborate synergies which enable a more holistic Systems Engineering view.

After this introductory section, definitions of the terms *system* and *complexity* are given and the problems associated with the development of complex systems are introduced. The third section presents existing development philosophies and procedures. Additionally the mentioned cross-disciplines are introduced together with international standards widely established in the respective fields. Because the selected standards are not only complementary but also overlapping, the fourth section describes the harmonization approach carried out, together with the resulting holistic view. This combination of the standards enhances the benefits of the "traditional" Systems Engineering approach and solves many of the mentioned problems associated to the development of complex systems by taking also project management and safety aspects into a deeper and therefore, more holistic, account.

2. Background

The concept *system* has been defined in multiple ways since Nicolas Carnot introduced it in the modern sciences during the first quarter of the 19th century. Most of the definitions assigned to it are based on the Greek concept of *"σύστημα systēma"*, which means: *a whole compounded of several parts or members, literally "composition"*. An example of the remanent influence of the original *system* concept on the modern one is the definition provided by Gibson et al. (Gibson et al., 2007) which defines a system as *a set of elements so interconnected as to aid driving toward a defined goal*.

As an extension to the concept *system*, the term c*omplex system* is interpreted very broadly and includes both physical (mostly hardware and software) groupings of equipment to serve a purpose, and sets of procedures that are carried out by people and/or machines (Eisner, 2005). In complex systems, characteristics and aspects belonging to different fields

of expertise interact with each other. The factors which make a system to be complex are the interactions and interdependencies between the different components of a system. Those dependencies are not always obvious, intuitive or identifiable in a straightforward way. Especially, keeping a perspective of the whole system, together with all its implications in big projects, is complicated if not almost impossible at all. Even if the size is not a determinant factor for complexity, complex systems tend to be relatively large, with lots of internal and external interfaces. Additionally, in complex systems other kinds of considerations than those purely technical come frequently into play like political interests, international regulations, social demands, etc.

2.1 Problems associated with complex systems development

The development of complex systems implies other kinds of problems apart from those directly related with the different technical fields involved in it. Eisner summarizes in (Eisner, 2005) some of the problems associated with the design and development of complex systems. Eisner further classifies those problems into four different categories: Systems-, Human-, Software- and Management-related problems. Fig. 1 lists the mentioned problem categories together with their respective problems associated with the development of complex systems.

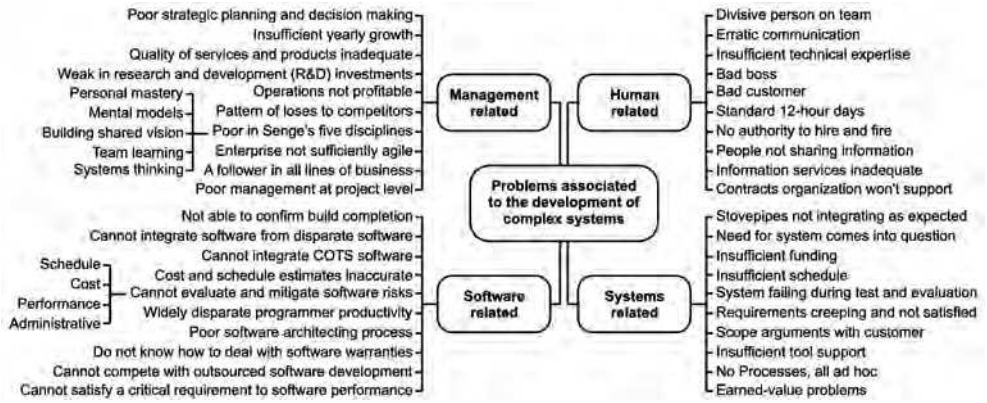


Fig. 1. Problems associated with the development of complex systems

As a consequence of all those problems, the efficiency during system development process decreases, which in fact can lead to a loss of money or project cancellation, both due to lower productivities. Besides, this efficiency decrease can result in higher project risks i.e. violation of deadlines or project failure during system verification phase due to poor system quality.

Another critical point associated with problems which belong to the previous classification like: *Erratic communication, People not sharing information, Requirements creeping and not validated, No processes, all ad hoc* and *Poor software architecting process* is the fact that they make the achievement and maintenance of traceability very difficult. Traceability is a key source of know-how in every company since it condensates the rationale behind every decision made during the system design process. Traceability is also vital for finding the location of design and production failures in case they are detected internally or reclamations from

customers take place. Finally, in the case of safety-related systems, it is a mandatory requirement for system certification as well as for failure and accident investigation.

All these problems result in a poor execution of system development processes which, in case they get established in the every-day working methodology of a company, could even threat the profitability and continuity of the company itself.

3. Existing development philosophies and procedures

3.1 System development philosophies

The development of systems in general, and of technical systems in particular, has been carried out since the foundation of engineering sciences, or even earlier. During that time, many different terms like systems analysis and systems integration have been used to make reference to the concept represented by the modern Systems Engineering approach. Currently two philosophies with different focuses are applied in the development of technical systems, the analytic and the holistic approach (Jackson, 2010).

On the one hand, the traditional approach taken for the development of systems is the *analytic approach,* which concentrates on the development of each system's element independently, without paying any attention neither to the system as a whole, nor to the interactions among the different elements conforming the system once they are assembled together. This design process is carried out according to the problem solving methodology stated by Descartes, which consists on dividing the complex problems into smaller and simpler problems. Once the top problem has been decomposed into a collection of atomic entities, the problems are solved hierarchically in an ascent way until a solution for the complex problem on the top is achieved. This kind of methodology, applied in the conventional engineering design, is suitable and valuable for the design of systems where the technological environment is subject to minor changes, system's goals are clear, and the amount of uncertainties is low.

On the other hand, the *holistic approach* is based on the *Systems thinking* philosophy which considers a system as a whole rather than as simply the sum of its parts, and tries to understand how the different parts of a system influence each other inside the whole. This approach also takes into consideration the boundaries and environment of the system-of-interest by determining which entities are inside the system and which are not, as well as by analysing the influence of the operating environment on the system to be developed. The *holistic approach* has also been considered as a problem solving method in which the different aspects of a problem can most effectively be understood if they are considered in the context of interactions among them and with other systems rather than in isolation. This problem solving nature has been also stated by Sage and Armstrong in (Sage & Armstrong, 2000). According to them, the *holistic approach* stresses that *there is not a single correct answer or solution to a large-scale problem or design issue. Instead, there are many different alternatives that can be developed and implemented depending on the objectives the system is to serve and the values of the people and organizations with a stake in the solution.*

The principles of *Systems thinking* state that events can act as catalysts which can heavily influence complex systems. Thereby, the events as well as the systems can be completely different. The events can have a technical, natural or timely source amongst others, while the systems can be from technical, political, social, or any other kind. In fact, identifying the

so-called *emergent* properties of a system that cannot be predicted by examining its individual parts is an exclusive feature of the *holistic approach* not provided by the *analytical approach*. This kind of methodology is suitable and valuable for the design of systems where the technological environment is subject to significant changes, system's goals are not clear, and the amount of uncertainties is high.

According to the provided definition of *complex system* and the description of its characteristics, it can be stated that the features of the *holistic approach* make it to be best suited to the characteristics required for the process of developing this kind of systems. Table 1 maps the specific challenges associated with the development of complex systems to the characteristics and features provided by the holistic system design approach. It shows how *holistic approach* provides measures to manage all the concerns present in a typical development process of complex systems.

The argument of the *holistic approach* being more suitable for developing complex systems is supported by the statement made by Gibson et al. in (Gibson et al., 2007) in which *system team members are supposed to be able to work across disciplinary boundaries toward a common goal when their disciplinary methodologies are different not only in detail but in kind*. A design process based on the *analytical approach* cannot fulfill this requirement since the system team members work exclusively in their own disciplines and they do not have access neither to a vision in perspective of the whole system nor to the context information related to the other elements in the system. The former is necessary for identifying the interacting elements while the latter is necessary for assessing the way the different elements interact with each other.

The characteristics of the *holistic approach* described above may propitiate the assumption that this approach remains pretty much superficial and that it does not get very detailed or specific. This assumption is incorrect in the sense that, inside the *holistic approach*, there is much effort devoted to in-scoping, high-fidelity modeling, and specification of system requirements and architecture (Sage & Armstrong, 2000).

Mapping of characteristics	
Complex systems	**Holistic approach**
• Difficulty to maintain whole system under perspective	• Systems considered as a whole, not as a sum of parts
• Big amount of internal and external interfaces	• Focus on understanding how the different parts of a system influence each other inside the whole
	• System aspects considered in the context of interactions among components and with other systems rather than in isolation
• Implication of different technical fields	• Identification of *emergent* properties that cannot be predicted by examining individual parts of a system
	• Analysis of unexpected interactions and cause-effect events
• Broad and heterogeneous stakeholders	• Consideration of system boundaries and operating environment

Table 1. Mapping of characteristics of complex systems and holistic approach

3.2 Standardized procedures as a means for managing complexity

As in any other field of life, the experience and knowledge acquired with the time plays a vital role in the design of complex systems. Past experience provides the system engineer with a set of rules of thumb, intuition and sense of proportion and magnitude, which combined together, result in a very valuable toolbox to be applied for proposing solutions, supporting judgements and making decisions during the development of complex systems. Those design principles, guidelines, or rules that have been learned from experience, especially with respect to the definition of the architecture of a system, have been considered by Jackson to constitute which is called heuristics (Jackson, 2010).

