Capgemini Agile Framework (CAF) Agile Project Governance Plan

**(Project Name and Id)**

**(Client)**

**9-Dec-16 Version 1.1**

| **Entity** | **Receiving Letters on Dec. 15** | **Receiving Letters on Dec. 22** |
| --- | --- | --- |
| **ACIS** | P0, P1, P2, P3, P4, P5, M1 | M2, M3, M4, M5, M6 |
| **FS** | Software associate  Software engineer  Senior Software Engineer  Associate Consultant  Consultant  Senior Consultant  Lead Consultant | Manager  Senior Manager  Principal |
| **IGATE** | L2, L3, L4, L5, L6, L7, L8 | L9, L10, L11, L12, L13 |
| **BSv (ACIS)** | B1, B2, B3, B4, B5, B6, B7  and  P0, P1, P2, P3, P4, P5, M1 | B8, B9  and  M2, M3, M4, M5, M6 |
| **BSv (CG BSIL)** | O1, O2, O3, O4, O5, O6 | O7, O8, O9, O10 |
| **BSv (IGATE)** | L1, L2, L3, L4, L5, L6, L7 | L8, L9, L10, L11, L12, L13 |

<https://spring.io/guides/gs/spring-boot/>

<https://docs.spring.io/spring-boot/docs/current-SNAPSHOT/reference/htmlsingle/#boot-documentation>.

Table of Contents

[Table of Contents 1](#_Toc467746816)

[1 Project Governance Plan Overview 6](#_Toc467746817)

[1.1 Purpose Of The Project Governance Plan 6](#_Toc467746818)

[1.2 Scope Of The Project Governance Plan 6](#_Toc467746819)

[1.3 Control Of The Project Governance Plan 6](#_Toc467746820)

[1.4 Deviations From Standard Structure 7](#_Toc467746821)

[1.5 References 8](#_Toc467746822)

[2 Project Overview 9](#_Toc467746823)

[2.1 Project Description 9](#_Toc467746824)

[2.2 Project Objectives 9](#_Toc467746825)

[2.3 Project Scope 10](#_Toc467746826)

[2.4 Project Critical Success Factors 11](#_Toc467746827)

[2.5 Contractual Deliverables 11](#_Toc467746828)

[2.6 Contractual Milestones 12](#_Toc467746829)

[2.7 Client Obligations 12](#_Toc467746830)

[2.8 Assumptions And Constraints 12](#_Toc467746831)

[2.9 Dependencies 13](#_Toc467746832)

[3 Project Approach 14](#_Toc467746833)

[3.1 Iterations And Activities 14](#_Toc467746834)

[3.2 Definition Of Done 15](#_Toc467746835)

[3.3 Definition Of Ready 15](#_Toc467746836)

[3.4 Deliverables 16](#_Toc467746837)

[3.5 Verification And Validation 18](#_Toc467746838)

[3.6 Notification Of Deliverables 18](#_Toc467746839)

[3.7 Replication 18](#_Toc467746840)

[3.8 Installation And Implementation 18](#_Toc467746841)

[3.9 Servicing 18](#_Toc467746842)

[4 Project Governance 18](#_Toc467746843)

[4.1 Project Management Method 19](#_Toc467746844)

[4.2 Project Organisation 19](#_Toc467746845)

[4.3 Project Roles and Responsibilities 19](#_Toc467746846)

[4.4 Project Monitoring And Reporting 20](#_Toc467746847)

[4.4.1 Project Monitoring 20](#_Toc467746848)

[4.4.2 Project Reporting 20](#_Toc467746849)

[4.5 Meetings 20](#_Toc467746850)

[4.6 Client Billing 21](#_Toc467746851)

[4.7 Supplier Invoices 22](#_Toc467746852)

[4.8 Monitoring Of Actions 22](#_Toc467746853)

[4.9 Client Sign-Off And Acceptance 22](#_Toc467746854)

[4.10 Governance Handover 22](#_Toc467746855)

[4.11 Warranty 22](#_Toc467746856)

[5 Planning & Financial Management 24](#_Toc467746857)

[5.1 Estimating (Sizing) 24](#_Toc467746858)

[5.2 Project Plan 24](#_Toc467746859)

[5.3 Project Cost And Budget Management 24](#_Toc467746860)

[6 Resource Management 25](#_Toc467746861)

[6.1 Team Organisation 25](#_Toc467746862)

[6.2 Team Roles and Responsibilities 25](#_Toc467746863)

[6.3 On-Boarding 25](#_Toc467746864)

[6.4 Training and Coaching 25](#_Toc467746865)

[6.4.1 Capgemini Team Members 25](#_Toc467746866)

[6.4.2 Client Team Members 26](#_Toc467746867)

[6.5 Team Member Evaluations 26](#_Toc467746868)

[6.6 Team Member Release 26](#_Toc467746869)

[7 Scope And Requirements Management 27](#_Toc467746870)

[7.1 Requirements Traceability 27](#_Toc467746871)

[8 Change Control 27](#_Toc467746872)

[8.1 Impediment Procedure 27](#_Toc467746873)

[9 Risk Management 28](#_Toc467746874)

[9.1 Business Continuity Planning 28](#_Toc467746875)

[10 Issue Management 28](#_Toc467746876)

[11 Client Relationship Management & Communication 29](#_Toc467746877)

[11.1 Client Profile 30](#_Toc467746878)

[11.2 Client Steering Board Description 30](#_Toc467746879)

[11.3 Client Kick-Off Meeting 30](#_Toc467746880)

[11.4 Client Satisfaction And OTACE 30](#_Toc467746881)

[12 SUPPLIER AND PROCUREMENT MANAGEMENT 31](#_Toc467746882)

[12.1 Supplier Selection 31](#_Toc467746883)

[12.2 Supplier Agreements 31](#_Toc467746884)

[12.3 Supplier Tracking 31](#_Toc467746885)

[12.4 Supplier Audits 31](#_Toc467746886)

[12.5 Supplier Billing 31](#_Toc467746887)

[12.6 Supplier Performance 31](#_Toc467746888)

[12.7 Supplier Acceptance 31](#_Toc467746889)

[13 Communication Management 32](#_Toc467746890)

[13.1 Progress Meetings 32](#_Toc467746891)

[13.2 Correspondence Handling 32](#_Toc467746892)

[13.3 Communication Plan 32](#_Toc467746893)

[13.4 Inter-group commitments 32](#_Toc467746894)

[13.5 Grassroots Support 33](#_Toc467746895)

[14 Infrastructure Management 34](#_Toc467746896)

[14.1 Project Management Office Infrastructure 34](#_Toc467746897)

[14.2 Project Delivery Infrastructure 34](#_Toc467746898)

[14.3 Infrastucture Reviews 34](#_Toc467746899)

[14.4 Security Management 34](#_Toc467746900)

[14.4.1 Security Baselines 35](#_Toc467746901)

[14.4.2 Site Access 35](#_Toc467746902)

[14.4.3 System Access 35](#_Toc467746903)

[14.4.4 Storage and Archiving 35](#_Toc467746904)

[14.4.4.1 Storage 35](#_Toc467746905)

[14.4.4.2 Archiving 36](#_Toc467746906)

[14.4.4.3 Backup Procedures 36](#_Toc467746907)

[15 Configuration Management 37](#_Toc467746908)

[15.1 Configuration Identification 37](#_Toc467746909)

[15.2 Configuration Control 37](#_Toc467746910)

[15.3 Configuration Status Reporting 37](#_Toc467746911)

[15.4 Configuration Reviews and Audits 37](#_Toc467746912)

[16 Quality Management 38](#_Toc467746913)

[16.1 Project Reviews 38](#_Toc467746914)

[16.2 Deliverable Reviews 38](#_Toc467746915)

[16.3 Project Audits 38](#_Toc467746916)

[16.4 Quality Audits & Schedule 38](#_Toc467746917)

[16.5 Defect Tracking 39](#_Toc467746918)

[16.6 Defect Prevention 40](#_Toc467746919)

[16.7 Continuous Improvement 40](#_Toc467746920)

[16.8 Final Evaluations 40](#_Toc467746921)

[17 Knowledge Management 41](#_Toc467746922)

[17.1 Engagement Profile 41](#_Toc467746923)

[17.2 Knowledge Object Reuse About Release Retrospective 41](#_Toc467746924)

[17.3 Knowledge Object Sharing 41](#_Toc467746925)

[18 Quantitative Engagement Management 41](#_Toc467746926)

[18.1 Project Metrics Management 41](#_Toc467746927)

[19 Appendices 42](#_Toc467746928)

[19.1 Appendix A - Deliverable Descriptions 42](#_Toc467746929)

[19.2 Appendix B – UPM Deliverables 45](#_Toc467746930)

[19.3 Appendix C – Agile Tooling 47](#_Toc467746931)

[19.4 Appendix D – Glossary 48](#_Toc467746932)

[19.5 Appendix E – Tailoring for Engineering Documents 49](#_Toc467746933)

[19.6 Appendix F – Decision Analysis and Resolution Plan 49](#_Toc467746934)

[19.7 Appendix G - Document Control 50](#_Toc467746935)

<Instrsuctions:

*This document should be modified with specifics in regard to this engagement’s tailored methods and tool-choices.*

* *Most sections have information provided as part of the template which can be used as-is or modified as per the project requirements. Help text provides guidance notes/examples (wherever applicable) to facilitate the same.*
* *If any section / sub section is not applicable to your engagement, please write ‘Not Applicable’ and delete any text in that section/sub section.*
* *The section headings from this document must not be deleted under any circumstances. If a particular section isn’t applicable then ‘N/A’ should be entered under the appropriate heading.*
* *<Blue text> means more information can be provided, in addition to the information already present in that section by default (in black). Red text means more information in the form of an example. After adding information, change font to black and remove “<>”*
* *<Refer to........> means reference to relevant document(s) should be given. Delete the original reference placeholder after adding the link to the document(s).*
* *Delete these instructions after preparing the document.>*

1. Project Governance Plan Overview

## 1.1 Purpose Of The Project Governance Plan

*The objectives of the Project Governance Plan are as follows:*

* *To give [the Client] confidence in the quality of the work that Capgemini will perform on the project.*
* *To define [the Client's] rights and obligations in relation to quality.*
* *To make visible all the means to be applied in meeting [ the Client's] technical and quality requirements.*
* *To provide the quality authority with information necessary to organise quality assurance and quality control activities, including transfer of information, verification actions, etc.*
* *To identify all the components to be used in the project; procedures, rules and applicable methods, etc.*

This Project Governance Plan provides a high-level view of the **<PROJECT>** project. It details the scope, objectives and overall approach for planning, designing, developing and implementing the solution for .

