

Chapter 2

A Systems Engineering Framework for Reliability Assurance of Subsea Oil and Gas Production Systems



Sirous F. Yasseri

Abstract Capital projects, such as deepwater offshore oil and gas production systems (SPS), require a large investment, thus high availability to recover the investment is vital. The costs of intervention (recovery of failed equipment, repair, and replacement) and the loss of revenue will add to the problem. Thus, the reliability of the production system must be assured by reliability analyses and testing. A Systems Engineering (SE) approach is described in this chapter that links the client's reliability needs to the system's performance, hence permitting the specification of appropriate strategies and procedures for verification and validation while accounting for all constraints, including the costs of maintenance, possible intervention requirements, and downtime, and relates these to the equipment performance that is needed to achieve the desired availability of the SPS. In addition, the possibility of constructing the field in stages and expanding it as needs arise must also be considered. It is shown how to relate the equipment performance to the Client's requirements. The procedure described in this chapter can also assist with project risk management by blending the reliability analyses, testing, and various risk analysis methods for the system verification and validation procedures. The system engineering V-model is augmented by reliability assurance requirements to assure sustained operation by ensuring the robustness and resilience of the production system.

Keywords Subsea production system (SPS) · Capital projects · Reliability assurance. Systems engineering · Verification · Validation & testing · Qualification & certification · Failure mode effect critically analysis · Hazard identification · Hazard of operation · Mean time to failure · Mean time to repair · Mean time between failures

S. F. Yasseri (✉)
Brunel University London, London, UK
e-mail: Sirous.Yasseri@Brunel.ac.uk

Abbreviation

CAPEX	Capital Expenditure
ConOps	Concept of Operations
CR	Client Requirement
EFAT	Extended Factory Acceptance Test
FE	Finite Element
FAT	Factory Acceptance Test
FFP	Fit-For-Purpose
FFS	Fit-For-Service
FMECA	Failure Mode Effect Critically Analysis
HAZID	Hazard Identification
HAZOP	Hazard of Operation
MTTF	Mean Time to Failure
MTTR	Mean Time to Repair
MTBF	Mean Time Between Failures
OPEX	Operation expenditure
RAM	Reliability, Availability, and Maintainability
SAT	Site Acceptance Test
ROV	Remotely Operated Vehicle
SPS	Subsea Production System
TRL	Technology Readiness Level
V&V	Verification & validation

2.1 Introduction

The petroleum industry requires a detailed comprehensive framework for delivering high reliability and availability systems. “*Reliability is taken as the probability that a system will operate satisfactorily under specific operating conditions for a given time. System maintainability is defined as the ability of a system to be operable without failure for a given duration in the future, and the system can be restored easily if a breakdown occurs*” (Rausand & Høyland, 2004). A system is considered ‘not available’ if it is shut down for unplanned or planned maintenance or component failures since the outcome is the same. “*The reliability analysis is used to judge a system’s maintainability. Reliability, in turn, is dependent on the system architecture, material selection as well as design details; and it is only achievable if the availability is at its highest level*” (MIL-HDBK-217 Rev. F, 1995).

“*Performing System reliability, availability, and maintainability (RAM) analyses early in the phases of a project development provides a metric for comparison of alternative architectural concepts*” (Yasseri & Bahai, 2018). At the concept generation time (Yasseri, 2012), several functional architectures are considered, it is useful at this

phase to model the functional components, as rough building blocks, without reference to their physical properties for an early estimate of the reliability using historical data. Functional architecture is an idealized abstraction of a system, which identifies functional elements without a precise description of their physical properties and their implementation.

The client decides on the desired target reliability level of the system, which is used by the designers for the allocation of reliability requirements for every piece of equipment (it may also include software) as a target. Then the system engineer's objective is to demonstrate "*by examination and provision of evidence that the hardware (as well as software) meets the Client's specified requirements for the intended use*" (DNVGL-RP-A203, 2019). For novel hardware, the failure data is likely to be non-existent or insufficient, thus tests may be essential to enhance confidence in the reliability of results.

Reliability can also be affected by an ill-defined specification or mismatch between specification and design. Bad manufacturing processes, unsuitable materials, poor installation, inadequate or irrelevant tests, and incorrect use of the system will also influence the system's performance. These lead to the estimated performance, which is demonstrated by analyses, to be different from the actual performance of the as-built system. Other explanations are "*emergent behavior, undetected faults, unanticipated operating conditions, unanticipated failure mechanisms & their causes, epistemic and aleatory uncertainties*" (Pecht, 1993). Unforeseen and unexpected operational conditions are because of insufficient or incorrect specifications, user errors, or as a result of incorrect implementation changes due to inadequate change control management and lack of oversight. A scenario-based approach and what-if analyses can help to minimize the impact of any uncertainties. The "*results of functional failure analyses and testing, are complemented by field experience obtained from observation of proven technologies as well as physics-based analyses*" (Viola et al., 2012).

A reliability analyst who uses only generic historical data, (e.g., OREDA, 2009), to determine the probability of mechanical failures cannot account for the impact of design errors and poor manufacturing on reliability. It may be incorrectly assumed that all errors will be detected and rectified during the development of the system. Thus, "*the reliability predictions based only on historical data is not highly dependable, and hence must be augmented by other types of analyses and tests*" (Feiler et al., 2012). It is not realistic to assume that modern fabrication methods and material qualities are the same as they were in the past.

The equipment reliability may even change from project to project. Components that are designed to perform a specific functionality by different manufacturers could have different failure modes and routes to failure. V&V and testing must be used to fill the knowledge gap.

A more dependable framework is needed for validating and qualifying a system, economically and quickly, rather than "test and test again until time and budgets are exhausted". The objective is to outline a methodology for the detection of all types of errors early in the Development Phase and to "*furnish the system with good quality attributes, such as high performance, safety, sufficient reliability, resilience,*

robustness, and defensible (adequate installation security) (Yasseri & Bahai, 2018). It is prudent to build resilience into the system at the design stage to counter unforeseen, undetected, and emergent behavior. It is also crucial to assure that unavoidable, undetected, and unanticipated failure modes are managed by a well-organized and robust risk management plan during the operational phase. This framework aims to identify failure modes at the architectural level, the approach is also can deal with issues that are not easy, or possible, to test unless the whole system has been installed.

A framework for achieving a reliable SPS is described in this chapter. A parallel V is proposed which shadows the SE’s V-model (see Fig. 2.18). This ties the reliability assurance to the system development process efforts and minimizes downtimes by embedding robustness and resilience into the system. The framework enables the delivery of reliable systems while respecting all constraints and requirements. The subsea battery limit in this chapter is from the down-hole valve to the seabed production equipment, to the topside equipment, (and possibly, to an onshore receiving terminal), in their operational environment employing the notion of “Fit-For-purpose”.

2.2 System Thinking in SE

A system is an assembly of components and linkages, and linkages allow the system’s components to interact with each other (Fig. 2.1). How components of a system are arranged, interact, and influence each other determines the property of that system. A collection of components, without linkages and relationships, does not make a system.

Accordingly, a system is a set of objects ate are organized in a specific way, with a certain relationship between the objects that work together in some manner to perform a function (the purpose). Systems can accomplish tasks that would be impossible if the same elements were put together in random order, or if there is

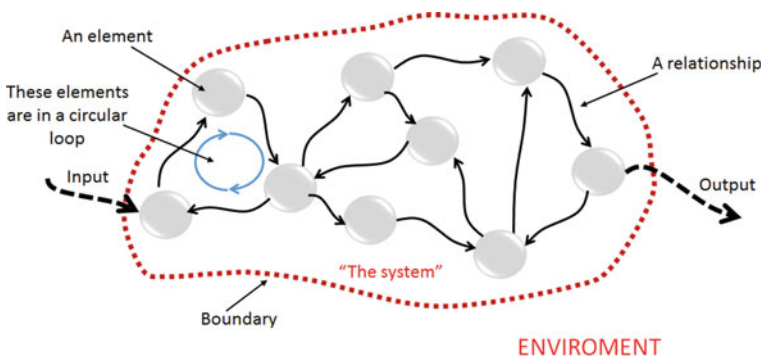


Fig. 2.1 A system consists of three elements: components, linkages, and relationships

no logical relationship between them. Humanity benefits continually from various clever ways of putting together the resources that provide us with food, transportation, education, goods, and services.

The characteristics of a system are (Fig. 2.1):

- **Purpose:** A system can only be visioned when it has a clear purpose and provides a desired function. This purpose usually governs the arrangement of elements, their connectivities, as well as the strength of their relationship and the interactions between the system and its environment.
- **Boundaries:** The boundary determines the extent of influence of a system. The boundary stops where the impact of the environment on the system becomes marginal, and vice versa. Judgments as to where the boundary lies, are necessary constituents of the Systems Thinking.
- **Coherence:** (A sense of belonging). Every interaction within a system must be coherent.
- **Emergence:** A characteristic of systems is that they cannot be identified solely by their parts. This wholeness causes behaviors to emerge that are known as emergent characteristics.
- **Hierarchy:** Any system should consist of at least three levels of hierarchy; System of systems (SoS), systems, and sub-systems, which determine how changes at one level can influence other levels.
- **Sub-systems:** These are the parts of the system that must interact to achieve a balance to the purpose of the system. A sub-system or a component is a system the vendor.
- **Environment:** All things not included in the system that may affect its purpose. Some aspects of a system's environment may be closely associated with the system, while other aspects are less relevant or unrelated.

The “systems thinking focus is on relationships between the system's elements, (not on the elements as unrelated objects), objectives (not the structure), the whole (not its constituent parts), the context (rather than the contents) of a system, and patterns” (Royal Academy of Engineering, 2014). Engineers for a long time have taken any complex system (like a transportation system), separated it into its parts, and then tried to manage each part as best as they can. Parts could in the context of transportation refer to different means of transport (road, rail, air, etc.), hardware, or people. If that was done, engineers believed that the system would behave well. “Thinking in systems requires shifts in perception, which lead to diverse ways to perceive, and different ways of organizing a system” (Edson, 2008). “It is possible to improve the performance of many system's components and yet disable or destroy the system in its entirety” (Senge, 1990).

2.3 System Architecture

“*System Architecture is an abstraction of the vision of how a system should hang together, which is an arrangement of its components and their relationships to each other and the environment*” (Sillitto, 2014). The system architecture is used as a plan (blueprint) for the definition of subsystems and components, their design, manufacture, and integration with the system’s operational environment so that the elements of the installed system will work in unison to deliver the intended functionality.

A system architecture is presented at two levels of abstraction hierarchy, which are known as functional and physical. The first level is the functional architecture, which is also known as the “conceptual design”, it is still an abstract view of the system but may have more details (Yasseri & Bahai, 2018). In software engineering, another layer is added between these two and call it the logical architecture. In this chapter functional and logical architectures are used interchangeably.

The functional architecture is a representation of the system independent of suppliers, and equipment is named by its functions. Each piece of equipment is represented by a box and identified by its function. At this early stage of development, what a component must deliver is known but its physical properties (dimensions, sizes, footprint, material, weight, and so forth) are not known until more definitions are added by identifying suppliers and deciding which equipment to procure. This takes place in the next phase of the project development.

Equipment manufactured by two different suppliers delivers the same function, but their physical properties are quite different. Two pieces of equipment designed and manufactured by two suppliers will share many common functional characteristics, but they will have many different physical characteristics. A component in the functional architecture represents its function (what it does), but some properties, and interfaces may be similar to a range of products supplied by different vendors. The functional architecture remains static and independent of technologies and vendors and will provide a relatively stable baseline to proceed to the system design, vendor selection, and fabrications.