It is common that companies rely on heuristics-dominant system teams for the development of systems in areas considered as sensitive for the companies. However, this is a very individual-centred approach, in which system's or even company's know-how is concentrated in specific people and thus dependent on them. This kind of know-how is critical, since in the case of one key person leaving the team or the company, the know-how it possesses leaves with him or her, thus creating a loss of knowledge with two different consequences: On one side, the company loses all the existing information, creating a regression of company's know-how in the field. On the other side, it takes a lot of time to determine exactly which specific know-how has been lost and to assess which part of the know-how still remains in the company.

Another aspect of heuristics to be considered is that human beings unconsciously make use of the knowledge they possess in a specific situation in order to interpret the reality they confront. In other words, heuristics provide background information and helps to put the facts and figures in context and to interpret them. This means that two different members of the same system team might interpret in a different way and derive different conclusions from the same information just because they possess different background knowledge.

A standardized know-how management system can help making company's dependency on individuals' heuristics unnecessary or at least, less critical. The generation of standard documentation with predefined structure and contents allows condensing the most important information about projects and its transmission. A key piece of information that must be included in the standard documentation is the rationale behind the different decisions made in the project, in order to provide traceability. Standardized documentation means that anyone working in a company knows exactly which documents are available inside a project and which information do they contain. This makes possible to minimize the consequences of a key person leaving the team, since its successor ideally would be able to achieve the same knowledge status about the project in a fast and efficient way thanks to the traceability of decisions made. For the same reasons stated before, the information contained in the standardized documentation can be transmitted to every other member of the team or the company in a transparent way, thus enabling the achievement of homogeneous background information about the project that can be shared by all team members.

In the modern and globalized industrial market, where trends, products and technologies change very rapidly and companies worldwide compete fiercely for the same business niche, the reputation of a company frequently plays a determinant role. This reputation basically depends on the quality of the products they produce or the services they provide, which at the same time, greatly depends on the quality of the processes used during the

whole product's life-cycle. The definition of efficient and high-quality working methodologies and best practices takes place as a result of an iterative learning process which refines itself making use of the lessons learned during the development of past projects. All this know-how is considered as a strategic business active of every company and therefore it is condensed in standard practices and regulations that become mandatory for every employee of the company. Every time a new employee joins the entity, he or she must get started with those internal regulations and assimilate them.

Nowadays, the system development strategies based on the black box approach, which uses in-house developed proprietary technologies, has been substituted by a white box approach based on Commercial-of-the-Shelf technologies, where most of the system development workload is subcontracted to external entities. This subcontracting strategy has many associated advantages like the reduction of development costs and risks (derived from delegating the development of specific system parts to companies with more experience in that type of elements) among others. However, this strategy has also associated risks that must be correctly managed in order not to become drawbacks with highly negative effects. One of those risky factors is a higher communication flow between at least two different entities, which in general possess different working methodologies and tools. A standardized system development process, makes the exchange of information effective and efficient, since on one side, there is no risk of misinterpretation of the transmitted information and on the other side, the number of required transactions decreases due to the fact that every part knows which documents with which specific content must be delivered in every phase of the development process.

All these aspects have been also considered by Sage and Armstrong (Sage & Armstrong, 2000) who stated that the development process of any system in general, and of complex systems in particular, should fulfil amongst others, the following requirements:

- Systems engineering processes should be supportive of appropriate standards and management approaches that result in trustworthy systems.
- Systems engineering processes should support the use of automated aids for the engineering of systems, such as to result in production of high-quality trustworthy systems.
- Systems engineering processes should be based upon methodologies that are teachable and transferable and that make the process visible and controllable at all life-cycle phases.
- Systems engineering processes should be associated with appropriate procedures to enable definition and documentation of all relevant factors at each phase in the system life cycle.

In summary, standardized processes help to increase the productivity in system development activities by improving the transparency of all team members' work, which eases and advances communication and collaboration. They also help to increase the quality of working methods and products, as well as to manage company's know-how by enabling traceability of requirements, decision, rationales and deliverables. This traceability makes all working steps reproducible and improves consistency and integrity of all deliverables, contributing to the management of knowledge created during the process. Additionally, standardized processes help to mitigate risks by enabling comparability with previous development projects, amongst others, which supports monitoring and controlling of cost and schedule.

3.3 Fundamental development disciplines

The fundamentals of building and managing complex systems at the top level have been identified by Eisner in (Eisner, 2005). According to him, there are three areas which are critically important in building and managing complex systems: Systems engineering, project management and general management. The importance of these three areas has also been identified by Sage and Armstrong in (Sage & Armstrong, 2000) in which they state that, *Systems engineering processes should enable an appropriate mix of design, development and systems management approaches*.

Additionally, in the special case of developing systems whose failure could imply catastrophic consequences like big economic losses or human casualties, the concepts, methods and tools belonging to the safety engineering discipline must also be considered as fundamental.

The area of general management is an extremely broad topic, which is out of the scope of the current chapter and therefore the chapter's contents will concentrate on the other disciplines mentioned, i.e. Systems engineering, project management and safety engineering.

3.3.1 Systems engineering

Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems (Haskins, 2010). It is based on well-defined processes considering customer needs and all other stakeholders' requirements and it always profits from providing a holistic view on all problems across the whole development life-cycle. It has progressively attracted the attention in different fields of industry, as a methodology for managing the design and development of complex systems in a successful, efficient and straightforward way. According to (Gibson et al., 2007), *it is a logical, objective procedure for applying in an efficient, timely manner new and /or expanded performance requirements to the design, procurement, installation , and operation of an operational configuration consisting of distinct modules (or subsystems), each of which may embody inherent constraints or limitations.*

This conceptual definition of Systems engineering, states implicitly that the development process is defendable against external critics and that all the decisions made inside are objective and traceable. As it has been reasoned previously, traceability is a fundamental characteristic that must be present in every development process because of the multiple benefits it has associated with it, i.e. project reproducibility or the creation of know-how by means of stating the rationale behind the design decisions made, or listing and describing the risks found out and resolved during the development process.

Additionally, previous definition of Systems engineering also describes implicitly its holistic nature, by taking in consideration all the phases of a system's life-cycle and the interfaces and interactions between the system of interest and the systems related to it.

The field of Systems Engineering has published an international standard called *ISO/IEC 15288 – Systems and software engineering* (ISO 15288, 2008). It provides a *common framework for describing the life-cycle of systems* from conception up to retirement and defines associated processes and terminology. Processes related to project management are specified therein, but because of standard's scope focusing on Systems engineering, those processes do not cover the complementary domain of project management. The last update of the ISO 15288

Standard was released in 2008[1] which points to it as an active standard which is still in an iterative improvement status. Nevertheless, the standard has been consolidated with the INCOSE Handbook (Haskins, 2010) which is broadly established worldwide.

3.3.2 Project management

Project Management is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements (PMI, 2008). It is also based on well-defined processes regarding planning, executing, monitoring, and controlling of all working activities and the effective application of all assigned project resources. Project management profits from an always transparent status of all activities and deliverables and from the early identification of any risks.

It must be remarked that project management consists not only on applying the specific skills necessary for carrying out a project once it has been accepted, but also on managing the systems team itself on an effective manner.

Gibson et al. identify in (Gibson et al., 2007) some requirements for building an effective systems team. Aspects like having a leader, defining a goal and using a common working methodology with a well-balanced set of skills among members who pull together towards the goals have been identified as critical for achieving project's goal on schedule.

Sage and Armstrong (Sage & Armstrong, 2000) state in addition to this that systems engineering processes should possess following characteristics from the point of view of project management: 1) *they should support the quality assurance of both the product and the process that leads to the product*, 2) *they should be associated with appropriate metrics and management controls* and 3) *they should support quality, total quality management, system design for human interaction, and other attributes associated with trustworthiness and integrity*. These statements support the idea of a holistic design process for developing complex systems.

The Project Management Institute (PMI) has published the guide to the *Project Management Body of Knowledge (PMBOK)* (PMI, 2008). This document is recognized as a standard by classical standardization entities like ANSI and IEEE. It covers all management topics completely, without taking engineering aspects into scope.