The Project Governance Plan provides for a common understanding between all project participants, helps manage expectations, and becomes the standard against which changes and deviations to process or product are identified and formally managed.

## 1.2 Scope Of The Project Governance Plan

*<*This section should describe the scope of the Project Governance Plan, making clear both what is covered by the document and what is not.>

The Project Governance Plan covers the processes, means and personnel for the following domains:

* the planning activities (project plan, effort , and delivery dates, etc.);
* the production activities (writing of a document, producing software, modelling, etc.);
* the management activities (project management, preparation of reviews, training, installing tools, configuration management, etc.);
* the verification activities (walk-through, review, audit, test by sampling, etc.).

<When the Project Governance Plan is first created, a full understanding of all phases of the project will not usually be available. A completion schedule should therefore be incorporated in this section, which identifies the dates planned for further additions.>

## 1.3 Control Of The Project Governance Plan

1.3.1 Preparation

*It is normally the Engagement Manager who prepares the Project Governance Plan. The Engagement Manager may choose to delegate this task to another resource, such as the quality authority. However, the ultimate responsibility for its completion and maintenance remains with the Engagement Manager. Also, as the Client must approve the Project Governance Plan, the Client should therefore be involved in its formulation, albeit through discussion, to agree the approach, and to partake in the composition.*

*The Project Governance Plan should always be a co-ordinated effort with a broad range of inputs and contributions, even if the Engagement Manager or one other delegated person is responsible for combining, moulding and presenting these inputs. The use of a table will simplify the summary of the activities and the roles responsible for the development and production of the Project Governance Plan, for example:*

|  |  |
| --- | --- |
| *PGP Development Activity* | *Responsible* |
| *DEFINE THE MODEL* | *Engagement Manager* |
| *Define contents of section 3* | *Team Leader 1* |
| *Define contents of sections 4.1 to 4.10* | *Team Leader 2* |
| *Define contents of remaining sections* | *Engagement Manager* |
| *Produce initial version* | *Documentation Specialist* |
| *Update the Project Governance Plan* | *Engagement Manager* |

1.3.2 Approval

*This section should describe the procedure for internal approval and external approval. The complexity of the approval process will be dependent upon the nature of the local Operation Unit procedures, the Client procedures and the project.*

The Project Governance Plan has received "internal approval" by the Capgemini Delivery manager and Independent Quality Advisor and has received "external approval" where applicable by the [client Engagement manager]

*When a Project Governance Plan is a contractual requirement, an authority stipulated in the contract (usually the Client) must perform the external approval. When this is not the case, the Client must clearly indicate to whom this authority is given.*

1.3.3 Updating

The Capgemini Engagement Manager is responsible for maintaining the PGP.

Once this PGP has been agreed, by signature of both Capgemini and [CLIENT] representatives (as applicable), it may be deviated from by either Capgemini or the [CLIENT], only under Engagement Change Management Procedures.

*As the Project Governance Plan contains elements which should not vary with time, only exceptional events (modification of the contract, deviations to the Project Governance Plan due to impossible application of its requirements, new production conditions, etc.) are grounds for modification of the Project Governance Plan. The Client needs to be informed of any modifications made to the Project Governance Plan during the life of the project and obtain client sign-off in case of contractual deliverable*

*The updating procedure should be described in detail. Depending on the Client requirement for a Project Governance Plan, the following type of information should be given:*

* *authority responsible for the updating process,*
* *version control definition,*
* *request for updating coming from the Client,*
* *request for updating coming from Capgemini (usually the Project Manager),*
* *decision-making process,*
* *the Capgemini responsibilities,*
* *the Client responsibilities,*
* *organisation of meetings for decision-making,*
* *recording of decisions,*
* *responsibilities for performing the update,*
* *information for the Client,*
* *re-issuing of the document,*
* *responsibilities associated with the follow-up of decisions,*
* *actual details of the update to be performed,*
* *approval of the update.*

## 1.4 Deviations From Standard Structure

Refer [Tailoring Guidelines](https://iqms.fs.capgemini.com/FSQMS/UPM-FSSBU/index.htm#UPM-FSSBU/customcategories/Tailoring%20Guidelines_E2026175.html) to identify the process/artefacts options available in iQ. Provide the selected options in the table below

|  |  |  |
| --- | --- | --- |
| **Process** | **Tailored Process** | **Remarks, if any** |
| Project Governance |  |  |
| Planning And Financial Management |  |  |
| Resource Management |  |  |
| Scope And Requirements Management |  |  |
| Engagement Change Management |  |  |
| Risk Management |  |  |
| Issue Management |  |  |
| Client Relationship Management |  |  |
| Supplier And Procurement Management |  |  |
| Communication Management |  |  |
| Infrastructure Management |  |  |
| Configuration Management |  |  |
| Quality Management |  |  |
| Knowledge Management |  |  |
| Decision Analysis and Resolution | Triggers identified based on project methodology and requirements |  |
| Causal analysis and Resolution | Triggers identified based on project methodology and requirements |  |

Mention the deviations applicabale to the project below

|  |  |  |  |
| --- | --- | --- | --- |
| Deviation No | Deviation Title | Deviation Link | Deviation Expiry Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 1.5 References

In this document there are references to the following documents:

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Document-Name | Remarks | Link |
| 1 | Contract **[Project Name]** | All contractual documents related to the engagement |  |
| 2 | CAF\_Contract Guidelines |  | https://troom.capgemini.com/sites/ESC/Shared%20Documents/Published%20Content/CAF/CAF\_Contract%20Guidelines\_v%201.0.pdf |
| 3 | CAF\_Distributed\_Delivery | The Distributed CAF chapter highlights the challenges of distributed delivery, defines a number of extensions and provides set  of best practices and guidance on how to overcome these challenges. | <https://troom.capgemini.com/sites/ESC/Shared%20Documents/Published%20Content/CAF/CAF_Distributed_Delivery_v1.0.pdf> |
| 4 | CAF\_MetricsGuide | This document describes the various metrics that can be tracked in agile engagements. The metrics  suggested in this document are in-line with the activities defined in Prepare, Build and Release iterations of  Capgemini Agile Framework (CAF). | https://troom.capgemini.com/sites/ESC/Shared%20Documents/Published%20Content/CAF/CAF\_MetricsGuide\_v1.0.pdf |
| 5 | Organisation Baselines and Goals | Refer to the FS goals for Agile metrics in this document. | https://iqms.fs.capgemini.com/documents/quality\_documents/others/Organizational%20Baselines%20\_%20Goals.xlsm |

1. Project Overview

This section provides an executive summary of the project, describing the client overview and the project overview.

* 1. Project Description

*<This section should comprise the definition and background of the project, including:*

* *the identity of the project,*
* *the name of the Client,*
* *a brief description of the Client’s activities in relation to the project,*
* *Pricing Mechanism (Fixed Fee, T&M, T&M with Capgemini responsibility)*
* *An explanation of the overall environment of the project.>*
  1. Project Objectives

*<This is a key section in which the fundamental objectives of the project should be specified. Both the Clients high-level and project-level objectives should be outlined. The high-level objectives, in the context of the business case, which will have identified the need for the project in the first place, should first be stated>*

*<State the summary of product vision. This should be the vision statement of a top executive or the sponsor from the customer organization.*

* *Example 1: Develop and rollout an e-commerce website that can sell merchandise in b2b and b2c models.*
* *Example 2: Build a mobile banking application that will enable retail customers to perform mobile banking transactions from their smart phones.>*
  1. Project Scope

*<Specify the subject matter boundary of the project, the subset of the organization that will be involved in the project, the time frame from project start to completion, and the deliverables to be produced during the project. The responsibility of cost management lies with the Engagement Manager and reference to Contract/SOW need to be given. Describe the purpose and scope of the project and clearly identify what is covered and exclusions, if any. Define In-scope and Out of scope sections explicitly for various kinds of scope e.g.: Functional scope, Technical scope, Geographic scope and Services scope.>*

*<Refer to the Contract>*

**Functional Scope**

**In-scope**: *<List all business needs in terms of features known at that time. If a tentative release planning is already done, then list the features against a corresponding planned release.*

* *Example 1: The consumer should be able to register to the e-commerce website*
* *Example 2: The consumer should be able to select merchandise from various categories such as electronics, apparel, furniture and gift articles.*
* *Example 3: The retail banking customer should be able to check account balance from his smart phone.>*