The lefthand side of Fig. 2.2 shows, a deepwater subsea system to deliver certain functionality (extract gas from six wells and send it to shore for preparing it for sale), consisting of several sub-functions. At this stage of hierarchy, the function of all equipment, their relationship & connectivity, and how they should communicate are defined, which are mostly diagrammatic and descriptive. This is to make sure that all required components are present and logic for their inclusion is well understood.

Several concepts are developed, prioritized, and the front runner is taken forward for greater definition. The physical architecture gradually evolves to the middle section of Fig. 2.2. The middle section of the drawing in Fig. 2.2 is similar to the lefthand side, but with more information, and “*there is a one-to-one relationship between the functional components and their physical realization.*” (Yasseri, 2014a). All major components of the physical architecture (middle part of Fig. 2.2) are defined by their physical properties, suppliers, position in the system, and relationship and communication between them. It must include all known data such as the concept of

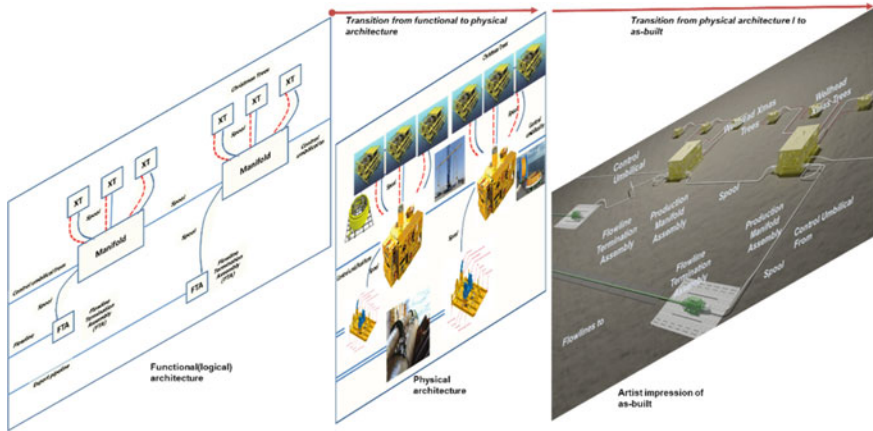


Fig. 2.2 A typical subsea system and its functional and physical architecture

operation (How the system should operate), system configuration, supplier’s operating instructions, materials, and means of communication (flow of fluid, signals, and energy) & control. “All physical constraints or limitations are also identified, e.g., physical solution for interfaces, fluid flow requirements, size (geometric compatibility), footprint, weight, and installation barges & cranes requirements are also decided” (Yasseri, 2015b).

“The functional architecture is a plan that enables each function of a system to be allocated to a physical component (Fig. 2.2). The functional design will remain almost unchanged, but the physical design will change throughout the lifecycle” (Yasseri & Bahai, 2018). The choice of physical components is governed by the available suppliers and needs to improve or modify the installation during its life-cycle, and hence the physical system will be changed to suit the new conditions. Physical architecture would also change with the introduction of new capabilities (e.g., debottlenecking or expansion), new technologies (e.g., new control systems), hardware innovations, software upgrades, the necessity of replacing obsolete equipment (e.g., no spare is available), or acquiring a piece of equipment from a different vendor.

Figures 2.3 and 2.4 show examples of the functional architecture of two types of deepwater fields.

2.4 Phase-Gated-Incremental Commitment

The development of a project is a sequential process that takes several years from its inception to its completion. The time from the inception to decommissioning is known as the life cycle. Life cycle models vary according to the project’s nature, purpose, use, and the procedures of the Client’s organization. There are many forms

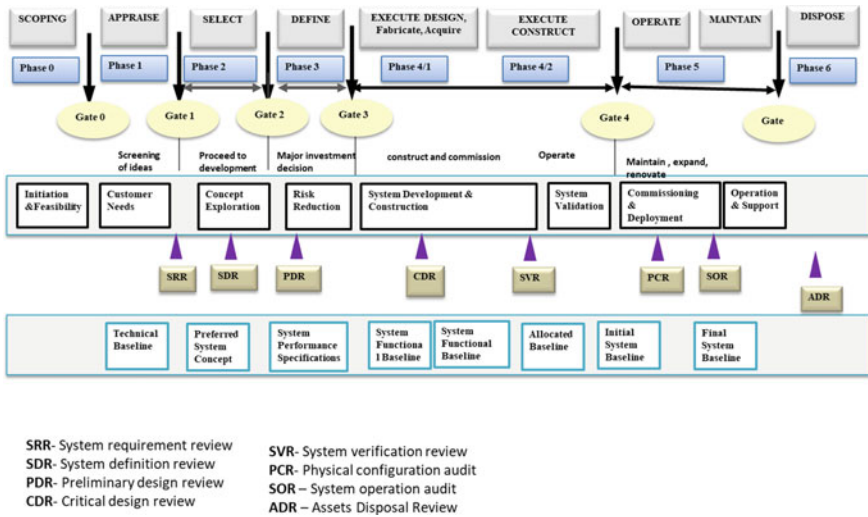


Fig. 2.5 A typical life cycle model with phase-gate and review milestones

of life cycle models, however, they all share a similar set of phases. The development life is divided into several phases (Fig. 2.5), and sometimes each phase is broken down into several stages—(some authors swap around phases and stages). “Each phase has a distinct and definite purpose and position in the life cycle and represents an identifiable period in the life cycle of a system” (ISO/IEC 15288, 2008). These phases also mark major milestones in the development process.

The subsea project life cycle begins with exploration and scoping (initiation)—(Phase 0)—the concession to explore by the government is excluded here. Phase 1 (the Appraisal Phase) focuses on identifying the Client’s needs and objectives, exploring diverse ways of extracting the hydrocarbon, and transporting it to the shore. The focus of Phase 2 (the Select Phase) is identifying, refining, and verifying the system requirements, generating a few concepts with enough detail for decision-making, choosing a front-runner, and taking it to Phase 3 (the Define Phase) for more definition. Afterwards, the project progresses to Phase 4 (the Execution Phase), Phase 5 (the Operations Phase), and finally to Phase 6 (decommissioning or retiring). A life cycle model shows how early choices would impact what can be done further along a project’s life cycle, thus enabling sensible trade-offs, and can beneficially influencing its viability.

Commitment to the capital investment in any large complex project is incremental. The lifecycle approach enables one to commit to the project incrementally (*incremental commitment*). This is achieved by inserting gates between phases. The gates are milestones and key decision-making points of the project development. The gate enables sponsors to review progress, decide on the commercial and technical viability of the project, and whether it is logical to proceed to the next stage by committing

more funds. Phase-gated processes (Fig. 2.5) allow the timely accrual of required information for the decision-makers.

Each phase is designed to collect specific information or meet specific goals (Table 2.1). There is a major *review gate* at the end of each phase, where the Client's team gets the opportunity to assess whether the phase objectives are met and decide if and how the project should continue. Reviews are a formal means that allow project sponsors to control risks (commercial and technical) and monitor changes in the project scope. Based on the deliverables and decision criteria for the phase, sponsors can also validate the business case. This is an external review by the executive sponsor, stakeholders, and others who were not involved with the design.

At each gate, Project managers and sponsors should review the following:

- Identify and manage risk in each phase.
- Whether the phase met its objectives.
- Approve any changes in scope or schedule since the last gate review.
- Abandon the project, or proceed (with or without modification in scope).

The criteria for successful gating are:

- Gates must control decisions, not activities. Deliverables, decision criteria, and decision-makers must be clearly defined.
- Division of the project into a suitable number of phases that are structured, scalable, simple, and adaptable.
- The gate must be for transitioning a project to the next phase and must logically be a milestone in the development process.

2.5 Fitness-For-Service

The notion of Fitness-For-Service (FFS) and Fit-For-Purpose (FFP) is promoted to assess a system's or a product's suitability for service—i.e., it does the job. Two phrases of FFS and FFP are used interchangeably in this chapter. These phrases are used to mean that a system, based on rational reasoning, is suitable for a specified purpose. The proof can be qualitative as well as quantitative. If a system is poorly assembled, sustained some damage, or is not suitable for its intended purpose, then it is considered as not 'Fit-For-Purpose'. That is the system cannot reliably deliver what is expected of it; either it fails frequently, or it doesn't function as it is supposed to. For example, if a component is '*bolted on*'—(added as an afterthought)—to a system to enhance the system somehow (e.g., to become more reliable or safer), but the add-on component does almost nothing, then that component is not FFP, meaning quality, is decided based on FFP (or FFS). This means that quality is not a system's intrinsic property; but is assessed in the context of what it must deliver (namely if it provides the required functionality). For instance, hardware cannot be judged to be of a 'high quality' product, because the quality is an attribute of the relationships among the system's components and the purpose for which a component is inserted into a system. But a product can be judged as 'low quality' because it can be shown that it

Table 2.1 Activities in the design phases of a large capital project**Appraise:**

- Feasibility and economics Studies
- Alignment with business strategy
- What are the project drivers?
- Is there a viable opportunity to pursue?
- Engage with regulators

Select:

- Develop a few concepts (options) and prioritize
- Develop an initial cost estimate and schedule for each option
- Compare options focusing on risk, economics, and uncertainties (trade-offs)
- Recommend a preferred option. Provide improved cost estimate, schedule, and production forecast

Define:

- Develop the selected option with enough details and planning requirements to enable freezing the scope
- Alignment with business strategy
- What are the project drivers?
- Refine, costs, schedule, and production estimates
- Vendor assessment & Selection
- Contacts for long lead items
- Inform Regulator

Execute:

- Detail engineering
- Procurement
- Testing (Factory, quayside assembly, installed assemblies and Integration)
- Site support
- Project management and System engineering
- Accounting
- Document control & management
- Submit the 'Formal Safety Case' to obtain permit
- Sparing policy

Operate:

- Evaluate installation against requirements
- Revise RAM studies
- Personal training
- Support the hand over team
- Check the system's performance
- Plan sparing
- Intervention policy

is not good for any use. The purpose of a system (its function), and the quality that is needed for delivering the stated function, must be well-defined such that to enable one to make a judgment on the system's quality. It is difficult to imagine a purpose for a subsea control system that cannot perform emergency shutdown in an orderly manner and as safely as reasonably practicable within an acceptable time. Some systems may have many functionalities (purposes), and over time, some original functionalities may not be needed anymore. The systems' envisioned operating conditions (normal

and abnormal) and the system's intended purpose are coupled and must be completely identified and well-defined, leaving no room for interpretation.

With the notion of "Fitness-For-Purpose", one can judge, as well as question, the completeness, and relevance of the defined purposes to ensure compliance and possible enhancement. FFP equates quality with the fulfillment of a specification or stated objectives. The intention is to validate a system for its intended use, nothing less or more. There may be multiple purposes such as safety, reliability, availability, or some specific quality requirements (possibly all), which are requested by the client. Thus, FFP is a practical concept for assuring a system meets quality, measured against the client's requirement. This definition gives the impression that '*value for money*' is a synonym for Fitness-for-Purpose, however, neither affordability nor cost-effectiveness criteria are necessary elements of FFP.

The notion of Fitness-For-Purpose has emerged as a guide to direct efforts toward an installation with zero faults (i.e., no downtime). The ultimate measure of perfection is 'zero faults', which is an excellent goal but achieving it is impractical. Perfection is meaningless if a system does not deliver the required service.

Proving a system is 'Fitness-For-Service' requires gathering many kinds of evidence, which involves collecting data while the system development is in progress. Such evidence includes reviews, V&V of requirements and design, using the analytical methods, simulation, and particularly test results to support justifiable confidence in the as-built system.

Generally, a 'claim' is made that a piece of equipment is FFS, then the claim is qualified by assembling relevant evidence (Yasseri, 2015a) that supports "the equipment would function within defined limits and with a sufficient confidence level" (Woody et al., 2014). Such confidence is assured by prototyping, simulation, physics-based analyses (analytical and numerical), reliability analysis methods (FMECA, RAM, etc. (see e.g., IAEA, 2001), risk assessments, visual inspection, and of course testing.

