**Deployment Package**

**Needs and Requirements Engineering**

**Systems Engineering Basic Profile**

**Notes:**

This document is the intellectual propriety of its author’s organization. However, information contained in this document is free of use. The distribution of all or parts of this document is authorized for non-commercial use as long as the following legal notice is mentioned:

© International Council on Systems Engineering (INCOSE) 2013

Commercial use of this document is strictly forbidden. This document is distributed in order to enhance exchange of technical and scientific information.

This material is furnished on an “as-is” basis. The author(s) make(s) no warranties of any kind, either expressed or implied, as to any matter including, but not limited to, warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material.

The processes described in this Deployment Package are not intended to preclude or discourage the use of additional processes that the user may find useful.

|  |  |
| --- | --- |
| **Authors** | G. Fanmuy, INCOSE, France  Joseph MARVIN – INCOSE, USA  Ronald HOUDE, Mannarino Systems & Software, Canada |
| **Editors** | Martin GEISREITER – INCOSE, Germany  C. Y. LAPORTE, École de Technologie Supérieure |
| **Creation date** | 5 January 2011 |
| **Last update** | 24 January 2017 |
| **Version** | 2.0 |

Versions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date**  (dd/mm/yyyy) | **Version** | **Description** | **Author** |
| 05/01/2011 | 0.1 | Initial Release | Gauthier FANMUY  Joe MARVIN |
| 03/08/2013 | 0.2 | Revision | Ronald HOUDE |
| 09/08/2013 | 0.3 | Updated Task Lists to  ISO/IEC 29110-5-6-2, 2013-07-23 | Ronald HOUDE |
| 10/31/2013 | 0.4 | Task/Activities Completion  Alignment with INCOSE Guide for Writing Requirements | Ronald HOUDE |
| 11/7/2013 | 0.5 | Updated Section 1 | Ronald HOUDE |
| 11/14/2013 | 0.6 | Incorporation of review comments | Ronald HOUDE |
| 12/20/2013 | 0.7 | Incorporation of VSME DP Comments | Ronald Houde |
| 7/4/2014 | 2.0 | Publish draft for distribution after IS 2014 | Joseph Marvin |

Abbreviations/Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| AIAA | American Institute of Aeronautics and Astronautics <http://www.aiaa.org> |
| ANSI | American National Standards Institute <http://www.ansi.org> |
| CMMI | Capability Maturity Model Integration <http://cmmiinstitute.com> |
| CMMI-DEV | Capability Maturity Model Integration for Development |
| DP | Deployment Package |
| EIA | Electronic Industries Alliance <http://www.eciaonline.org/eiastandards/> |
| EMC | Electro-Magnetic Compatibility |
| GEIA | Government Electronics and Information Technology Association <http://www.geia.org> |
| HMI | Human Machine Interface |
| IEC | International Electrotechnical Commission <http://www.iec.ch> |
| IEEE | Institute of Electrical and Electronic Engineers <http://www.ieee.org> |
| ISO | International Organization for Standardization, <http://www.iso.org> |
| INCOSE | International Council on Systems Engineering, <http://www.incose.org/> |
| PM | Project/Program Manager/Management |
| PMBOK | Project Management Body of Knowledge, <http://www.pmi.org/> |
| SE | Systems Engineering |
| SR | System Definition and Realization |
| StRS | Stakeholder Requirements Specification |
| SyRS | System Requirements *Specification* |
| TBD | To Be Defined/Determined |
| TBR | To Be Resolved |
| TBS | To Be Specified |
| TR | Technical Report |
| VSE | Very Small Entity (ISO/IEC 29110) |
| VSMEs | Very Small and Micro Entities (INCOSE) |
| V&V | Verification and Validation |

Table of Contents

[1. Technical Description 6](#_Toc392250125)

[Purpose of this document 6](#_Toc392250126)

[Why is Systems Engineering important? 7](#_Toc392250127)

[Why is Requirement Engineering important? 8](#_Toc392250128)

[Characteristics of Good Requirements 9](#_Toc392250129)

[Characteristics of Good Specifications 11](#_Toc392250130)

[2. Definitions 12](#_Toc392250131)

[Generic Terms 12](#_Toc392250132)

[Specific Terms 13](#_Toc392250133)

[3. Relationships with ISO/IEC-29110 15](#_Toc392250134)

[4. Description of Processes, Activities, Tasks, Steps, Roles and Products 21](#_Toc392250135)

[SR.1.1 – Review Project Plan with the Work Team members 21](#_Toc392250136)

[SR.2.1 - Elicit Acquirer and other stakeholders requirements and analyze system context 22](#_Toc392250137)

[SR.2.2 - Review Stakeholders Requirements Specifications with PM 24](#_Toc392250138)

[SR.2.3 - Baseline Stakeholders Requirements Specification with the Acquirer and Stakeholders 25](#_Toc392250139)

[SR.2.4 - Capture System Requirements and Interfaces 27](#_Toc392250140)

[SR.2.5 - Capture System Element(s) and Interface Requirements 28](#_Toc392250141)

[SR.2.6 - Verify and obtain Work Team (WT) agreement on the System and System Elements Requirements Specifications 29](#_Toc392250142)

[SR.2.7 - Validate that System Requirements Specification satisfies Stakeholders Requirements Specification 31](#_Toc392250143)

[SR.2.8 - Define or update traceability between Requirements 32](#_Toc392250144)

[Roles 34](#_Toc392250145)

[Artefacts 34](#_Toc392250146)

[5. Templates 35](#_Toc392250147)

[6. Example of Lifecycle 37](#_Toc392250148)

[Example 1 of Requirement Practices Lifecycle 37](#_Toc392250149)

[Example 2 of Requirement Practices Flow diagram 37](#_Toc392250150)

[7. Checklist 39](#_Toc392250151)

[Requirement checklist 39](#_Toc392250152)

[Specification Checklist 40](#_Toc392250153)

[8. Tools 41](#_Toc392250154)

[Requirements Management Tool 41](#_Toc392250155)

[Requirements Traceability Tool 46](#_Toc392250156)

[9. References to Other Standards and Models 50](#_Toc392250157)

[ISO 9001:2008 Reference Matrix 50](#_Toc392250158)

[ISO/IEC 15288:2008 Reference Matrix 52](#_Toc392250159)

[CMMI-DEV, V1.3 Reference Matrix 52](#_Toc392250160)

[10. References 54](#_Toc392250161)

[11. Evaluation Form 55](#_Toc392250162)

# 1. Technical Description

## Purpose of this document

A Deployment Package (DP) is a set of artifacts developed to facilitate the implementation of a set of practices in a Very Small Entity (VSE). A DP is not a process reference model (i.e. it is not prescriptive). The elements of a typical DP are: roles and products, description of processes, activities, tasks, template, checklist, reference to standards, etc.

This Deployment Package (DP) supports the Systems Engineering Basic Profile as defined in ISO/IEC TR 29110-5-6-2, the Management and engineering guide [ISO/IEC 29110]. The Basic Profile is one profile of the Generic profile group. The Generic profile group is applicable to VSEs that do not develop critical systems. The Generic profile group is composed of 4 profiles: Entry, Basic, Intermediate and Advanced. The Generic profile group does not imply any specific application domain. The Basic profile is targeted to VSEs working on one project at a time.

The Basic profile is composed of two processes: the Project Management Process and the System Definition and Realization (SR) Process.

The processes, activities and tasks described in this DP are consistent with those listed in ISO/IEC TR 29110 5-6-2 Systems Engineering — Lifecycle Profiles for Very Small Entities (VSEs) — Part 5-6-2: Management and engineering guide – Generic profile group: Basic profile.

The INCOSE Systems Engineering Handbook [INCOSE] has been used as the main reference to develop this DP. The INCOSE Handbook is consistent with ISO/IEC 15288:2008 – *Systems and software engineering – System life cycle processes* [ISO 15288].

Information contained in this DP is applicable to VSEs performing general systems engineering functions using conventional Systems Engineering techniques and tools and a Waterfall system life-cycle model. VSEs performing systems engineering in specialized industries and domains (aerospace, atomic energy, medical instrumentation, government, etc.) should be aware of compliance (process objectives, domain-specific requirements and outcomes/artefacts) requirements specific to their area of work. VSEs applying advanced techniques such as Model-Based Systems Engineering (MBSE), Agile or Lean life-cycle models should refer to the Advanced Requirements Engineering DP (to be developed).

This document is intended to be used by a VSE to establish processes to implement any development approach or methodology including, e.g., agile, evolutionary, incremental, test driven development, etc. based on the organization or project needs of a VSE.

The content of this document is entirely *informative*.

ISO/IEC TR 29110-5-6-2 is available at no cost on the ISO web site at the following URL: <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>

## Why is Systems Engineering important?

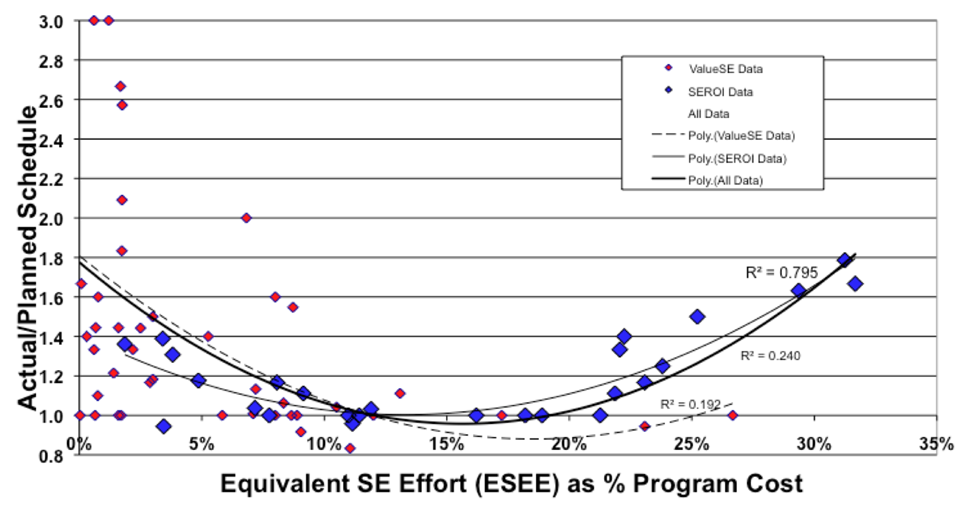
Modern man-made systems, and systems-of-systems, continually increase in complexity. Those systems are developed, more and more, by partnerships involving multiple suppliers and developers and, very often, geographically dispersed teams. This increase in complexity severely challenges organizations’ to achieve project success (i.e. delivering the quality expected by the acquirer and stakeholders, on time and within the established budget. In response to the challenge, governments and industry have fostered the development of the Systems Engineering discipline.