**Out of scope:** *<List all specific functional features that are out of scope of this contract. In some cases, the customer team or a 3rd party supplier team might be developing such features of related applications. Such cases should be explicitly mentioned.*

* *Example 1: On the e-commerce website, payment thru PayPal is out of scope.*
* *Example 2: On the e-commerce website, display list of 2 categories viz. Jewellery and Books is out of scope (They are probably planned during subsequent releases or being developed by other partners)>*

**Technical Scope**

**In-scope**: *<List all technical requirements including non-functional requirements known at that time.*

* *Example 1: The e-commerce website should be able to support the current 100,000 users and potential 500,000 users by 2016.*
* *Example 2: Concurrency aspect to be taken care when up to 5,000 users are logged on at the same time.*
* *Example 3: Only Microsoft stacks of software and tools to be used.*
* *Example 4: The architecture of the software system should follow TOGAF standards.>*

**Out of scope:** *<List all specific technical features that are out of scope of this contract.>*

**Services Scope**

**In-scope:** *<List all in-scope services known at that time.*

* *Example 1: Build & Unit testing, Integration testing, Performance testing, Implementation, UAT support of the product are provided as part of this contract.*
* *Example 2: Deployment support is provided during the product rollout. Warranty support is provided up to 3 months post deployment date.>*

**Out of scope:** *<List all specific services that are out of scope of this contract.*

* *Example 1: Database administration and Data testing are out of scope.*
* *Example 2: The documentation of end-to-end architecture of the application is not updated as part of this contract.>*

**Geographic Scope**

**In-scope:** *<List all geographic locations that the product or application will be rolled out.*

* *Example: This application will be rolled out to NA, France and UK locations. However, rest of the world will not be covered under the current contract scope>.*

**Out of scope:** *<List all specific locations that are not agreed upon.*

* *Example: This application will be not be rolled out to Mexico and Australia locations due to legal constraints imposed by the corresponding government regulations.>*

**Language Scope**

**In-scope:** *<List all languages in which the product will be deployed.*

* *Example: This application will be rolled out in English, French and Spanish languages in the scope of this contract.>*

**Out of scope:** *<List all specific languages that the product will not be deployed in.*

* *Example: This application will be not be deployed in any other languages other than those listed in the ‘In-scope’ section above.>*
  1. Project Critical Success Factors

A critical success factor is an internal, business-related item that is measurable and will have, on an ongoing basis, a major influence on whether or not an enterprise or process meets its objectives. These will often be identified in the original Clients Request for Proposal (RFP).

The critical success factors for the project/objective/milestones/waves etc. are:

Example:

* **Maintainable Solution:**  
  Objective: The objective is to develop an easy maintainable solution during the build Iteration.  
  Critical Success Factor: writing code and measuring code against a good coding standard.  
  Measurement: Not more that 10% of total defects found during build, should be related to coding standard.
* Xxxx
* Xxxx
  1. Contractual Deliverables

The following table defines the contract deliverables to be produced throughout the project. Contract deliverables require a formal review and acceptance by **[CLIENT]**.

Refer to the Contract for the contractual deliverables.

|  |  |  |
| --- | --- | --- |
| Workflow | Deliverable | Responsible Party |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. Contractual Milestones

*<The contractual milestones, which the project will deliver, should be described in this section. The following about project milestones should be shown in the Project Plan:*

* *Start Date.*
* *Contractual deadline.*
* *Intermediate dates linked to validation, reviews, visible stages, etc.*
* *Events linked to Client's obligations, such as providing equipment, approving deliverables, payment tied to a milestone etc.*
* *Delivery dates.*
* *Go/no-go decision points.>*
  1. Client Obligations

*<List the general obligations of the client. Cross reference to the Contract if appropriate. Consider:*

* *Provisioning of working environment*
* *Provisioning/maintenance of development environment*
* *Provisioning/purchasing of licences and software*
* *Making key personnel available*
* *Training, if not already covered by the project*
* *Maintenance of the live system, if not already covered by the project*
* *Operation of the live system, if not already covered by the project>*
  1. Assumptions And Constraints

*<In this section all key assumptions made about the project should be listed. The initial list may be taken from the proposal and supporting documentation. It is particularly important to include all assumptions relating to deliverables, resources, and other factors to be supplied by the Client, upon which the project will be dependent. These dependencies should be clearly documented and carefully monitored.>*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Assumption No. | Assumption and Assessment of Impact if Assumption is Invalid | Action  if Assumption is Invalid | Responsible for Validation | Estimated Date when Validity of Assumption Needs to be Confirmed |
|  | Assumption:  Impact if Invalid: | e.g.  Action/Issue/Risk added in Action Log / Issue Log Register/Log |  |  |
|  | Assumption:  Impact if Invalid: |  |  |  |
|  |  |  |  |  |

*<This section should also include constraints. Constraints may be ‘external’, that is to say, those imposed by Client, for example, limitations on the hardware to be used, or restricted access for Capgemini personnel to areas or information. There may also be ‘internal’ constraints, that is to say, those originating from Capgemini, for example, limited availability of personnel.>*

|  |  |
| --- | --- |
| Constraint  No. | Constraint |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* 1. Dependencies

This section defines the known external dependencies associated with this project i.e. dependencies this project has with other projects, areas of work, people or events outside of this project.

**a) Projects/Events that this Project is Dependent On**

*<Delete this section if not required>*

|  |  |  |  |
| --- | --- | --- | --- |
| Provider (Details of who/what this project is dependent on) | | | |
| Provider Id | Name of Provider | Description of Dependency | Date Required |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**b) Projects/Events that are Dependent on this Project**

*<Delete this section if not required>*

*<If the Dependencies have not been defined in the contract refer to the standard draft contract schedule for the areas to include in this section. Agree the Dependencies with the Delivery Manager (DM).>*

|  |  |  |  |
| --- | --- | --- | --- |
| Receiver (Details of who/what is dependent on this project) | | | |
| Provider Id | Name of Provider | Description of Dependency | Date Required |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

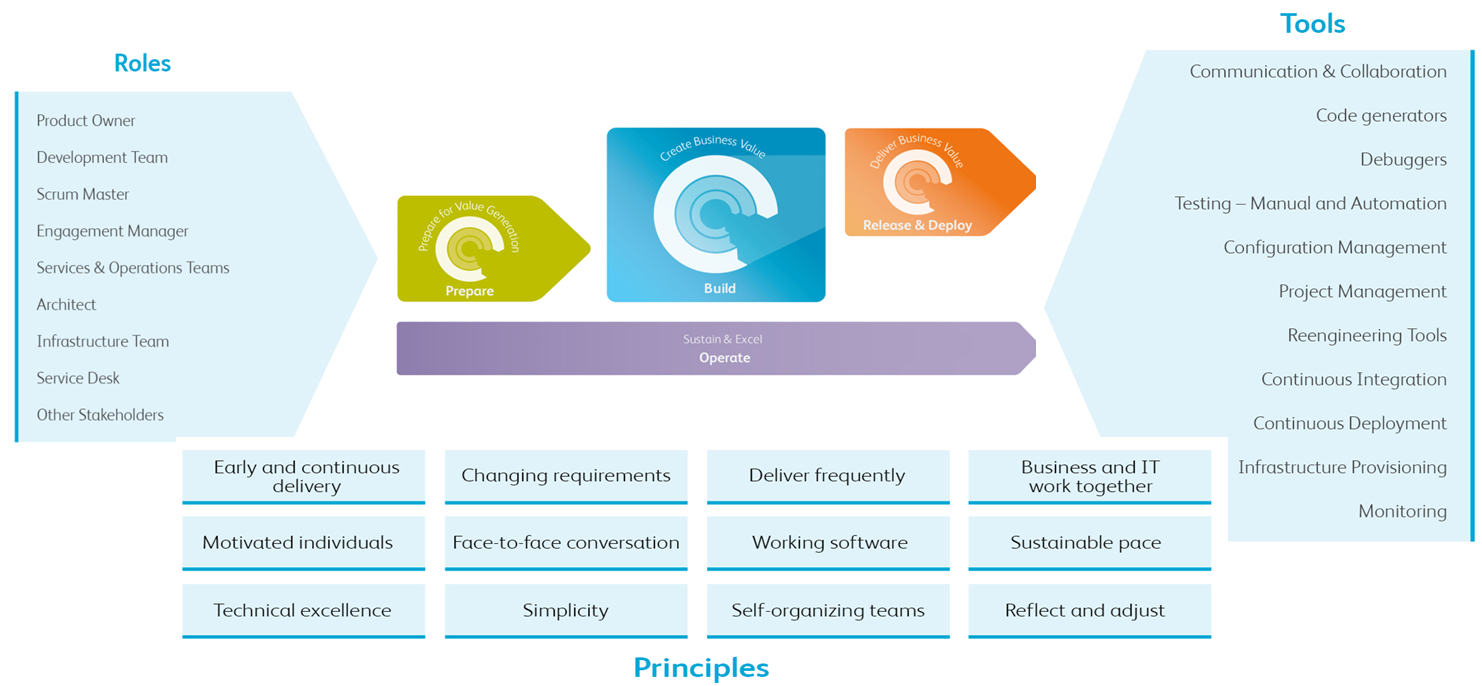
1. Project Approach

The project delivery approach is based on the Capgemini Agile Framework (CAF), which is Capgemini’s standard way of delivering agile projects. CAF is aligned with UPM

*< If the project delivery approach has not been defined clearly in the contract this must be raised with your Delivery Manager and the way forward jointly agreed.>*

* 1. Iterations And Activities

*<Refer to the Capgemini Agile Framework (*[*CAF*](http://deliver2.capgemini.com/components/Capgemini_Agile_Framework/#Capgemini%20Agile%20Framework/guidances/supportingmaterials/CAF%20Overview_B368AD2D.html)*) for Iteration types and Activities.>*

**

The CAF lifecycle includes an initial Prepare, followed by iterative Build & Release. For each of these, a set of activities are defined.

