IT Project Delivery

Management Processes

# Document

## Description

Title: IT Project Deliverables to Consider

Author: Sky Sigal, Solution Architect

Date: 18/05/2021

## Distribution

Distribution/Review List:

* Amy Orr, Data Architect
* Diane Simpson, Infrastructure Architect
* Brad Atte Le Crouche, Solution Architect
* Stuart McGrigor, Solution Architect
* Dries Venter, Cloud Infrastructure Specialist
* Werner Van der Merwe, Automation Lead
* Yujia Huang, Security Specialist, Contractor
* Jeff Grove, Security Specialist, Contractor
* Sean Torley, ICT Change & Transition
* Sally Murrey, ICT Change Advisor
* Grayson Mitchell, Manager, BE&S ICT Strategy Planning & Architecture

## Conventions

### Terms

The Appendices include a Glossary of Terms to assist understand this document.

### Diagrams

Where applicable, diagrams are developed using ISO-\*, Archimate, UML or appropriate industry standards and conventions.

## Table of Contents

[1 Document 2](#_Toc74483114)

[1.1 Description 2](#_Toc74483115)

[1.2 Distribution 2](#_Toc74483116)

[1.3 Conventions 2](#_Toc74483117)

[1.4 Table of Contents 2](#_Toc74483118)

[1.5 Synopsis 3](#_Toc74483119)

[1.6 Background 3](#_Toc74483120)

[1.7 Objective 4](#_Toc74483121)

[1.8 Scope 4](#_Toc74483122)

[1.9 Applicability 4](#_Toc74483123)

[1.10 Responsibility of Delivery 5](#_Toc74483124)

[1.11 Completeness 5](#_Toc74483125)

[2 Processes 6](#_Toc74483126)

[2.1 Background 6](#_Toc74483127)

[3 Software Lifecycle Standards 9](#_Toc74483128)

[3.1 ISO-15288: System & Software engineering – System Life cycle Processes 9](#_Toc74483129)

[3.2 ISO-12207: System and Software Engineering – Software life cycle processes 9](#_Toc74483130)

[3.3 ISO-29110: System and Software Life Cycle Profiles and Guidelines for Very Small Entities (VSEs) 10](#_Toc74483131)

[4 Appendices 17](#_Toc74483132)

[4.1 Terms 17](#_Toc74483133)

## Synopsis

The success of a programme/project which contains technical elements depends on the timely delivery of deliverables expected and usable by different stakeholder groups.   
It is unlikely to meet these expectations without a clear idea of what is to be delivered by one delivery stakeholder group or another. Additionally, it is unlikely to meet the deliverable expectations within expected timelines without processes to minimise friction and uncertainty as to required tasks. This document lists and presents internationally accepted standards for processes to deliver the expected deliverables.

## Background

IT Project Delivery Management is a specialised subtype of general Project Management, with a distinct set of expected deliverables.

Market scarcity leads to a relatively common condition where IT projects are being delivered via generalist project managers with limited prior exposure to successful IT delivery processes.

### Problem

Delivery Managers are reliant on a reactive, ad hoc, process of incomplete discovery of expected deliverables – rather than having a clear understanding of objectives, expected deliverables and planned processes to deliver them. This leads to unnecessary remedial costs (between 17% and 40%), while significantly increasing the risk of the project failing to deliver some or all the expected deliverables within budget and/or schedule.