2.6 State of Practice

Classification societies have published recommended practices and guidance notes on the qualification of subsea production systems. For example, API-RP-17N (2023) and API-RP-17Q (2023), DNVGL-RP-A203 (2019), Bureau Veritas-NI525 (2020), ABS (2017), and Lloyds Register (2017). Figure 2.6 shows the DNVGL-RP-A203 (2019) procedure. API recommendations are similar with some variations (Fig. 2.7).

The primary target of these codes of practice and guidance notes is "New Technology", but their definition of new technology is quite wide and includes almost everything if the site is greenfield and even includes some brownfield sites. The term "Technology" in these codes refers to a piece of "*equipment that uses a physical law to satisfy a purpose*". They recommend that both the underlying physics and equipment be qualified. For example, if existing topside equipment is modified for the subsea application (i.e., marinized), it must be qualified. Generally, if no new

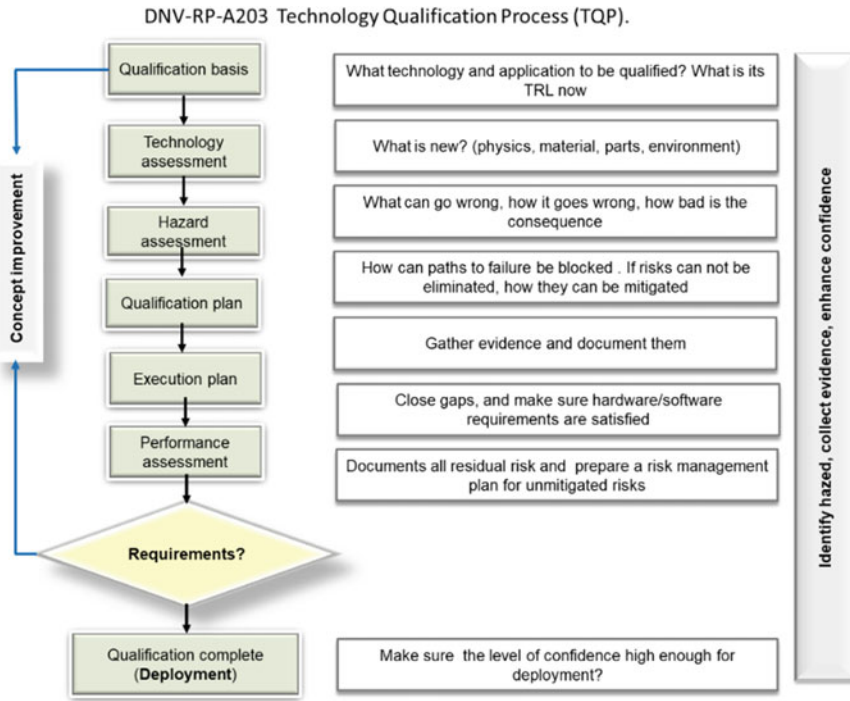


Fig. 2.6 DNVGL-RP-A203 (2019), Technology qualification process (TQP)

physics is involved, then it is only required that the equipment be qualified for its new working environment (effects of corrosion, marine fouling, etc.).

These codes, require reliability analyses to be conducted in Phase 1 for the entire system, with a level of detail that is commensurate with the definition of the system at that phase. The stated purposes to do so are:

- Identify possible design weaknesses.
- Compare and contrast alternative designs, architecture, equipment, materials, etc.
- Estimate costs at each phase of the lifecycle, with sufficient accuracy necessary for the decision-making.
- Perform availability assessments and check if the architecture would meet the client’s target.
- Define requirements, procedures, tooling, and required results for performing reliability testing.
- Specify sparing requirements and sparing policy.
- Probable intervention needs, its practicality, and intervention tooling & methods.

More than one reliability analysis method must be used to tease out all failure modes. Another tool to be used together with reliability methods to mitigate technical risks is the Technology Readiness Level (TRL).

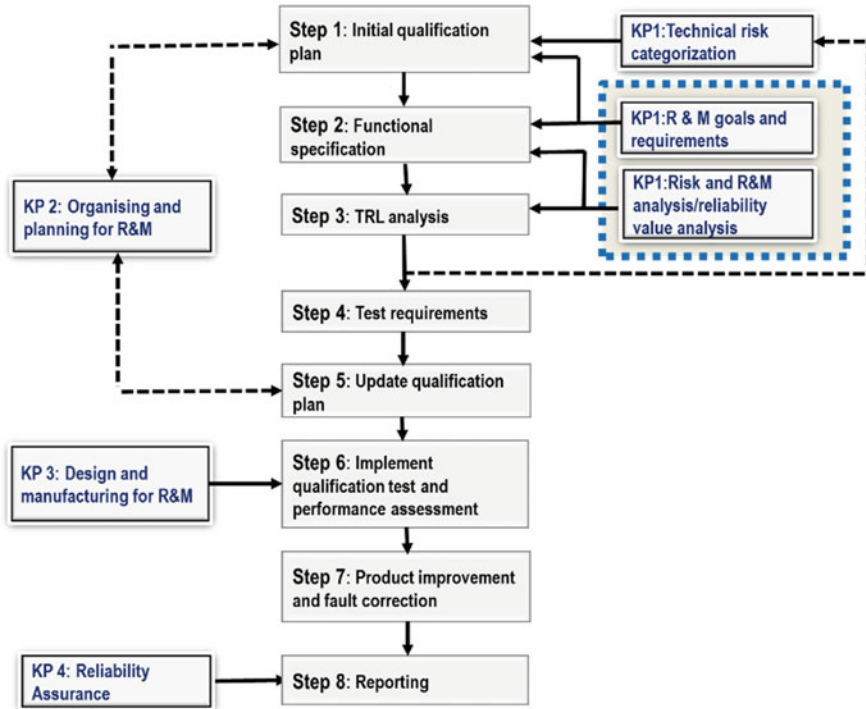


Fig. 2.7 API-RP-17N (2023) Technology qualification process flow chart

DNVGL-RP-A203-2019 and API-RP-17N-2023 recommendations define “technology as ‘new’ when it is not used (i.e., its suitability is not proven) in a similar field under similar conditions”. Thus, if Commercial Off-The-Shelf technology (COTS) is used in an environment that was not used before, it must be considered new, but not unknown. This implies that the TRL of every subsea equipment for a new field at best, at TRL = 5, (Yasseri, 2013).

2.7 Systems Engineering V-Model

Systems Engineering (SE) provides processes for developing a system that can satisfy the client’s requirements and needs against the background of conflicting constraints. “SE is an all-encompassing integrative activity, which encourages and coordinates the collaboration of several disciplines, to deliver a coherent operable system that is not dominated by the perspective of any single discipline” (NASA, 2007). INCOSE (2015) gives this definition: “SE is an iterative process of top-down sequential synthesis and development to produce a system that meets, (in a near-optimal manner), the full range of the client’s requirements”.

SE does not deal with the physics of the problem but provides processes, which can be employed to meet both the client's business needs and the technical requirements in engineering the system. System Engineering processes have been successfully applied for many purposes such as (NASA, 2007):

- **Definition of systems of systems**—identification of system(s) that satisfies the client's needs.
- **Development of system requirements**—development of conceptual architecture, concepts trade-offs, configuration management during development, and system integration.
- **Validation, verification**—operability evaluation and acceptance tests, sparing policy, and planning for maintaining the system over the whole lifecycle, including interventions, expansion, and refurbishment.

The process starts left-hand side of the SE's V-Model (Fig. 2.8) with the definition of the ConOps and the client's operational needs. Namely how the system is supposed to operate and function. Then the system is deconstructed (decomposed) into functional components or subsystems and components, for the ease of managing its development. The aim of breaking down a system into its constituent components is to create a logical chain by linking the operational needs to system requirements, to the specification of subsystems, then to the specifications for their integration, and then to acceptance testing. Moving along the left-hand side of the V one can partition the system hierarchy into functional, and physical collections of components which can be designed by the discipline experts and tracked to the logical conclusion. The use of conceptual models early in the project development is encouraged by SE processes to gain insights into the technical feasibility of a concept. A better understanding of the client's requirements enhances the chance of succeeding in delivering what the client asked for. The V-model allows concurrent activities.

The horizontal line, in the middle of the "V" in Fig. 2.8, depicts the handover of the design activities to the specialized disciplines, or engineers, who specialize in specific engineering disciplines, to produce the physical system. The position of this dividing line determines the overlap between discipline engineers and the systems engineer engaged in the integration processes. The horizontal line as drawn shows a modest overlap; a total separation is not implied. Interface management and some integration and qualification activities take place during the design.

The right-hand side of the V-model depicts the integration, Verification & Validation, and qualification activities. Integration involves the assembly of parts into components, the assembly of components into subsystems, the assembly of sub-subsystems into higher-level subsystems, and the assembly of subsystems into the final system. These parts, components, and assemblies must be qualified which could involve testing of the newly assembled sub-subsystems to check their compliance with the requirements; this process is known as verification (Grady, 2007). After verifying the system against the system requirements, the system must be validated.

The V diagram graphically shows how the design activities flowed down from system requirements to functional design, and finally to the physical design in an iterative loop of interrelated activities. Several factors, such as technology selection

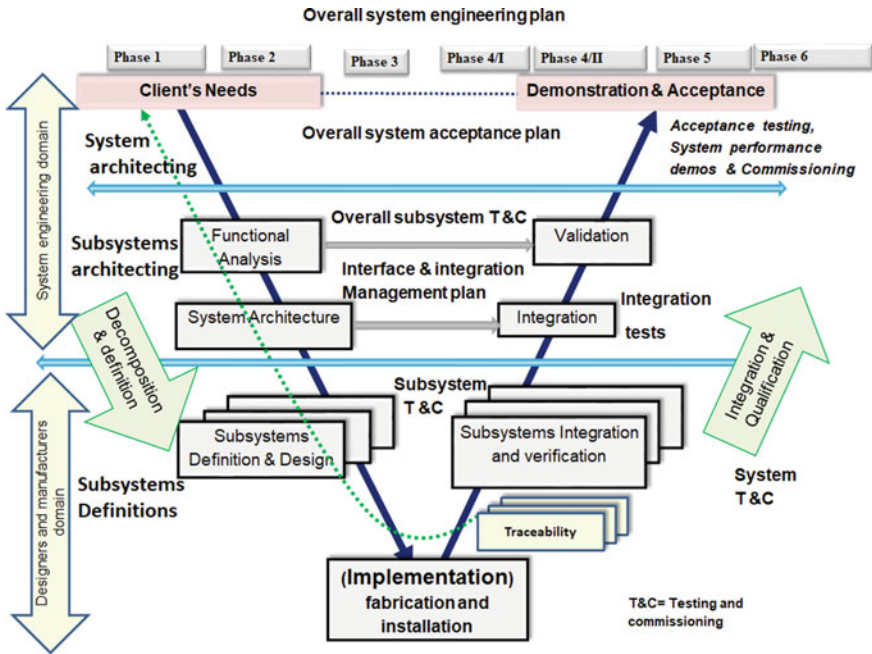


Fig. 2.8 Systems engineering V-model

(Yasseri, 2012), degree of standardization, hardware interface requirements, as well as the choice of concept would influence the nature and the level of iteration and possibility of concurrent engineering (Yasseri & Bahai, 2019).

2.8 Primary Loops of Development Process

The V-model requires that a system be decomposed into functional subsystems, which can be designed with fewer complications. This allows subsystems to be designed and fabricated in parallel (concurrent engineering) according to verified and validated system specifications developed in the previous phase. The SE processes allow the concurrent development of subsystems of a large system which accelerates the project development by involving many disciplines concurrently and encourages the engagement of vendors.