* **Prepare**, the objectives of which are to, validate the project scope, establish the environments required at this stage of the project, identify functional and technical risks & risk mitigation strategy, validate the initial estimates and the high-level Product Backlog plan and to develop a candidate high level design for the Solution.
* **Build**, the objectives of which are to incrementally design, develop and test the remaining functionality of the Solution (that can be developed within the remaining budget, the project timescale and the existing resources).
* **Release & Deploy**, the objectives of which are to gain final Acceptance by Client of the Solution, to implement the Solution and handover the Solution to the Support Organisation.
* **Operate**, the objectives of which are to maintain and operate the application – monitor services, create SLA reports etc.

*< Some of the deliverables in Prepare will be repeated iteratively/incrementally in Build and Release. Define which* [*life cycle pattern*](http://deliver2.capgemini.com/components/Capgemini_Agile_Framework/#Capgemini%20Agile%20Framework/deliveryprocesses/CAF%20Lifecycle_A86CFAF2.html) *is applicable for this project.>*

* 1. Definition Of Done

*<Provide “Definition of Done” - an agreed definition of criteria that must be satisfied by the team in order to complete the Product Backlog items and for the Product Owner to accept them. Refer to the Definition of Done* [*guideline*](http://wiki.capgemini.com/wiki/index.php/Agile:DefinitionOfDone) *and* [*example*](http://deliver2.capgemini.com/components/Capgemini_Agile_Framework/Capgemini%20Agile%20Framework/guidances/guidelines/resources/Definition%20of%20Done.docx) *in CAF.>*

*<Example DoD:*

*A (Smart) Use Case is ‘Done’ when the following is applicable for the:*

*Requirements:*

* *The (Smart) Use Case is worked out in the design tool, both the basic and the alternative fows;*
* *The Smart Use Case is available as a Word document (generated from the tool;*
* *The non-functional requirements which the (Smart) Use Cases should satisfy are worked out in the Software Architecture Document (SAD);*
* *The Smart Use Case is geared to the Product Owner;*
* *Agreed Output Documentation is updated;*
* *.......*

*Functional Acceptance Test:*

* *For every Smart Use Case that is detailed in the design tool, there is a test script available.*
* *The Smart Use Case is tested by a tester in the test environment and the test is walkthroughed successfully (no blocking or serious errors).*
* *For every Smart Use Case that is marked in the tool for Automatic Testing, an automated test script is available and added to the regression test.*
* *The regression test (all automated test scripts) is executed and this test has been successfully walked through (all green).*
* *.....*
  1. Definition Of Ready

*<Provide ‘Definition Of Ready’ - an agreed definition of the readiness criteria for a Product Backlog item, which must be satisfied for the team to start development (i.e. to be able to convert Product Backlog item into a working solution. Refer to the Definition of Ready* [*overview*](http://www.romanpichler.com/blog/the-definition-of-ready/) *and* [*guideline*](http://systemagility.com/2011/05/17/definition-of-ready/) *in CAF.>*

*<Example:*

*A Smart Use Case is Ready when the following is applicable:*

* *The Size in (Points) is known*
* *There are no open functional issues*
* *There is a trace to the requirements*
* *A short description is available*
* *It is known what test types are necessary (determined during poker sessions)*
* *It is known who are the subject matter experts*
* *Agreed Output Documentation is updated>*
* *...*
  1. Deliverables

Refer to the Contract/Proposal for an overview of the deliverables.

Appendix A contains a Deliverable Description for all deliverables from the project, both contractual and non-contractual. The method of delivery to [CLIENT] is defined in that description.

Below is an example of the recommended deliverables for an engagement that fully follows the CAF lifecycle. These are Capgemini deliverables.

|  |  |  |
| --- | --- | --- |
| Iteration Type | Deliverables | Frequency |
| Prepare\*\* | Staffing Plan | Beginning of every Project |
|  | Risk Register / Risk Log | Weekly |
|  | Definition of Ready for Product Backlog | Every iteration |
|  | Definition of Done for Product Backlog | Every iteration |
|  | Stakeholder Map | Beginning of every PIR\*\*\* |
|  | Coding Standards | End of first PIR |
| Build | Code | Every iteration |
|  | Burn-down Chart | Anytime |
|  | Product Documentation (Relevant) | Every iteration |
|  | Team Capacity\* | Beginning of every iteration |
|  | Iteration Backlog | Every iteration |
|  | Metrics, as applicable | Every month |
|  | Improvement plan (Action items from retrospectives) |  |
| Release & Deploy | Bundled Package | Every release |
|  | Product Documentation, as applicable | End of every release |
|  | User Manual | End of every release |
| Operate | SLA reports | Continuous |
|  | Defect fixes | Continuous |
|  | Metrics, as applicable | Continuous |

\* Team Capacity: Number of team’s available hours for the iteration. The hours for vacation, leaves, and holidays must be reduced while determining team capacity.

\*\* Some of the deliverables in Prepare will be repeated iteratively/incrementally in Build and Release. For example, Risk Register needs to be updated and delivered at the beginning of every iteration.

**\*\*\*** Product Increment Release (PIR) – A set of iterations that consists of Prepare, Build and Release iterations - the number of iterations depends on the project complexity. The objective of each Product Increment Release is to deliver a product increment that adds tangible business value to the user.

**Mandatory UPM deliverables:**

Please refer to Appendix C – UPM Deliverables for a list of mandated UPM deliverables. Though the list has all mandated deliverables as per UPM, only the relevant deliverables need to be included in the final Project Governance Plan.

The following is a list of deliverables that the Client would provide to enable Capgemini to perform the services listed in Section 3.4.

|  |  |  |
| --- | --- | --- |
| Iteration Type | Deliverables | Frequency |
| Prepare\*\* | Product Vision Statement | Beginning of every PIR |
|  | Product Backlog | Every iteration |
|  | High-level Product Release plan, if any | Beginning of every PIR |
|  | Project environments (such as Dev, SIT, UAT etc) where not provided by Capgemini | Beginning of every PIR |
| Build | Formal Acceptance of build iteration deliverables | Every iteration.  Specify the maximum delay for acceptance |
|  | Test data | Every iteration |
| Release & Deploy | Formal Acceptance of release deliverables | Every release iteration.  Specify the maximum delay for Release acceptance |
| Operate | Metrics, as applicable |  |

\*\* Some of the deliverables in Prepare will be repeated iteratively/incrementally in Build and Release.

* 1. Verification And Validation

*<This section should list the verification activities that have to be set up for assuring that the results of the Iterations meet their quality requirements. These requirements should be consistent with, but not restricted to, the Client's quality requirements and acceptance criteria.>*

Appendix A contains a Deliverable Description for all deliverables from the project, both contractual and non-contractual. The method of verification and validation by Capgemini is defined in that description.

* 1. Notification Of Deliverables

Appendix A contains a Deliverable Description for all deliverables from the project, both contractual and non-contractual. Delivery arrangements are defined in the Deliverable Description.

* 1. Replication

*Appendix A contains a Deliverable Description for all deliverables from the project, both contractual and non-contractual. Any method of replication (or reproduction) to ensure that a replicated object is identical to the master and of the same quality is defined in the Deliverable Description.*

* 1. Installation And Implementation

*<If installation is to be performed within the project, provide details of the installation process.>*

* 1. Servicing

A Handover Plan to the ongoing support organisation *< Support Organization Name>*, will be provided as defined in Appendix A.

If there is to be any subsequent service management provided by Capgemini, the relevant service management method to be implemented should be specified here <e.g. Unified Service Management (USM), Application Maintenance (AM) method or ADM iMethod>. The procedures for knowledge transfer and for formal hand-over should also be stated in such case.

1. Project Governance
   1. Project Management Method

The project will be managed using the Capgemini Agile Framework (CAF). CAF is compliant with the Unified Project Management (UPM) method.

*<In this section a reference should be made to the Capgemini Agile Framework quoting the version number. Any tailoring or deviation applied to the method, to meet the specific needs of this project, should be detailed here. This project is governed based on processes defined in the iQ, utilizing Capgemini Agile Framework. If the client has mandated a specific methodology to be followed then the team follows the steps given in tailoring and deviation method for client mandated process and raises an appropriate tailoring or deviation.*

*Capgemini Agile Framework (CAF) is integrated with UPM guidelines and available on Deliver. Any further tailoring done to UPM to suit Agile/CAF way of work should be explicitly described here>*

* 1. Project Organisation

The following diagram summarises the links that connect the parties involved in the project, and indicates the specific functions associated with the project:

*<Insert Chart/Diagram>*

* 1. Project Roles and Responsibilities

*<This section should list, for each project stage and activity identified, the personnel (indicated by job title, not by name) responsible, in terms of those accountable, those responsible and those consulted. Other levels of responsibility, such as those who verify, inform or sign-off can be indicated as appropriate. This may be done in the form of* [*roles and responsibilities*](http://deliver2.capgemini.com/components/Capgemini_Agile_Framework/Capgemini%20Agile%20Framework/guidances/templates/resources/CAF_Roles_Ceremonies_v1.0.pdf) *matrices, in the Capgemini Agile Framework.>*

Typically the structure includes, the Development Team, Scrum Master, Product Owner, Engagement Manager, Proxy Product Owner, Services & Operations teams and other stakeholders like: customer sponsor, customer IT managers, functional managers, end-users, SMEs, services teams, SQAs, infrastructure teams, finance teams and legal teams.