## Objective

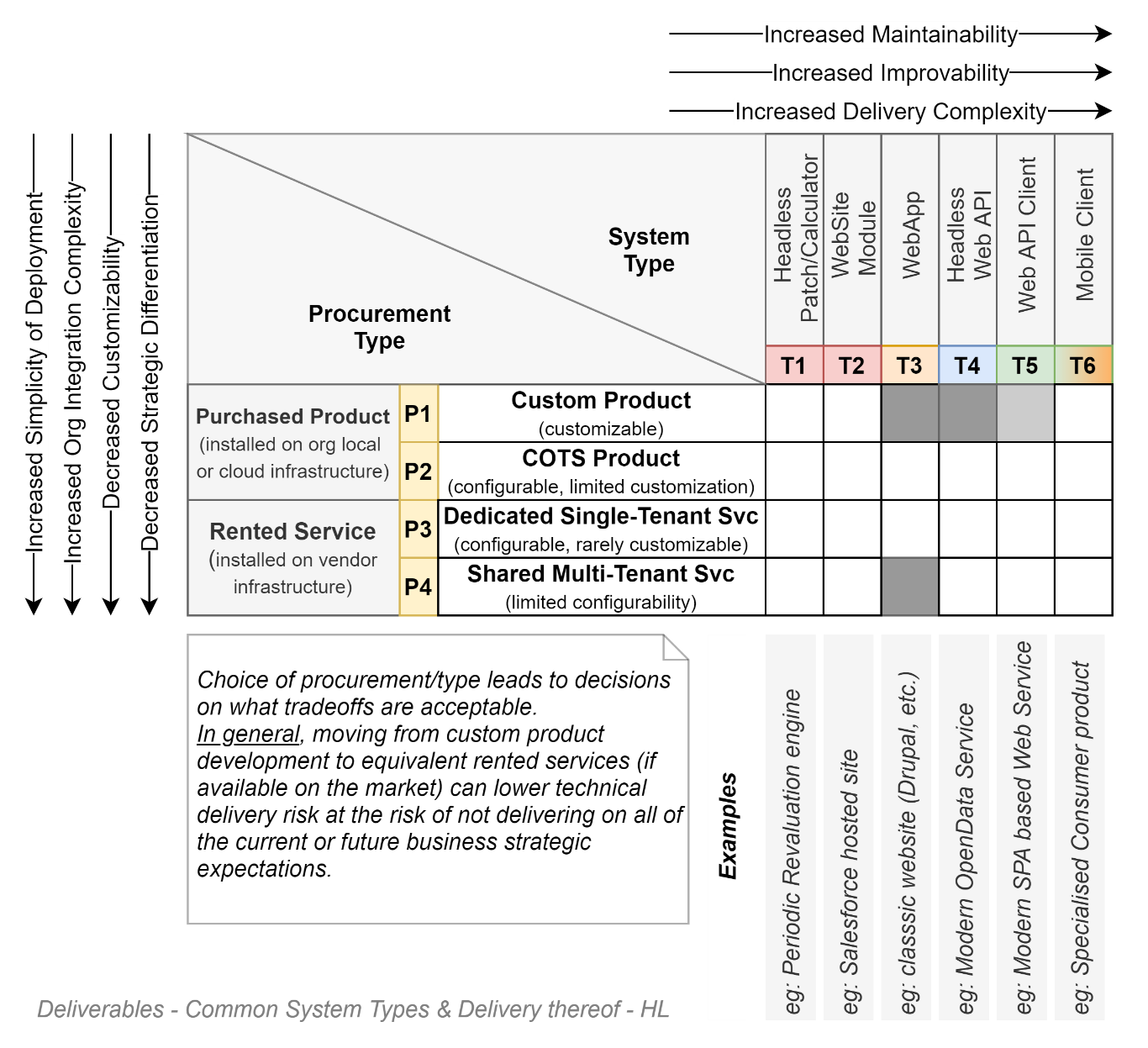
The purpose of this document is to provide managers exposure to internationally accepted and recommended processes to decrease risk of failure to deliver expected deliverables.

## Scope

The scope of this document includes describing the types of delivery methodologies and processes to follow.

It is not within this document’s scope to describe the Deliverables themselves.

## Applicability



Rather than trying to cover all types of IT projects, this document lists the deliverables for the most common types commissioned within organisations.

The final system type is – at a high level -- determined by the organisation’s procurement (buy versus rent), hosting preferences (on-prem versus cloud), technological capability (low to medium) – as per the diagram above.

The less capability an organisation has, the more it sees customisation a risk, preferring to stick to buying mature custom products, even if the product is strategically poorly aligned[[1]](#footnote-1). The more digitally mature an organisation, the more it can take advantage of customization abilities to deliver outcomes that are better strategic fits.

## Responsibility of Delivery

Depending on the procurement process used, the responsibility of delivering the listed Deliverables is balanced differently between suppliers and this organisation.

## Completeness

Not all system types will require all deliverables.

Whomever is responsible for delivering the deliverable, it remains the accountability project manager’s responsibility to ensure processes are followed and roles are aware of the responsibility to deliver the deliverables.

