**Deployment Package**

**Configuration Management**

**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 process described in this Deployment Package is not intended to preclude or discourage the use of additional processes that Very Small Entities may find useful.

|  |  |
| --- | --- |
| **Author(s)** | Jimm HUMMER - INCOSE, USA |
| **Editors** | Claude Y. LAPORTE – École de technologie supérieure (ÉTS), (Canada)  Martin GEISREITER – INCOSE, Germany  Joseph Marvin – Prime Solutions Group, Inc, INCOSE, USA |
| **Creation date** | *12 April 2013* |
| **Last update** | *04/07/2014* |
| **Version** | *1.0* |

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date**  (dd/mm/yyyy) | **Version** | **Description** | **Author** |
| 12/04/2013 | 0.5 | Initial Release | Jimm HUMMER |
| 04/07/2014 | 1.0 | Version 1 created for distribution post IS 2014 | Joe Marvin |
|  |  |  |  |
|  |  |  |  |

Abbreviations/Acronyms

|  |  |
| --- | --- |
| **Abre./Acro.** | **Definitions** |
| DP | Deployment Package - a set of artefacts developed to facilitate the implementation of a set of practices, of the selected framework, in a Very Small Entity. |
| INCOSE | International Council on Systems Engineering (<http://www.incose.org>) |
| ISO | International Organization for Standardization (<http://www.iso.org>) |
| SW | Software |
| SY | System |
| TR | Technical Report |
| VSE | Very Small Entity – enterprise, organization, department or project having up to 25 people. |
| VSEs | Very Small Entities |
| CM | Configuration Management |
| V&V | Verification and Validation |
| PMBOK | Project Management Body of Knowledge, <http://www.pmi.org/> |
| *<details>* | *<details>* |
|  |  |

Table of Contents

[1. Technical Description 5](#_Toc355199258)

[Purpose of this document 5](#_Toc355199259)

[Why is Systems Engineering important? 5](#_Toc355199260)

[Why is cooperation between Systems Engineering and Project Management important? 6](#_Toc355199261)

[Why is Configuration Management Important? 7](#_Toc355199262)

[Tailoring this Deployment Package 9](#_Toc355199263)

[2. Definitions 10](#_Toc355199264)

[Generic Terms 10](#_Toc355199265)

[Specific Terms 10](#_Toc355199266)

[3. Relationships with ISO/IEC 29110 11](#_Toc355199267)

[4. Description of Activities, Tasks, Steps, Roles and Products 13](#_Toc355199268)

[Establish the CM Strategy 13](#_Toc355199269)

[Conduct CM during Project Execution 14](#_Toc355199270)

[Role Description 15](#_Toc355199271)

[Product Description 15](#_Toc355199272)

[Artefact Description 16](#_Toc355199273)

[5. Template 17](#_Toc355199274)

[6. Example of Activity Lifecyle 18](#_Toc355199275)

[7. Checklist 19](#_Toc355199276)

[8. Tool 20](#_Toc355199277)

[9. Reference to other Standards and Models 22](#_Toc355199278)

[ISO 9001 Reference Matrix 22](#_Toc355199279)

[ISO/IEC 12207 Reference Matrix 22](#_Toc355199280)

[CMMI for Development Reference Matrix 22](#_Toc355199281)

[10. References 23](#_Toc355199282)

[11. Evaluation Form 24](#_Toc355199283)

# 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 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 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 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 that do not develop critical products that require intense verification and validation (V&V) activities. Those projects could use of the appropriate standards and guides (e.g. ANSI/GEIA EIA-632, MIL-STD-499, etc.)

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*.

Once published by ISO, ISO/IEC TR 29110-5-6-2 will be available at no cost on the following ISO site: <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>

## Why is Systems Engineering important?

The way to effective Systems Engineering (SE) is not “in the direction of formal, formidable, massive documentation” [Chase]. Systems Engineering is a perspective, a process, and a profession [INCOSE]. SE has an iterative nature that supports learning and continuous improvement. SE has a horizontal orientation which means SE is a mechanism to establish agreements for the creation of products and services in a web of contractors and subcontractors. Therefore SE is the link between contractors, and PM, and the organizational parts of enterprises and single technical disciplines (e.g. Software, mechanics, HMI, EMC, etc.).