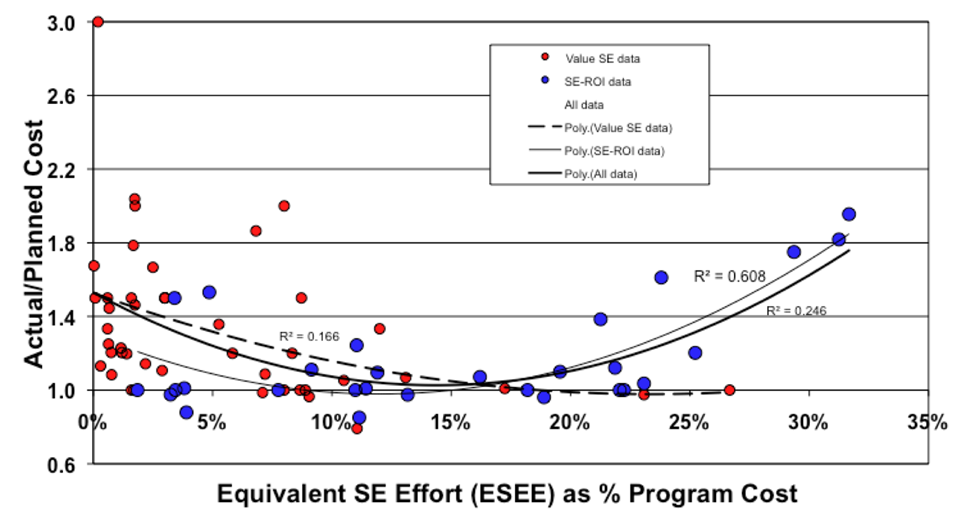
The International Council on Systems Engineering (INCOSE) defines Systems Engineering as[[1]](#footnote-1):

*… an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining Acquirer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem.*

Recent research work performed by Eric Honour[[2]](#footnote-2) as demonstrated that:

1. There is a quantifiable relationship between Systems Engineering effort levels and program success (i.e. meeting the program schedule and within budget), as shown in the graphs below;
2. Systems Engineering has a significant, quantifiable Return on Investment, achieving, optimally, a ROI of 3.5:1;
3. There is an optimum amount of Systems Engineering for best program success, representing an “investment” of 14.4% of the total program cost on Systems Engineering activities.





Increasingly, the responsibility to develop complex systems has flowed down through the supply chain to smaller and smaller organizations. It is not surprising, then that Very Small Entities (VSE) are seeking ways increasingly to achieve program success through the implementation of Systems Engineering best practices.

## Why is Requirement Engineering important?

Requirements Engineering provides a means of defining and communicating the capabilities, conditions and constraints which a system is required to satisfy. To paraphrase Lewis Caroll in Alice in Wonderland[[3]](#footnote-3):

“If you don’t know where you’re going, any road will get you there.”

Requirements are considered to be the cornerstone of the Systems Engineering effort, therefore modern Systems Engineering could not exist without first establishing quality requirements. In his Systems Engineering ROI paper, Eric Honour found an optimal distribution of the Systems Engineering budget over the life-cycle of a project, as shown in the figure below.



In the graph, Requirements Engineering activities[[4]](#footnote-4) are represented by two areas:

* Mission/Purpose Definition (MD), by which Stakeholder Requirements are elicited and developed; and
* Requirements Engineering (RE), by which System and System Elements Requirements are elicited and developed.

Whereas the early emphasis of the Requirements-related activities focused on technical (i.e. System and System Element) requirements, recent research and experience indicate that the most important requirements (i.e. those driving system quality and system success) are Stakeholders requirements expressing their needs and expectations.

Depending on the complexity of the system under development, the optimal combined MD/RE level-of-effort should represent, according to Eric Honour’s SE-ROI paper, approximately 1.5% to 2% of the overall program budget. The optimal level of MD activities peaks at approximately 2.5% of the overall leading to the System Requirement Review (SRR) milestone, whereas the RE activities peak at approximately 1.8% of the overall effort leading to the System Design Review (SDR) milestone.

## Characteristics of Good Requirements

In defining requirements, care should be exercised to ensure each is crafted correctly. ISO/IEC/IEEE 29148:2011 defines a well written requirement as one possessing the following characteristics[[5]](#footnote-5):

1. *Necessary* – The requirement defines an essential capability, characteristic, constraint, and/or quality factor. If it is removed or deleted, a deficiency will exist, which cannot be fulfilled by other capabilities of the product or process. The requirement is currently applicable and has not been made obsolete by the passage of time. Requirements with planned expiration dates or applicability dates are clearly identified.
2. Singular - The requirement statement includes only one requirement with no use of conjunctions.
3. *Implementation Independent* –The requirement, while addressing what is necessary and sufficient in the system, avoids placing unnecessary constraints on the architectural design. The objective is to be implementation independent. The requirement states what is required, not how the requirement should be met.
4. *Clear and Concise* –Is the requirement clear and concise? Is there only one possible interpretation of the statement? Are the terms defined? A glossary should be used to precisely define often‐used terms or terms, such as “process,” that could have multiple interpretations. Is the requirement written in the active (vs passive) form? Is the use of synonyms or aliases avoided? Since many readers master English only as a second language, the terms and syntax used must be simple, clear and exact.
5. *Complete* – The stated requirement needs no further amplification because it is measurable and sufficiently describes the capability and characteristics to meet the parent need.
6. *Consistent* – The requirement is free of conflicts with other requirements.
7. *Feasible* – The requirement is technically achievable, does not require major technology advances, and fits within system constraints (e.g., cost, schedule, technical, legal, regulatory, etc.) with acceptable risk.
8. *Traceable* – Do all requirements trace to the higher level specification and/or user need? Are there requirements at the higher level not allocated (or allocated, but not picked up)? Those with no allocation may be satisfied at that level of the specification. Requirements with either deficiency should be corrected.
9. *Verifiable* – Each requirement must be verified at some level by one of the four standard methods (inspection, analysis, demonstration, or test). An Acquirer may specify, “The range shall be as long as possible.” This is a valid but unverifiable requirement. This type of requirement is a signal that a trade study is needed to establish a verifiable maximum range requirement. Each verification requirement should be verifiable by a single method. When the system hierarchy is properly designed, each level of specification has a corresponding level of verification during the verification stage. If element specifications are required to appropriately specify the system, element verification should be performed.

## Characteristics of Good Specifications

Requirements are never generated individually. When taken as a set, requirements are expected to possess certain characteristics of its own[[6]](#footnote-6):

1. Complete. The set of requirements needs no further amplification because it contains everything pertinent to the definition of the system or system element being specified. In addition, the set contains no To Be Defined (TBD), To Be Specified (TBS), or To Be Resolved (TBR) clauses. Resolution of the TBx designations may be iterative and there is an acceptable timeframe for TBx items, determined by risks and dependencies.
2. Consistent. The set of requirements does not have individual requirements which are contradictory. Requirements are not duplicated. The same term is used for the same item in all requirements.
3. Affordable. The complete set of requirements can be satisfied by a solution that is obtainable/feasible within life cycle constraints (e.g., cost, schedule, technical, legal, regulatory).
4. Bounded. The set of requirements maintains the identified scope for the intended solution without increasing beyond what is needed to satisfy stakeholder needs.

The Appendices to this document propose a criterion by which requirements should be assessed, individually and collectively. The criterion is taken from the INCOSE Guide to Writing Requirements.

# 2. Definitions

This section proposes two sets of definitions. The first set defines the terms used in all Deployment Packages, i.e. generic terms. The second set of terms used in this Deployment package, i.e. specific terms.

## Generic Terms

***Activity:*** a set of cohesive tasks of a process [ISO/IEC 15288].

***Artefact:*** information, which is not listed in ISO/IEC 29110 Part 5, but can help a VSE during the execution of a project.

***Process:*** set of interrelated or interacting activities which transform inputs into outputs [ISO/IEC 15288].

***Product:*** An artefact that is produced, is concrete, and can be either an end item in itself or a component item.

***Role***: a defined function to be performed by a project team member, such as testing, filing, inspecting, coding. [ISO/IEC 24765]

***Step:*** In a deployment package, a taskis decomposed in a sequence of steps.

***Sub-Task:*** When a task is complex, it is divided into sub-tasks.

***System:*** combination of interacting elements organized to achieve one or more stated purposes [ISO/IEC 15288]

NOTE 1 A system may be considered as a product or as the services it provides.

NOTE 2 In practice, the interpretation of its meaning is frequently clarified by the use of an associative noun, e.g. aircraft system. Alternatively, the word “system” may be substituted simply by a context-dependent synonym, e.g., aircraft, though this may then obscure a system principles perspective

Design Constraints coming from the Acquirer/stakeholders.

***Task:*** required, recommended, or permissible action, intended to contribute to the achievement of one or more outcomes of a process[ISO/IEC 15288].