### Constraints

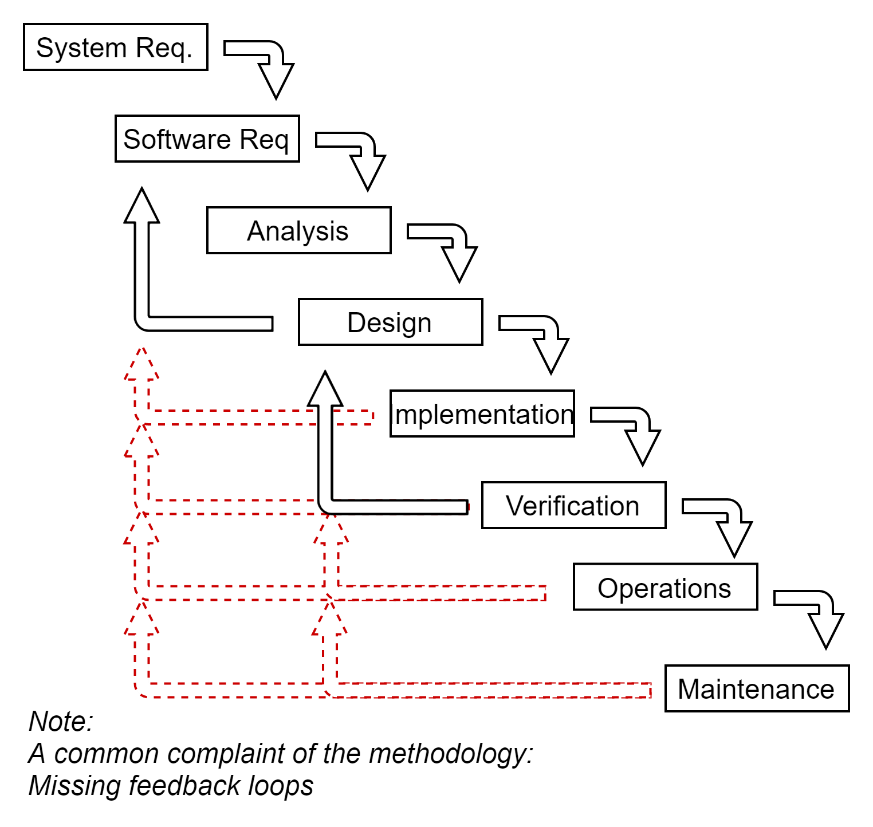
Government organisations are mandated by regulation to use international standards (ISO) when they are available[[2]](#footnote-2).

# Processes

## Background

Almost all IT project stakeholders have had some passing exposure to two dominant IT project delivery methodologies:

### “Waterfall” Methodology



The Waterfall Methodology is a well-known iteration through the SDLC Software construction Phases defined within ISO-12207.

Common issues with the way the way the methodology was implemented include:

* No feedback loop
* Heavy upfront design
* Primarily intended to be a single “Go-Live” deployment -- albeit with subsequent touch ups.

*Note:  
Other Phases of the ISO standard are not reflected in most applications of the waterfall methodology, wasting the knowledge already available in the ISO.*

### Agile Methodology

The Agile Methodology was developed to address some of the deficiencies in how the waterfall methodology was being applied in most instances.

Specifically, the Agile Methodology emphasises the following:

* Either a continuous or alternatively a rapid iterative delivery process,
* A reduction of time consumed on delivering deliverables for non-user stakeholders (eg, support documentation), by focusing on using
* Self-describing tests
* maintainable code development patterns, all while using
* Automation to reduce the time to build, integrate, test, deploy environments & systems

Unstated Consequences of the above devolution include:

* The autonomy and self-reliance required of development stakeholders implies a necessity for atypically experienced development resources.
* A focus on Development as opposed to upfront Design, leads to poor planning and a high degree of re-work.
* A focus on Development as opposed to Operations. Hence the later emergence of the DevOps delivery methodology.

#### Scrum

Scrum is an accredited formalization of Agile development patterns. At its very essence it relies on designated scrum masters to lead daily developer standups to promote communication amongst delivery members of work still to be done and any required dependencies delaying their delivery, and a focus on delivering quickly to business & user stakeholders a Minimum Viable Product (MVP).

Scrum has steadily been losing favour in the industry due to not being able to resolve what DevOps and SecOps has been able to address better.

#### Agile MVPs

A reduction of processes, a de-emphasis on delivering other deliverables other than the system itself, a focus on rapid delivery of a business and user focused Minimum Viable Product (MVP) at the cost of first using system engineering best practices to develop a firm yet flexible foundation, a lack of architectural planning before effort begins, often lead to undesired outcomes.   
  
When “Fools rush in” (Pope, 1711)., MVP stands for *Missing Valuable Planning*.

### DevOps/SecOps

The Agile DevOps Methodology is a variation of the basic Agile development methodology to address its too narrow focus on development only, including the processes and people from a different group – Operations.

Sec-[Dev]-Ops is a newer superset of DevOps that expands this cross-group integration, starting with the people & processes of the Security group.

Both DevOps & SecOps take to heart Agile’s recommendation to put automation at the centre and uses it for as much as possible.