Ensure that members of the development team can fulfill the following roles of: Requirements Analyst, Solution Architect, Lead Programmer, Programmer, Tester and Configuration Manager.

**Project RACI:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Activity | Responsible | Accountable | Consulted | Informed |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

This section should also state the purpose of the Internal Steering Board and identify each of its members. The purpose of the Internal Steering committee is to ensure that the appropriate Preparation tasks have been completed and to direct the management of potential risks, problems, issues or concerns that may impede the successful completion of the project.

The composition of the Internal Steering Board may be represented in a table format or graphically, in a chart or overview diagram and it is advisable to include role and contact details of each member.

* 1. Project Monitoring And Reporting

This section details the reporting structure / mechanism that is followed in the project. This would carry details around the sender, the receiver(s), contents that would be shared, frequency and mode of distribution and so on.

* + 1. Project Monitoring

Project progress is primary based on the [Burn-up](http://deliver2.capgemini.com/components/Capgemini_Agile_Framework/#Capgemini%20Agile%20Framework/guidances/templates/Iteration%20Planning%20And%20Backlog%20Example%20With%20Burndown%20Chart_97EC890D.html) of the project, a chart that shows how fast we are going within an iteration; the burn down chart shows the progress of work that is done. It tends toward a straight line intersecting the axis at the end of the iteration. The slope of the line can soon show a predicted intersection elsewhere, indicating that remedial action may be necessary. They can be used to predict intersection with other future events not just end of iteration.

A burn-up of the used budget can be used to check if the project stays withing budget.

The tool used for project monitoring is: <Name Tool>.

* + 1. Project Reporting

*<Example>*The following reports could be provided. *<Adjust the table where applicable to determine which reports will be delivered.>*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Report | Objective | Delivered To | Delivered By | Frequency |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

* 1. Meetings

This section describes the various regular key meetings that shall be held during the project. There could be various types of meeting such as technical meeting, status meeting, query resolution meeting, causal analysis meeting, knowledge sharing meeting, stand up meeting, retrospection meeting and so on which could be listed as part of this section.

<Example:> The tables below describes the various meeting structures with objectives, frequency and participants. *<Table given below is an example, update the table as per applicability to project>*

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting Name | Focus / objectives | Frequency | Participant |
| Steering Committee | * Coordinate contractual consequences of decisions and actions * Steer the project and monitor project goals * Decision taking on exceptions (included RFC) | * Monthly * Ad-hoc | * Members of Steering Committee * EM |
| Progress Meeting | * Discuss progress, issues, risks, quantitative aspects * Prepare Steering Committee meeting * Practical issues | Two-weekly | * <Name/Role Client> * EM |
| Daily Stand-up Meetings | * Synchronize individual’s work * Report to each other on obstacles * Plan activities for the next 24 hours | Daily | * Delivery Team * Scrum Master |
| Iteration Planning | * Plan the work to be executed during a Iteration * Estimate user story | Start of Iteration | * Delivery Team * Product Owner * Scrum Master |
| Iteration Review | * Inspect what has been built in the Iteration * Adjust the Product Backlog | End of Iteration | * Delivery Team * Scrum Master * Product Owner |
| Iteration Retrospective | * Inspect oneself, how well Iteration went on * Plan how to improve oneself during next Iteration | End of Iteration | * Delivery Team * Scrum Master * Product Owner |
| Backlog Refinement Session | * Refine back log items so that they can be picked up during next Iteration | Weekly | * Delivery Team * Scrum Master * Product Owner |
| Campfire Meeting | * Agree on details of work items in the stream | Weekly | * Delivery Team * Scrum Master * Product Owner |

* 1. Client Billing

*<Amend details as appropriate>*

Amend this section to ‘Invoices will be sent as defined in the Contract if invoicing arrangements are clearly specified in the Contract (or perhaps the overall Programme Plan if this project is part of a programme)

* 1. Supplier Invoices

Supplier invoices received by the project are checked by the Engagement Manager. Valid invoices are passed for payment authorisation and payment within Capgemini.

* 1. Monitoring Of Actions

Actions arising from meetings, reviews, audits, risk management, problem management are reviewed on at least a monthly basis by the Engagement Manager.

* 1. Client Sign-Off And Acceptance

Reflect/cross reference the Contract if appropriate.

The acceptance criteria against which the deliverables will be measured are described in the Deliverables section of Appendix A.

Acceptance criteria should be defined at a feature level as well as the contract level (as applicable) along with the Definition of Done.

<Examples of Acceptance Criteria:

* Example 1: Every feature must be tested and successfully pass the UAT.
* Example 2: The product increment is working as per the original description and has no more than 5 minor and 1 major defects.
* Example 3:
* **Acceptance of Work done by Capgemini**  
  Work done by Capgemini team members is accepted by monthly signing the time sheets by <Name / Role Client>.
* **Acceptance of Functionality**During every Iteration the <Client Name> is executing an acceptance test on the delivered Smart Use Cases/Work Items. After that the Product Owner decides if the Smart Use Case actively complies to the Definition of Done. In case it complies, the status on the dashboardchanges to Done (accepted). In case it does not comply, to change the status of the Smart Use Case before the end of the Iteration to ‘Done’, then the item will stay on the product backlog. The item will not be made available for users.
* **Decharge to the EM for the project**The EM will be decharged from the project if fase <Iteration Name> of the <Project Name> project is completed.>
  1. Governance Handover

*<Provide a brief description of how and to whom the project will be handed once the team’s responsibilities have been completed. This may be to a Capgemini team or to the client. It may be, for example, to a testing team or a support team.>*

**Handover Roles and Responsibilities:**

Handover roles and responsibilities will be defined and documented during the Implementation Iteration.

* 1. Warranty

*<These details are covered in the contract and could be stated here, in case of no warranty state here has no warranty provided by client.>*

1. Planning & Financial Management
   1. Estimating (Sizing)

*<State the estimation model that will be used by the project to arrive at size, effort, cost (if applicable) estimates and also arrive at schedule estimates using effort and resource availability and task dependency/constraints etc.*

*If the Capgemini Agile Framework doesn’t satisfy the project estimation model needs or if there is a client mandated estimation model that needs to be used, it shall be done by taking a deviation in Deviation Management System.>*

*<Refer to ‘*[*FS Story Point Estimation Tool*](https://km3.capgemini.com/system/tdf/book/2012/12/14/574514/FS%20Story%20Point%20Estimation%20Tool.xlsx?file=1&type=document)*’ technique for size estimation of user stories.>*

* 1. Project Plan

*<Amend details as appropriate. Describe the procedure used for managing the Project Plan and the tasks contained therein. It may be necessary to be explicit about how the plan is to be incorporated into an overall programme>.*

*<The Project Plan consists of a Release Level Schedule (two-level plannings) and Iteration Plannings.*

*Releases are either planned on the basis of time (a business critical date) or on the basis of features (a minimum usable set of features).*

*Iteration planning is done on iteration level. At iteration level, once a set of backlog items have been selected, the work items/tasks needed to implement them (to get to the definition of done) are sized in hours (Actual Effort). Also work items for sourcing the project, meetings and backlog refinement are part of the iteration Planning. Iteration Planning is supported by* [*burn down charts*](http://wiki.capgemini.com/wiki/index.php/Burn_Down_Chart)*.*

* 1. Project Cost And Budget Management

*<Amend details as appropriate. Define processes for controlling the budget as cost, revenue and margin. Revenue to complete will not be from the current Project Plan if the project is fixed price, it will be the remainder of the agreed budget}*

1. Resource Management

The Capgemini Engagement Manager will liaise with Capgemini’s Resource Management Team to obtain resources to deliver the project. [CLIENT] will be advised, at project meetings, of planned project joiners/leavers

*<Amend details as appropriate. If the team is joint Capgemini/client, the Capgemini Project Manager may be asked to provide appraisals for the client staff. If this is the case, the details of the appraisal process for the client staff must be included. The activity must be included in the Project Plan. If the project is fixed price and this is not a contractual deliverable or is not included in the budget, then Change Request must be raised by the Capgemini Project Manager to cover the cost of this activity. It’s the responsibility of the EM to ensure that the team with the required capabilities and capacity is in place and ready to work*. *Refer to FSSBU UPM Stream ’Resource Management’.>*

*Team Information/details will be maintained using* [*Team Information Sheet*](https://iqms.fs.capgemini.com/FSQMS/UPM-FSSBU/index.htm#UPM-FSSBU/guidances/templates/Team%20Information_D3536DC3.html)

* 1. Team Organisation

*<Insert a detailed organisation chart or refer to section 4.2 of this document if this includes the whole project team.>*

* 1. Team Roles and Responsibilities

The following table describes the responsibilities of the roles within the project team:

A sample of a table to populate is shown below. Check the table and text below against your contract and the project and delete or update as needed.

[*<Refer to Roles & Responsibilities in CAF>*](http://deliver2.capgemini.com/components/Capgemini_Agile_Framework/#Capgemini%20Agile%20Framework/rolesets/Roles_DF090A46.html)

|  |  |
| --- | --- |
| Role | Responsibilities |
| Development Team |  |
| Scrum Master |  |
| Product Owner |  |
| ....... |  |
|  |  |

* 1. On-Boarding

New Capgemini team members are given a brief overview of the project by the Capgemini Engagement Manager <and Technical Manager for technical aspects of the project >.