## Specific Terms

***Condition:*** measurable qualitative or quantitative attribute that is stipulated for a requirement [ISO/IEC/IEEE 29148:2011]

***Constraint***: externally imposed limitation on system requirements, design, or implementation or on the process used to develop or modify a system [ISO/IEC/IEEE 29148:2011]

***Mode:*** Set of related features or functional capabilities of a product. [IEEE Std 1362-1998]

Non Functional Requirement: a system requirement that describes not what the System will do but how the System will do it. ISO/IEC 24765, Systems and Software Engineering Vocabulary. Synonym: design constraint. See also: functional requirement. NOTE for example, software performance requirements, software external interface requirements, software design constraints, and quality attributes. Non-functional requirements are sometimes difficult to test, so they are usually evaluated subjectively. [ISO/IEC24765]

***Operational concept:*** verbal and graphic statement of an organization's assumptions or intent in regard to an operation or series of operations of a system or a related set of systems [ANSI/AIAA G-043-1992]

***Operational scenario:*** description of an imagined sequence of events that includes the interaction of the product or service with its environment and users, as well as interaction among its product or service components [ISO/IEC/IEEE 29148:2011]

***Requirement:*** statement which translates or expresses a need and its associated constraints and conditions. [ISO/IEC/IEEE 29148:2011]

Requirements analysis: The process of studying user needs to arrive at a definition of system, hardware, or software requirements. [ISO/IEC 29148]

***Requirements elicitation***: process through which the acquirer and the suppliers of a system discover, review, articulate, understand, and document the requirements on the system and the life cycle processes. [ISO/IEC/IEEE 29148:2011, adapted from ISO/IEC/IEEE 24765:2010]

***Requirements engineering***: Interdisciplinary function that mediates between the domains of the acquirer and supplier to establish and maintain the requirements to be met by the system, software or service of interest. NOTE Requirements engineering is concerned with discovering, eliciting, developing, analysing, determining verification methods, validating, communicating, documenting, and managing requirements. [ISO/IEC/IEEE 29148:2011]

Requirements phase: the period of time in the software life cycle during which the requirements for a software product are defined and documented. [ISO/IEC 24765]

Requirements specification: a document containing any combination of recommendations, requirements or regulations to be met by a system. [ISO/IEC 24765]

**Requirements traceability matrix:** Table that links requirements to their origin and traces them throughout the project life cycle. This can be done in a simple relational database, a requirements management application or system engineering tool. [ISO/IEC/IEEE 29148:2011]

***Requirements validation***: confirmation by examination that requirements (individually and as a set) define the right system as intended by the stakeholders [ISO/IEC/IEEE 29148:2011, adapted from EIA 632:1999]

***Requirements verification***: confirmation by examination that requirements (individually and as a set) are well formed [ISO/IEC/IEEE 29148:2011, adapted from EIA 632:1999]

Stakeholder: individual or organization having a right, share, claim, or interest in a system or in its possession of characteristics that meet their needs and expectations. [ISO/IEC 15288:2008]

Stakeholder Requirements Specifications (StRS): document which captures the System Operational Concept. The StRS may be written by one or more representatives of the supplier, one or more representatives of the Acquirer, or by both. [ISO/IEC/IEEE 29148:2011] NOTE: In the Basic Profile, it is assumed the StRS is prepared by the Acquirer and provided to the Supplier either before or during the Requirements phase.

***State***: condition that characterizes the behaviour of a function/subfunction or element at a point in time [ISO/IEC 26702]

System Requirements Specifications (SyRS): document which captures the structured collection of requirements (functions, performance, design constraints, and attributes) of the system to develop and its external interfaces

Well-formed requirement: a requirement statement that: 1) can be verified; 2) has to be met or possessed by a system to solve a stakeholder problem or to achieve a stakeholder objective; 3) is qualified by measurable conditions and bounded by constraints; and 4) defines the performance of the system when used by a specific stakeholder or the corresponding capability of the system, but not a capability of the user, operator, or other stakeholder. [ISO/IEC/IEEE 29148:2011]

# 3. Relationships with ISO/IEC-29110

This deployment package covers the activities related to requirements analysis of the ISO Technical Report ISO/IEC 29110 Part 5-6-2 for Very Small Entities (VSEs) – Basic Profile [ISO/IEC29110].

The concepts promoted in this Deployment Package are drawn from the ISO/IEC/IEEE 49128:2011 standard (Systems and software engineering — Life cycle processes — Requirements engineering).

In this section, the reader will find a list of applicable System Definition and Realization (SR) process, activities, tasks and roles that are directly related to this topic. The Task List will help the Project Manager scope and plan the System Requirements Engineering effort.

At the Basic Profile level, an organization will be expected to work from an existing textual Acquirer/stakeholder requirements documentation set. The bulk of the effort defined hereafter is meant to parse through the documentation/material, extract system requirements, ensure they are stated such as to be “well-formed” requirement statements, meet the quality criteria and are traceable to the source document or material.

The accomplishment of the tasks defined in this Basic Profile will also allow a VSE to fulfill all Performance and Enabling Objectives of the CMMI for Development (CMMI-DEV), Version 1.3 Requirements Management (REQM) Process Area, a requirement towards the achievement of CMMI Level 2 Process Maturity or Capability.

The development of a Stakeholder Requirements Document and the analysis and development of text-based system requirements from a non-existent set of documents or materials will be covered in the Intermediate Profile.

The definition of stakeholder or system requirements using model-based techniques will be covered in the Intermediate or Advanced Profile.

The following Figure shows the relationship between the System Requirements Engineering Tasks of the Basic Profile and the other System Engineering Activities defined in ISO/IEC 29110 Part 5-6-2, including:

* Project Management;
* Functional & Physical Architecture;
* Development; and
* Configuration Management.

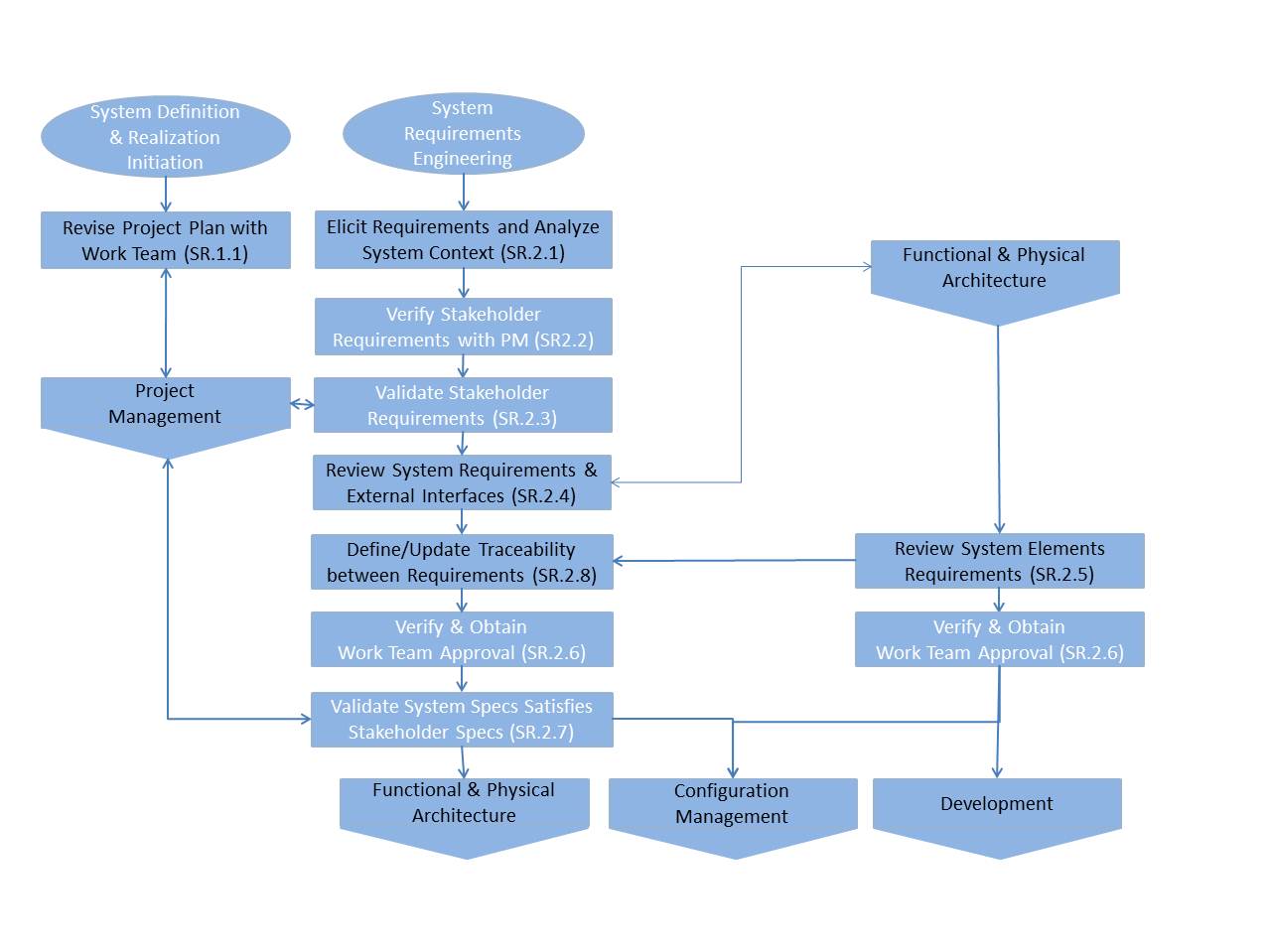


Figure 6 - System Requirements Engineering Tasks of the Basic Profile

* **Process: System Definition and Realization**
* **Activity: SR.1 System Definition and Realization Initiation**
* **Tasks and Roles:**

| **Task List** | **Role** |
| --- | --- |
| **SR.1.1 Revise the Project Plan with the Work Team**  In *order to achieve a common understanding and get their engagement with the project.* | TL  WT |