3.3.3 Safety engineering

Safety engineering can be seen as *a set of well-defined processes aiming at achieving freedom from unacceptable risk* (ISO 61508, 2009), together with the application of methodologies in order to quantify and to prove it. Due to the fact that not only the complexity of modern systems increases, but also their capabilities, the amount of functions performed by a system also raises. Inside those functions, there are safety-related functions performed by specific systems included whose failure would lead to important economical and material damages, severe injuries, or even fatalities. The increase of capabilities in systems, together with the growing humankind's dependency on them, leads to the fact that more often the safety depend directly on a fail-safe operation of systems. Furthermore, the more safety-related

[1]The references made to ISO 15288 relate to the 2008 version of the standard, if not explicitly mentioned otherwise.

equipment is integrated into a system, the bigger is the probability that one system element fails. This, in turn, increases the concern about safety among the society. Additionally, due to the overall complexity of systems, assessing the impact of single failures on them and setting up preventive or corrective actions is a very challenging task. All these facts mentioned above have contributed to making safety considerations become more and more essential to modern development processes.

In the field of Safety engineering, a widely considered international standard is the *ISO/IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems* (ISO 61508, 2008) which sets out a generic approach for all safety life-cycle activities for systems whose elements perform safety functions. This standard is field-independent and sets the basis framework in which additional, branch-specific industrial safety standards are based, e.g. ISO 26262, the new safety standard for the automotive domain, or EN 50128 for railway systems.

3.4 Drawbacks of standards involved in complex-systems development

The historical evolution of every of the standards presented above has grown independently from each other. This fact implies that, even if the participation of the three cross-disciplines and their combined use has been recognized as critical for the development of complex systems by the industry, the different standards are poorly connected or not connected at all among them. This leads to a situation in which the standards overlap with each other in many processes and activities, and in the worst case they even could contain conflicting directives. Additionally, there is a lack of consolidated set of terms used inside the standards. Every standard makes its own definition of terms which creates confusion and misunderstandings and makes the cross-disciplinary communication difficult.

Besides, the standards themselves possess some deficiencies that difficult their interpretation and understanding, and consequently, their implementation. On one side, the ISO 15288 standard does not provide any sequence diagrams showing the relationships between the processes and activities contained in it. On the other side, the ISO 61508 standard lacks of a detailed description of the inputs and outputs associated with the different activities it describes.

4. Systems engineering approach based on international standards

4.1 General description

The holistic Systems engineering view described in this work takes the ISO 15288 standard as its core and tries to combine it with the other two standards introduced above. Some of the technical processes contained in the ISO 15288 are also addressed by the safety and project management standards respectively, providing interfaces where information can be exchanged among them or even where processes can be merged together. This combination of standards can be noticed in the case of the project related processes of ISO 15288, which are completely replaced by those defined inside the PMBOK standard, due to the fact that this standard considers them in a much more detailed way. The agreement processes defined by the ISO 15288 standard are also considered by the PMBOK standard inside the procurement area, but in this work, merging the agreement processes of both standards has been considered out of scope.

From the five organizational project-enabling processes defined by the ISO 15288 standard, only the *Human resource management* and *Quality management* processes are explicitly addressed by the PMBOK standard. The remaining three processes are not explicitly treated by the project management standard and therefore they are not considered inside the present work. Fig. 2 shows the process groups defined by the systems engineering standard together with an overview of the process groups also addressed by the project management and safety standards.

4.2 Harmonization process

The analysis and comparison of different items like the standards mentioned above, is logically impossible without a common reference framework in which all the items to be compared can be represented.

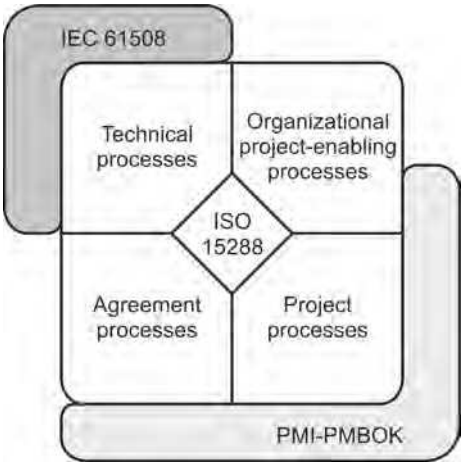


Fig. 2. Overlapping of considered standards regarding process groups

A detailed analysis of the three international standards has revealed that no common reference framework exists among them. This fact implies that before any task of the merging process can be carried out, e.g. comparison and identification of interfaces among the standards, a reference framework must be defined. The PMBOK standard provides a clear overview of its management processes structured in a two-dimensional matrix, representing different process groups in its columns against specific knowledge areas in its rows. This kind of representation based on a matrix has been considered by the authors as a clear and valuable means for analysing, comparing and merging the different international standards and consequently, it has been selected as the reference framework for the merging process.

None of the ISO standards analysed defines process groups or knowledge areas in the way that PMBOK does. The PMBOK standard defines process groups according to a temporal sequence while the ISO 15288 standard defines the process groups on a purpose basis. As a consequence, their respective reference matrices of both ISO standards need to be created from the scratch. Instead of the process groups used by PMBOK standard, the different life-cycle stages named by the ISO 15288:2002 standard have been taken. In the case of the

knowledge areas, if the ones from the PMBOK standard were not appropriate, new ones have been defined.

This approach showed that the matrices of both ISO standards can be merged to one unique matrix while the mapping of management process groups of the PMBOK standard into the life-cycle stages of the ISO standards is not possible. This is due to the fact that project management activities are carried out during the whole life-cycle of the system-of-interest and not just during a specific stage. Besides, there are also several knowledge areas regarding management, e.g. procurement, which cannot be considered together with technical processes. In consequence, the management and life-cycle stages have to be considered as parallel stages and two different process matrices have been created; one for management processes and another one for technical processes, respectively.

Finally, the processes being assigned to the same stage and knowledge area inside the technical processes' matrix are good candidates for interfacing or merging. After the description of the two matrices, a detailed analysis of the processes follows based on the matrices.

4.2.1 Management processes

The matrix shown in Fig. 3 is taken from the PMBOK standard. The columns represent process groups which can also be seen as project management stages starting with *Initiation* and ending with *Closing*. Each of the rows represents a typical project management topic, which is further called knowledge area. All of the forty two management processes specified by the PMBOK standard are classified into the cells resulting from the crossing of five process groups' columns with the nine knowledge areas' rows.

4.2.2 Technical processes

In the case of technical processes, the ISO 15288 standard does not define stages for the life-cycle of systems. However, a division of the life-cycle in various stages was provided in its previous version, ISO 15288:2002. These life-cycle stages have been assigned to the columns of the respective matrix. For the rows, ISO 15288 standard defines four knowledge areas (as shown in Fig. 2), in which the life-cycle processes are grouped by their purpose. However, these knowledge areas are not useful for comparing the processes with those contained in the other standards. Therefore, those used in the project management matrix were considered. Only two knowledge areas, *Scope* and *Quality,* were found to be also relevant for technical processes. Two further knowledge areas have been defined by the authors. On one hand, *Realisation* represents all activities which elaborate the outputs of the *Scope* area, which then can be quality-checked. On the other hand, *Service* describes all the activities to be carried out during the operating life of a system.

The ISO 15288 standard does not explicitly assign any processes to any life-cycle stages. In fact, the processes are initiated in one or more stages and some can be executed sequentially or in parallel. In this work, an interpretation process has been carried out in which the processes of the standard have been assigned to the cells of the matrix described above. The aim of this interpretation work was to enable the comparison and analysis of the processes and activities of the three standards in order to facilitate the identification of possible interfaces and overlapping areas between the different standards.