## Why is cooperation between Systems Engineering and Project Management important?

Systems engineers and program managers bring unique skills and experiences to the programs on which they work. There is also a “shared space” (PM/SE) where program managers and systems engineers collaborate to drive the program team’s performance and success. Therefore they have to collaborate.

Figure 1 shows a concept how systems engineering (SE) and project management (PM) might relate to each other. The basis for this concept is the project lifecycle as proposed by *ISO 21500 – Guidance on project management*. But it’s too simple just to consider the pure project time span for a product development. SE has to consider the whole life cycle of a product in the product concepts until product disposal. Therefore SE has to contribute in all project control activities and provide relevant inputs.



Figure 1 Overview of a concept for SE – PM cooperation

Because a deployment package is not a complete process reference model a VSE might need guidance about how they might perform a project.

To consider the idea for the ISO/IEC TR 29110 simplified technical processes have been defined (see the 9 coloured blocks in Figure 2). Each of these blocks consists of business aspects and technical aspects. Just the degree of involvement for PM and SE changes. Interface management or requirements engineering are commonly understand as SE activities. But they are also influenced by business aspects, enterprise interests or simply by available resources which are more in the PM domain. Therefore the addressed technical processes in Figure 2 might be understood as common (PM&SE) activities.

Configuration management (CM) might be understood as an enterprise oriented task and used in every project. The activities of CM should start with the earliest project activities (the first idea for a project) and will not end with a project. The stored information must be available after a project is finished for several purposes (e.g. following project, legal issues, etc.).

Each of the technical process blocks includes activities which might be performed in different project phases. Figure 2 shows an example to map project process steps to single technical processes. The details for the technical processes are described in different DP packages.



Figure 2 DP structure and linkage to project steps

## Why is Configuration Management Important?

The Functional Analysis and Physical Architecture design is performed during the development phase of the project (see Figure 1). Looking deeper into the development activities depicted in Figure 3 you can see that the refinement and allocation of requirements as well as identifying the design solution are tightly coupled. That is where the Functional Analysis and Architectural Design activities are taking place.

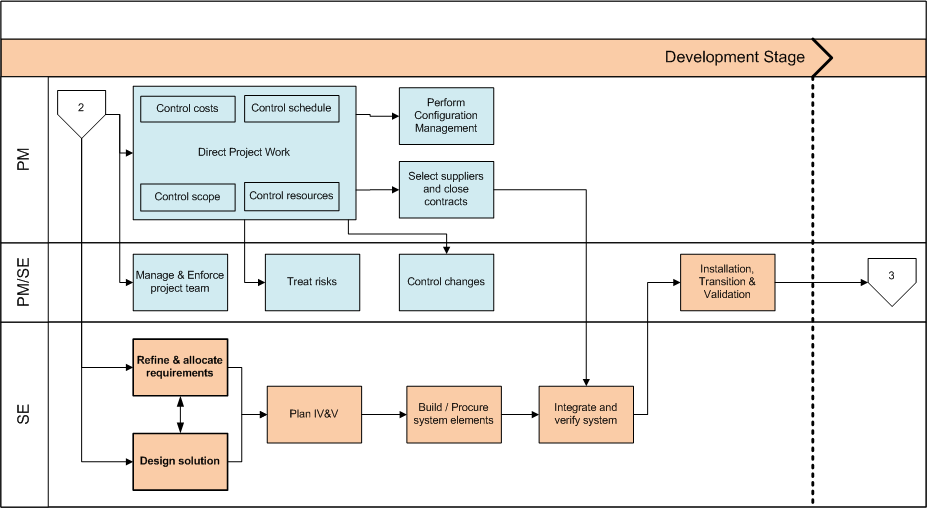


Figure 3 Overview of a concept for SE –Development Stage

During the Functional Analysis, which has to be seen as a part of the “Refine and allocate requirements” activity, the stakeholder needs and stakeholder requirements (not just the customer!) are analyzed to identify the system functions. These system functions subsequently are grouped into functional entities and their logical collaboration (among them and the environment) is described. It is important to notice that the Functional Analysis as such is an implementation free activity, allowing to completely understanding the problem space before selecting a possible implementation (solution). The Functional Analysis is therefore closely related to the Requirements Analysis process as described by ISO/IEC15288 allowing defining a set of system requirements.