The V-model breaks down system definitions into three separate loops (Fig. 2.9). These three main loops, (Fig. 2.8), are the three main loops in the system development. The first loop is the design loop which deals with components, assemblies, and subsystems. If the system is a modular type, then the subsystem design and verification can be done in parallel. The installation '(implementation) takes place

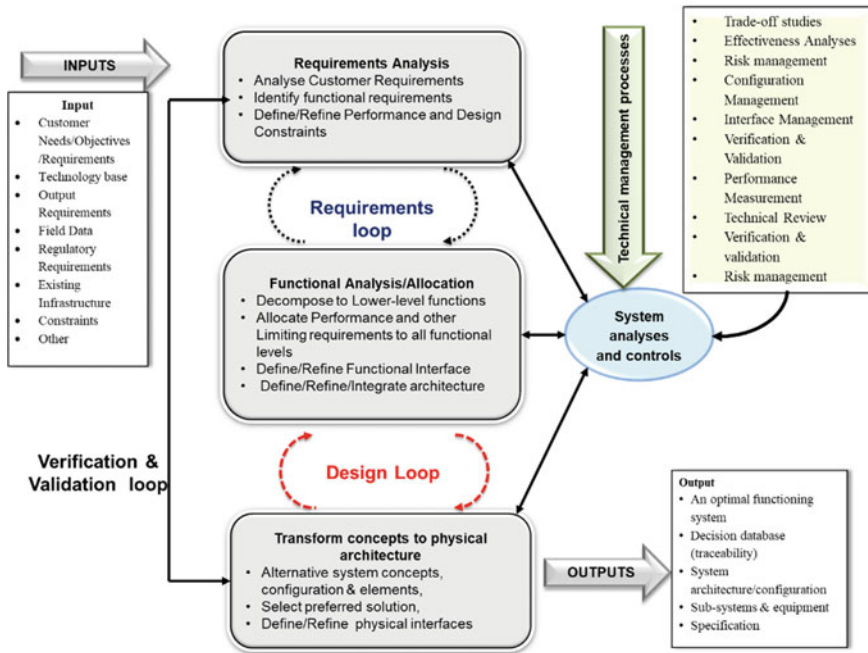


Fig. 2.9 Three loops of the development process-Left-hand side of the V-model (adapted from NASA, 2007)

at the bottom of the V, which is a collection of many ‘parallel Vs’, which are equal to the number of subsystems that are inserted into the system.

“The Systems Engineering Processes are iterative, which is applied top-down sequentially by system engineers to decompose a complex system into manageable parts, for which an expert can be found. The client’s goals, requirements, and needs are described in a set of top-level system requirements that are input for the next level of decomposition” (NASA, 2007). A complex system is decomposed sequentially to several levels. At each level, more definitions are added, and performance requirements cascaded down. This process leads to nested loops (Fig. 2.9) indicating the repetitive nature of the process. “The loops are the **requirements loop, design loop, verification loop, and control loop**. It also includes **input & output definitions**” (NASA, 2007). “These loops link **requirement analysis, functional analysis & functional allocation, and synthesis**” (see NASA, 2007 for more details).

The **Inputs** are the customer’s requirements, objectives, needs, and the list of all constraints. The design process starts with understanding the client’s needs and wants, the system operating environment and the battery limit. Before searching for a concept, it must be determined if the client’s inputs are primary requirements or nice-to-have features (wants). Separating needs from wants allows the system engineer to concentrate on needs as the primary objectives and define a system that satisfies requirements rather than the implementation of the directed by the client’s wants.

Requirements analysis is the elicitation and validation of the client's requirements and needs, which is the basis of the system's functional and performance requirements. The client's requirements are translated into the system requirements, namely what and how the system must function. Development starts by translating the Client's need into a set of agreed requirements, from which the system requirements are established. The system requirements are then flowed down to establish requirements for subsystems and equipment. "*Parts, assemblies, and subsystems are successively qualified against their requirements*" (Bahill & Henderson, 2005).

Functional Analysis & Allocation's purpose is to allocate functions and performance requirements to lower-level subsystems, which defines the system successively to its lowest level. "*High-level system requirements are flowed down for allocating them to subsystems and components. Defining allocated functions in adequate detail provides design specifications and verification criteria to support the development of the entire system*" (INCOSE, 2015). Functional and performance requirements for lower-level subsystems must be tied to higher-level requirements. Functional analysis and allocation activity will ensure consistency of the requirements and may require another iteration of the requirement's analysis. This is the **Requirements Loop**, which is iterative.

Synthesis defines the property of the hardware that makes subsystems, which leads to the complete description of the physical architecture. Every hardware (part) must support at least one of the functional requirements, however, several functions can be delivered by a single part.

Design Loop is the process of inspecting and assuring that the functional architecture leading to the physical system can deliver the desired functionalities at the desired performance levels. The **design loop** allows the revisiting of how the system would function and if it is desirable to optimize the system further.

The verification Loop is for verifying if the solution satisfies the requirements. System requirements at each level of the hierarchy must be verified. During the functional analysis and allocation baseline documents are developed which define how every requirement must be verified. As each component is integrated into the system, it is verified for compliance with all higher-level requirements. Visual inspection, demonstration, simulation, or test are used for the verification. Verification strategies and plans are to support the requirements. Validation is a system-level activity in which the system performance is compared with the requirements.

2.9 Requirement Analysis

The quality of a product only has meaning if it fulfills the service provider's needs. If a piece of equipment performs well the function for which it is inserted into a system, then it is considered as a quality product (i.e., Fit for Service), Thus "*what the system must deliver, its performance and availability*" must be defined (FAA, 2008). "*Requirements are linked to ConOps via traceability matrix*", (INCOSE,

2015) and “*cascaded down into requirements for subsystems, sub-subsystems, and components*” (Hull et al., 2002).

System requirements are defined for two operational conditions:

- The capabilities under normal operational regimes, which specifies the expected behavior, and desired performance.
- Desired expected behavior during upset conditions (abnormal conditions) i.e., the required resilience and survivability (robustness), and how to control the system during an upset condition and return it to normal operation.

The first loop of product development (Fig. 2.9) is about the requirements that define what is required of a system and its purpose. How well a system must fulfill its functions, or how well it must suit its purpose, which is an indication of how good the system is.

Requirements engineering is the systematic effort to collect, verify, specify, agree, validate, and manage the client’s needs and goals while considering the user’s interaction with the system, technical issues, and economic & business concerns. These envelop the whole lifecycle, involving dispersed teams of specialist engineers and several supply chains over many regions for a few years. Thus, requirements that are complete, verified, and stable are important tasks of systems engineers, since all design activities are cascaded from the high-level requirements.

The following three concepts are helpful when dealing with large capital projects:

- **Abstraction:** i.e., seeing the big picture, not details. The functional architecture is an abstraction of the system’s functions without much detail.
- **Decomposition:** i.e., decrypting a system into its subsystems and components, so that they can be studied in isolation by relevant engineers. A system is decomposed along the line of suppliers’ specialization. In decoupling between parts no decomposition is perfect, however, it enables the identification of specialist engineers and competent suppliers.
- **Projection:** i.e., an understanding of how the system should work (a perspective of view of the system) and describing only the pertinent aspects. While constituent (decomposed) components are designed independently, they share a common mission (purpose) as members of one system.

Requirements analysts use these concepts to decide what requirements are necessary and sufficient and how to satisfy them. The system engineer by abstraction, decomposition, and projection reduces a complex problem to its simplest form and investigates if existing solutions or off-the-shelf items can be used. Ideally, the decomposition must be directed toward components (or solutions) that already exist, which can be used albeit with some modifications. However, adopting existing solutions and off-the-shelf items could require substantial work to integrate them into a system in a different context.

“*Each requirement may impact many parts of a system and may need several test cases to verify it*” (INCOSE, 2015). The integration of a system with its environment and user interaction are also requirements.

There are two types of requirements (NASA, 2007):

- Functional requirements define the system’s purposes, i.e., what services it provides and how.
- Non-functional requirements address the practicalities, which is how the system must operate and, the regulations and standards that must be obeyed. Other attributes cannot be expressed as functions—for example, the installation security, reliability maintainability, and availability.

Non-functional requirements may also include the following:

- Can the system be expanded, or adapted to suit new conditions?
- Can the system be fabricated in existing construction yards?
- Can the system be broken down to suit road and sea transportation restrictions?
- Can the system be installed using existing barges and lifting capabilities?
- Can the system be shut down fast in an emergency and startup with a reasonable effort and time?
- Are human–machine interfaces suitable and are users’ access acceptable?
- Are suitable materials, skill sets, and manufacturers available? and
- any other constraints.

2.10 Concept of Operations (ConOps)

The operation (ConOps) document (Fig. 2.10) describes how the system should operate (Frittmann and Edson (2010) and GOES-R, 2020). According to IEE (1362 and 1220), “*the ConOps is a “user-oriented document”, that describes how a system will be used, and includes: who will use it; when they will use it; how they will use it, and for what purpose they will use it.*”

The ConOps document defines the user’s needs and expectations for the system developer, the procurement team, and the other stakeholders. ConOps establishes a shared understanding among all stakeholders. The ConOps document (Fig. 2.10) is prepared at the beginning of the requirements analysis, describing what the system should do (not how it will do it) and its rationale (why). It should also identify any critical, top-level performance objectives and requirements as well as the system rationale. The human–machine interface must also be defined.

The primary considerations are (GPO, 2005):

- The client’s team must be involved.
- The ConOps must be mature as the project moves through the project lifecycle.
- Must allow performing “what if” analysis.
- Should help to reach a consensus before the requirements process begins.

The principal function of ConOps is to have a collective understanding among all stakeholders regarding the expected functionality and level of expected performance. It is also used to describe/define some of the high-level concepts in support of detailed engineering, installation, integration, verification, and validation processes.

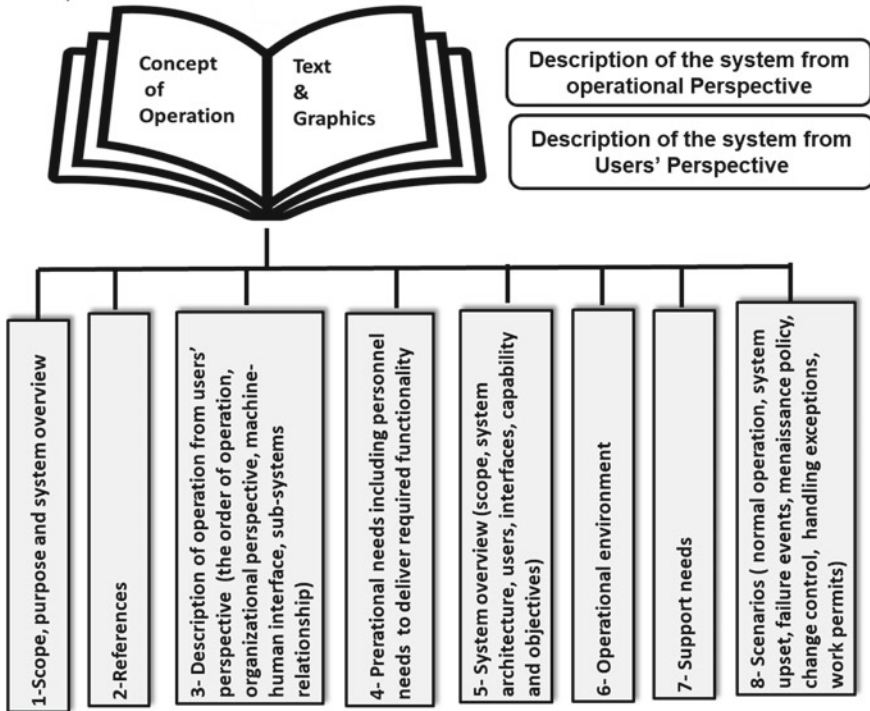


Fig. 2.10 Content of the ConOp document

2.11 Baselineing

Paraphrasing Barry Boehm “Constructing an installation from a specification is like walking on the water—It is safer if it is “frozen”.