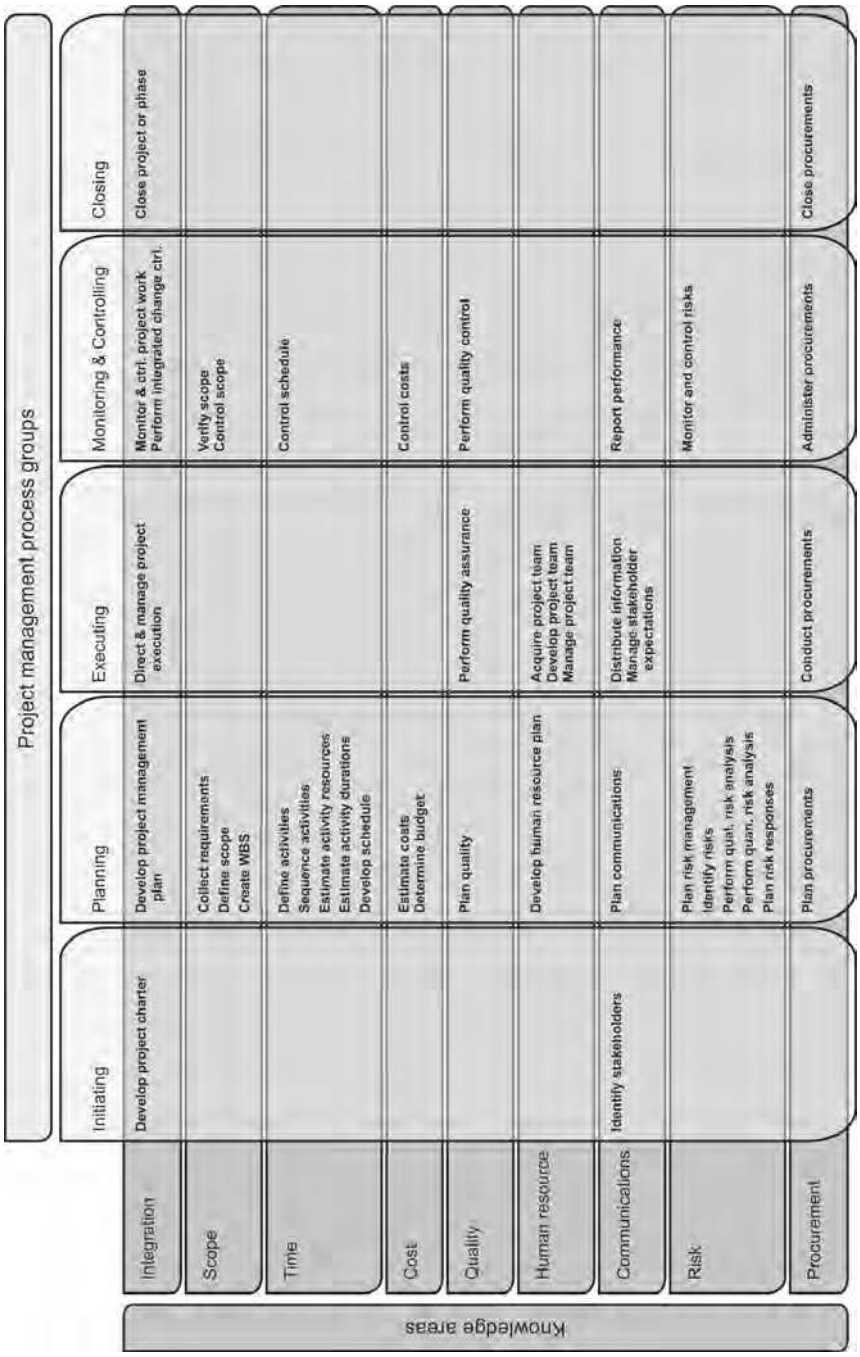


Fig. 3. Project management processes assigned to process groups and knowledge areas

The eleven technical processes specified inside ISO 15288 have been spread over the matrix using a black font, as depicted in Fig. 4. Inside the *Conception* stage, three different processes have been assigned to two different knowledge areas. The two processes dealing with requirements have been assigned to the *Scope* area, while the *Architectural design* process has been assigned to the *Realisation* area. In the first case, requirements specify the scope of the system. In the second case, the process was assigned to that specific area because one of the process' activities is to evaluate different design candidates, which cannot be done in the development stage or later ones. Besides, the process generates a system design based on the requirements elicited in the scope area, which supports its assignment to the *Realisation* row.

The *Production* stage contains *Transition* and *Validation* processes in two different knowledge areas. *Transition* process has been assigned to the *Production* stage because the development ends before the transition of the system (ISO 24748-1, 2010). In the same way as with *Verification*, *Validation* has also been seen in the *Quality* area. It must be remarked that the *Validation* process has been considered by the authors to take place at the end of the transition, in which at the end, the customer accepts the system delivered and installed in its operational environment. *Operation* and *Maintenance* belong to *Utilization* and *Support*, while *Disposal* can be found in the *Retirement* stage. All of them are assigned to the *Service* area. The activities of the Disposal process can also be seen as a service in the widest sense.

The ISO 61508 standard defines sixteen so-called life-cycle phases. In this work, they are interpreted as activities because for each of them, its inputs and outputs are defined and in the corresponding standard's chapters, tasks are indicated that have to be carried out. The standard neither defines any superior life-cycle stages comparable to ISO 15288:2002 nor defines any knowledge areas. For this reason, and because the activities are also of a technical kind like the processes of ISO 15288, they have been assigned to the same matrix shown in Fig. 4.

The matrix contains all of the sixteen activities defined by ISO 61508, illustrated by a grey font. Six activities are assigned to the *Conception* stage divided in two different knowledge areas. *Concept*, *Overall scope definition*, *Hazard and risk analysis* and *Overall safety requirements* have been assigned to the *Scope* area because they contribute to defining the scope and safety related requirements for the design. The *Overall safety requirements allocation* and *System safety requirements specification* have been assigned to the *Realisation* area. This is due to the fact that both processes specify and allocate safety requirements to designed system elements during the *Architectural design* process.

Inside the development stage five different processes have been assigned into three different knowledge areas. First, *Realisation,* O*ther risk reduction measures* and *Overall installation and commissioning planning* have been assigned to the *Realisation* area because they address questions related to the physical implementation of the system. The two remaining planning activities, i.e. *Overall safety validation planning* and *Overall operation and maintenance planning* have been assigned to the *Quality* and *Service* knowledge areas respectively. The planning activities typically take place in parallel to the implementation and they must be carried out before the system is installed and validated in its operational environment.

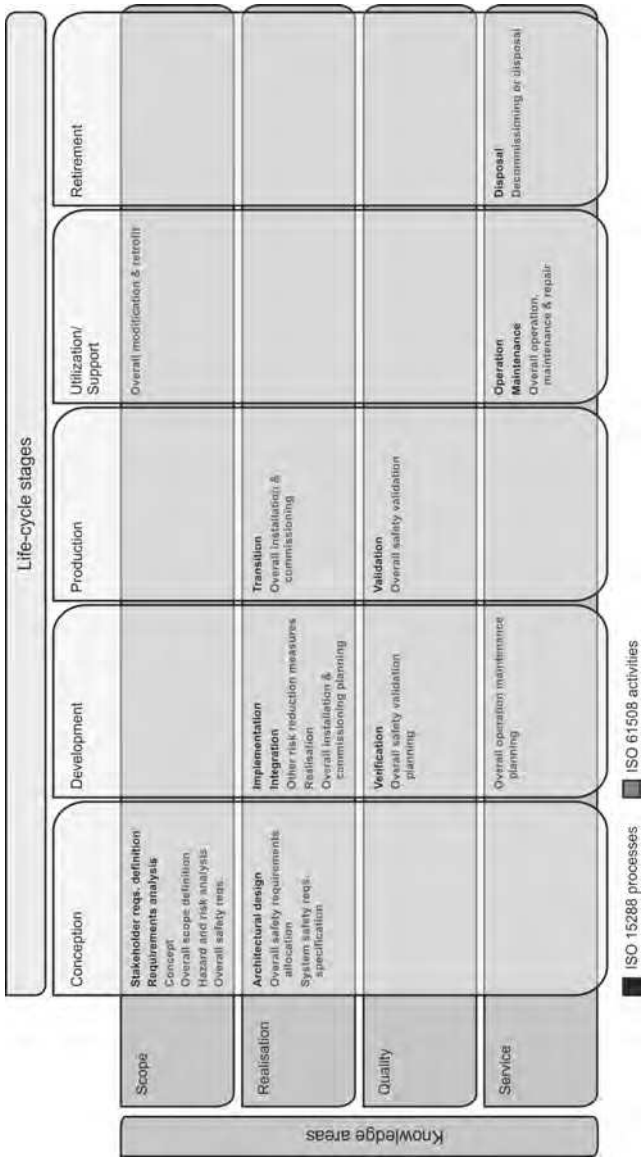


Fig. 4. ISO 15288 technical processes and ISO 61508 activities assigned to life-cycle stages and knowledge areas.

Inside the *Development* stage, another three processes have been assigned to the *Realisation* and *quality* areas. On one hand, the processes *Implementation* and *Integration* have been assigned to the *Development* and *Realisation* area because the physical creation of the real system takes place inside them. On the other hand, the *Verification* process is part of the *Quality* area because it contributes to guarantee the quality of the system-of-interest under development.

The *Overall modification and retrofit* activity has been assigned to the *Support* stage because this activity is typically initiated when the system is under operation and its support is active. Due to the fact that the output of this activity can affect all knowledge areas including the scope, it has been assigned to this overall area. The last two activities of the ISO 61508 standard, i.e. *Overall operation, maintenance and repair* and *Decommissioning or disposal* can be found in the *Service* area, assigned to the corresponding life-cycle stage.

4.3 Detailed standards interfacing and merging process

Those processes which are in the same life-cycle stage or knowledge area or both, bear potential for being harmonized. After an in-depth analysis of the three standards, eleven information and twelve process interfaces have been respectively identified. On one hand, information interfaces represent some kind of information generated by any of the standards, which is provided to the other standards for its use. E.g. safety requirements provided by the ISO 61508 are merged into the *System requirements* document generated by the ISO 15288 standard. On the other hand, process interfaces represent similar activities that are carried out in at least two of the standards, which in consequence, can be put together in order to avoid duplicities that constitute a waste of resources.

Because processes basically describe a sequence of activities, they are typically represented by some kind of flow diagram. For this reason, a standardized graphical notation for process diagrams has been selected to represent the relevant process parts and the outcome of their merging.