* **Process: System Definition and Realization**
* **Activity: SR.2 System Requirements Engineering**
* **Tasks and Roles:**

| **Task List** | **Role** |
| --- | --- |
| **SR.2.1 Elicit Acquirer/Stakeholders Requirements and Analyze System Context**  Identify and consult information sources (Acquirer, users, stakeholders, previous systems, documents, etc.): Statement of Work (SOW), Concept Documents, previous system description, etc.  Analyze the context of use of the system with Acquirer and other stakeholders:   * Review the stakeholders * Review the concepts of use of the system; * Review scenarios, business processes   Identify and analyze requirements to   * Determine the scope and system boundary; * If applicable, identify the strengths and the weaknesses of the previous system; * Ensure that the Stakeholder requirements are complete and consistent; * Elicit missing Stakeholder requirements;   Resolve conflicting, duplicate and out-of-scope Stakeholder requirements.  Update the Stakeholders Requirements Specification. | SYS  ACQ  STK |
| **SR.2.2 Verify the *Stakeholders Requirements Specifications* with PM**  Obtain Work Team agreement on the *Stakeholder Requirements Specification* | PM  SYS  WT |
| **SR.2.3 Validate the Stakeholders Requirements Specification with the Acquirer and Stakeholders**  Obtain Acquirer and Stakeholder agreement on the Stakeholder Requirements Specification | PM  SYS  ACQ  STK |
| **SR.2.4 Capture System Requirements and Interfaces**  - Review the system boundary.  - Capture the System Requirements, system design constraints  - Capture the interface requirements between the System and its external environment  - Release a draft *System Requirements Specification for review* | SYS  DES |
| **SR.2.5 Capture System Element(s) and Interface Requirements**  Note: System Element requirements are generally elaborated in parallel with the System Logical and Physical Architectural Design Activity (see Activities SR.3.1 and SR.3.3)  Review the allocation of System requirements to System Elements in conformity with the functional and physical architecture and decompose System requirements so that System Element requirements are distinctively and clearly defined. Review System element requirements derived from the System architectural design but that cannot be traced to a specific parent System requirement  Refine as necessary external interface requirements and identify internal interface requirements between System Elements.  Update a System Element Requirements Specification for each System Element defined in the System Design Document.  Note: interface requirements are included in System Elements Requirements Specifications. Separate specification document can be established.  Note: System elements requirements become needs and expectation in input of the system elements implementation. | SYS  DES |
| **SR.2.6 Verify and obtain Work Team (WT) agreement on the System and System Elements Requirements Specifications**  Ensure with WT that requirements are SMART. In particular   * are precise, concise, non-ambiguous * are consistent * are properly traced * can be implemented * can be verified and validated * fall within project cost and schedule constraints (PM)   The results found are documented in a Verification Report and corrections are made until the document is approved by PM. If documents are under configuration, identify and characterize the impact of the change and initiate if necessary (i.e. change approved) a Change Request. | PM  WT |
| **SR.2.7 Validate that System Requirements Specification satisfies Stakeholders Requirements Specifications.**  For the Acquirer and Stakeholders to verify and validate the System requirements, resolve any issues raised during the review and validate assumptions made by the Team Members during the internal review.  Obtain System Requirements buy-in from the Acquirer. | PM  SYS  ACQ  STK |
| **SR.2.8 Define or update traceability between Requirements**  According to the data model defined in SR.1.2, at each level of decomposition of the system, define or update traceability between   * System requirements, interface requirements and their parent Stakeholder requirements * System elements requirements, interface requirements and their parent system requirements. | SYS  DES |

# 4. Description of Processes, Activities, Tasks, Steps, Roles and Products

**This Section expands the activities identified in the previous section. It provides the following information:**

* **Activity objective**
* **Rationale behind activity**
* **Roles involved in the execution of the tasks**
* **Artefact(s) produced at the end of the activity**
* **Task Steps**
* **Task Step Descriptions**

**Process: System Definition and Realization**

**Activities:** SR.1 System Definition and Realization Initiation

SR.2 System Requirements Engineering

## SR.1.1 – Review Project Plan with the Work Team members

|  |  |
| --- | --- |
|  | |
| ***Objectives:*** | The objective of this activity is to review, and revise if necessary, the Project Plan with the Work Team as it relates to Requirements Engineering tasks, scope of the effort, the tailoring of tasks, as required, and task assignments. |
| ***Rationale:*** | It is important that all members of the Work Team clearly understand the planned effort and operate “from the same sheet of music”. |
| ***Roles:*** | Team Leader (TL) |
| Work Team (WT) |
| ***Artefacts:*** | Revised Project Plan, as required |
| ***Steps:*** | 1. Assemble Work Team |
| 1. Assign Tasks |
| 1. Distribute Project Plan |
| 1. Hold Requirements Analysis kick-off meeting |
| 1. Identify Needed Changes to Project Plan |
| 1. Update and Release Updated Project Plan |
| ***Step Description:*** | ***Step 1. Assemble Work Team***  The Team Leader identifies the member of the Requirements Engineering Team and their respective roles based on the Project Plan.  ***Step 2. Assign Tasks***  The Team Leader assigns each Requirements Engineering role to one or more team member, as required.  ***Step 3. Distribute Project Plan***  The Team Leader distributes the Project Plan to the Work Team for everyone to review the scope of the planned effort, whatever tailoring might have been planned and their respective role within the team.  ***Step 4. Hold Requirements Analysis Kick-off Meeting***  The Team Leader convenes and holds the Requirements Analysis Kick-off to review the Project Plan, task assignments and raise any issues for the Work Team as to needed revisions to the Project Plan.  ***Step 5. Review Project Plan Changes***  The Team Leader reviews proposed changes and issues raised during the Requirements Analysis Kick-off and analyses possible Project Plan changes.  **Step 6. Update and Release Updated Project Plan**  The Team Leader incorporates needed changes to the Project Plan and releases the updated plan. |

## SR.2.1 - Elicit Acquirer and other stakeholders requirements and analyze system context

|  |  |
| --- | --- |
|  | |
| ***Objectives:*** | The main objective of this activity is to:   1. gain a comprehensive view of the Acquirer and Stakeholder needs; 2. clearly define the scope of the project; and 3. Elicit and analyze Acquirer and stakeholder requirements source material. |
| ***Rationale:*** | It is important to clearly define the project scope (boundaries), identify all sources of requirements and to identify key requirements of the future system with the stakeholders to avoid problems like forgotten key functionalities/characteristics or requirements creep. |
| ***Roles:*** | Systems Engineer (SYS) |
| Acquirer (ACQ) |
| Stakeholders (STK) |
| ***Artefacts:*** | Stakeholder Requirements Specification (StRS) placed under configuration control |
| ***Steps:*** | 1. Identify Stakeholders |
| 1. Collect information about the project and application domain. |
| 1. Define project’s scope |
| 1. Identify and capture Stakeholders’ source requirements |
| 1. Structure and prioritize Stakeholders’ requirements documents/materials |
| ***Step Description:*** | ***Step 1. Identify Stakeholders***  The Systems Engineer, with the assistance of the Project Manager and the Acquirer, identifies and establishes contact with the project’s stakeholders. This can occur in a single meeting, a series of meeting or through individual, one-on-one meetings.  Example of Stakeholders  Final users or final users representative  Community of users  Acquirers or Acquirers representative  Executive board  Regulatory agencies (national, regional, local)  Standardization agencies (international, national, regional)  Employee unions  Partners  Testers  Manufacturers  Distributors  Logistics specialists  Maintenance, Repair and Overhaul organization  Environmental protection organizations  ***Step 2. Collect information about the domain***:  The Systems Engineer captures the key concepts of the business domain of the Stakeholders. The Project Manager, Acquirer and Stakeholders assist the Systems Engineer by providing all the information (existing documentation or explanation) that will facilitate this understanding.  ***Step 3. Review project’s scope***  Review the project objectives and priorities as defined by the PM. Review system boundaries and context as identified by the F&PA DP.  ***Step 4. Identify and capture source requirements***  Having in mind key concepts related to the Stakeholder business domain, the Systems Engineer starts identifying requirements. None of the situations in projects are identical. In some cases, most of the requirements are already identified in a document (call for tender in case of fixed priced projects). However, in many cases requirements are implicit and not written. Some requirements can contradict themselves between Stakeholders.  The Systems Engineer identifies and lists the key requirements of the system to be built. During this Step, the Systems Engineer should not start detailing identified requirements. The main goal is to gain a comprehensive view of the needs.  The Systems Engineer captures existing requirements source documents, placed under configuration control by the Team Leader or Project Manager. Each requirement/statement/paragraph may be assigned a project-unique identifier at this step to facilitate the creation of traceability links later on. The Stakeholder source documents are typically not reformatted in any way, so as to facilitate communications with the Acquirer and Stakeholders as well the future updating of the document if changes are released. |