* 1. Training and Coaching
     1. Capgemini Team Members

The Capgemini Engagement Manager is responsible for ensuring that Capgemini team members have received the necessary training to complete their tasks, and that coaching is given where necessary.

* + 1. Client Team Members

The [CLIENT] Engagement Manager is responsible for ensuring that the [CLIENT] Project team members receive appropriate training to enable them to complete their tasks.

* 1. Team Member Evaluations

The Capgemini Engagement Manager will ensure that Assignment Details are produced for each Capgemini team member. An Assignment Appraisal will be held either every six months/year according to his or her grade and/or at the end of the assignment for each Capgemini team member.

The Project Team Leads are responsible for holding the Assignment Appraisals for their team members. The Engagement Manager holds the Assignment Appraisals for the Project Team Leads.

* 1. Team Member Release

The Capgemini Engagement Manager is responsible for assessing the impact of a team member release. Before a team member is released, the Engagement Manager makes sure the Assignment Appraisal, knowledge capture, and time records are completed. Prior to releasing a team member the Capgemini Engagement Manager (or nominated Work stream lead) will discuss the reasons for leaving with the [CLIENT] Engagement Manager / Product Owner. When a team member is released the Engagement Manager will communicate the release to the [CLIENT] Engagement Manager /Product Owner and to the appropriate Capgemini business unit. At the end of the project all team members will be released to their respective Capgemini business units.

1. Scope And Requirements Management

*<This section defines the agreed scope and how traceability of requirements will be maintained throughout the project. If this is not fixed, then it needs to be managed specifically for each assignment. However, this section shall define the procedure to be followed for agreeing on the scope, base-lining it, and maintaining bi-directional traceability of requirements.>*

* 1. Requirements Traceability

Requirements are documented in the requirements repository of the tool <Tool Name>.

The deliverables that are part of the Scope Baseline are kept in the <Tool Name>, and are placed under configuration and change control.>

1. Change Control

*<This section gives guidance on how to develop Change Control Procedures and to manage changes on a Capgemini project. Change Control allows for proper handling of changes to fundamental aspects of the project such as Scope, Requirements, Information Security, timescales, costs, and ultimately contract.>*

Capgemini Engagement Manager will be accountable for setting up and monitoring the change management framework. The Product Owner will be accountable for (re-)prioritization of the changes in the Product Backlog. The Scrum Master has a supporting role towards the Product Owner and is responsible for notifying the Engagement Manager when changes will impact contractual agreements.

<Example:

There are two types of changes, namely changes for which the Product Owner thinks it is required to create a Request For Change (RFC) and changes for which it is not required.

Changes without a RFP are directly submitted on the Product Backlog, e.g.:

* Addition of a Use Case (+)
* Changes on a Use Case that has not been realized yet (+/-) or on a Use Case that already has been realized (+).
* Removal of a Use Case that has not been realized yet (-) or on a Use Case that already has been realized (+).

An increase of the Product Backlog (+) means that it is possible that one or more Uses Cases cannot be realized. A decrease (-) just offers more capacity.

Changes for which there is a RFC are running through the following steps:

1. *Identify the Change:* The change requester describes the required adjustment. After that the Scrum team gives an estimation of the time needed for the impact analysis.
2. *Impact Analysis:* After approval of the Product Owner of the estimation of the impact analysis, the team updates the RFC with the impact. This includes the scope and solution of the change, the effort needed and other costs to realize the change and the impact of the change on the planning.
3. *Decision about Change:* The Product owner decides based on the impact if the change will be executed or not. If the decision is positive the change is added to the Product Backlog.
   1. Impediment Procedure

The impediment procedure will come into effect, on request of the Scrum Master, on the moment that the Scrum Team is blocked to deliver the committed product backlog items for the iteration. By mutual agreement the reasons and causes are discussed by Scrum Master and Product Owner. If needed, the Engagement Manager will be involved for additional actions which are considered out of scope of the Scrum Team.

Decision is taken leading to one of the following situations:

1. *take measures to prevent or cancel out the reason.*
2. *Decide to do not take any action, because the tolerance will not be exceeded.*
3. *Increase/decrease the planning or budget*
4. *Take consession with regards to the scope of what needs to be delivered.*
5. *Involve the Steering Committee if applicable to take decisions, because financial and/or planning consequences are to big.*

*The first four situations are stated in the next progress report. In the last situation a (new) steering committee meeting is planned to take a decision about continuation of the project and any further conditions for continuation.*

1. Risk Management

*This section should specify the procedure and responsibilities with regard to each of the four principal elements of Risk management: risk identification, risk assessment/ analysis, risk containment / management (mitigation, contingency) and risk monitoring. The procedure should be consistent with the* ***unit/region*** *Risk Management Procedures. In particular the following aspects of the procedure should be highlighted:*

Risk assessment and monitoring will take place at regular intervals throughout the project. The tasks are:

|  |  |  |
| --- | --- | --- |
| Task | Timescale | Responsibility |
| Initial Risk Assessment | Build Iteration | Capgemini Engagement Manager/ Scrum Master |
| Subsequent Risk Assessment | Weekly / on scope change/ beginning of each Iteration | Capgemini Engagement Manager/ Scrum Master |
| Technical Risk Assessment | Daily as Required/ During stand up meeting | Capgemini Engagement Manager/ Scrum Master |

The Risk Register will be maintained using Unified Risk Register /Clarity.

Where necessary, the Project Plan will be updated to reflect the impact of any agreed containment actions.

* 1. Business Continuity Planning

*Account needs to prepare its Business Continuity Plan to ensure continuity of project services in the event of a critical incident to any of the operational facilities.*

1. Issue Management

Procedures for Issue management are described within FSSBU UPM (Unified Project Management) method.

The responsibilities / timescales for effecting these procedures are:

|  |  |  |
| --- | --- | --- |
| Task | Timescale | Responsibility |
| Raise Issue Report | As they occur | Any person working on or related to the project |
| Assessment/ management of issues | Weekly/periodically | Capgemini Engagement Manager  Capgemini Project Team Lead |
| Reporting | Periodically | Capgemini Engagement Manager / Client Engagement Manager |

Escalation of Issues will be as follows:

|  |  |
| --- | --- |
| Type | Escalation to Whome |
| Escalation within Client | Scrum Master to Product Owner |
| Product Owner to Client Sponsor (in case internal issue)  Product Owner to Project Manager (e.g.technical security or resourcing issue) |
| Client Sponsor to Client Manager (internal)  Project Manager to Client Manager |
| Escalation within Capgemini | Scrum Master to Capgemini Project Manager |
|  | Capgemini Project Manager to Capgemini Delivery Manager |
| Capgemini Delivery Manager to Capgemini Delivery Director |
| Capgemini Delivery Director to Intergroup escalations (DHO, Finance SSC, HR, STAFFING, ITICS,L&D) |

1. Client Relationship Management & Communication
   1. Client Profile

<\**Provide a brief summary of the client and where the client representatives are based.>*

* 1. Client Steering Board Description

*<If the Client Steering Board has not been defined in the contract refer to the standard draft contract schedule for the areas to include in this section. Ensure the way forward is agreed with the Delivery Manager>*

* 1. Client Kick-Off Meeting

The Kick-Off meeting will be held on / was held on xx in xx. The objective of the Kick-Off meeting is /was to communicate the objective and the high level approach of the project.

* 1. Client Satisfaction And OTACE

*<If it is agreed with the Capgemini Account Manager and Delivery Manager that no OTACE will be carried out for this project, state that there will be no OTACE for this project. This may be appropriate for a small/short project.>*

It is Capgemini policy to measure Client Satisfaction throughout the life of each project so that [CLIENT]’s aims and objectives are understood. This process also provides feedback on areas that are going well and those areas where improvements are needed. The Capgemini program is entitled OTACE (On Time and Above Client Expectations) and procedures are described in the OTACE rules and Client overview. OTACE results will be distributed between the EM and the Steering Committee members.

{If OTACE for this project is encompassed by OTACE for the Programme/Service within which this project fits, delete the following and refer to the Programme/Service level OTACE}

{Describe any other client satisfaction methods employed on the project e.g. end of project survey}

.

The responsibilities / timescales for affecting these procedures are:

|  |  |  |
| --- | --- | --- |
| Task | Timescale | Responsibility |
| Initial criteria defined | During Preparation Iteration | Capgemini Account Manager/ Capgemini Delivery Manager & <Client> Product Owner |
| Assessment | {at major Milestones or max every 3 months and at the end of the project} | <Client>Product Owner |

1. SUPPLIER AND PROCUREMENT MANAGEMENT

*<If there are no Third Party Suppliers involved in the project, state that there are none. If it aids clarity, include the section if the client is responsible for Third Parties but restate that their management is the client’s responsibility. Other Capgemini Units may be regarded as any other external supplier or subcontractor. However, in general, the client should perceive Capgemini as a whole unit and care should be taken to avoid showing Capgemini as fragmented.>*

* 1. Supplier Selection
  2. Supplier Agreements
  3. Supplier Tracking
  4. Supplier Audits
  5. Supplier Billing

Supplier bills received by the project are checked by the Project Manager. Valid invoices are passed for payment authorisation and payment within Capgemini.

* 1. Supplier Performance
  2. Supplier Acceptance

1. Communication Management

Communication Management will establish and implement effective communication to the [CLIENT], Capgemini and Suppliers in the progress and overall outcome of the project.