4.3.1 Business Process Model and Notation (BPMN) specification

The Object Management Group (OMG), a non-profit consortium dedicated to developing open computer industry specifications, took over the development of the BPMN specification in 2005. BPMN's primary goal is to provide a notation that is readily understandable by all business users, from the business analysts, to technical developers, and to managers who will manage and monitor those processes. (OMG, 2011)

The notation used in Fig. 5 to Fig. 8 corresponds to BPMN. The processes defined in the different standards (*activities* in BPMN) are represented as boxes, their outputs (*data objects*) are depicted by the leaf symbol, and the arrows illustrate the sequence flow. Circle symbols represent either the start or end event, or they describe an incoming or outgoing link to another diagram or (not depicted) process. In BPMN, a diamond symbol illustrates a gateway control type which marks the point where sequence flow paths are joined or divided. Gateways that initiate or merge a parallel sequence flow are expressed by a diamond containing a *plus* symbol. In the following diagrams, those gateways have been mostly omitted for the sake of simplicity and size. Gateways that introduce a conditional sequence flow are expressed by an empty diamond. Horizontal *pool lanes* represent a categorization of activities.

4.4 Harmonization result: The Holistic Systems Engineering view (HoSE)

Fig. 5 to Fig. 8 represent the product life-cycle stages defined in ISO 15288:2002 respectively. Every figure contains the project management as well as technical processes corresponding to the specific life-cycle stage. Due to length constraints a complete in depth representation

of all standard's levels is not possible, thus only the top level view has been provided. The processes of every standard are contained in a pool lane. The ISO 15288 standard is depicted in the middle pool lane of each figure. In case of the PMBOK standard, only the processes related to technical activities have been considered. Every sequence flow arrow crossing a lane represents an information interface between the corresponding standards.

4.4.1 HoSE conception stage

In Fig. 5, the corresponding processes of the three international standards for the *Conception* stage are shown. This includes eight of the technical processes already assigned to the conception stage as depicted in Fig. 4 as well as three related management processes.

In every standard, one initiating process is defined. Regarding the PMBOK, the first process is the *Identify stakeholders* process, for the ISO 15288 it is the *Stakeholder requirements definition* process, and so on. In this case, the *Identify stakeholders* process has been selected. Looking at the activities of the ISO 15288 *Stakeholder requirements definition* process and its outputs, it shows that it includes a sub-activity called *Identify the individual stakeholders*. This activity matches exactly with the *Identify stakeholders* process from PMBOK which identifies the related stakeholders and which lists them in an output document called *Stakeholder register*. As a consequence, the ISO 15288 sub-activity has been merged together with the PMBOK process and the *Stakeholder register* document it produces has been provided as an input to the remaining activities inside the *Stakeholder requirements definition* process of ISO 15288.

In the PMBOK lane in Fig. 5, the next process is the *Collect requirements* process. This can be merged with the activity *Elicit stakeholder requirements* of the *Stakeholder requirements definition* process from ISO 15288. At this point, a distinction between product and project requirements, as explicitly recommended by the PMBOK, helps to differentiate between project's progress and system-of-interest's advancements. In this way, PMBOK's activity of eliciting product requirements is merged into the ISO 15288 process, which also includes merging the techniques of facilitated workshops and prototypes into the ISO standard. In consequence, the output documents of the *Collect requirements* process are changed to project (only) requirements, project (only) requirements traceability matrix, and an (unchanged) requirements management plan. The sequence flow of the documents is kept as defined in the PMBOK, as illustrated by the grey lines.

The separation of the requirements into stakeholder and system requirements, as explicitly recommended by ISO 15288, enables the consideration of different views on the requirements. Stakeholder requirements define high-level functions from the point of view of client's expectations, while system requirements define functions in more detail from a technical perspective. Both kinds of requirements belong to the problem domain and not the solution domain. In other words, they try to specify what should be developed and not how it should be done. The stakeholder requirements constitute the input for the *Concept* activity of ISO 61508 and provide the level of understanding of the system-of-interest and its environment, required by this task. The *Concept* activity includes performing a *Functional hazard analysis (FHA)* which contributes together with safety-related requirements to the stakeholder requirements by identifying the likely sources of top-level hazards for the system. Those enhanced stakeholder requirements complement the requirements flowing to further PMBOK or ISO 15288 processes.