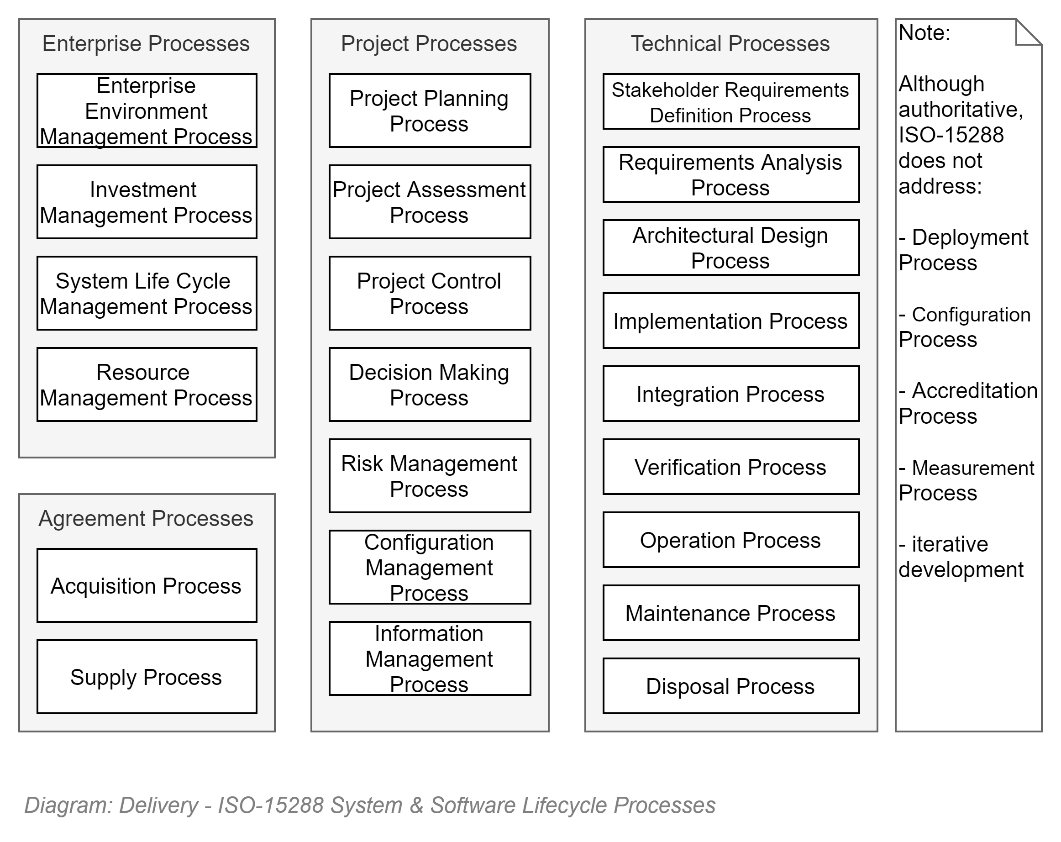
A consequence of including the Operations Group’s activities (and later the Security group’s activities – referred to as the SecOps delivery methodology), is a reassessment of the importance of documentation and messaging between groups.

The added complexity of communication and cross-group deliverables leads to a reappraisal of Agile’s principle to focus on development artefacts.

The Sec/DevOps process has proven itself at reducing the risk of non-delivery and/or meeting the expectations of multiple stakeholder groups. So much so that it has been developed into an ISO standard: ISO-29110-5-5/-4-5.