A design, product, or procedure at the end of a phase is called a “baseline”, provided it has been reviewed and agreed upon, and then that level of progress is frozen. Any change thereafter can only take place through the project’s formal change control management. A design that is baselined becomes the basis for the next stage of improvement evolving toward the final stage of development. For example, a System Requirement Specification (SRS) is frozen (i.e., baselined) to move to the next phase of development as a basis for completion. Once a design (or product) is baselined then no change can take place haphazardly, thus providing a stable reference for further improvement.

Thus, a baseline is a frozen picture of the design evolution at a specific time (generally the project’s milestone point) in the system development lifecycle, signaling the end of a phase. It becomes a basis for improvements under change control management in the following phase, and hence it needs to be a stable reference for design

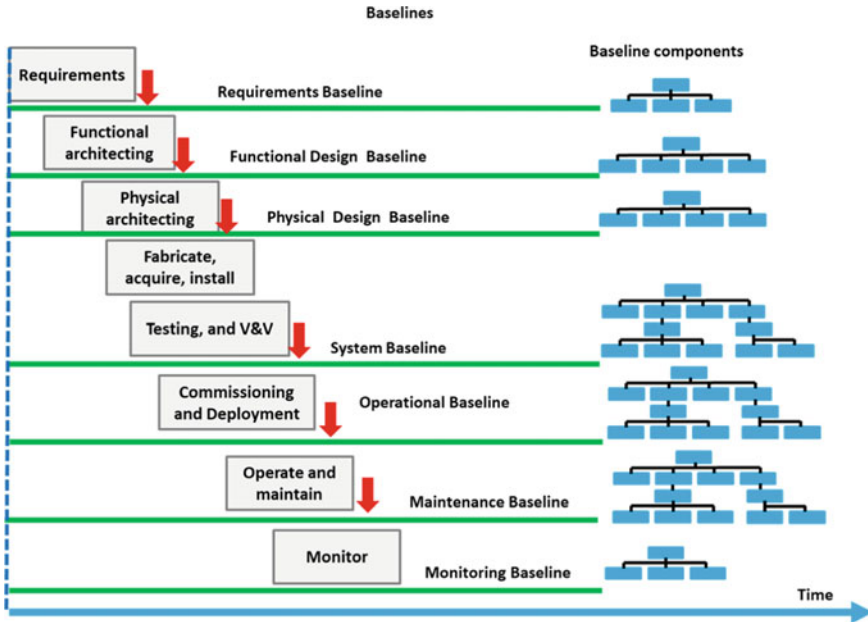


Fig. 2.11 Specifying a baseline

evolution. The objectives are to mitigate the vulnerability of all key deliverables to haphazard uncontrolled changes.

Figure 2.11 shows a typical baseline waterfall, which includes.

2.12 Requirements Traceability

“The purpose of requirements traceability (Fig. 2.12) is to ensure every low-level requirement is linked to the higher-level requirements” (Dick, 2002, 2012), however, some high-level requirements may impact many low-level requirements. Everything should be traceable from requirement specifications to design documents, interface control documents, and down to test procedures for acceptance. “It is important to establish the link between requirements, design specifications, and supporting data for design (known as the design basis) since providing the original context in which a requirement was defined enables any future modification of the requirement to be checked to see if the originally defined constraints are still controls” (Königs et al., 2012).

The traceability aims to create consistent links between test cases, user requirements, and project specifications. It should be possible to consistently cross-reference between components and system requirements, namely the functionality of any equipment is traceable to the client’s requirements. For this purpose, the Client’s

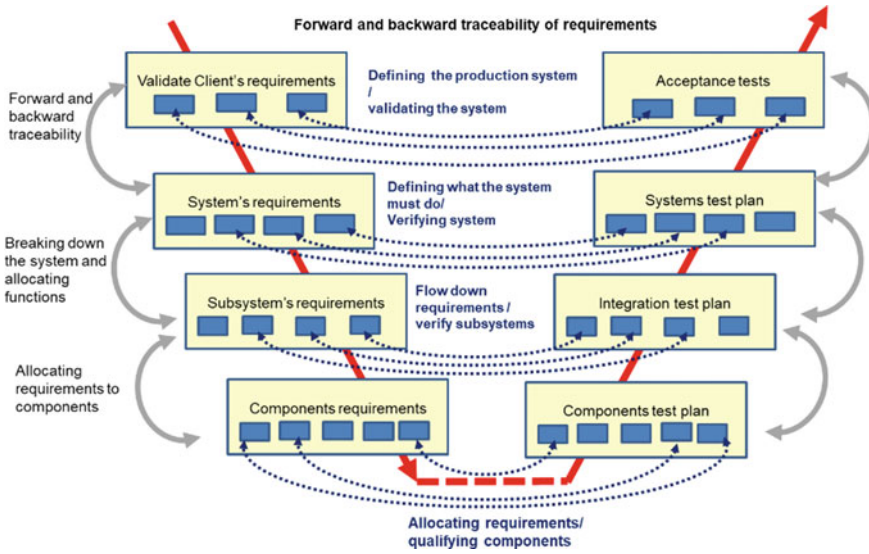


Fig. 2.12 Requirements traceability (see also Fig. 2.8)

requirements are assigned a distinct identification number for designation, which enables referencing.

Each client’s requirement is tied to at least one system requirement, and vice versa. The system engineer enters these links in the tracing matrix to demonstrate that all client requirements have been considered. This is also used to show the completeness of the system specifications and the correspondence between the technical implementation and the requirements. The tracing matrix is also used for the compilation of the test plans to demonstrate that all requirements have been tested.

The primary purpose is to establish links between V&V tests and system requirements (Fig. 2.12). After cascading down higher-level requirements to equipment appropriate test plans are defined to verify whether the system will meet requirements. The following three items must be considered:

- If the Client’s requirements and needs are accounted for
- All components are necessary, and in combination are sufficient, for adequately meeting the client’s needs.
- The test plans will unambiguously verify them.

The circular traceability links enable precisely assessing what will be impacted if a requirement changes, and if there is a choice to avoid the proposed change.

Figure 2.13 gives an example of the decomposition of a system requirement into many component requirements. The two essential sufficient and necessary conditions are:

- **Sufficient:** if the compiled low-level requirements are sufficient? and
- **Necessary:** if every low-level requirement is necessary?

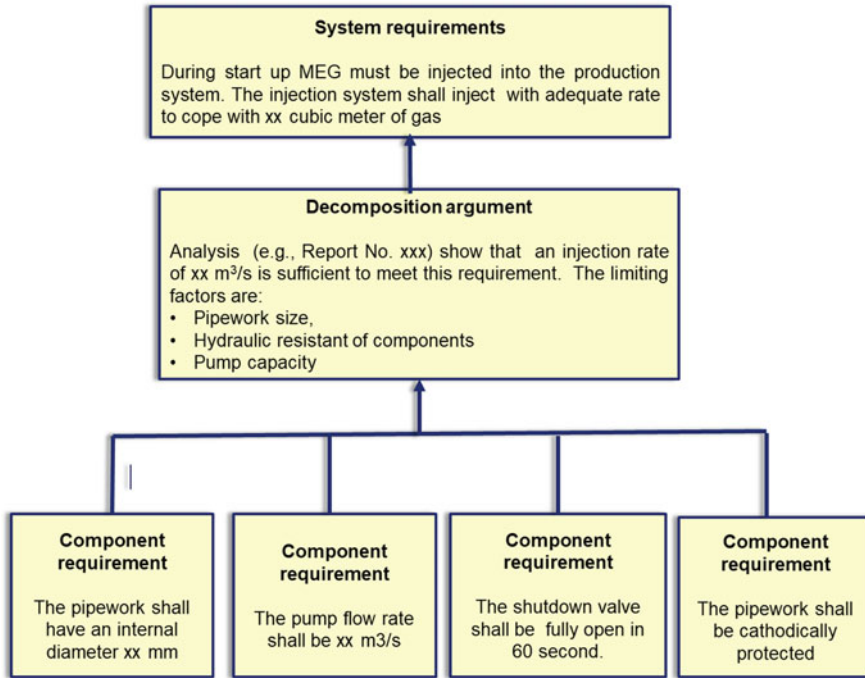


Fig. 2.13 Tracing requirements through a satisfaction relationship

2.13 Reliability Assessment for Assurance

The reliability of any modern equipment that is well-designed and manufactured, with materials that are suitably chosen, and tested is generally very good. However, in practice, several items are bundled together to make an assembly, hence the assembly’s reliability is lower, and the Mean Time to Failure (MTTF) for the assembly would be shorter because there is more equipment that may fail. The Choice of architecture based on reliability can help to optimize MTTF.

Reliability, Availability, and Maintainability (RAM) analyses are used to obtain a functional architecture with the most advantages, considering all constraints. There is a multitude of methods such as MTBF (Mean Time Before Failure), MTTR (Mean Time to Repair), and the Reliability Block Diagram (RBD) that can be used to achieve the client’s goals within reason (Fig. 2.14). These techniques are used to determine the most promising functional architecture for a field. At the early phase of development, historical failure data, (OREDA, 2009) is employed to estimate the availability of a system in pursuit of meeting the project’s target availability.

At the physical design phase, the supplier and the client’s failure database become available, and they are added to the historical data for a more accurate estimation of system reliability. The primary tool to capture all probable failure modes, their

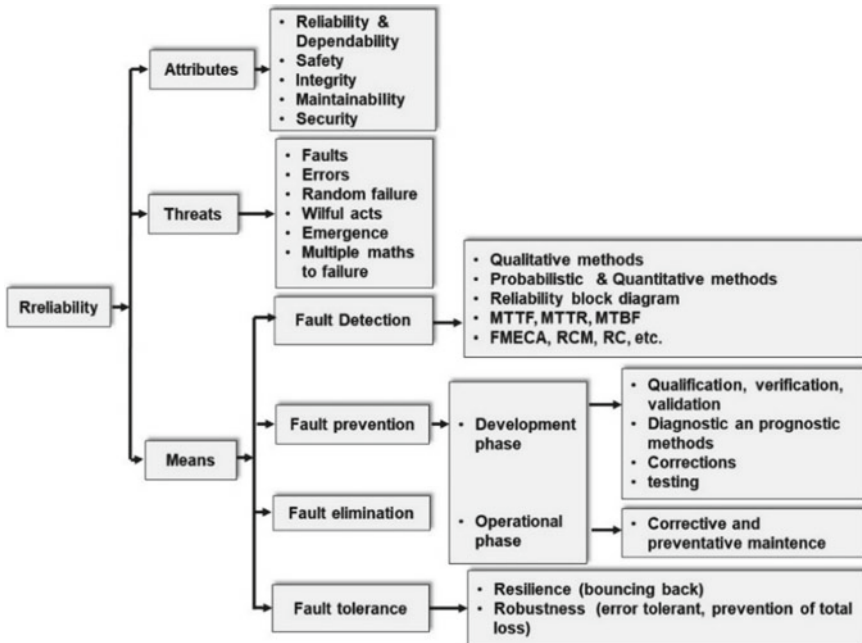


Fig. 2.14 Components of reliability analysis

effects, and criticality is FMCA (the Failure Modes, Effects, and Criticality Analysis). The Define Phase is a suitable time to perform the first FMECA and should be revisited in the Execution Phase when the physical architecture is almost complete. The aim is to identify the weaknesses and potential failure modes, rank them, assess their criticality, and suggest design modifications to avoid them, and if modification is not possible then mitigate their effect. The level of detail of FMEA must be commensurate with the project development phase. It is challenging to undertake RAM analysis at an early phase of development since little is known about the physical system. Nevertheless, even a rough RAM analysis is useful in assessing if the target availability is achievable. Before performing the RAM analysis, a Systems Description Document must be prepared to enable a common understanding among system engineers and designers. The content of this document is a description of all components and their functions as well as their interfaces. This document also defines the expected level of performance for all components, which are used in the system’s RAM analysis at the Define Phase.