* 1. Progress Meetings

*<The frequency and responsibility of planning and conducting the progress meetings are to be specified.>*

* 1. Correspondence Handling

*<All memoranda, letters and faxes to the [CLIENT] or received from the [CLIENT], will be logged and filed within the Project Repository.>*

*<If email is used to communicate with the [CLIENT], all major incoming and outgoing emails that contain important project-related decisions or information will be filed within the Project Repository>*

* 1. Communication Plan

To ensure that the project communicates with all parties, internal and external, effectively a Communication Plan will be written, agreed and followed.

Also Describe how to communicate within the team and which tool/media to use for Information exchange.

* Documentation::
* (Smart) Use Cases:
* Progress
* ….

The Scrum Master communicates with the Product Owner. The EM communicates with the <Name/Role Client>.

* 1. Inter-group commitments

|  |  |  |  |
| --- | --- | --- | --- |
| Item/Commitments/ Dependencies | Provided By | Planned Duration | Remarks |
| Required Associates | Staffing | Across Project when ever required |  |
| Desktops / Laptops | ITICS | Across Project when ever required |  |
| Visa Processing | People Processes | Across Project when ever required |  |
| Travel arrangements | ISA | Across Project when ever required |  |
| Security audits/updates(As applicable) | BRiCS | Across Project when ever required | Capgemini Security Manager |
| Standard Processes & Templates | PEG | Across Project when ever required | Process Improvement & Excellence Group |
| Environment infrastructure | Respective Support teams | Across Project when ever required | *Refer to* [**Work Environment Standards**](https://iqms.fs.capgemini.com/documents/support_functions/india/process/work_environment_standards/Work%20Environment%20standard.xls). |
| Internal Audits |  | Every … months |  |
| Metrics Data | Metrics Group (PEG) | Monthly |  |

* 1. Grassroots Support

For a high acceptance level it is important that resources of <Client Name> in the project and possibly other stakeholders are sufficiently involved with the project.

< Describe the Grassroots Support procedure>.Teaming

Make agreements about how to effectively and efficiently deal with each other in the engagement to simplify achievement of the goals.

* *Quickly give feedback*
* *Be directly and have an open mind*
* *Quickly look for agreement*
* *Quickly look for support of other project members if you don’t know how to proceed*
* *Be clear about your attendance*
* *Talk about facts*
* *Focus on the end goal*
* *Prevent surpises*
* *Celebrate successes*
* *Good is Good enough*
* *The team creates a procedure for expressing wishes and ideas from outside the team, and submission for the benefit of the Product Owner.*
* *It is the ambition of the Product Owner to change the status of a story to ‘Done”shortly after delivery.*

1. Infrastructure Management

The project will use various sorts of infrastructure, For instance the OTAP, Cloud and/or Premise environment.

Capgemini standard tooling will be used to submit and maintain all items that will used within the engagement. Refer to appendix D: ‘Agile Tooling’, for a list of recommendable tools.

* 1. Project Management Office Infrastructure

*<This section details the infrastructure such as tools, hardware, software, access required etc. required for this project. These details may be summarized in a tabular format. Details of Tools from all the streams are identified as part of this section.*>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Category (Hardware Software / Tools) | Dependencies | Schedule  (Start Date / End Date | | Status | Acceptance Criteria | Responsible Group |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

* 1. Project Delivery Infrastructure

*<An overview of the development environment of the project would be useful with an architecture type of diagram. List all hardware used in the delivery environment. Note: where hardware items are remote from the team location. Refer to* [**Work Environment Standards**](https://iqms.fs.capgemini.com/documents/support_functions/india/process/work_environment_standards/Work%20Environment%20standard.xls).*for the environment infrastructure requirements.>*

Addendum C shows an overview of CAF recommended tools

* 1. Infrastucture Reviews

The infrastructure for the project will be regularly reviewed and any additional infrastructure required will be reviewed as an action plan agreed with the [CLIENT].

* 1. Security Management

*<Some of the clients specify the specific security requirements that they require as part of the contract. This section should detail such security requirement apart from any other security practices. The details could be mentioned for physical security, mandatory security trainings, restrictions mandated by client, safety rules and so on. Mention the applicability of these requirements across various locations. For e.g the physical security requirement could be specific to India location and may not cover onsite.>*

* + 1. Security Baselines

Based on the chosen infrastructure and application packages or software components the relevant Capgemini Application Services Technical security baselines and, if present, client’s security baselines must be selected for implementation.

<Example Text

Based on the chosen operating system and the chosen application the security controls defined in Capgemini Application Services Technical baselines for Linux and SAP will be used and implemented.

Additionally since there is a client requirement that the information needs to be protected according to the standard NEN7510. This standard, where it deviates from the above mentioned technical baselines, will be implemented additionally.>

* + 1. Site Access

For all work carried out on a Capgemini site normal working practices will apply.

For work carried out at the [CLIENT] site, the [CLIENT’S] working practices defined in x will be followed.

<Example Text

Normal Working Hours are considered to be between 9am and 5:30pm, Monday to Friday, excluding Bank Holidays. Unlimited access to the [CLIENT’S] office will be required for all Project staff during these times and during outside normal working hours. CLIENT will supply Access Passes granting access to the main project area and any other building where project work may take place as required. All Project staff will adhere to the CLIENT policies listed below, which can be found on xx.

The following are the pertinent on-site working practices: ...........>

* + 1. System Access

The Engagement Manager ensures that system access is based on the project roles and is not available for any unauthorized personnel.

The Engagement Manager uses a Configuration Management tool to grant access to system according to different roles described in the Engagement.

All access to the shared folders will be maintained by the PMO, if applicable.

* + 1. Storage and Archiving
       1. Storage

*<Describe where documents or software are stored, saying who is responsible for the original or master copy and how long they are kept. For hardware the special conditions for storage are indicated (humidity, temperature, etc.).>*

All documentation and information regarding the management of the project is filed in the Project Repository.

All signed off paper originals of document Deliverables will be filed in a hard copy repository. The location for archiving these artifacts would be addressed by the Configuration Manager. Signed off paper originals also have to be scanned and added to the Project Repository.

Software and Hardware will be stored and managed according to the Configuration Management Procedure.

* + - 1. Archiving

Enter details of the items that must be archived, related to development and management activities. This should include the following:

* Project Repository;
* Deliverables, including signed paper copies;
* quality activity documentation (for example minutes of reviews and audits, trace of the configuration status, validation results, acceptance report);
* acceptance certificates;
* emails and other key correspondence.

Define media and format of archival and the location and duration of archiving.

.

Legal requirements for archiving vary from one country to another - note any special requirements.

* + - 1. Backup Procedures

X (for e.g. ICT) are responsible for the back-up and any restore of the Project Repository, the Development Environment and all other data on the Y network. If the network is outside the purview of Capgemini, then the strategy should be determined.

Project information held on a Capgemini networked server is backed up by standard Capgemini CIM procedures.

Enter details of what is to be backed up. Consider:

* the frequency of back-up
* location of back-ups (storage premises including off-site copies)
* scope of backup (part or total)
* verification of back-ups (for example by periodical restoration tests)
* back-ups of information not held on the normal development media such as information held on Capgemini laptops.

1. Configuration Management
   1. Configuration Identification

Configuration identification means the determination of the constituent parts of a system or product, the recording of the characteristics of those parts, the identification of their relationships, the assignment of a unique name to each part, and the graphical or tabular depiction of the whole system/product.

A list of items under configuration management will be maintained. Details of these information assets will be maintained using [Information Asset Register](https://iqms.fs.capgemini.com/FSQMS/UPM-FSSBU/index.htm#UPM-FSSBU/guidances/templates/Information%20Asset%20Register_C14B8AD4.html).

Capgemini documentation standard guideline will be used for document naming convention and version control.

The following Configuration Management tools will be used:

* Document Versioning: <TeamForge>
* Code Versioning: <Subversion and Rational ClearCase and/or ……>

If no Configuration Management tool is used then this section contains precise details of how Configuration Management will be achieved.

* 1. Configuration Control

*<Amend the steps of Configuration control as appropriate>*

Further details on configuration control can be found in FSSBU UPM and the Configuration Management Plan

* 1. Configuration Status Reporting

*<Amend the steps of Configuration Status Reporting as appropriate>*

Further details on configuration status reporting can be found in the Configuration Status Accounting checklist.

* 1. Configuration Reviews and Audits

*<Amend the steps for configuration reviews and audits as appropriate>*

*F*urther details on configuration reviews and audit can be found in FSSBU UPM and the Configuration Management Plan.

1. Quality Management

This section contains guidance on how to implement Quality Management on the project. The focus is on the quality assurance and quality control aspects related to the project approach and the project Deliverables. The purpose is to implement the management processes which will contribute to the overall quality of the project, when detailed verification and validation activities (such as testing) are described in appropriate delivery processes.

EM is responsible for delivering the agreed deliverables in the specified time & meeting quality requirements.

* 1. Project Reviews

Quality Management (QM) within the project is provided by a number of roles operating independently of the project team, and each providing a different perspective to the overall Quality Management role.

The quality status of any findings is also reported to the Capgemini leadership team. Any non-compliance with QM requirements is escalated to the Delivery Manager.

* 1. Deliverable Reviews

This section could contain the deliverables that are to be reviewed, the inputs to be referred, method of review, reviewer, outcome of review and so on. These details may also be included as part of Appendix A – Deliverables Descriptions.

* 1. Project Audits

Audits are carried out by appropriate Capgemini staff who are not involved in the project (by the PEG Team ). The approach taken normally involves interviewing selected members of the Project Team (arranged with the Capgemini Project Manager) and, where relevant, it is preferred that key < The Client > staff involved with the project are included. A report is produced from the audit, which is internal to Capgemini, although key findings will be discussed with the < The Client > Project Manager.