# Software Lifecycle Standards

## ISO-15288: System & Software engineering – System Life cycle Processes



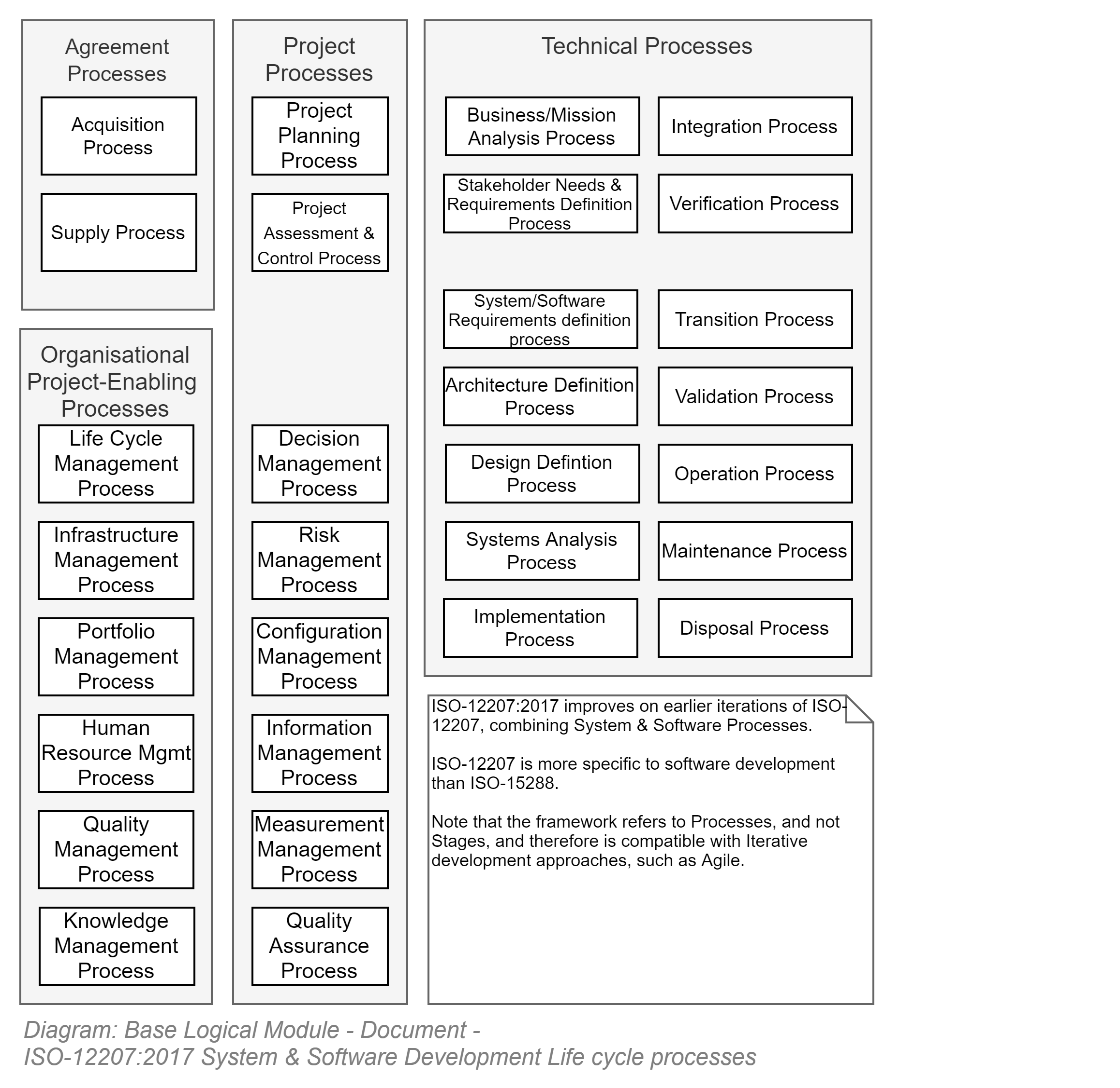
ISO-15288 can be considered the original standard that is the foundation on which later simplified standards have built.

It provides a framework of described *Activities*, grouped into different categories: *Processes* (Enablement, Management, Specialist/Technical, etc.).

Key points include:

* As per its title, the standard is for physical systems, which may have software aspects to it.
* As such, it’s a very complete standard, appropriate for the most robust projects
* Due to its completeness, and therefore complexity, the standard is only recommendable for the most complex of software only projects.
* Due to its age, it does not address well more modern development concepts.
* The ISO defines a set of *Process* groups, each which contain one or more *Processes*, composed of one or more *Activities*, which are in turn composed of *Tasks*.

## ISO-12207: System and Software Engineering – Software life cycle processes



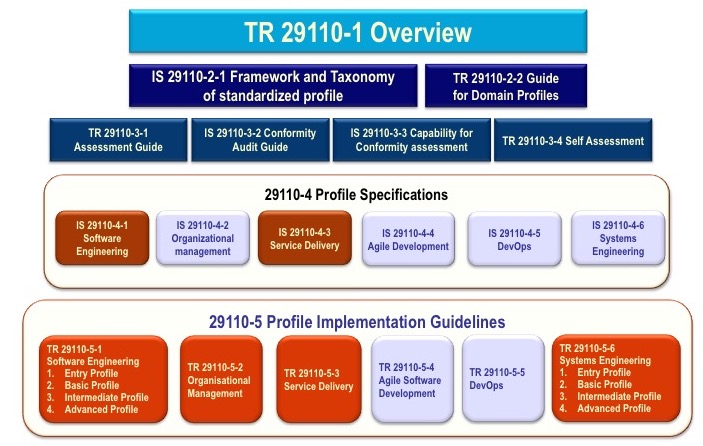
More focused on delivering software solutions – and therefore simpler to implement – is ISO-12207.