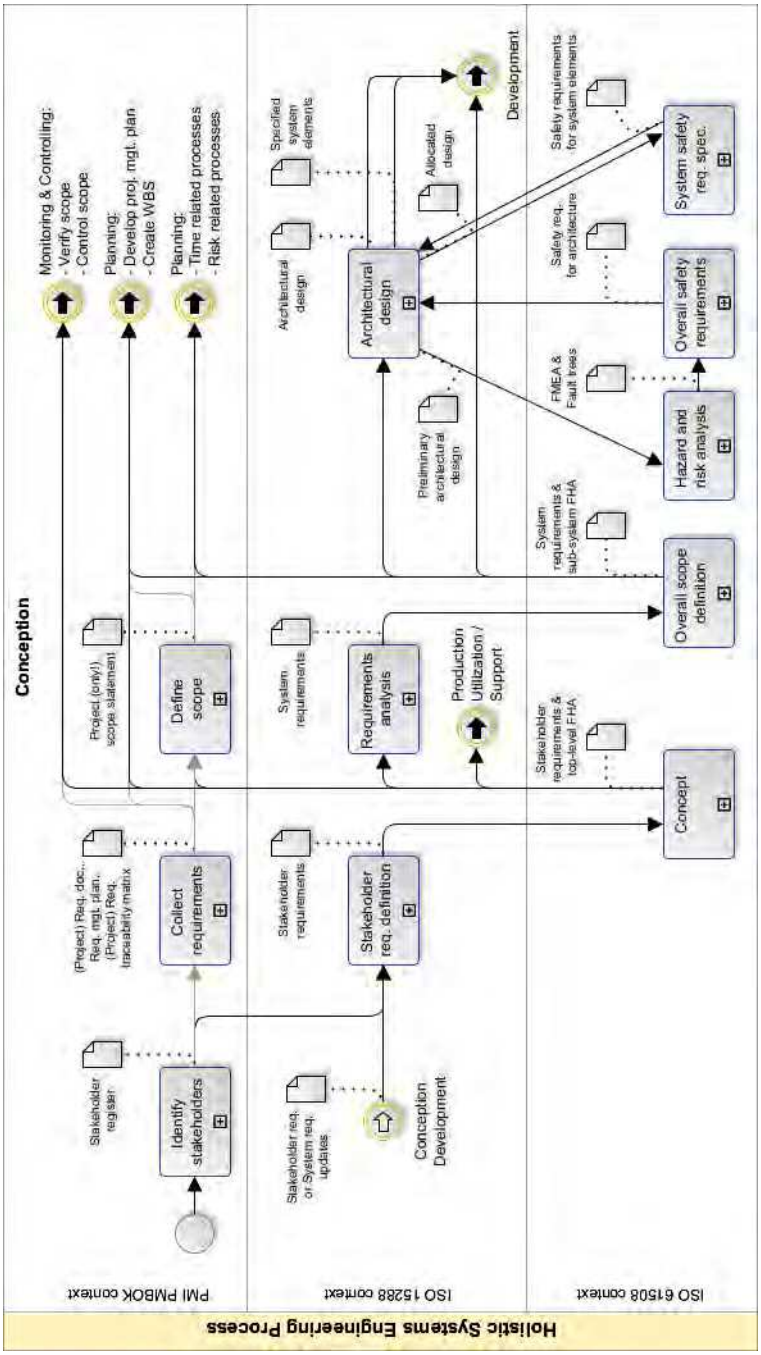


Fig. 5. Conception stage of the holistic systems engineering view

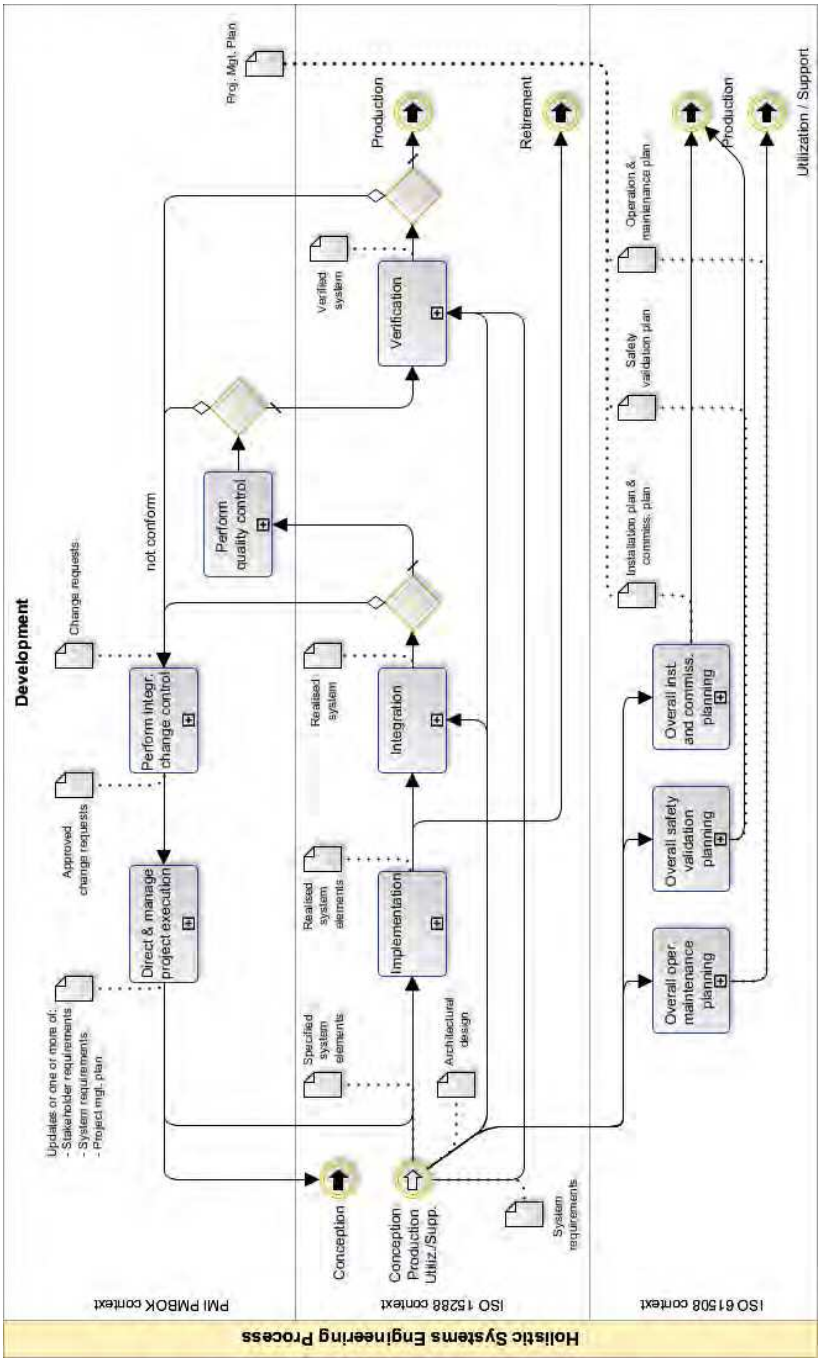


Fig. 6. Development stage of the holistic systems engineering view

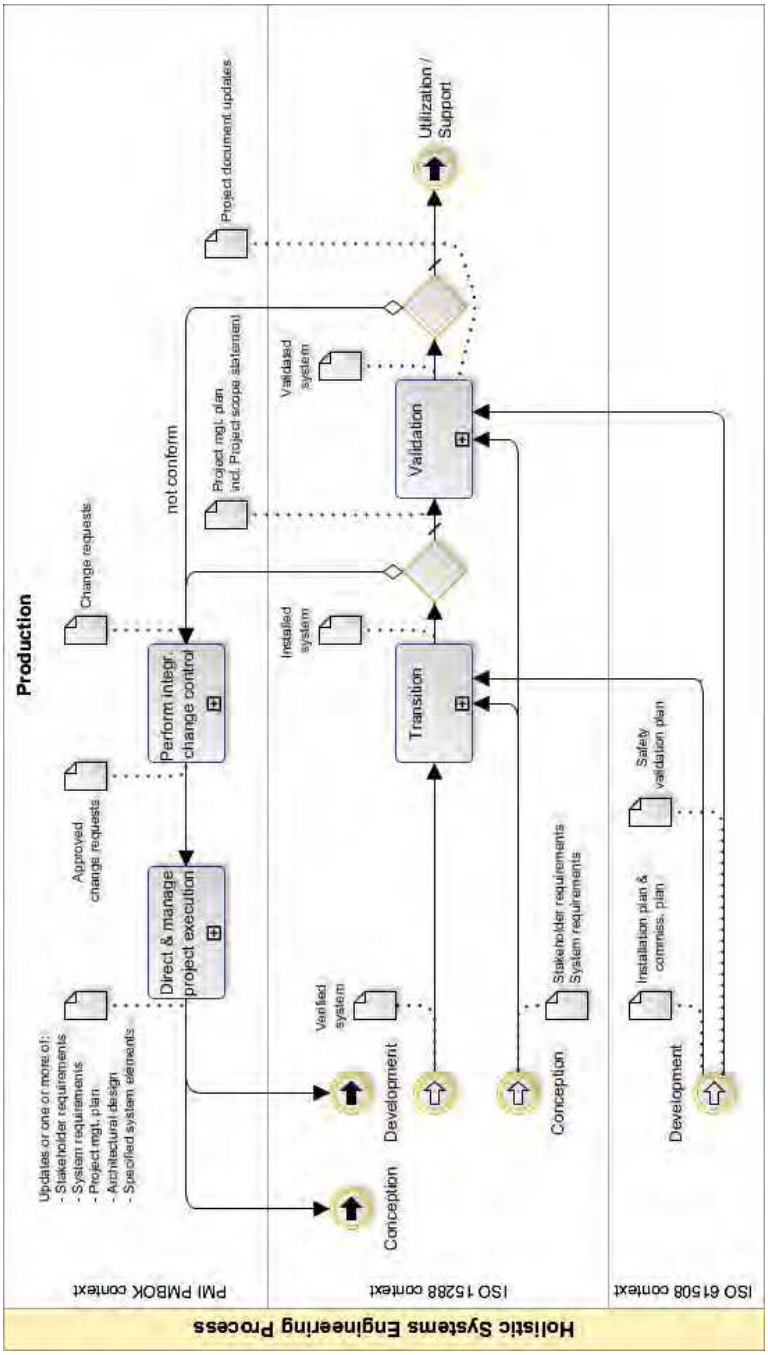


Fig. 7. Production stage of the holistic systems engineering view

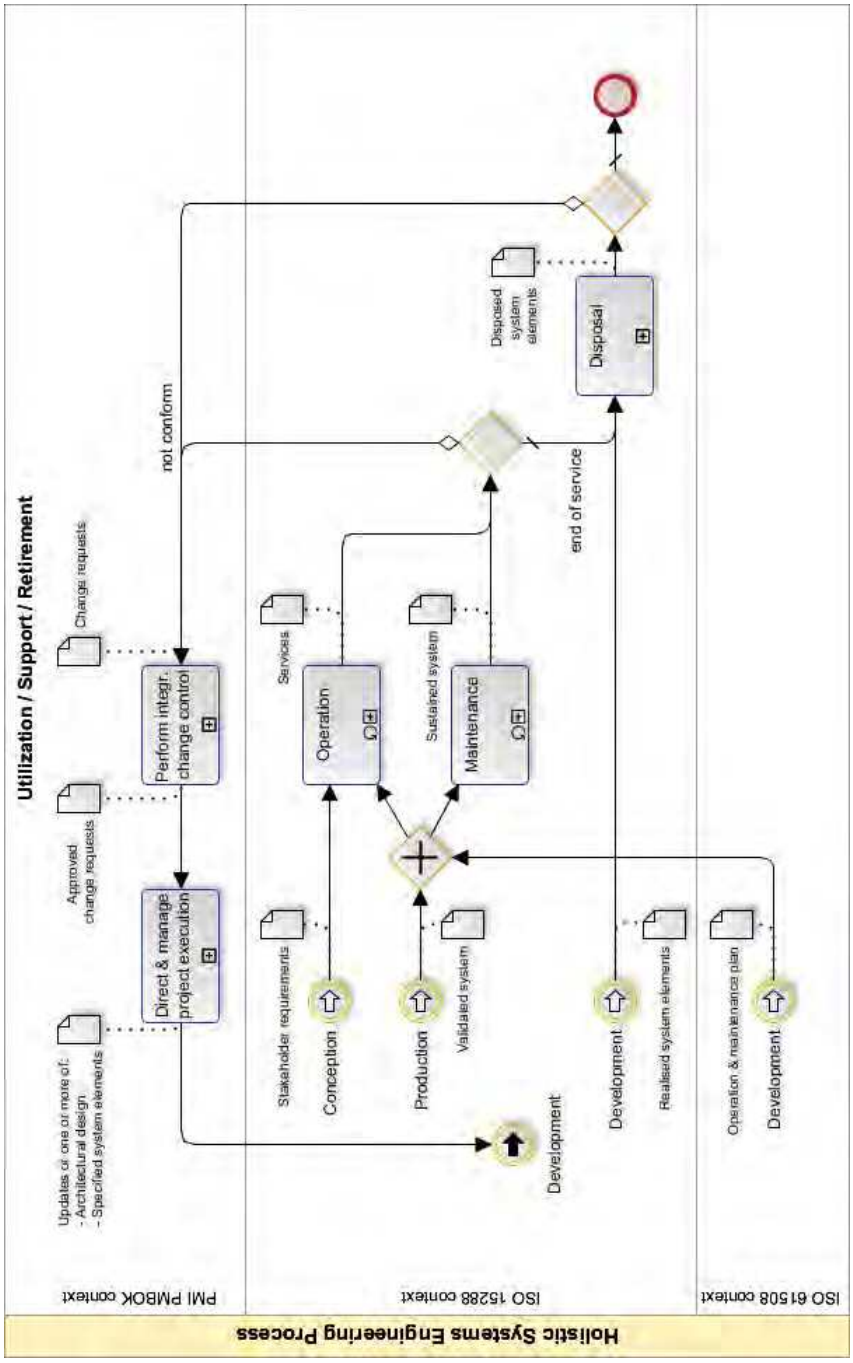


Fig. 8. Utilization, support, and retirement stages of the holistic systems engineering view

The *Requirements analysis* process of the ISO 15288 refines stakeholder requirements into technical system requirements. In the holistic view, the technique of P*roduct analysis*, specified in PMBOK's *Define scope* task, and the product related *Scope statement* are moved into this process. The complete system requirements are used by the *Overall scope definition* activity of ISO 61508 to refine the identified hazards and to specify the boundary and scope of the system-of-interest from the safety perspective. Both, *Requirements analysis* and *Overall scope definition* processes, could disclose weaknesses in the stakeholder requirements, which enforce the revision of the requirements (not depicted in the figure for the sake of clearness). The resulting enhanced system requirements flow into related PMBOK processes and into the *Architectural design* process of ISO 15288.