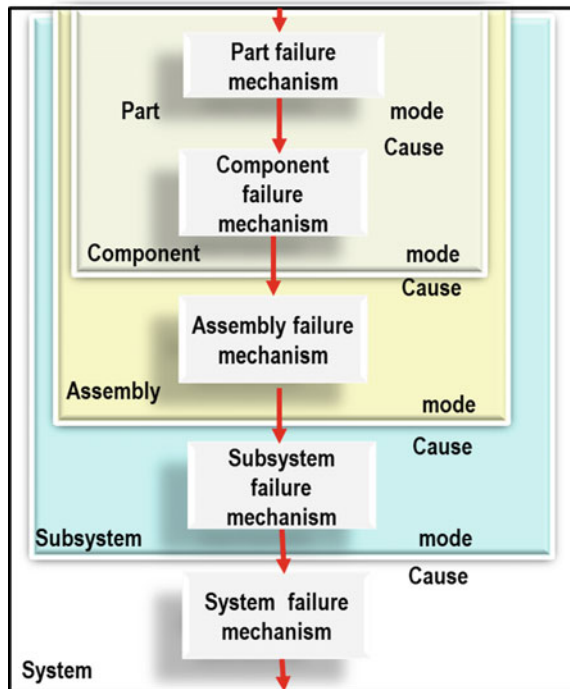
The primary objective is to identify all possible ways that a system can fail to perform. “A failure state results if one or more components malfunction (e.g., not performing well or exceeding their acceptable limit). The resulting state is called a fault or a failure mode” (Rausand & Høyland, 2004). “A component may have several failure modes and each failure mode may have many causes, mechanisms, and effects” (Rausand & Høyland, 2004).

Early in the Define Phase, only suppliers of long lead items are selected and very few pieces of equipment are known with enough detail, it is useful to perform functional FMECA to identify potential failures for each function according to their hierarchy of functions, because a failure of a lower function leads to failure of a higher-level function. After Phase 4 when preferred suppliers are selected and the physical design has taken shape, an FMECA is performed for interfaces to verify compatibility across all interfaces of the system’s components. Then specifications for equipment are prepared (Datasheet), and the preferred suppliers are invited to tender. Towards the end of the Define Phase vendors are selected and possibly contracts to supply are placed.

When contacts are placed for all hardware (the Execution Phase), a System Break-down Structure (SBS) is constructed showing the hierarchy of components and subsystems, which is like the Function Trees Fig. 2.15. With the SBS as input, a detailed FMECA is performed to identify system failures based on the failure modes of lower-level components and step by step moving toward the higher levels in the functional components hierarchy. The FMECA is performed by posing the following questions (Rausand & Høyland, 2004):

- *Credible failure of each part, component, and assembly.*
- *Possible failure mechanisms of identified failure modes? And their possible effect.*
- *Is the failure on the safe or unsafe side? (The concept of a “fail-safe” system.)*

Fig. 2.15 Failure hierarchy (adapted from IEEE 1220)



- *How to detect failure?*
- *What provisions are provided to stop the failure progression or mitigate its effect?*

RAM analysis is deployed for verification and validation of the system's components at every level of evolving development (Using TRL as an indicator) and compared against the agreed client's operational requirements. Complications in manufacturing and system integration could lessen the system availability. Therefore, to offset the influence of manufacturing errors on the system availability, designers deliberately aim at availability above the agreed operational availability target, while addressing every manufacturing limitation.

2.14 Technology Readiness Level

Tests and simulations can only eliminate some of the uncertainties. The Technology Readiness Level (TRL) scale is another tool to manage technical risks. TRL is a useful tool for tracing the progress of technology toward readiness and maturity (API-RP-17N, 2023). However, "*TRL is not a measure of the quality of technology to be inserted into a system*" (API-RP-17N, 2023).

"The TRL for a piece of existing equipment that is inserted in a new system is assumed to be at TRL = 4, or at best at TRL = 5" (Yasseri, 2014b). The logic behind this decision is that a new subsea field is not the same as an old one; they are similar but not the same. Consequently, every piece of equipment must be qualified for use in the new environment and operational conditions.

The notion of TRL was advanced by NASA in the '70s. Later, NASA rehashed the idea and published this metric as a 9-point scale. Many industries have adopted NASA's 9-point TRL scale but modified it to suit their needs (Yasseri, 2013). Table 2.2 is adopted from API, which shows API's definitions alongside a NASA-type TRL. TRL = 1 in The NASA scale is a technology as a basic idea probably supported by basic science. The development is pushed along the TRL ladder until it reaches maturity, then readiness which is proven by working in its intended operating environment.

Table 2.3 shows processes that are used to reduce uncertainties of the technology during its development phase.

2.15 Verification and Validation

Components are tested for acceptance at the factory, known as Factory Acceptance Tests (FAT). Some components may require extended factory tests (EFAT). Tested components are delivered to the fabrication yard to produce bigger assemblies or modules for ease of transportation and installation. Modules are then transported to the quayside for integration. They are tested at quayside before installing them in their

Table 2.2 API definitions of TRLs and its equivalent NASA's scale (based on API-RP-17N, 2023)—(Yasseri, 2013)

	API 17N's TRL	API 17N's TRL	NASA type TRL	Development stage completed	Definition of the development stage
Concept	Initiation	0	1	Unproven concept (basic R&D, paper concept)	Basic scientific/engineering principles observed and reported; paper concept; no analysis or testing completed no design history
	Concept	1	2	Proven concept (as a paper study or R&D experiments)	(a) Technology concept and/or application formulated (b) Concept and functionality proven by analysis or reference to features common with/to existing technology (c) No design history; essentially a paper study not involving physical models but may include R&D experimentation
Proof-of-concept	Proof-of-concept	2	3	Validated concept (experimental proof of concept using physical model tests)	Concept design or novel features of the design is validated by a physical model, a system mock-up, or a dummy and functionally tested in a laboratory environment; no design history; no environmental tests; materials testing, and reliability testing are performed on key parts or components in a testing laboratory before prototype construction
	Integration	3	4	Prototype tested (system function, performance, and reliability tested)	(a) Item prototype is built and put through (generic) functional and performance tests; reliability tests are performed including reliability growth tests, accelerated life tests, and robust design development test programs in relevant laboratory testing environments; tests are carried out without integration into a broader system (b) The extent to which application requirements are met is assessed and the potential benefits and risks are demonstrated

(continued)

Table 2.2 (continued)

	API 17N's TRL	API 17N's TRL	NASA type TRL	Development stage completed	Definition of the development stage
	Demonstration	4	5	Environment tested (pre-production system environment tested)	Meets all requirements of TRL 3; designed and built as a production unit (or full-scale prototype) and put through its qualification program in a simulated environment (e.g., hyperbaric chamber to simulate pressure) or intended environment (e.g., subsea environment) but not installed or operating; reliability testing limited to demonstrating that prototype function and performance criteria can be met in the intended operating condition and external environment
	Commissioning	5	6	System tested (production system interface tested)	Meets all the requirements of TRL 4; designed and built as a production unit (or full-scale prototype) and integrated into the intended operating system with a full interface and functional test but outside the intended field environment
Field qualified	Production	6	7	System installed (production system installed and tested)	Meets all the requirements of TRL 5; production unit (or full-scale prototype) built and integrated into the intended operating system; full interface and function test program performed in the intended (or closely simulated) environment and operated for less than three years; at TRL 6 new technology equipment might require additional support for the first 12–18 months
	Field-proven	7	8&9	Field proven (production system has been field proven)	Production unit integrated into the intended operating system, installed and operating for more than three years with acceptable reliability, demonstrating a low risk of early life failures in the field

Table 2.3 Uncertainty reduction at various levels of the TRL scale (adapted from Yasseri, 2015b)

Phase	API TRL	Development stage completed	Reduction of uncertainties
System validation	7	Field Proven SPS is field-proven (several months in operation)	Maintaining the aging system's reliability
	6	System installed SPS is installed and tested for operational requirements. Start of commissioning and hand over	Validating and commissioning of SPS using 'use cases' and the system's operational requirements
Technology verification	5	System tested Testing of SPS in its environment is complete	Validation testing; final RAM analysis of the as-built SPS
	4	Tested in the operational environment SPS is tested in its environment	Verification testing
	3	Prototype tested The system's functions, performance, and reliability are investigated	RAM analysis using the vendor's and client's failure data
Concept validation	2	Validated concept Experimental proof of physics is performed using laboratory models	HAZID, HAZOP, FMECA, fault tree analysis, event tree analyses, Bow-Tie analysis
	1	Fundamental concepts are demonstrated Proof of concept as desk study or R&D by experimentation	What-if analysis, scenario building, logical architecture level reliability analyses; RAM analysis using generic data
	0	Unproven concept Basic ideas in research papers	Preliminary HAZID, HAZOP, FMECA, and Operability analyses

working environment and integrating them with the previously installed modules until the integration and verification of the entire 'as-built system' is complete, the entire system is tested and commissioned for handing it over to the client's team. When the entire system is installed on the seabed, and commissioning tests are complete, the responsibility of operation is gradually handover to the client's operations team. The handover includes providing support, devising a sparing policy, instructions for operation, operator training, and all other enabling items that assure the smooth running of the operation and maintaining the system in good working condition. During the handover period acceptance tests are organized by the client's team to confirm that the system complies with the client's requirements. The handover

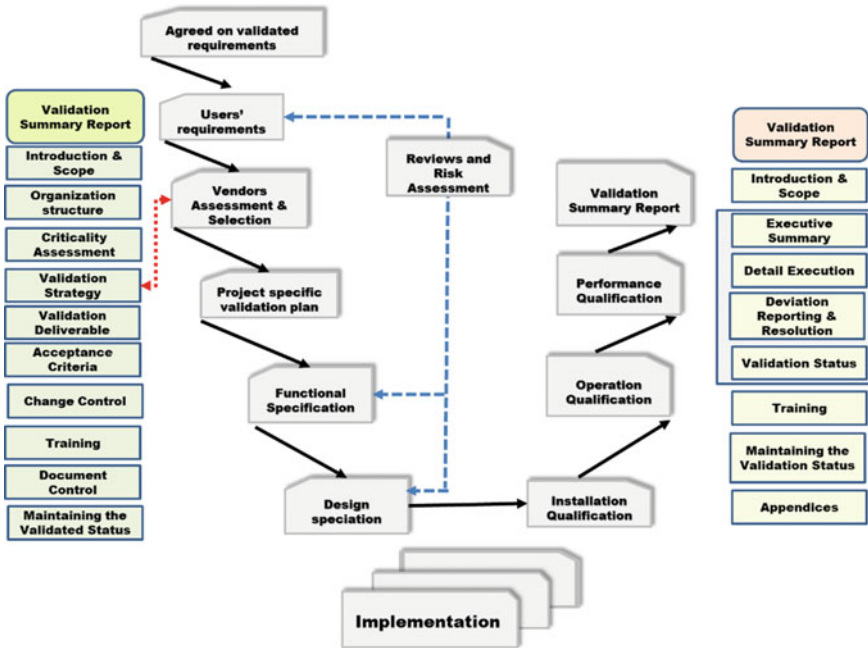


Fig. 2.16 V&V life cycle

period and warranty period are intended for a smooth transition of responsibilities from the primary contactor to the client’s operation team. The entire process is called “Verification” and “Validation” (V&V), which are carried out through a myriad of tests at every stage of system integration (Fig. 2.16).