The resulting logical architecture provides the foundation for the definition of the physical system architecture as defined during the Architectural Design process described by ISO/IEC15288.

During the design activities a physical architecture is identified and selected, implementing the functional entities identified before.

Design decisions, such as implementation of functions by separate components (e.g. computers) and the selection of specific interface types are taken during the definition of the Physical Architecture.

Note, that such design decisions might also require additional (derived) functionality, e.g. bus controller capability, which requires the return to the Functional Analysis activity for completion. Systems Engineering is an iterative business and in the least cases straight forward.

Finally it has to be mentioned that the aspects of the functional analysis and the physical architecture design already have to be considered and documented during the planning phase (refer to Figure 1). They are typically documented in the SEMP (which is part of the PM plan). A template on the structure of such a technical planning document can be found in 5.1.

## Tailoring this Deployment Package

This DP describes the minimum set of Configuration Management activities and tasks that should be implemented by a VSE. A VSE may have existing processes that can be substituted for these activities, tasks, steps, products and roles.

# 2. Definitions

In this section, the reader will find 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

***Process:*** set of interrelated or interacting activities which transform inputs into outputs [ISO/IEC 12207].

***Activity:*** a set of cohesive tasks of a process [ISO/IEC 12207].

***Task:*** required, recommended, or permissible action, intended to contribute to the achievement of one or more outcomes of a process[ISO/IEC 12207].

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

***Step:*** one element (numbered list item) in a procedure that tells a user to perform an action (or actions) [ISO/IEC 26514]. In a deployment package, a taskis decomposed in a sequence of steps.

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

***Product:*** piece of information or deliverable that can be produced (not mandatory) by one or several tasks. *(e. g. design document, source code)*.

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

***System:*** combination of interacting elements organized to achieve one or more stated purposes. [ISO/IEC 15288:2008]

***Software:*** all or part of the programs, procedures, rules, and associated documentation of an information processing system. [ISO/IEC 2382-1]

## Specific Terms

*<details>*

# 3. Relationships with ISO/IEC 29110

This deployment package covers the activities related to Configuration Management of the ISO Technical Report ISO/IEC 29110 Part 5-1-2 for Very Small Entities (VSEs) – Basic VSE Profile [ISO/IEC 29110].

In this section, the reader will find a list of Project Management (PM) and System Implementation process, activities, tasks and roles from Part 5 that are directly related to this topic. This topic is described in details in the next section.

* **Process:** PM.1 Project Planning, (PM.O1, PM.O5, PM.O6, PM.O7)
* **Activity:** Project Planning
* **Tasks and Roles:**

|  |  |
| --- | --- |
| **Tasks** | **Roles[[1]](#footnote-1)** |
| * PM.1.13 Document the Configuration Management Strategy in the Project Plan. * Identify the Configuration Items. * Define the applicable configuration status * Define the tasks and actors to manage the changes and the configuration**.** | PM |

* **Process:** PM.2 Project Plan Execution (PM.O2, PM.O3, PM.O4, PM.O5, PM.O7)
* **Activity:** Project Plan Execution
* **Tasks and Roles:**

|  |  |
| --- | --- |
| **Task** | **Roles** |
| * PM.2.5 Perform configuration management * According to the configuration management Strategy, manage in configuration the different artifacts of the project. * Generate System Configuration at the planned configuration status. * Identify Changes (e.g. architecture, requirements) and/or Project Plan to address major deviations, potential risks or problems concerning the accomplishment of the project. * Initiate Change Requests on baselined artifacts and analyze impacts (technical cost, quality) before change approval by PM. * Track the changes to closure. | PM  WT |

* **Process:** SR.3 System Architectural Design (SR.O3, SR.O6, SR.O7)
* **Activity:** System Architecture Design
* **Tasks and Roles:**

|  |  |
| --- | --- |
| **Task** | **Roles** |
| * SR.3.5 Verify and obtain approval of the System Design. * If System Design is under configuration management, identify and characterize the impact of the change and initiate if necessary (i.e. change approved) a Change Request. | SYS  DES  DEV |