## SR.2.2 - Review Stakeholders Requirements Specifications with PM

|  |  |
| --- | --- |
|  | |
| ***Objectives:*** | Obtain Work Team consensus agreement on the Stakeholder Requirements Specifications. Identify issues that need to be resolved with the Acquirer and Stakeholders. Prepare for the formal review with the Acquirer/Stakeholders. |
| ***Rationale:*** | Prior to seeking approval from the Acquirer/Stakeholders, it is critical that all internal work team members review the Stakeholder Requirements and have a chance to expose their issues, assumptions and concerns with regards to the requirements before they are agreed with the Acquirer and baselined. |
| ***Roles:*** | Project Manager (PM) |
| System Engineer (SYS) |
| Work Team (WT) |
| ***Artefacts:*** | Stakeholder Requirements Database or Stakeholder Requirements Specification (StRS) |
| Requirements issues |
| ***Steps:*** | 1. Review Stakeholder Requirements Specification (StRS) |
| 1. Identify requirements issues |
| 1. Hold internal requirements review meeting |
| ***Step Description:*** | ***Step 1. Review Stakeholder Requirements Specification:***  The reviewers verify that the requirements, taken individually and as a set, possess the characteristics of well written stakeholder requirements.  ***Step 2. Identify Requirements Issues:***  The Team Members review stakeholder requirements in order to detect those that do not possess the desired characteristics. Assumptions about requirements that need to be verified with the Acquirer or Stakeholders are identified and noted.  ***Step 3. Hold Internal Requirements Review Meeting:***  The Project Manager convene a review meeting to go over the findings from the internal review, resolve internal issues and prepare external issues for the external review with the Acquirer/Stakeholders. |

## SR.2.3 - Baseline Stakeholders Requirements Specification with the Acquirer and Stakeholders

|  |  |
| --- | --- |
|  | |
| ***Objectives:*** | Obtain Acquirer and Stakeholder agreement on the Stakeholder Requirements Specification. |
| ***Rationale:*** | It is essential that the Acquirer and Stakeholders review the Stakeholder requirements, resolve issues raised by the Work Team and have a chance to expose their own issues, assumptions and concerns with regards to the Stakeholder Requirements.  Once agreed upon and approved, the Stakeholder Requirements are placed under configuration control and become the basis upon which System Requirements can be developed and the formal validation cornerstone of the System development life-cycle. |
| ***Roles:*** | Project Manager (PM) |
| Systems Engineer (SYS) |
| Acquirer (ACQ) |
| Stakeholders (STK) |
| Work Team (WT) Members (as required) |
| ***Artefacts:*** | Approved Stakeholder Requirements Database or Stakeholder Requirements Specification (StRS) |
| Stakeholder Requirements Review Meeting Minutes |
| ***Steps:*** | 1. Convene Stakeholder Requirements Review Meeting |
| 1. Hold Stakeholder Requirements Review Meeting |
| 1. Publish Meeting Minutes |
| 1. Dispose Issues and Action Items |
| ***Step Description:*** | ***Step 1. Convene Stakeholder Requirements Review Meeting:***  The Project Manager issues the Stakeholder Requirements Review meeting invitation and agenda. The internally approved Stakeholder Requirements Specification is issued to all invitees with sufficient lead-time to allow them to review the Stakeholder requirements and those raised during the Internal Stakeholder Requirements Review.  ***Step 2. Hold Stakeholder Requirements Review meeting:***  The Project Manager, System Engineer, Acquirer, Stakeholders and Team Members as required, meet to review all issues raised both internally during the internal requirements review and through the Acquirer’s and stakeholder’s own review of the requirements.  ***Step 3. Publish Review Meeting Minutes:***  The Project Manager collates and publishes the Minutes of the Stakeholder Requirements Review Meeting.  ***Step 4. Dispose Issues Resolution and Action Items:***  The Project Manager plans for the disposition of issues resolution and action items raised during the Stakeholder Requirements Review meeting, allocate disposition actions to the Team Members and monitor the progress. A target date or milestone for resolution of issues and action items is established.  NOTE: It is not necessary that all issues and action items be disposed prior to the start of the next phase of the system life-cycle. Requirement issues or actions that cannot be resolved before the next phase of activities should be reviewed as to the risk the delay may present to the project. The results of the risk analysis will be merged into the project’s risk management activities. |

## SR.2.4 - Capture System Requirements and Interfaces

|  |  |
| --- | --- |
|  | |
| ***Objectives:*** | The objective of this Step is to:  - Review the system boundary.  - Capture the System Requirements, system design constraints  - Capture the interface requirements between the System and its external environment  - Release a draft *System Requirements Specification for review* |
| ***Rationale:*** | The Systems Engineers needs to establish a clear and complete definition of the System Requirements. |
| ***Roles:*** | Systems Engineer (SYS) |
| Designers (DES) |
| ***Artefacts:*** | System Requirements Database |
| System Requirements Specification (SyRS) |
| System prototype (optional) |
| ***Steps:*** | 1. Produce a prototype (optional) |
| 1. Capture System requirements |
| 1. Prepare draft System Requirements Specification (SyRS) |
| ***Step Description:*** | ***Step 1. Produce demonstrators, prototypes, analysis models or simulations (optional)***  The Systems Engineer and designers can use demonstrators, prototypes, analysis models or simulations to facilitate the comprehension and scoping of the stakeholder needs and system requirements (i.e. Acquirer/Stakeholder and development team side). A prototype may be used to implement only the subset of the intended system functionality which represents the highest risk or is the most poorly understood. It must be understood that these mechanisms **ARE NOT** the intended solution.  ***Step 2. Capture system requirements:***  The System Engineer extracts system requirement statements from the source documentation, stakeholder interview notes or as the result of the prototype production and evaluation. Typically, the source material/document paragraphs consist of many sentences and statements, not all of which are meant to be system requirements. The Systems Engineer may use keywords such as “shall” and “must” in source document sentences to identify potential system requirements. Also, the Systems Engineer sorts Program/Project requirements (i.e. the Supplier shall…) from technical requirements (i.e. the System shall…) and retains only the technical requirements. Program/Project requirements are passed on the Project Manager for him to action as required.  If the source documents include a compliance matrix associated with a Proposal document to the Acquirer, the compliance matrix is used to convert each statement in the document referenced by the compliance matrix into fully compliant requirements (i.e. non-compliances are removed and partial compliances are applied so that only the “compliant” portion of the statement becomes a requirement).  The Systems Engineer interacts with Acquirer representatives, subject matter experts and stakeholders in order to clarify specialty requirements such as human factor engineering issues, whenever relevant (e.g. if a Graphical User Interface must be developed).  ***Step 3. Prepare draft System Requirements Specification:***  The Systems Engineer organizes the system requirements into a draft System Requirements Specification (SyRS) in accordance with the guidelines of ISO/IEC/IEEE 29148, Section 8.3. |

## SR.2.5 - Capture System Element(s) and Interface Requirements

|  |  |
| --- | --- |
|  | |
| ***Objectives:*** | The objective of this Step is to allocate the System requirements to System Elements and decompose the system requirement statement so as to clearly scope each respective System Element’s contribution to the fulfillment of the parent System requirement. |
| ***Rationale:*** | It is important to ensure that:   1. the system requirement be clearly and fully satisfied; and 2. that the interface between related System Elements are scoped. |
| ***Roles:*** | Systems Engineer (SYS) |
| Designers (DES) |
| ***Artefacts:*** | System Requirements Database, updated as required |
| System Requirements Specification (SyRS), update as required |
| Draft System Elements Requirements Database |
| Draft System Elements Requirements Specification |
| ***Steps:*** | 1. Allocate/Decompose System Requirements to System Elements |
| 1. Write System Elements requirements |
| 1. Prepare draft System Elements requirements specification document |
| ***Step Description:*** | ***Step 1. Allocate/Decompose System Requirements to System Elements***  The Systems Engineer uses the Functional/Physical Architecture to:   1. allocate system requirements to the System Elements. This step may require that feedback be provided to the Functional/Physical Architecture activity in order to adjust the architecture as issues are identified, analysed and resolved; 2. define System Element requirements derived from the Functional/Physical Architecture that are not traceable to a parent System requirement. The rationale for the derived requirements are documented.   ***Step 2. Write System Elements requirements:***  The Systems Engineer decomposes the system requirement statement into preliminary System Element Requirements statements. The Designer participates in the System Element requirements definition to ensure it meets the scope and understanding of the System Element role.  ***Step 3. Prepare Draft System elements Requirements Specifications:***  The Systems Engineer organizes the System Elements requirements into specifications in accordance with the guidelines of ISO/IEC/IEEE 29148, Section 8.3.  If the System Element is a software entity, this Step can be substituted by the VSE Software Life-cycle Software Requirements Specification preparation activity. |

## SR.2.6 - Verify and obtain Work Team (WT) agreement on the System and System Elements Requirements Specifications

|  |  |
| --- | --- |
|  | |
| ***Objectives:*** | For the Work Team members to review and approve System and System Element requirements as well as interface requirements between System Elements ; and  Obtain System and System Element Requirements buy-in from the Work Team. |
| ***Rationale:*** | Prior to seeking approval from the Acquirer/Stakeholders, it is critical that all internal team players review the requirements and have a chance to expose their issues, assumptions and concerns with regards to the requirements before they are agreed with the Acquirer and deposited into the baseline.  **NOTE:** Although this Review is identified as one task, it is typically scheduled to occur in distinct occurrences, held at different times over the life-cycle of the System:   * One System Requirements Review will be held for the System Requirements; and * Multiple System Element Requirements Reviews will be held, one for each of the defined System Elements. |
| ***Roles:*** | Project Manager (PM) |
| Team Leader (TL) |
| System Engineer (SYS) |
| Work Team (WT) |
| ***Artefacts:*** | Requirements Database or Specification (System or System Element) |
| Review Meeting Minutes |
| Requirements issues |
| ***Steps:*** | 1. Review System/System Elements Requirements Specification |
| 1. Identify requirements issues |
| 1. Hold Requirements Review meeting |
|  | 1. Publish Review Meeting Minutes |
|  | 1. Action and Resolve Raised Issues |
| ***Step Description:*** | ***Step 1. Review System/System Elements Requirements Specification:***  The reviewers verify that the System or System Element requirements, individually and as a set, satisfy the criterion defining well written requirements.  ***Step 2. Identify Requirements Issues:***  The Systems Engineer and Designers document issues arising from the review. Assumptions about requirements that need to be verified are also captured.  ***Step 3. Hold Requirements Review Meeting:***  The Project Manager convenes a review meeting to go over the findings from the review, identify and assign action items and prepare external issues for a review with the Acquirer or Stakeholders.  ***Step 4. Publish Review Meeting Minutes:***  The Project Manager collates and publishes the Minutes of the System or System Element Requirements Review Meeting.  ***Step 5. Action and Resolve Action Items:***  The Project Manager plans for the disposition of issues resolution and action items raised during the System or System Elements Requirements Review meeting, allocates actions to the Team Members and monitor the progress. A target date or milestone for resolution of issues and action items is established.  NOTE: It is not necessary that all issues and action items be disposed prior to the start of the next phase of the system life-cycle. Requirement issues or actions that cannot be resolved before the next phase of activities should be reviewed as to the risk the delay may present to the project. The results of the risk analysis will be merged into the project’s risk management activities. |