“Verification and Validation procedures are used to confirm that a product, service, or system meets its defined specifications and judged it is FFS. Verification is a quality control process that is used to evaluate whether a product, service, or system complies with regulations, specifications, or conditions requested by the client at the beginning of the Development” (Babuska & Oden, 2004). “Validation is a quality assurance process for obtaining evidence that with a high degree of confidence proves a product, service, or system delivers the agreed specified requirements” (Plant & Gamble, 2003). The ISO 9000 (2015) definition is based on the general field of quality and the focus is on providing “objective evidence” which proves that all requirements have been satisfactorily satisfied. According to ISO 26262 (2011), “the validation is focused on providing proof that the system will meet its intended purpose.” ISO defines the verification process in broad terms.

Figure 2.17 shows a possible flow diagram for the V&V activities. The process begins with reliability analyses and ends with V&V by testing, prototyping, simulation, and analytical approaches. The approved system’s requirements are used to define the subsystems’ requirements and specifications, which are then validated to

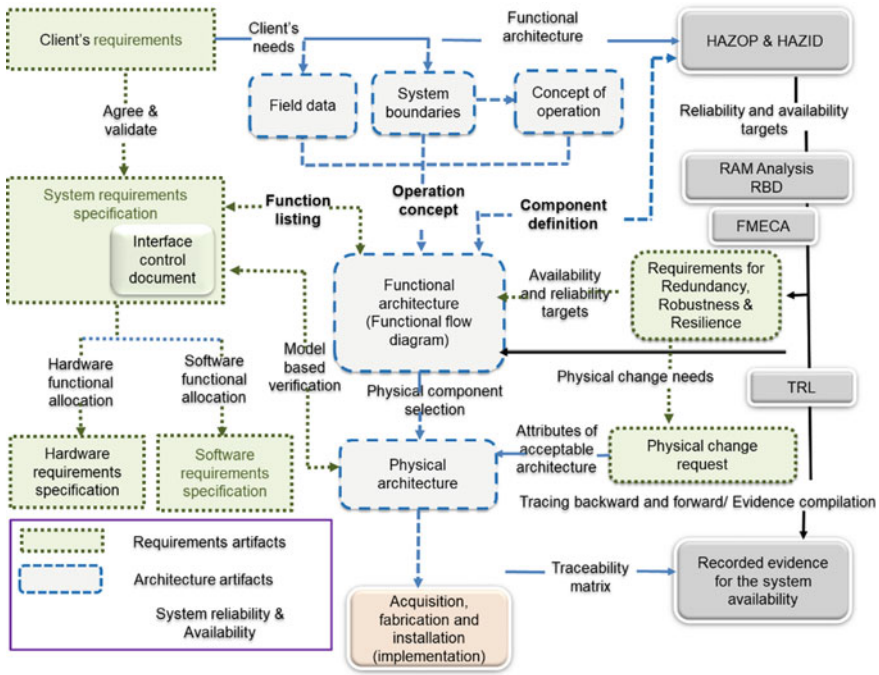


Fig. 2.17 A possible flow chart for verification and validation

assure that they are feasible, necessary, and exhaustive, in the light of the notion of ‘necessary and sufficient’ condition.

Tools are qualified but processes are validated according to this definition, qualification is considered as a subset of validation. All fasteners (rivets and nuts & bolts) including welding are considered tools for joining, but their FFS must be evaluated. In this respect:

- Fasteners, weldments, and materials as well as procedures using them are qualified as a tool for system building. They are procured from trusted suppliers and may be accompanied by a certificate of FFS. However, basic verification, based on the statistical sampling method, should be undertaken.
- Fasteners and weldments in assembled equipment must be validated to assure that they are capable enough to allow the equipment to fulfill its purpose. Results from the fastener’s qualification tests are appended in the equipment’s validation report.

The Verification and Validation strategy is a set of actions, consisting of tests, inspections, and trials. Each requirement may require several actions.

Each action must be:

- Suitable to check the requirements under consideration.

- Timely-implementation at an early phase is preferable.
- Describe the necessary testing tools.
- Define the successful outcome.

Verification & Validation, and qualification are used interchangeably in some literature. For example, IEC 61508 (2010), defines the qualification process to encompass V&V.

Figure 2.18 shows the V&V activities in parallel with the development processes. Any requirement may give rise to several verification tests at every phase of the evolving project. If a requirement is fulfilled by chance due to the beneficial effect of emergent behavior, such a chance event must be confirmed by tests at the level of emergent behavior.

Evidence for quality assurance is collected throughout the development phase utilizing a combination of testing and simulation. Validation solely based on tests or analytical methods would let some faults remain undetected. A balanced approach to confirm compliance has a high chance to control costs and enhance confidence in the system’s performance. Simulation is preferable and testing is best used to fill the knowledge gap since simulation cannot detect manufacturing errors or visual inspection may not be suitable for accepting fabrication defects.

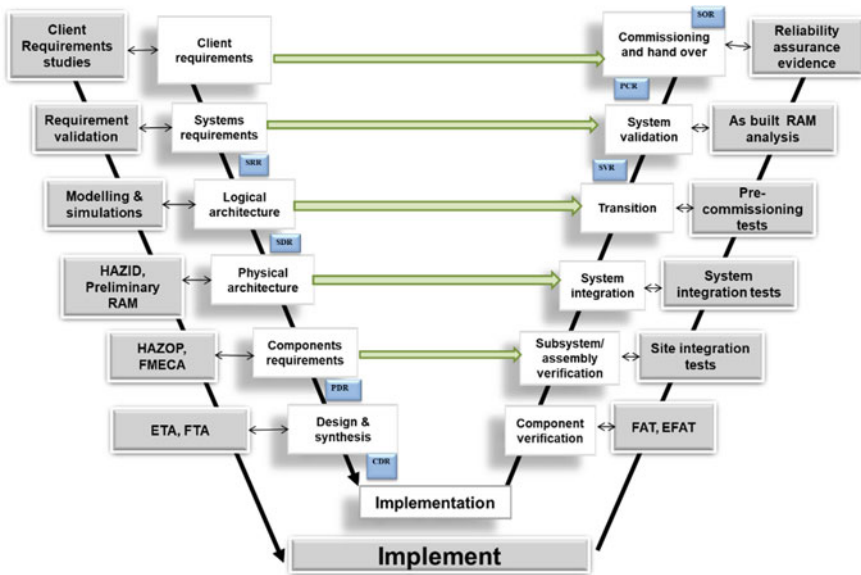


Fig. 2.18 The Qualification activities are shown in parallel with the development activities

2.16 Provision of Evidence

There are several methodologies for gathering evidence for supporting reliability assurance, and naturally, some overlap between them should be expected. Any chosen procedures (methods), and the depth and detail, are based on the “*need to know*” or “necessary and sufficient, and hence the choice depends on the problem at hand. There must be a purpose to gather information. Sometimes, evidence is collected for the design activities and hence is indispensable, since the design effort, however exhaustive, cannot reasonably detect all probable failures and their causes. The concept is founded on the principle of ‘beyond a reasonable doubt’, which is quite rigorous, not based on the balance of probabilities. However, reasonable doubt does not mean beyond all doubts. Sound engineering judgment is needed to avoid undertakings yielding little value. As a minimum two different methods should be used to detect all probable faults. A particular procedure, e.g., testing, may be necessary but is not sufficient (e.g., doing the same test twice), thus it must be complemented with another method to make sure that all faults are detected. Generally, simulations and analytical methods are used to lower the cost of testing needed for reliability assurance. Numerical approaches could replace the need for testing, for example when testing is almost impossible or very expensive. A few approaches that are in use for managing V&V are listed below:

Trust-based means that hardware is sourced from a trusted supplier or a design can be claimed to be compliant with codes and standards by the contractor, and it is taken on trust that the contractor’s claim is valid. Generally, any claim involving analysis or simulation is verified by a trusted third-party verifier.

The certification approach means that a third party has witnessed the performance of the finished product during some specified tests and that the third party awarded a certificate of performance. The certificate approach is commonly used for mass-produced items based on a standard or specification. The certificate is the qualification of the production facilities, as well as assuring that prescribed standards (depending on the application area), are followed and the product meets all stated requirements which means the product is FFP. This approach is also used to validate the claim of a manufacturer/fabricator that an item as sold is “fit-for-Purpose”. Representatives of a verifying consultancy witness tests organized by the manufacturer and issue a certificate of compliance for a particular application if convinced. For example, firewalls are qualified using this approach.

The current certification approach follows the prescribed process of an applicable standard. For example, IEC 61508 is designed for industrial purposes, ISO 26262 (2011) covers the automotive industry, and DO-178B/C (2012) focuses on software for airborne systems.

Competence Cost of compliance with ever-increasing requirements is not trivial. The capability to check weld quality demands management of personnel competence. ISO 3834:2008 defines the quality requirement for fusion welding, with an emphasis on the welder’s competence and inspection, supervision, and testing personnel (ISO

3834, 2008, and AS/NZS ISO 3834 (2021). Thus, competence assurance (e.g., certified operators) is essential in delivering reliable systems, to assure the delivered product is FFP, and should remain so for its design life. Personnel competence assurance is set out in ISO 9001 clause 6.2.1. Inspectors are also required to have a certificate of ‘competence’ issued by an authority.

2.17 Acceptance Testing

The purpose of acceptance testing is to validate the system assuring that it will deliver the required functionalities; that is FFS. At the start of the project, all Client requirements, the system purpose, key capabilities, use cases, (ConOps & usage scenarios), level of performance to be achieved, and the system’s acceptance criteria for validations are defined and documented. The *System Validation Plan* is produced and put under the change control process for monitoring to ensure that the test procedure (the verification plan) is relevant, up-to-date, and not changed without the approved change processes. “*The purpose of Test Plans is to demonstrate that a system satisfies the approved requirements-i.e., FFS*” (Engel, 2010)—Fig. 2.19

The Test Plans document is the overall testing strategy, which includes the general test procedures, what results are to be documented, and the procedures for dealing with test failures. The Test Plan will also include types of testing, describing the testing environments and tooling, the responsibility matrix, test equipment that will be used, and any other organizational procedures.

Test Protocols describe the specific testing requirements. Test Protocols are a collection of Test Cases (use cases) that validate a specific element of the system. Each test case includes the goal of the test, prerequisites, as well as acceptance criteria. Each test case is broken down into a series of steps. Each step includes detailed instructions, what result to expect, as well as the actual result, and what to document. The test procedures must have sufficient details so that a tester can perform the testing consistently without requiring interpretation.

The **Client Acceptance Criteria** are used for authorizing the shipment of parts, equipment, or assemblies that are tested and ready to be delivered to the Client site. That is, it is verified that the part, equipment, or assembly is constructed in a manner that has been defined by the flow-down of the client requirements and fabricated in a manner that meets the industry standards, good practices, and client standards. This is achieved by various procedures, such as using independent testers, witnessing the vendor’s test, or on the trust base.

User Acceptance Testing (UAT)—for operational needs—describes testing to prove the fulfillment of what the user expects the system to deliver, and how the system must function. UAT documents provide pertinent information, data, the operating environment, acceptable processes, and the system’s functionality to make tests meaningful, applicable, and repeatable. These tests are completed during the FAT (Factory Acceptance Test) as well as the SAT (Site Acceptance Test (SAT) (Rahimi, 2013). If a piece of equipment is developed by the vendor’s subcontractor, then it

A complementary purpose of testing is to check whether all interfaces comply with specifications and also all constraints have been accommodated. The integration plan, which was produced earlier in the project development, defines the order of components integration towards constructing the whole system. The functionality of every subsystem at every stage of integration is checked against the relevant approved requirements and must be verified following the ‘Subsystem Verification Plan.’ Tests for the verification of component-level requirements are necessary because many systems’ requirements are flowed down via several routes and levels of system decomposition (Yasseri, 2013). These efforts should ensure that the functionality of all parts of the system has been proven.