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

## Establish the CM Strategy

|  |  |
| --- | --- |
|  | |
| ***Objectives:*** | Decide on an acceptable CM Strategy consistent with the Project Plan, stakeholder expectations and project expertise. |
| ***Rationale:*** | The CM Strategy should provide an appropriate level of configuration control for the project. The CM function controls the project baseline of documents, data and product deliverables. |
| ***Roles:*** | PM. The Project Manager has a leadership role in the CM strategy and process. PM ensures customer CM requirements are addressed by the project team. |
| Systems Engineer (SE). The SE is responsible for CM. Depending on project size, the SE may perform CM functions or have a dedicated Configuration Manager. |
| ***Products:*** | CM Strategy (Include in Project Plan) |
| The CM Strategy discusses how CM is performed on the project and the expected results. How is the project managing and controlling the project baseline? Who does the CM function and what reporting is required. |
| ***Artifacts:*** | Project Baseline Configuration Items. List the Configuration Items and make visible to the project team in the project repository. |
| *<details>* |
| ***Steps:*** | 1. Develop the CM Strategy |
| 2. Define CM tasks and integrate into Project Plan |
| ***Step Description:*** | ***Step 1.*** Write the CM Strategy  Write down the CM objectives as derived from project description, customer needs and internal business requirements. The CM Strategy can be a section or attachment to the Project Plan. Include a list of the known Configuration Item deliverables. This is a list from the customer or stakeholder. Description of the step, input/output, form used, etc.  ***Step 2.*** Incorporate CMinto the Project Schedule  Built CM deliverables and activities into the top-level schedule. Allow enough time to complete CM activities. |
|  |  |

## Conduct CM during Project Execution

|  |  |
| --- | --- |
|  | |
| ***Objectives:*** | Execute CM on the project consistent with the strategy and schedule built in step 1. |
| ***Rationale:*** | Once the CM Strategy is developed, PM and CSE ensure CM functions are accomplished to meet project needs. CM is critical to project success, but need not be a complicated or involved process. |
| ***Roles:*** | PM. The Project Manager assigns CM function to responsible team member(s). Depending on size of the VSE, the CM function may be done by the PM or the CSE. |
| Systems Engineer (SE). The SE is responsible for CM. Depending on project size, the SE may perform CM functions or have a dedicated Configuration Manager. |
| ***Products:*** | CM Baseline |
| The CM Baseline is developed per the CM Strategy. The CM Baseline is approved by a project process and posted to the project repository. |
| ***Artifacts:*** | CM Baseline Management process. |
| Everyone on the project must know how the CM baseline is established and governed throughout the lifecycle of the project. |
| ***Steps:*** | 1. Write down the steps used on the project to develop the Project CM Baseline. |
| 2. Define how the CM Baseline is changed or modified by members of the project team. |
| ***Step Description:*** | ***Step 1.*** Develop the process of establishing the project baseline and how the project will manage the configuration controlled baseline. This may be a simple list of the steps the project team uses to document and govern the project baseline.  ***Step 2.*** Define and implement a change management process. This includes developing a change request and tracking of the change request through the process. |
|  |  |

## Role Description

This is an alphabetical list of the roles, abbreviations and list of competencies as defined in the engineering and management guide of ISO/IEC 29110..

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Role*** | ***Abbreviation*** | ***Competency*** |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |

**Example:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Role*** | ***Abbreviation*** | ***Competency*** |
| *1.* | *Analyst* | *AN* | *Knowledge and experience eliciting, specifying and analyzing the requirements.*  *Knowledge in designing user interfaces and ergonomic criteria.*  *Knowledge of the revision techniques and experience on the system/software development and maintenance.*  *Knowledge of the editing techniques and experience on the system/software development and maintenance.* |

## Product Description

This is an alphabetical list of the input, output and internal process products, its descriptions, possible states and the source of the product.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Description** | **Source** |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |

**Example:**

|  | **Name** | **Description** | **Source** |
| --- | --- | --- | --- |
| *1.* | *Change Request* | *It may has the following characteristics:*  *Identifies purpose of change*  *Identifies request status (new, accepted, rejected)*  *Identifies requester contact information*  *Impacted system(s)*  *Impact to operations of existing system(s) defined*  *Impact to associated documentation defined*  *Criticality of the request, date needed by*  *The applicable statuses are: initiated, evaluated and accepted.* | *System/Software Implementation*  *Customer*  *Project Management* |

## Artefact Description

This is an alphabetical list of the artifacts that could be produced to facilitate the documentation of a project. The artifacts are not required by Part 5, they are optional.

|  | **Name** | **Description** |
| --- | --- | --- |
| 1. | *<details>* | *<details>* |

# 5. Template

*<Template(s) is/are provided to help VSE produce a better product> <details>*

**Example:**

*Configuration Management Plan*

*Configuration Management Plan Outline*

***1. Introduction***

*1.1 Purpose*

*1.2 Scope*

*1.3 References*

*1.4 Configuration Item Nomenclature*

*1.5 Baselines*

*1.6 Change Control*

*1.7 Verification and Audits*

# 6. Example of Activity Lifecyle

*This section provides, for this topic, a graphical representation of a lifecycle. The example is provided to help the reader implement his own lifecycle fitting his IT project’s context and constraints.*

*<details>*

**Example:**

# 7. Checklist

*<Checklist(s) is/are provided to help VSE produce a better product> <details>*

**Example**:

# 8. Tool

*<List the reference(s) (e.g. address of web site) to tool(s) that can help implement this deployment package. If a tool is simple to describe, you may provide the description to use it in this section. See the example below>*

*<details>*

# 9. Reference to other Standards and Models

This section provides references of this deployment package to selected ISO and ISO/IEC Standards and to the Capability Maturity Model IntegrationSM for Development, Version 1.3 of the Software Engineering Institute (CMMI®[[2]](#footnote-2)).

Notes:

* This section is provided for information purpose only.
* Only 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 Reference Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Clause of ISO 9001** | **Coverage**  **F/P/N** | **Title of the Task and Step** | **Comments** |
| *<details>* | *<details>* | *<details>* |  |
|  |  |  |  |
|  |  |  |  |

## ISO/IEC 12207 Reference Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Clause of ISO/IEC 12207** | **Coverage**  **F/P/N** | **Title of the Task and Step** | **Comments** |
| *<details>* | *<details>* | *<details>* |  |
|  |  |  |  |
|  |  |  |  |

## CMMI for Development Reference Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Objective/ Practice of CMMI V1.3** | **Coverage**  **F/P/N** | **Title of the Task and Step** | **Comments** |
| *<details>* | *<details>* | *<details>* |  |
|  |  |  |  |
|  |  |  |  |

# 10. References

|  |  |
| --- | --- |
| **Key** | **Reference** |
| [ISO/IEC 12207] | ISO/IEC 12207:2008 Systems and software engineering - Software life cycle processes. |
| [ISO/IEC 15288:2008] | ISO/IEC 15288:2008 Systems and software engineering - System life cycle processes. |
| [ISO/IEC 26514] | ISO/IEC 26514, Systems and software engineering — Requirements for designers and developers of user documentation. |
| [ISO/IEC/IEEE 24765] | ISO/IEC/IEEE 24765:2010, Systems and Software Engineering - Vocabulary.  Available on line at:  <http://pascal.computer.org/sev_display/index.action> |
| [ISO/IEC 29110] | ISO/IEC 29110:2011-5-1-2 - Software Engineering — Lifecycle Profiles for Very Small Entities (VSEs) — Part 5-1-2: Management and Engineering Guide – Generic Profile Group - Basic Profile. |
| [ISO/IEC 2382-1] | ISO/IEC 2382-1:1993, Information technology — Vocabulary — Part 1: Fundamental terms. |
| *<details>* | *<details>* |

# 11. Evaluation Form

|  |
| --- |
| **Deployment Package *<Title>–* *Version <X.X>***  Your feedback will allow us to improve this deployment 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 |
| **6. 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. Roles are defined in a next section. Roles are also defined in the engineering and management guide of ISO/IEC 29110 [↑](#footnote-ref-1)
2. SM CMM Integration is a service mark of Carnegie Mellon University.

   ® Capability Maturity Model, CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University. [↑](#footnote-ref-2)