## SR.2.7 - Validate that System Requirements Specification satisfies Stakeholders Requirements Specification

|  |  |
| --- | --- |
|  | |
| ***Objectives:*** | For the Acquirer and stakeholders to verify and validate that System Requirements satisfy the Stakeholder needs, resolve any issues raised during the review and validate assumptions made by the Team Members during the internal review.  Obtain System Requirements buy-in from the Acquirer. |
| ***Rationale:*** | It is essential that the Acquirer and Stakeholders review the System requirements and have a chance to expose their issues, assumptions and concerns with regards to those requirements.  Once agreed upon and approved, the System Requirements placed under configuration control become the formal cornerstone of the System life-cycle. |
| ***Roles:*** | Project Manager (PM) |
| Acquirer (ACQ) |
| Stakeholders (STK) |
| Systems Engineer (SYS) |
| Work Team (WT) Members (as required) |
| ***Artefacts:*** | Requirements Database or Stakeholder and System Requirements Specifications |
| System Requirements Review Meeting Minutes |
| ***Steps:*** | 1. Convene System Requirements Review Meeting |
| 1. Hold System Requirements Review Meeting |
| 1. Dispose Issues and Action Items |
| ***Step Description:*** | ***Step 1. Convene System Requirements Review Meeting:***  The Project Manager issues the System Requirements Review meeting invitation and agenda. The draft System Requirements Specification is issued to all invitees with sufficient lead-time to allow them to review the System requirements and observations raised during the Internal System Requirements Review.  ***Step 2. Hold System Requirements Review meeting:***  The Project Manager, System Engineer, Team Leader, Team Members as required, Acquirer and Stakeholders as required meet to review all issues raised both internally during the internal System requirements review and through the Acquirer’s own review of the requirements.  ***Step 3. Dispose Issues Resolution and Action Items:***  The Team Leader and Project Manager plan for the disposition of issues resolution and action items raised during the Systems Requirements Review meeting, allocate disposition actions to the Team Members and monitor the progress. A target date or milestone for resolution of issues and action items is established.  NOTE: It is not necessary that all issues and action items be disposed prior to the start of the next phase of the system life-cycle. Any system requirements issue or action that cannot be resolved before the Functional/Physical Architecture activities begin should be reviewed as to the risk the delay may present to the project. The results of the risk analysis will be merged into the project’s risk management activities. |

## SR.2.8 - Define or update traceability between Requirements

|  |  |
| --- | --- |
|  | |
| ***Objectives:*** | According to the System architecture model defined in the Functional & Physical Architecture Activity, at each level of decomposition of the System, define or update traceability between  • System requirements, interface requirements and their parent Stakeholder requirements  • System elements requirements, interface requirements and their parent System requirements. |
| ***Rationale:*** | The establishment and maintenance of traceability between the Stakeholder, System and System Element Requirements is critical in keeping the life-cycle evolution of the system under control. |
| ***Roles:*** | Systems Engineer (SYS) |
| Designers (DES) |
| ***Artefacts:*** | Traceability Matrices |
| ***Steps:*** | 1. Establish bi-directional Stakeholder Requirements to source/reference material 2. Establish bi-directional System to Stakeholder Requirements Traceability Links 3. Establish bi-directional System Element to System Traceability Links |
| ***Step Description:*** | ***Step 1. Establish System to Stakeholder Requirements Traceability Links:***  The Systems Engineer creates traceability links between the captured Systems Requirements and their parent Stakeholder Requirements. By their nature, derived System Requirements are not traced to Stakeholder Requirements. The Systems Engineer ensure the rationale for derived requirements is captured.  ***Step 2. Establish System Element to System Requirements Traceability Links:***  The Designer creates traceability links between the captured System Elements Requirements and their parent System Requirements. By their nature, derived System Element Requirements are not traced to System Requirements. The Systems Engineer and Designed ensure the rationale for derived requirements is captured and is consistent with the defined System Architecture. |

## Roles

|  |  |
| --- | --- |
| **Role** | **Definition** |
| **Systems Engineer (SYS)** | Person in charge within the development team to gather, analyse and manage the requirements related to the System to be developed. |
| **Acquirer (ACQ)** | Person in charge within the Acquirer side to transfer and validate requirements to the development team. It can be the Acquirer or any representative. |
| **Stakeholder (STK)** | An individual or organisation that will be directly or indirectly affected by the use or deployment of the System. |
| **Designer (DES)** | Person(s) in charge of the design of the System or the System Elements. |
| **Reviewer** | Person reviewing one or more System Requirements Engineering artefacts. |
| **Project Manager (PM)** | Person in charge of managing the project (cost, schedule, tasks, contract…) |
| **Work Team (WT)** | All technical and specialists assigned to perform systems engineering, design engineering or specialty engineering (Safety, Human Factors, Reliability & Maintainability, etc.) activities on the project |

Table 1 - Definitions of Roles

## Artefacts

|  |  |
| --- | --- |
| **Artefacts** | **Definition** |
| Requirements Database | Repository of the project’s System and Sub-system requirements. |
| Requirements Specification | Document in which all identified requirements are grouped and structured. Three (3) types of requirements documents are produced in this DP: Stakeholder Requirements Specification (StRS), *System Requirements Specification (SyRS)* and System Element Requirements Specification (SeRS). |
| Requirements Review Meeting Minutes | Document compiling the discussion, observations, issues and action items raised during the requirements review. |

Table 2 Definitions of Artefacts

# 5. Templates

The templates provided with this deployment package should be customized for each project.

SyRS Template Table of Content

Adapted from ISO/IEC/IEEE 29148:2011, Section 9.4

**1. Introduction**

1.1 Purpose

1.2 Document conventions

1.3 Intended audience

1.4 Additional information

1.5 Contact information/SyRS team members

1.6 References

**2. System overview**

2.1 System context

2.2 System functions

2.3 User characteristics

**3. Functional Requirements**

**4. Usability Requirements**

**5. Performance Requirements**

**6. System Interfaces**

**7. System Operations**

7.1 Human system integration requirements

7.2 Maintainability

7.3 Reliability

**8. System modes and states**

**9. Physical characteristics**

9.1 Physical requirements

9.2 Adaptability requirements

**10. Environmental conditions**

**11. System security**

**12. Information management**

**13. Policies and regulations**

**14. System life cycle sustainment**

**15. Packaging, handling, shipping and transportation**

**16. Verification**

**17. Assumptions and dependencies**

**18. Other Requirements**

**Appendix A**: Terminology/Glossary/Definitions list

**Appendix B**: Requirements Traceability Matrix

This matrix traces each requirement to its parent/source requirement or material.

# 6. Example of Lifecycle

***Disclaimer***: *This section provides some graphical representation of example of requirement practices lifecycle. These examples are provided to help the reader to implement his own requirement lifecycle fitting their system project’s context and constraints.*