2.18 Insights and Implications for Practice

A practical framework was described for delivering reliable subsea production systems based on the system engineering processes. The objective is the assurance of uninterrupted operation and the robustness & resilience of the SPS. Although subsea production systems are used as a vehicle to explain the process, the method is equally applicable to the reliability assurance of any capital project (Okaro, 2017).

Reliability assurance is a useful framework, to build robustness and resilience into a system (e.g., security threats, Yasseri, 2019). Reliability assurance also relies on mitigative policies, such as using appropriate materials, corrosion & erosion protection, and prevention of accidents (e.g., dropped objects), other external hazards (e.g., boat impact, seismic event, storms, debris flow), geotechnical hazards (liquefaction, seabed movement), and so forth. The effectiveness of reliability assurance is judged by the availability of the system for continued operation when required.

The outlined method also supports the project’s risk control management, and it is also aligned with the owner’s strategic objectives. The described framework aims to achieve the following objectives:

- Meet the Client’s needs and goals.
- Control the project cost and schedule.

The method starts with the client’s requirements (needs and objectives). It was stated that each requirement shall be:

- *Traceable*—higher-level requirements are linked to one or more components’ requirements
- *Unique*—it should be associated with a paragraph in a document with an identifier
- *Single*—it should not concern more than one issue.
- *Verifiable*—can be verified using approved project’s verification procedures.
- *Unambiguous*—defined with an exact statement.
- *Correctly assigned* to applicable requirements, with unambiguous paragraph identifier.

A fundamental idea is that quality must be built into a system's components and processes at the start of development. The system design specifications must support the quality needs of all processes so that they can be judged as 'deemed' FFS. Reliability analysis will identify 'critical elements' of a system architecture, which then can be used to moderate the amount of testing. The term 'critical elements' means mitigation controls devices included during the design phase, which are hardware, not procedural controls. Risk analysis may also be used to identify critical elements.

A myriad of techniques is used for risk identification. HAZOP, HAZID, and FMECA are the most favored tools of hazard identification and assessment; all perform well in identifying failure modes. The original use of HAZID and HAZOP was to enhance system safety, but the reliability analysis also employs them.

It also emphasized the importance of tracing the requirements to their physical implementations (design solutions). It was shown how to translate the results of requirements analysis into the project-specific design requirements, from which technical specifications for equipment (data sheets) can be developed and used to prepare testing and acceptance criteria. Testing starts from the lowest level component, then progresses to assemblies and modules, and finally, the whole system is tested for compliance (Tehera et al., 2019)-The validation process.

Reliability and safety analysis address separate issues, but the safety-related system must be reliable, thus risk analyses are used for risk reduction and enhancement of the reliability of the safety system. Note that there are two sets of safety systems which are process safety and system safety; though both perform the same function but are independent of each other.

Designers should never intentionally create requirements and designs that result in the system operating at the "limit."; i.e., little or no margin. If a system is designed to meet performance specifications within an adequate margin, then it should be rare for the system to fail rapidly when excursions beyond normal operating conditions are minor. A key objective in developing a high-reliability system is, for the system to degrade gracefully without sudden, frequent failure, as well as OPEX, and unplanned intervention overrun.

References

- ABS. (2017). *Guidance notes on qualifying new technologies* (p. 45). American Bureau of Shipping.
- API-RP-17N. (2023). *Recommended practise for subsea production system reliability, technical risk & integrity management* (2nd ed.). API.
- API-RP-17Q. (2023). *Subsea equipment qualification—Standardized process for the documentation* (2nd ed.). API.
- AS/NZS ISO 3834. (2021). Standard: Australian Welding Institute. Last accessed 02/05/2023. <https://welding.org.au/certification/asnzs-iso-3834-awi/>
- Babuska, I., & Oden, J. T. (2004). Verification and validation. *Computational Engineering and Science: Basic Concepts. Computer Methods in Applied Mechanics and Engineering*, pp. 4057–4066. Last accessed 21/04/2023. Available at https://users.oden.utexas.edu/~oden/Dr._Oden_Reprints/2004-004.verification_and.pdf

- Bahill, A. T., & Henderson, S. J. (2005). Requirements development, verification, and validation exhibited in famous failures. *Systems Engineering*, 8(1), 1–14. https://www.researchgate.net/publication/227601137_Requirements_Development_Verification_and_Validation_exhibited_in_famous_failures
- Bureau Veritas-NI525. (2020). *Risk bases qualification of new technology methodological guidance* (p. 20). Guidance Note NI 5252 R00E.
- Dick, J. (2002). Rich traceability. In *Proceedings of the 1st International Workshop on Traceability in Emerging Forms of Software Engineering* (pp. 18–23). Last accessed 21/04/2023. https://www.researchgate.net/publication/239560722_Rich_Traceability
- Dick, J. (2012). Evidence-based development—Coupling structured argumentation with requirements development, system safety. In *7th IET International Conference on Incorporating the Cyber Security Conference, UK*.
- DNVGL-RP-A203. (2019). *Qualification of new technology*. Det Norske Veritas.
- DO-178B/C (2012). *Software considerations in airborne systems and equipment certification*. RTCA SC-205 and EUROCAE WG-12.
- Edson, R. (2008). *Systems thinking applied. A primer*. Analytic Services, Inc.
- Engel, A. (2010). *Verification, validation, and testing of engineered systems*. Wiley.
- FAA. (2008). Federal Aviation Administration. Requirements Engineering Management Handbook DOT/FAA/AR-08/32.
- Feiler, P. H., Goodenough, J. B., Gurfinkel, A., Weinstock, Ch. B., & Wrage, L. (2012). Special Report, CMU/SEI-2012-SR-013, Reliability Validation, and Improvement Framework Carnegie Mellon University.
- Frittman, J., & Edson, R. (2010). A systems thinking based approach to writing effective concepts of operations (ConOps). In *8th Conference on Systems Engineering Research* March 17–19, Hoboken, NJ. <http://anser.org/docs/asyst-doc/Systems-Thinking-and-Writing-Effective-CONOps.pdf>
- GOES-R Series. (2020). Concept of operations (CONOPS), 410-R-CONOPS-0008. Last accessed 21/04/2023. <https://www.goes-r.gov/syseng/docs/CONOPS.pdf>
- GPO (Government Printing Office). (2005). Systems thinking and writing effective ConOps. Concept of operations (ConOps v2). Last accessed 21/04/2023. https://www.govinfo.gov/media/FDsys_ConOps_v2.0.pdf
- Grady, J. O. (2007). *System verification: Proving the design solution satisfies the requirements*. Academic Press.
- Hull, E., Jackson, K., & Dick, J. (2002). *Requirements engineering*. Springer-Verlag.
- IAEA-TECDOC-1264. (2001). Reliability assurance program guidebook for advanced light water reactors.
- IEC 61508, SC 65A. (2010). Functional safety of electrical/electronic/programmable electronic safety-related systems in seven parts.
- IEEE 1220-2005. (2005). Reaffirmed in 2011, Standard for Application and Management of the Systems Engineering Process.
- INCOSE. (2015). *Systems engineering handbook—A guide for system life cycle processes and activities, version 4.0*. John Wiley and Sons, Inc., ISBN: 978-1-118-99940-0.
- ISO 26262. (2011). Functional Safety Standard. ISO organization.
- ISO 3834. (2008). *Quality requirements for fusion welding of metallic materials*. ISO Organization.
- ISO 9000. (2015). *Quality management systems: Fundamentals and vocabulary*. ISO Organization.
- ISO/IEC 15288. (2008). *Systems engineering-system life cycle processes*. ISO Organization.
- Königs, S. F., Beier, G., Figge, A., & Stark, R. (2012). Traceability in systems engineering—Review of industrial practices, state-of-the-art technologies and new research solutions. *Advanced Engineering Informatics*, 26(4) 924–940.
- Lloyds Register. (2017). *Guidance notes for technology qualification* (p. 43). Published by Lloyd's Register Group Ltd.
- MIL-HDBK-217 Rev. F. (1995). *Reliability prediction of electronic equipment*. US Department of Défense, MIL-HDBK.

- NASA. (2007). *Systems engineering handbook*. NASA Technical Report NASA/SP-2007-6105 Rev1, ISBN 978-0-16-079747-7.
- Okaro, I. A. (2017). *An integrated model for asset reliability, risk and production efficiency management in subsea oil and gas operations*, Ph.D. Thesis Submitted to the Newcastle University UK, <https://theses.ncl.ac.uk/jspui/bitstream/10443/3651/1/Okaro%2C%20I%20202017.pdf>
- OREDA. (2009). *Offshore reliability data* (5th ed.). Det Norske Veritas.
- Pecht, M. (1993). Design for qualification. In *Annual Reliability and Maintainability Symposium*, pp. 1–4.
- Plant, R., & Gamble, R. (2003). Methodologies for the development of knowledge-based systems. *Knowledge Engineering Review*, 18(1), 47–81.
- Rahimi, M. (2013). *A contribution to reliability qualification of new technical equipment: With focus on subsea production equipment*, Ph.D. This Norwegian University of Science and Technology. <https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/240961/Maryam%20Rahimi.pdf?sequence=3&isAllowed=y>
- Rausand, M., & Høyland, A. (2004). *System reliability theory: Models, statistical methods, and applications* (p. 664).
- Royal Academy of Engineering. (2014). *Thinking like an Engineer—Implications for the education. In System A report for the Royal Academy of Engineering Standing Committee for Education and Training Full report*. www.raeng.org.uk/thinkinglikeanengineer
- Senge, P. M. (1990). *The fifth discipline*. Random House.
- Sillitto, H. (2014). *Architecting systems. Principles and practice*. College Publications.
- Tehera, K., Wynn, D. C., Eral, C., & Ekert, C. (2019). Testing in the incremental design and development of complex products. *Research in Engineering Design*, 30, 291–316.
- Viola, N., Corpino, S., Fioriti, M., & Stesina, F. (2012). Functional analysis in systems engineering: Methodology and applications. In B. Cogan (Ed.), *Systems engineering—Practice and theory*. InTech. ISBN: 978-953-51-0322-6.
- Woody, C., Ellison, R., & Nichols, W. (2014). *Quality and reliability measures*. CMU/SEI-2014-TN-026, Carnegie Mellon University. Last accessed 21/04/2023. <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=428589>
- Yasseri, S. (2012). Subsea technologies selection using analytic hierarchy process. *International Journal of the Society for Underwater Technology*, 30(3), 151–164.
- Yasseri, S. (2013). Subsea system readiness level. *International Journal of the Society for Underwater Technology*, 31(2), 77–92.
- Yasseri, S. (2014a). Application of system engineering to subsea development. *International Journal of Underwater Technology*, 32(2), 93–109.
- Yasseri, S. (2014b). A measure of subsea readiness level development. *International Journal of Underwater Technology*, 33(4), 215–228.
- Yasseri, S. (2015a). Evidence-based subsea engineering. *International Journal of Underwater Technology*, 32(4), 231–244.
- Yasseri, S. (2015b). Interface management of subsea field development. *Underwater Technology*, 33(1), 41–57.
- Yasseri, S., & Bahai, H. (2018). Availability assessment of subsea distribution systems at the architectural level. *Ocean Engineering*, 153, 399–441.
- Yasseri, S., & Bahai, H. (2019). Interface and integration management for FPSOs. *Ocean Engineering*, 191(3), 106441.
- Yasseri, S. (2019). A systems engineering approach to physical security of oil & gas installations. *IJCOE*, 4(3).