As shown in Fig. 5, the *Architectural design* process of ISO 15288 is split into several parts for being able to accommodate the safety assessment related activities. First, a preliminary architectural design is created and passed to the *Hazard and risk analysis* activity of ISO 61508. In this process, a *Failure Modes and Effects Analysis (FMEA)* together with a *Fault Tree Analysis (FTA)* is performed based on the provided design. The FMEA table and the fault trees are used in the *Overall safety requirements* activity to create safety related requirements for the architecture like required reliability and redundancy levels.

Those requirements are fed back into the *Architectural design* process which provides a refined design where system elements are identified and all requirements are allocated to the related elements (*Allocated design*). The allocation activity also includes the allocation of safety requirements which means that the *Overall safety requirements allocation* activity of ISO 61508 standard can be merged into the *Architectural design* process. In the *System safety requirements specification* activity, safety requirements for the system elements are identified which again influence the design refined in the *Architectural design* process. Finally, an architectural design is created representing the whole system, its decomposition, and its interfaces. Additionally, all system elements are specified in detail to enable their realization in the next stage: the development stage.

4.4.2 HoSE development stage

Fig. 6 shows the six technical processes assigned to the development stage in Fig. 4 as well as three related management processes. The specified system elements created in the *Conception* stage are realized inside the *Implementation* process of ISO 15288. *Realization* and *Other risk reduction measures* activities of ISO 61508 have been merged into this process since both of them are related with the physical implementation of the system-of-interest. The realized system elements resulting from the *Implementation* process are passed to the *Integration* process for further development or to the *Disposal* process, in case that the production of the system-of-interest has been cancelled. On the sub-contractor side, verification, quality control, and validation tasks may also follow directly after or within the *Implementation* process.

During the Integration process, the physical system elements are assembled together according to the architectural design. This process ends with the physical implementation of the system-of-interest including its configuration. During system integration, problems or non-conformances may arise, which lead to change requests.

Those requests are explicitly managed by PMBOK's *Perform integrated change control* process. Approved change requests enforce corrective actions to be carried out within the *Direct and manage project execution* process of the same standard. This may include revising the corresponding requirements, updating the project management plan, implementing an improved system element, or cancellation of the project, in the worst case. *Overall modification and retrofit* activity of the ISO 61508 standard is also responsible for managing change requests with regard to safety aspects, thus it has been merged into the change control process of the PMBOK standard.

As shown in Fig. 6, a PMBOK process called *Perform quality control* follows a successful integration, but it can also be carried out after *Implementation* and/or *Verification* processes. The goal is to check the quality of the output provided by the related process. Any non-conformances are managed like described in the previous paragraph. The *Verification* process of ISO 15288 checks if the realized system meets the architectural design and the system requirements which can also include quality requirements. Again, non-conformances may arise in this process; otherwise, the verified system can be transferred into the *Production* stage.

During the implementation of the system or its elements, safety related planning must be performed according to ISO 61508. The corresponding outputs are plans regarding installation, commissioning, safety validation, operation, and maintenance. Those plans have to be integrated into the project management plan.

4.4.3 HoSE production stage

In the *Transition* process of ISO 15288, the verified system is set up in its operational environment. This is done under consideration of stakeholder and system requirements and the installation plan provided by ISO 61508 which contains a description of the operational environment. The *Overall installation and commission* activity of ISO 61508 also deals with the installation aspects of safety-critical systems. Therefore, it has been merged into the *Transition* process of ISO15288 standard.

After the transition, during the ISO 15288 *Validation* process, the installed system is validated against the requirements and the safety validation plan. PMBOK's *Verify scope* process and the *Overall safety validation* activity of ISO 61508 have been merged into this process due to their common goals. To enable the verification of project's scope as required by PMBOK, the *Validation* process is enhanced by the project validation task from PMBOK which requires the project scope statement as an input document. This additional task may lead to project document updates regarding the current state of the project or product.

Non-conformances during *Transition* or *Validation* are managed as already described. They can affect any requirements, designs, plans, or realized system elements which leads to a reiteration of the corresponding process. After a successful *Validation* process, the system, including its operational configuration, can be passed to the *Utilization* and *Support* stage.

4.4.4 HoSE utilization, support, and retirement stages

The validated system and the safety related operation and maintenance plan are the inputs for the next processes of ISO 15288. During the *Operation* process, the system is used to

deliver the expected services meeting the stakeholder requirements. The *Maintenance* process is typically applied in parallel to *Operation*. It enables a sustained system. The *Overall operation, maintenance and repair* activity of ISO 61508 is split in two and the corresponding parts are merged into the respective processes. *Operation* and *Maintenance* are carried out uninterruptedly until non-conformances arise or the end of service is reached.

During system operation and/or maintenance, change requests regarding the system or the services it delivers may arise. These must be evaluated through PMBOK's *Perform integrated change control* process. The *Overall modification and retrofit* activity of ISO 61508, responsible for guaranteeing the safe operation of the system, has been merged into this process. If the intended modification is unfeasible or system's end of service is reached, the *Disposal* process organizes the system's retiring and disposing. The *Decommissioning or disposal* activity of ISO 61508 has the same function, thus they have been merged together.

4.5 Harmonization summary

Fig. 9 illustrates a general overview of the harmonization work done. It shows the considered disciplines of project management, systems engineering and safety engineering together with their identified interfaces. There are two kinds of interfaces: On one side, *Information interfaces* express a dependency between information as well as documents of different standards. An information interface results in a merge or change of the information, or document flow. On the other side, *Process interfaces* represent a merge of whole processes or process parts of different standards.

It must be remarked that interfaces between the three standards are present in every of the life cycle stages. This reinforces the usefulness of consolidating the processes of those three standards into a holistic view.

4.6 Benefits of the holistic systems engineering view

The use of standardized procedures during the development of complex systems has many associated advantages. As previously stated in section 3.2, these advantages arise in different aspects of a company. From a commercial point of view, standardized procedures contribute to increase the efficiency of company's processes, to improve the communication with subcontractors and clients, and as a result of those, to increase the quality of the products or services a company offers. From a corporate point of view, standardized procedures provide the basis for traceability and storing of decisions' rationale, which constitute the fundamental factors for generating and managing company's know-how.

Most of the systems development problems mentioned in section 2 can be solved or at least reduced by applying the mentioned standards. However, some of the problems can be solved more effectively by applying the presented harmonized view on the standards. This is especially true for those problems which address the topics knowledge management, risk management, communication and systems thinking.

Using the classification of problems provided in Fig. 1, it can be stated that the use of the HoSE view contributes to solve problems in all the problem areas homogenously, thus reinforcing its holistic character.

In the case of *Human-related* problems, the *Bad customer* and *Erratic communication* problems are solved. On one side, in the bad customer case, a holistic approach based on standardized processes generates standardized documentation. One of those documents is the stakeholder requirements document which must be approved by all the stakeholders. Using this document, later discussions about uncovered topics or not fulfilled objectives can be rejected. The *Systems-related* problem of *Scope arguments with customer* is solved in the same way. On the other side, the problem of erratic communications is solved more effectively in the case that the project manager and the systems engineer are different people. Following the holistic view presented, the project manager and the systems engineer follow the same processes now, e.g. in the field of requirements definition, which avoid any misunderstandings.