Example 1 of Requirement Practices Lifecycle

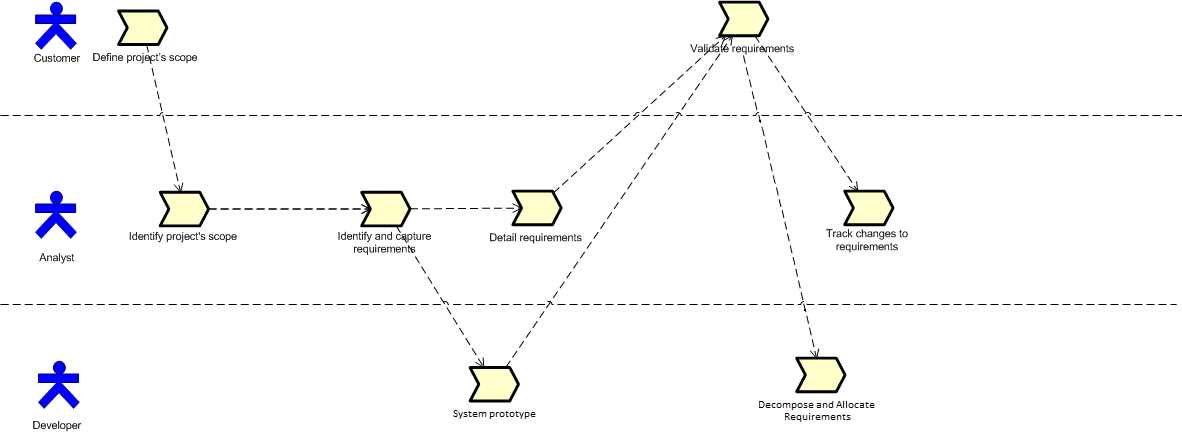


Figure 7 Example 1 of Requirement Practices Lifecycle

Example 2 of Requirement Practices Flow diagram

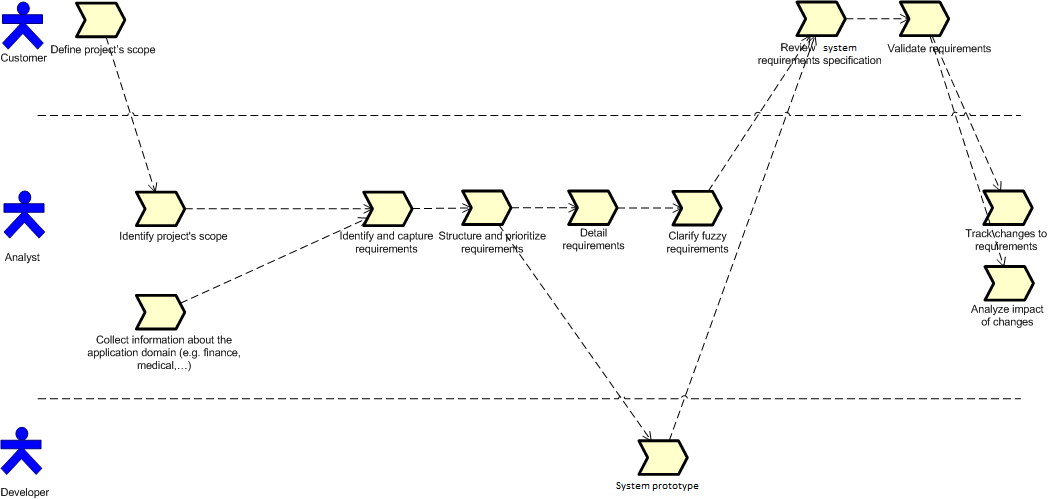


Figure 8 Example 2 of Requirement Practices Lifecycle

# 7. Checklist

## Requirement checklist

The following Checklist, drawn from the INCOSE Guide for Writing Requirements[[7]](#footnote-7), can be used to support a formal Requirements Review. Each requirement should be inspected to ensure they meet the criteria of the checklist.

More thorough criteria definitions are provided in the INCOSE Guide for Writing Requirements. The guide also provides rules and examples to help writers prepare quality requirement statements.

|  |  |
| --- | --- |
| **C1 Necessary** | The requirement statement is necessary. |
| **C2 Implementation Independent** | The requirement states *what* is required, not *how* it is to be satisfied/implemented. |
| **C3 Unambiguous** | The requirement lends itself only to a single interpretation. |
| **C4 Complete** | The requirement statement is complete in and of itself. |
| **C5 Singular** | The requirement statement addresses a single thought or capability/condition/constraint. |
| **C6 Feasible** | The requirement statement expresses a capability/condition/constraint that can be realized/implemented. |
| **C7 Verifiable** | The requirement statement can be verified objectively. |
| **C8 Correct** | The requirement statement is a correct expression of the stakeholder requirement and expectations |
| **C9 Conforming** | The requirement statement conforms to the requirements writing rules and guidelines standards applicable in the organization. |

## Specification Checklist

The following Checklist can be used to support a formal Requirements Review. Each Specification document, or set of related requirements, should be inspected to ensure they meet the criteria of the checklist below.

|  |  |
| --- | --- |
| **C10 Complete** | The set of requirements completely satisfies its parent set of requirements. |
| **C11 Consistent** | The set of requirements forms a coherent whole, void of conflicts and redundancies. |
| **C12 Feasible** | The set of requirements represents a feasible expression of the parent set of requirements. |
| **C13 Bounded** | The set of requirements represents a well-defined and bounded definition of capabilities, conditions and constraints. |
| **C14 Structured** | The set of requirements is structured such that sub-sets of related requirement statements can be identified. |

# 8. Tools

***Disclaimer***: *This section provides some suggestions of requirement management tools suitable for a typical VSE. These suggestions do not represent a formal endorsement of any specific tool. More detailed information about available Requirements Management tools can be found on the INCOSE web site at the following URL: http://www.incose.org/productspubs/products/rmsurvey.aspx*

## Requirements Management Tool

* Objectives:
  + To establish a repository for stakeholder, system or system elements requirements.
  + To ensure that all requirements are uniquely identified.
  + To track the life-cycle status of each requirement.
  + To support an initial allocation of system requirements to system elements.

**Eclipse Requirements Modeling Framework (RMF) and Requirements Engineering Platform (ProR)**

For larger projects, a more capable environment is highly recommended, especially in situations where the VSE is required to interact with a larger organization that uses a commercial requirements engineering tool. In some situations, the larger organization will be willing to exchange with the VSE using the OMG Requirement Interchange Format (ReqIF).

RMF/ProR is one example of an inexpensive Requirements Management tool well-suited for VSEs. It is an Open Source requirements engineering tool that supports the OMG Requirements Interchange Format (ReqIF) Standard natively. It is part of the Eclipse RMF project and is built for extensibility. ProR is the result of the European Union FP7 Research Project Deploy, where it is used to provide traceability between requirements and formal models.

More information on RMF/ProR can be found and the framework, installation instructions, sample projects and tutorials can be downloaded from:

[**http://www.eclipse.org/rmf/pror/**](http://www.eclipse.org/rmf/pror/)

Through the use of ReqIF, ProR/RMF requirements can be exchanged with other commercial Requirements Management tools such as IBM Rational DOORS, PTC Integrity, Atego Exerpt, Requisis ReX and Visure Requirements.

**MS-Office**

For very small projects (less that 10-20 stakeholder requirements, or less than 100-150 system requirements), it is possible to use the MS-Office Suite to capture the requirements and traceability. Templates are provided below that can be pasted into MS-Word or MS-Excel.

**Caution:** While MS-Office applications are often used for capturing requirements directly with the acquirer and stakeholders they are unsuitable to manage requirements and all its properties during the development lifecycle. The burden of generating and maintaining unique identifiers and traceability links, and managing change rest entirely on the shoulders of of the Systems Engineer with little to no tool support.

Template – Basic List of Requirements

To be used in an Excel sheet structured, to capture Stakeholder, System or System Element requirements, for example, as:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *ID* | *Requirement text* | *Compliance* | *Notes* | *Status* | *Priority* | *Verification Method* |
|  |  |  |  |  |  |  |

**ID** (required): Project-unique identifier which allows the unambiguous referencing of a requirement throughout the project. A proposed identifier schema is as follows:

ABCXYZ\_9999 where:

ABC identifies the requirement level, as follows:

STK: Stakeholder Requirement

SYS: System Requirement

SEL: System Element Requirement or, alternately

SRS: Software Requirement

HLR: High-level Software Requirement (DO-178B/C)

LLR: Low-Level Software Requirement (DO-178B/C)

HRS: Hardware Requirement

XYZ identifies the project, system or system element

9999: A sequence number serializing related requirements

Examples:

STKMCS\_0023: Stakeholder Requirement #23 for the Mobile Communications System (MCS)

SYSMCS-0970: System Requirement #970 for the Mobile Communications System (MCS)

SELNRU-2325: System Element Requirements #2325 for the MCS Node Radio – UHF (NRU)

**Requirement text** (required): The requirement statement text. Writing guideline: The statement should be in the form of a single sentence.

* When the statement calls for a compulsory requirement, it will be worded in the “shall” form;
* When the statement calls for a desirable feature, it will be worded in the “should” form.

**Compliance** (optional): The intended compliance status of the System or System Element with regards to the Requirement. The attribute can take one of the following values: Compliant; Partial; or Non-compliant. When the values Partial or Non-compliant are assigned, a rational for the status should be provided in the Notes.

**Notes** (required): This attribute serves to capture the collection of Work Team observations and comments. It is considered good practice to prefix a specific entry with the initials of the writer placed between square brackets.

**Status** (optional): This attribute follows the evolution of the requirement from its first elaboration to its final release. Typical values, aligned with the tasks identified in this DP include:

* Draft: the requirement is in draft form;
* Ready: the requirement is ready for internal review
* Reviewed: the requirement is ready for formal review
* Approved: the requirement has passed final review
* Released: the requirement has been released in the relevant configuration management baseline.

**Priority** (optional): This attribute rates the importance of the requirement in relation to the other requirements in the specification. The possible values can be enumerated (i.e. High, Medium, Low) or numeric (i.e. 1 to 10)

**Verification Method** (required): This attribute can be used to assist in the preliminary verification & validation planning task (See V&V Deployment Package). The process of thinking of a verification or validation approach for the requirement also helps ensuring the requirement is well written. The values typically used include:

* Test (T): the requirement will be verified or validated by the execution of a test procedure characterized by a clearly defined and objective pass/fail criteria, generally related to the condition or constraint stated in the requirement;
* Demonstration (D): the requirement will be verified or validated by the demonstration that the system can perform the function or capability stated in the requirement. In general, the pass/fail criteria will be more subjective than that associated with a Test;
* Inspection (I): the requirement will be verified or validated by a visual inspection of an artefact produced during the development life-cycle, or the system itself, to confirm the characteristic stated in the requirement is satisfied;
* Analysis (A): the requirement will be verified or validated by performing an analysis (e.g. reliability analysis, safety analysis, maintainability analysis, etc.) of a collection of artefacts or data generated during the development life-cycle or specific verification and validation activities.

## Requirements Traceability Tool

* Objectives:
  + To maintain the linkage from the source of each requirement through its decomposition to implementation and test (verification).
  + To ensure that all requirements are addressed and that only what is required is developed.
  + Useful when conducting impact assessments of requirements, design or other configured item changes.

|  |  |
| --- | --- |
| **Guideline** | Requirements traceability should:   * Ensure traceability for each level of decomposition performed on the project. In particular:   + Ensure that every lower level requirement can be traced to a higher level requirement or original source   + Ensure that every design, implementation, and test element can be traced to a requirement   + Ensure that every requirement is represented in design and implementation   + Ensure that every requirement is represented in testing/verification * Ensure that traceability is used in conducting impact assessments of requirements changes on project plans, activities and work products * Be maintained and updated as changes occur. * Be consulted during the preparation of Impact Assessments for every proposed change to the project * Be planned for, since maintaining the links/references is a labour intensive process that should be tracked/monitored and should be assigned to a project team member * Be maintained in an electronic document or database form |

**Traceability Linking with RMF/ProR**

The RMF/ProR environment provides one-to-one, one-to-many, many-to-one and many-to-many link creation, traceability link type definition (i.e. satisfies, verifies, traces to, etc.) and traceability reporting functions.