It provides a framework of described processes, grouped into different categories: Procurement, Enablement, Management, Specialist/Technical.

Key points include:

* Although more focused and lighter than ISO-15288, its still onerous compared for projects with less than 25 development stakeholders (generally referred to as Very Small Enterprise (VSE) projects).
* The ISO defines a set of *Process* groups, each which contain one or more *Processes*, composed of one or more *Activities*, which are in turn composed of *Tasks*.

## ISO-29110: System and Software Life Cycle Profiles and Guidelines for Very Small Entities (VSEs)



The ISO-29110 series of standards are focused on addressing the constraints of small projects a– as the majority of business organisation projects around the world are.

Key aspects of the ISO include:

* **Profiles:** project stakeholders start by choosing one of 4 available Profiles (see: ISO 291110-5-1) that best reflects their context (*Entry*, *Basic*, *Intermediate,* *Advanced*).   
  Small project-start teams might prefer the less tasks required defined for an *Entry* profile.   
  As the team grows and/or mature, the Profile that best suits their growing obligations and concerns may become *Basic*, and then *Advanced*.
* **Implementations:** the standard covers two methodologies: Agile (29110-5-3) and the more complete DevOps (29110-5-4). SecOps is not explicitly covered but the DevOps ISO can be used for this methodology.   
  The Profile chosen earlier defines more or less how complex ones implementation of a methodology (which the standard refers to as an *Implementation*) is required.   
  For example, an *Entry* Profile’s *Implementation* of the *DevOps* methodology will require less Processes and less Artefacts than a *Basic*, *Intermediate* or *Advanced* Profile.
* **Processes:** irrespective of the Implementation, the ISO defines a set of Processes.
* **Activities:** Processes are broken down into Activities groups, the same as in ISO-15288 and ISO-12207.
* **Tasks:** the Profile/Implementation choice determines the minimum recommended number of tasks/deliverables to plan for within each activity within each *Process group* (Project Management, Technical, etc – same as for ISO-15288, ISO-12207).

*Note:  
As per the earlier description of failings of a purely Agile approach, and the improvements that DevOps offer, only the DevOps standard is covered below.*

### Applicability

ISO-29110 is intended to be used by VSEs that do not have experience or expertise in adapting/tailoring ISO 12207 or ISO 15288 to the needs of the specific project.

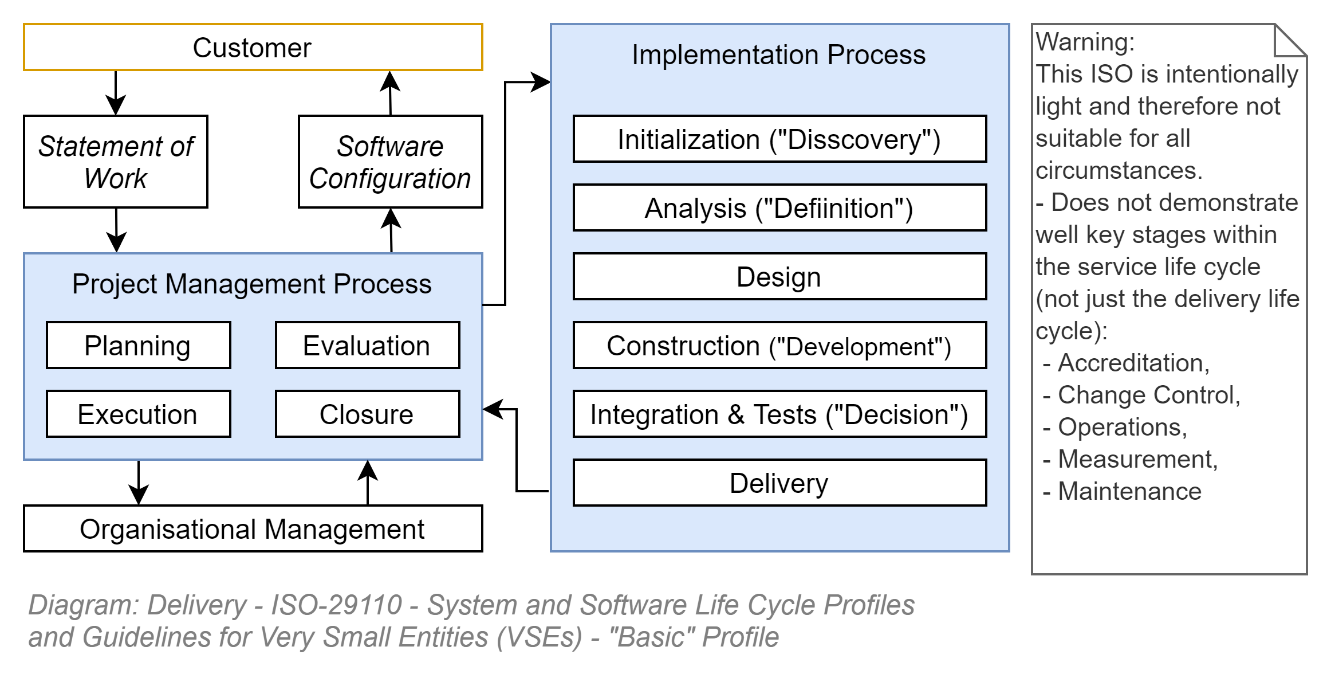
ISO 29110 primarily defines Processes, Activites & Tasks -- is not intended to preclude the use of different life cycles such as waterfall, iterative, incremental, evolutionary or agile.