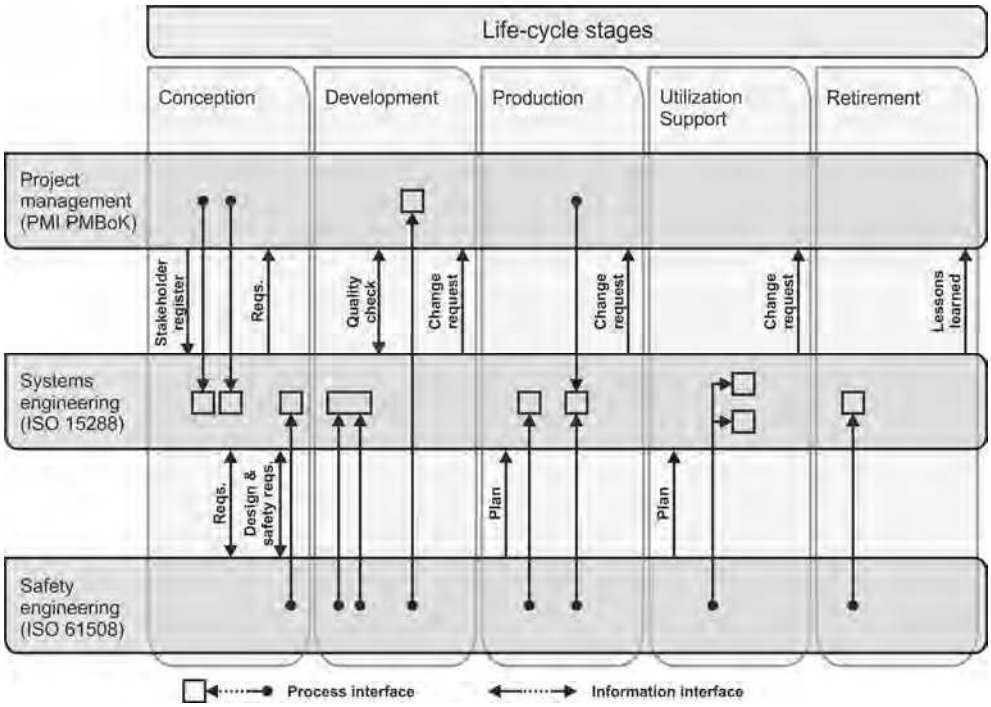


Fig. 9. General overview of the holistic Systems Engineering view

In the case of remaining *Systems-related* problems, *Insufficient funding* and *Insufficient schedule* problems are solved. All the different standards generate and store information during the whole life-cycle of previous projects. A holistic view condensates information from many different sources, thus providing an extremely valuable information source for the planning of further projects. This cumulated information supports an accurate and realistic calculation of resources during project planning.

In the case of *Software-related* problems, problems associated with risk management, performance and quality management like, *Cannot evaluate and mitigate software risks*, *Do not know how to deal with software warranties* and *Cannot satisfy a critical customer requirement to software performance* respectively, are solved due to the advantages provided by the HoSE view. In this case, the cross-discipline of safety engineering provides means for assessing risks, assessing the proper operation of the system and guaranteeing the satisfaction of critical requirements, which are all not present in a non-holistic approach. The same argument is applicable to the *Management-related* problem of *Quality of services and products inadequate*.

Finally, inside the *Management-related* problems, the HoSE view contributes to achieve one of the most important disciplines of a learning organization as stated by Senge in (Senge, 1994), Systems Thinking.

5. Conclusions

Increasing complexity of contemporary technical systems has led to several problems, inefficiencies and safety threats during their whole life-cycle. The system thinking philosophy, initiated as a consequence of the common need for a better understanding of multidisciplinary dependencies, surfaced the need of a holistic approach for the development of complex systems.

Standardized processes support the management of complexity in a critical way. Additionally, they improve risk mitigation, productivity and quality, and they serve as a basis for generating and managing the knowledge of a company.

Two different disciplines are considered to be essential in the development of modern complex systems: systems engineering and project management. In a reality were more and more responsibilities are being delegated to technical systems, the safety engineering discipline has become substantial also. For each of the three cross-disciplines, one internationally accepted standard has been chosen. ISO 15288 has been widely recognized as means for managing complexity and coping with uncertainties. The PMI PMBOK standard is comprised of detailed project management processes and activities and has gained the biggest support in the industry world-wide. Finally, ISO 61508 is a basic industrial standard which sets out a generic approach for developing safety-critical functions. This standard has been used as a reference for domain-specific safety standards.

Despite of the existing interdependencies regarding systems engineering, all three cross-disciplines have developed their corresponding standards with minimal consideration in form of referencing each other. This leads to a situation in which the standards overlap with each other in many processes and activities, and in the worst case, they even could contain conflicting directives. Additionally, some deficiencies like missing sequence diagrams or a clear description of inputs and outputs of the associated activities have been identified.

A unique kind of representation has been conceived in order to enable the comparison of the different standards. The processes belonging to different cross-disciplines have been arranged together in a matrix form, representing life-cycle stages and knowledge areas. Processes being assigned to the same stage and knowledge area were identified as possible candidates for being harmonized. Interacting processes and activities were either merged

together or their information flows were adapted into a holistic view. The resulting view, called HoSE view, has been illustrated using the standardized *Business Process Model and Notation (BPMN)*.

The results of the work carried out disclose that several interfaces and synergies do exist between the three standards. The holistic view arisen from this work aims to provide a good basis for further harmonization and consolidation within standardisation activities. Furthermore, it also makes a contribution to enhance the systems engineering approach by further improving its capabilities regarding productivity, quality and risk mitigation.

6. References

Eisner, H. (2005). *Managing Complex Systems: Thinking Outside the Box*, John Wiley & Sons, Inc., ISBN 978-0-471-69006-14, Hoboken, USA.

Gibson, J. E., Scherer W. T., & Gibson, W. F. (2007). *How to Do Systems Analysis*, John Wiley & Sons, Inc., ISBN 978-0-470-00765-5, Hoboken, USA.

Haskins, C. (Ed.). (2010). *Systems Engineering Handbook: A Guide for System Life-Cycle Processes and Activities v. 3.2*, International Council on Systems Engineering (INCOSE), San Diego, USA.

*ISO/IEC 51, Safety Aspects: Guidelines for their Inclusion in Standards (*1999*),* International Organization for Standardization *(ISO),* Geneva, Switzerland.

ISO/IEC 15288:2002, Systems and Software Engineering: System Life Cycle Processes (2002), International Organization for Standardization (ISO), Geneva, Switzerland.

ISO/IEC 15288:2008 Systems and Software Engineering: System Life Cycle Processes (2008), International Organization for Standardization (ISO), ISBN 0-7381-5666-3, Geneva, Switzerland.

ISO/IEC TR 24748-1, Systems and Software Engineering: Life cycle management -- Part 1: Guide for life cycle management (2010), International Organization for Standardization (ISO), ISBN 978-0-7381-6603-2, Geneva, Switzerland.

ISO/IEC 61508:2010, Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems (2010), International Organization for Standardization (ISO), ISBN 978-2-88910-524-3, Geneva, Switzerland.

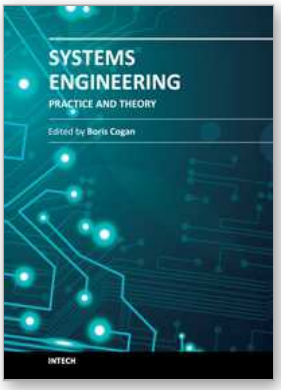
Jackson, S. (2010). *Architecting Resilient Systems: Accident Avoidance and Survival and Recovery from Disruptions*, John Wiley & Sons, Inc. ISBN 978-0-470-40503-1, Hoboken, USA.

Object Management Group (2011). *Business Process Model and Notation (BPMN) v. 2.0*, Needham, USA.

Project Management Institute, Inc. (2008). *A Guide to the Project Management Body of Knowledge (PMBoK Guide), 4th ed.*, ISBN 978-1-933890-51-7, Newtown Square, USA.

Sage, A. P. & Armstrong, J. E. Jr. (2000). Introduction to Systems Engineering, John Wiley & Sons, Inc., ISBN 0-471-02766-9, Hoboken, USA.

Senge, P. M. (1994). The Fifth Discipline: The Art & Practice of the Learning Organization, Doubleday Business, ISBN 0-385-26095-4, New York, USA.



Systems Engineering - Practice and Theory

Edited by Prof. Boris Cogan

ISBN 978-953-51-0322-6
Hard cover, 354 pages
Publisher InTech
Published online 16, March, 2012
Published in print edition March, 2012

The book "Systems Engineering: Practice and Theory" is a collection of articles written by developers and researches from all around the globe. Mostly they present methodologies for separate Systems Engineering processes; others consider issues of adjacent knowledge areas and sub-areas that significantly contribute to systems development, operation, and maintenance. Case studies include aircraft, spacecrafts, and space systems development, post-analysis of data collected during operation of large systems etc. Important issues related to "bottlenecks" of Systems Engineering, such as complexity, reliability, and safety of different kinds of systems, creation, operation and maintenance of services, system-human communication, and management tasks done during system projects are addressed in the collection. This book is for people who are interested in the modern state of the Systems Engineering knowledge area and for systems engineers involved in different activities of the area. Some articles may be a valuable source for university lecturers and students; most of case studies can be directly used in Systems Engineering courses as illustrative materials.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821