**Traceability with MS-Office Suite**

For very small projects and modest traceability needs (i.e. only one type of traceability links such as requirement to parent requirements) it is possible the maintain traceability between requirements using a MS-Excel spreadsheet. The Unique Identifier of each requirement is used to identify the requirement in traceability matrices.

**Caution:** The maintenance of traceability links with tools such as MS-Excel can rapidly become a heavy burden on the Work Team if too many types of traceability links are captured and maintained in the spreadsheet.

**Traceability Matrix Template**



|  |  |
| --- | --- |
| **Instructions** | |
| The above table should be created in a spreadsheet or database such that it may be easily sorted by each column to achieve bi-directional traceability between columns. The unique identifiers for items should be assigned in a hierarchical outline form such that the lower level (i.e. more detailed) items can be traced to higher items. | |
| Unique Requirement Identification (ID) | The Unique Requirement ID / System Requirement Statement where the requirement is referenced, and/or the unique identification (ID) for decomposed requirements |
| Requirement Description | Enter the description of the requirement (e.g., Change Request description). |
| Design Reference | Enter the paragraph number where the CR is referenced in the design documentation |
| Module / Configured Item Reference | Enter the unique identifier of the System module or configured item where the design is realized. |
| Release Reference | Enter the release/build version number where the requirement is fulfilled |
| Test Script Name/Step Number Reference | Enter the test script name/step number where the requirement is referenced (e.g., Step 1) |

# 9. References to Other Standards and Models

This section provides references and level of coverage achieved by this deployment package against the objectives and requirements of the main internationally recognized development process life-cycle Standards, including:

* the ISO 9001:2008 Quality Management System standard
* the ISO/IEC 15288:2008
* the Capability Maturity Model IntegrationSM for Development(CMMI-DEV), Version 1.3

Notes:

* This section is provided as a guideline for creating a company specific process compliance matrix.
* Only those activities and tasks covered by this Deployment Package are listed in each table.
* The tables use the following convention:
* Full Coverage = F
* Partial Coverage = P
* No Coverage = N

## ISO 9001:2008 Reference Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Coverage**  **F/P** | **Clause of ISO 9001** | **Comments** |
| SR.2.1 | F | 7.2.1 IDENTIFY YOUR UNIQUE PRODUCT REQUIREMENTS  • Identify the unique requirements that your Acquirers want you to comply with.  • Identify the requirements that are dictated by your product's intended use or purpose.  • Identify the requirements that are imposed on your products by external agencies.  • Identify any additional requirements that are important to your organization and must be met. |  |
| SR.2.2 | F | 7.2.2 REVIEW ACQUIRERS' PRODUCT REQUIREMENTS  • Review Acquirers' product requirements.  • Maintain a record of product requirement reviews.  • Control changes in Acquirers' product requirements. |  |
| SR.2.3 | F | 7.2.3 COMMUNICATE WITH ACQUIRERS  • Establish Acquirer communication arrangements.  • Implement Acquirer communication arrangements. |  |

## ISO/IEC 15288:2008 Reference Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Coverage**  **F/P** | **Clause of ISO/IEC 15288** | **Comments** |
| SR.2.2 | F | 6.4.1 Stakeholder Requirements Definition Process  6.4.1.3 a) Elicit Stakeholder Requirements |  |
| SR.2.2 | F | 6.4.1 Stakeholder Requirements Definition Process  6.4.1.3 b) Define Stakeholder Requirements |  |
| SR.2.3, SR.2.4, SR.2.8 | F | 6.4.1 Stakeholder Requirements Definition Process  6.4.1.3 c) Analyze and Maintain Stakeholder Requirements |  |
| SR.2.5 | F | 6.4.2 Requirements Analysis Process  6.4.2.3 a) Define System Requirements |  |
| SR.2.7, SR.2.8 | F | 6.4.2 Requirements Analysis Process  6.4.2.3 b) Analyze and Maintain System Requirements |  |
| SR.2.6 | P | 6.4.3 Architectural Design Process  6.4.3.3 a) Define the Architecture | Tasks 2) and 3) covered |
| SR.2.7, SR.2.8 | P | 6.4.3 Architectural Design Process  6.4.3.3 c) Document and Maintain the Architecture | Tasks 1) and 3) covered |

## CMMI-DEV, V1.3 Reference Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Coverage**  **F/P** | **Objective/Practice of CMMI-DEV** | **Comments** |
| SR.2.1, SR.2.4, SR.2.5 | F | PA REQM  SG 1 – Manage Requirements  SP 1.1 – Understand Requirements |  |
| SR.2.3, SR.2.7 | F | PA REQM  SG 1 – Manage Requirements  SP 1.2 – Obtain Commitment to Requirements |  |
| SR.2.8 | F | PA REQM  SG 1 – Manage Requirements  SP 1.4 – Maintain Bidirectional Traceability of Requirements |  |
| SR.2.2, SR.2.6 | F | PA REQM  SG 1 – Manage Requirements  SP 1.5 - Ensure Alignment Between Project Work and Requirements |  |

# 10. References

|  |  |
| --- | --- |
| **Key** | **Reference** |
| [ISO/IEC 29110] | System Engineering — Lifecycle Profiles for Very Small Entities (VSEs) — Part 5-1: Management and Engineering Guide - Basic VSE Profile |
| [OWPL-EN] | Renault A., Habra N., Alexandre S., Deprez J.-C., OWPL. System Process Improvement for VSE, SME and low maturity enterprises. Version 1.2.2, FUNDP-CETIC, 2000.  (<http://www.cetic.be/internal393.html> ) |
| [INCOSE-HDBK] | INCOSE Systems Engineering Handbook, V3.2.2, October 2011  <http://www.incose.org/ProductsPubs/products/sehandbook.aspx> |
| [INCOSE-GUIDE] | INCOSE Guide for Writing Requirements <http://www.incose.org/ProductsPubs/products/guideforwritingrequirements.aspx> |
| [ISO/IEC12119] | ISO/IEC 12119:1994 Information technology – System packages -- Quality requirements and testing. |
| [ISO/IEC12207] | ISO/IEC 12207:2008 Systems and software engineering - Software life cycle processes. |
| [ISO/IEC15288] | ISO/IEC 15288:2008 Systems and software engineering -- System life cycle processes |
| [ISO/IEC24765] | ISO/IEC 24765, Systems and System Engineering Vocabulary |
| [ISO/IEC29148] | ISO/IEC/IEEE 29148:2011 |
| [SELB07] | Selby, P., Selby, R.W., Measurement-Driven Systems Engineering Using Six Sigma Techniques to Improve System Defect Detection, Proceedings of 17th International Symposium, INCOSE, June 2007, San Diego. |
| [STAN02] | Standish Group – Chaos report 2002. |
| [SPEM05] | System Process Engineering Metamodel Specification, OMG, 2005. |

# 11. Evaluation Form

|  |
| --- |
| **Deployment Package – System Requirements Analysis – Version 0.2**  Your feedback will allow us to improve this package, your comments and suggestions are welcomed |
| **1. How satisfied are you with the CONTENT of this deployment package?**   *Very Satisfied*  *Satisfied*  *Neither Satisfied nor Dissatisfied*  *Dissatisfied*  *Very Dissatisfied* |
| **2. The sequence in which the topics are discussed, are logical and easy to follow?**   *Very Satisfied*  *Satisfied*  *Neither Satisfied nor Dissatisfied*  *Dissatisfied*  *Very Dissatisfied* |
| **3. How satisfied were you with the APPEARANCE/FORMAT of this deployment package?**   *Very Satisfied*  *Satisfied*  *Neither Satisfied nor Dissatisfied*  *Dissatisfied*  *Very Dissatisfied* |
| **4. Have any unnecessary topics been included? (please describe)** |
| **5. What missing topic would you like to see in this package? (please describe)**   * Proposed topic: * Rationale for new topic |
| 1. **Any error in this deployment package?**    * Please indicate:      + - Description of error :        - Location of error (section #, figure #, table #) : |
| **7. Other feedback or comments:** |
| **8. Would you recommend this Deployment package to a colleague from another VSE?**   *Definitely*  *Probably*  *Not Sure*  *Probably Not*  *Definitely Not* |

**Optional**

* Name:
* e-mail address : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Email this form to**: [joseph.marvin@incose.org](mailto:joseph.marvin@incose.org) or: [claude.y.laporte@etsmtl.ca](mailto:claude.y.laporte@etsmtl.ca)

1. INCOSE Web Site http://www.incose.org/practice/whatissystemseng.aspx [↑](#footnote-ref-1)
2. HONOUR Eric, “Systems engineering return on investment”, Defence and Systems Institude, School of electrical and Information Engineering, University of South Australia, January 2013 [↑](#footnote-ref-2)
3. http://philosiblog.com/2011/07/13/if-you-dont-know-where-youre-going/ [↑](#footnote-ref-3)
4. The other activities shown in the graph are : Scope Management (SM); System Architecting (SA); System Integration (SI); Technical Analysis (TA); Technical Leadership/Management (TM); and Verification & Validation (VV). [↑](#footnote-ref-4)
5. INCOSE Guide for Writing Requirements <http://www.incose.org/ProductsPubs/products/guideforwritingrequirements.aspx> [↑](#footnote-ref-5)
6. INCOSE Guide for Writing Requirements <http://www.incose.org/ProductsPubs/products/guideforwritingrequirements.aspx> [↑](#footnote-ref-6)
7. INCOSE Guide for Writing Requirements <http://www.incose.org/ProductsPubs/products/guideforwritingrequirements.aspx> [↑](#footnote-ref-7)