### Profiles

There are 4 Profiles defined (*Entry*/*Basic*/*Intermediate*/*Advanced*).

*Note: It can be safe to say this organisation is not Advanced, or even Intermediate.*

#### Basic Profile



The Basic Profile is defined with only two phases: Project Management *and* Implementation*).*

Although far simpler, one can still distinguish the legacy of ISO-15288 and ISO-12207, in that there are still Processes – just fewer of them.

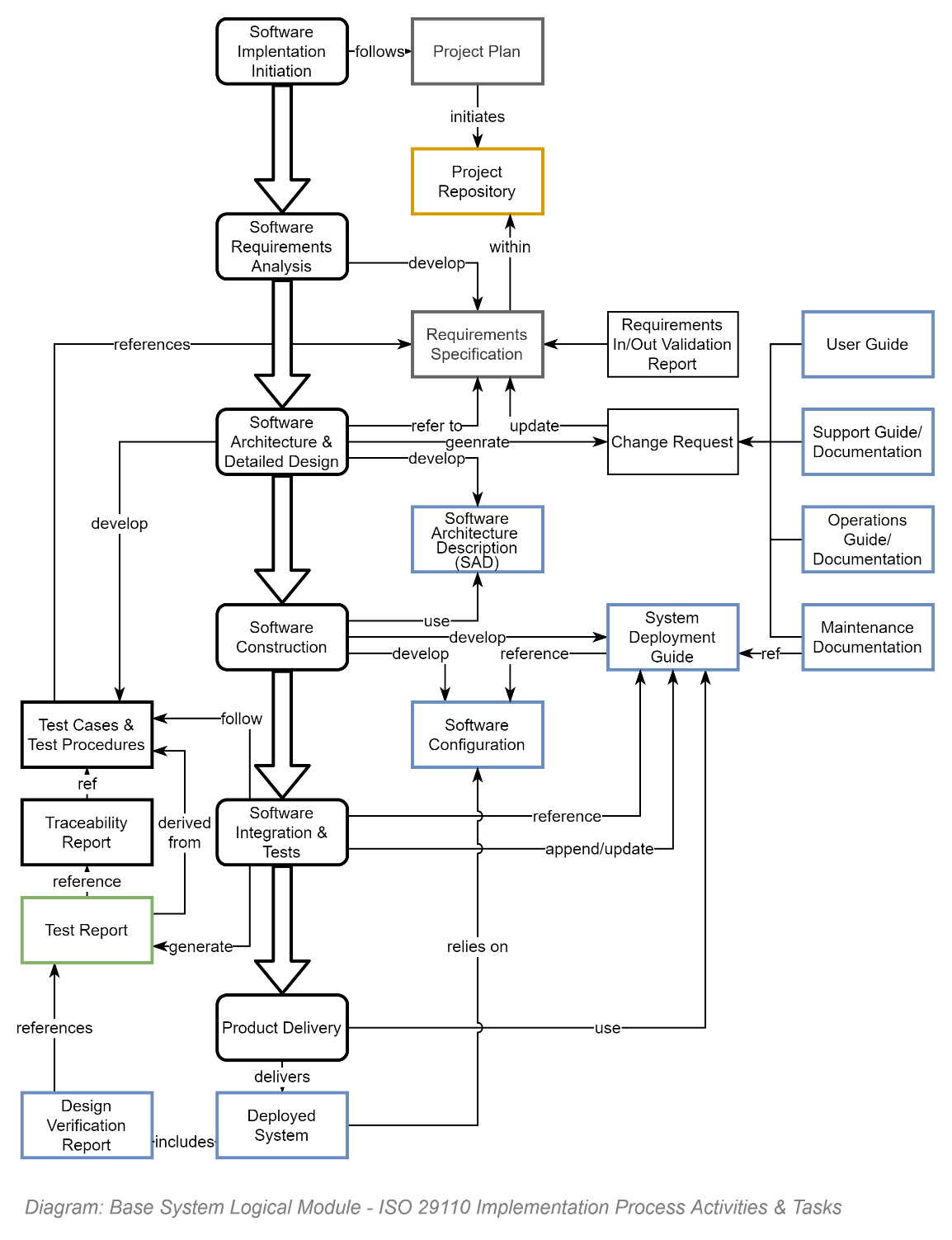
The two processes, and their child Activities and Tasks are described below.

### Project Management Process Activities

The above diagram outlines the 5 Activities that occur within the Project Management Process, and Tasks that belong to each Activity.

The diagram’s Tasks are self-explanatory.

### Implementation Process Activities



The Basic Profile’s activities and tasks within the Implementation Process are only slightly more complex that the Project Management Process Activities and Tasks.

Again, the diagram’s tasks should be self-explanatory.

### Phases versus Stages

The first time the ISO was published the term Stages was used instead of Phases.   
The consequence was that project managers expected the project to proceed in an orderly set of stages – much like a Waterfall SDLC based project.   
Nothing could have been further from the intentions of the publisher of this ISO.   
Hence their correction and using the term Stages to imply that Stages are progressed through *iteratively*. In other words, PMs will be continually or iteratively returning to their Activities and Tasks -- and the same for the Implementors – until Stakeholders are satisfied, or available Resources are depleted.

### Value the ISO brings to Projects

Activities are logical, and not difficult to understand. The same applies for the Tasks of each Activity.

In fact, the simplicity of the Processes, Activities & Tasks diagrams above may obfuscate the obvious value they bring.

The value that the ISO brings to projects – other than meeting this organisation’s compliance obligations to use standards when available -- is that the ISO provides a clear upfront list of Tasks to better plan their timely completion at the right time. *As opposed to only discovering Tasks in an ad hoc, reactive, manner when project stakeholders bring them up.*

### Planning & Deliverable Variation

Variation is Permitted by Decision.   
Not all projects have to deliver all artefacts – and some projects may have more deliverables. But the ISO provides a central backbone of plannable Tasks.

### Implementation

All the above Phases, Activities and Tasks are the Basic minimum to consider and plan for when delivering small projects.

But *how* you do it will depend on the Implementation chosen.

In our case we will concentrate on the DevOps implementation.

#### ISO-29110-5-5/-4-5: DevOps profile Guidelines & Specifications

The DevOps implementation

* Emphasises the iterative aspect to performing each phases tasks
* emphasis on using Automation at the centre of the artefact delivery process,
  + in the process adds a few more iteratively developed deliverable artefacts

##### Live Documentation

ISO-29110 does not specify how the Tasks are accomplished – the value comes from understanding the specific needs of stakeholders on the project at hand.

In the case of Documentation, one should take learnings from the Agile Manifesto and where possible prefer light, live documentation (eg: in a shared Wiki) as opposed to full blown Word documents that once circulated for review will rarely be updated.

In the case of Tests, instead of developing documents and excel based testing scripts, the preference is for a Test Analyst to develop scripts which developers can develop into coded tests within the automation pipeline. As such, the Test Cases become part of the Work Items and Test Procedures document becomes a thin document that explains how tests are developed into automation.

The SAD document requires circulation, review and acceptance by Governance boards, so it should be a word document or similar.

The System Deployment Guide is probably also a thin document that simply points to the project teams shared wiki, where there is an explanation as to how the development pipeline is developed.

##### Automation

The Automation is addressed by ensuring a Delivery Pipeline is begun first, to in turn run Automated Tests, which in turn validate the Pipeline’s compilation and deployment of the business system.

1. Note that “mature products”, practically by definition, have been around a long time and most probably developed before Integration & Reuse was considered a Strategic requirement – therefore missing APIs. [↑](#footnote-ref-1)
2. As the public paid for the development of the standards to encode best practices for obtaining highest long term value, it is appropriate that government agencies use the best practices to the public’s benefit. [↑](#footnote-ref-2)