* 1. Quality Audits & Schedule

<Agree types and dates of audits and reviews with Quality Manager/Leader and enter details and/or delete items from the table below. Consider also whether the client will audit the project and ensure the objectives/process/duration/follow up are defined.

If Contract Review and/or Delivery Solution Review have been undertaken this section must state when they occurred and who was involved>

The following reviews and audits are planned to take place during the course of the project according to the Quality Assurance procedures:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type of Review or Audit** | **Responsibility for Organising Review/Audit** | **Target Date of Review/Audit** | **Client Involvement**  **(Y/N)** | **Output from the Review/Audit** |
|  |  |  |  |  |

* 1. Defect Tracking

Defect Management Process Overview

|  |  |
| --- | --- |
| Process Step | Description |
| Identify Defect | Testers find defects during functional, regression or integration testing. Developers find defects during unit/system testing. |
| Open Defect | Tester or developer logs a defect and assigns an ‘Open’ status to the defect. |
| Defect Review Meeting | Defect Triage Team reviews the defect, adjusts the defect type, responsible group, severity, priority and other attributes of Defect. Responsible Team Leads assign the defect to a resource and provide an Estimated Fix Date.  A “risk or issue” review is also conducted within this forum and escalated as appropriate. |
| Track Defect | All Open defects are tracked on a daily defect report. Filtering and Prioritization will determine the appropriate order – but all are to appear. |
| Defect Resolution | Assigned resource resolves the defect and assigns a Resolution Status of ‘Fixed’ and a Status of ‘Awaiting Retest – System’ |
| System Test | Solution Team performs system test; Upon successful test, initiates the migration of the fix to Functional Retest. |
| Migrate Change | Configuration Management Team migrates the changes to User Test environment and changes the status to ‘Migrated’;  Upon assuring the fix migration and preliminary test (if it necessitates), assigns the defect to ‘Awaiting Retest |
| Re-Test Defect | Test Lead reviews the Awaiting Retest defects on a daily basis to plan and communicate retest of the corrected defects |
| Close Defect | Tester verifies the defect has been resolved and changes the status to ‘Closed’. |
| Re-Open Defect | If the defect has not been resolved tester changes its status to ‘Re-Opened’. |
| Discuss during next Defect Tracking Meeting and follow above process to Close Defect | ‘Re-Opened’ defects are discussed during the next Defect Review meeting. |

* 1. Defect Prevention

*<Defect prevention activity is carried-out throughout the project life cycle. Defect Prevention activities may include Causal Analysis, Quality Circles, Task Kick-off meeting, Lessons learnt consolidation>*

*Document the Defect Prevention activities that will be done for this account / project, with high level timelines>.*

|  |  |  |
| --- | --- | --- |
| Activity | Project Iteration / Task | Frequency / Timeframe |
|  |  |  |
|  |  |  |

* 1. Continuous Improvement

*<This section details the improvement activities planned during the execution of the project. It also talks about the reusability of the same which can be used during further Iterations or iteration of the project. This includes any changes to the project process step(s) that result from Causal Analysis Resolution (CAR) actions.>*

* 1. Final Evaluations

*<This section details the consolidated view of how quality assurance will be managed in the project. It also talks about the relevant KPIs and how they will be managed and tracked throughout the project.*

*It produces a detailed summary of the project (link to closure report).>*

1. Knowledge Management
   1. Engagement Profile

At the end of the project a brief summary of the engagement will be documented for internal Capgemini use.

* 1. Knowledge Object Reuse About Release Retrospective

Capgemini has a policy of re-using best practices gained in projects and from alliance partners. Each project has access to a range of functional descriptions, system designs, frameworks, components, and code.

* 1. Knowledge Object Sharing

Certain documents are identified as re-usable under Capgemini processes. This project will be expected to create and retain the documents, relevant to the type of project, like Project Contact List , Lessons Learned, Methods and Tools, architectural patterns, infrastructure frameworks, latest product developments, technology feasibility documents, how to guides, useful web sites etc ..

1. Quantitative Engagement Management

*This chapter should not be considered as required for all Agile projects. The metrics described here should be tracked on any Agile project, but if you don't do formal quantitative engagement management, this is done as standard project tracking/monitoring within the Governance stream.*

*<For each type of work (Customized Software Development, Package Solutions, Testing, Application Maintenane), specify Quality Objectives & Improvement Plan. Define what measures (Metrics) will be used to evaluate them.*

*Distinguish the ones which are shared with the client (Product Quality, Cost, and Timelines.) and Ones which are shared internal.>*

KPI’s & metrics are tracked regularly to get an early alert of the situation and it is acted upon.

* 1. Project Metrics Management

< Add applicable Text>

Establish strategy for project metrics which is derived from client KPI and Organizational goals (Agile metrics from metrics guidelines). The strategy provides project metrics, baselines, tracking and reporting of metrics, casual analysis based on metrics analysis, revision to metrics etc

*<Use organizational performance baselines, and the project objectives and expectations (described in Organisation Baselines and Goals) to decide target performance level for selected metrics>*

*<For those selected measures / metrics, define*

* *data needed to measure them,*
* *data collection and consolidation procedure*
* *Analysis tools and techniques*
* *Action triggers, and possible actions (corrective and preventive)>*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric Name** | **Project/Business/Customer Requirement** | **Frequency** | **Name of Tool/Template used** | **If Template, stakeholders who would have access to it** |
| Velocity | <Fill> | End of Iteration | <Fill> | <Fill> |
| Feature Points Acceptance | <Fill> | End of Iteration | <Fill> | <Fill> |
| Total Weighted Defect Index | <Fill> | End of Iteration | <Fill> | <Fill> |
| UAT DD | <Fill> | End of Iteration | <Fill> | <Fill> |

1. Appendices
   1. Appendix A - Deliverable Descriptions

*<Complete one Deliverable Description for every deliverable, whether or not contractual and whether delivered from Capgemini, the client or a third party.*

*If there are many deliverables, consider inserting an index at the start to help find the individual Deliverable Description. If a category of information of the Deliverable Description is not applicable state ‘Not applicable’, or if the information is to be provided during a later update of the PGP, state when this will take place.*

*Do not leave a category of information blank. Alternatively a transposed version can be created in excel detailing all the project deliverables.>*

|  |  |
| --- | --- |
| **Deliverable ID** | *<Enter the identifying code by which the deliverable will be known in here e.g. D0002. The identifier is referenced elsewhere in the document, so it must be unique>* |
| **Deliverable Name** | *<Enter the name by which the deliverable will be known in here e.g.*  *Project Governance Plan, Functional Specification>* |
| **Abbreviation** | *<Enter the abbreviation by which the deliverable is normally known e.g. PGP, FS>* |
| **Contractual Deliverable** | Contractual / Not Contractual *<delete one of these as appropriate>* |
| **Type** | *<Enter type of deliverable e.g. Document, Software, Hardware>* |
| **Source/Destination** | *<Enter e.g. Delivered from Capgemini to [CLIENT]*  *Or Delivered from [THIRD PARTY SUPPLIER] to Capgemini, then to [CLIENT]*  *Or Delivered from [THIRD PARTY SUPPLIER] to [CLIENT]*  *Or Delivered from [CLIENT] to Capgemini>* |
| **Purpose** | *<Enter the purpose of the deliverable, why has it been produced, what information it is it intending to convey>* |
| **Composition** | *<Enter e.g. the contents of the deliverable (i.e. the Table of Contents for a document, with a brief description of the contents of each section.*  *If the deliverable is software, list the main functional elements of the software and a brief description of each.*  *If the deliverable is a software product from third party supplier state the name and version of the product and a brief description.*  *If the deliverable is a Release of software list the components of the Release, version numbers and brief description of each>* |
| **Form** | *<Enter the form that the deliverable will be prepared in (e.g. Word 2007>* |
| **Deliverable Standard** | *<Enter the e.g. the documentation standard (e.g. Q004 Capgemini Document standard).*  *Development standards e.g. which coding standard, which tools or hardware, include version numbers as appropriate>* | |
| **Derivation** | *<Enter the derivation of the deliverable e.g. the documents which are referred to when preparing the deliverable e.g. the derivation for a PGP may be the Contract, the Proposal>* | |
| **Acceptance Criteria** | *<Provide the criteria against which the deliverable will be measured, The criteria should be as measurable as possible, not subjective*  *If this is software to be delivered for testing, this must state the acceptable levels of each fault classification>*  *If this is Iteration acceptance, this must state that: The development team will demonstrate the Product Backlog Items (PBI) that were committed for the iteration and documented in the Iteration Backlog. These items should have approved ‘Definition of Ready’ prior to commitment of Iteration Backlog. Only those PBIs which satisfy ‘Definition of Done’ will be demonstrated during the Iteration Review. The team from the Customer organization including Product Owner and other business stakeholders must attend this review meeting in order to review and sign-off the deliverables and product features. Also feedback will be noted during the review meeting. If the feedback results in changes of scope or Product Backlog, the change management procedure in Section 6 will be invoked.*  *All those items which are reviewed and signed-off in the Iteration Review meeting are deemed as accepted and should be formally documented with appropriate signatures from agreed client representatives. Wherever feasible, such acceptance should be gathered during the Iteration Review itself and in all cases a maximum delay period for client feedback must be documented and agreed in the contract to avoid unnecessary delays/review cycles in the project for getting acceptance.*  *If this is Release acceptance, this must state that: All Build iterations should conclude with an Iteration Review, followed by Iteration acceptance as described above. The purpose of the Release acceptance is to test the interaction, operation of the PBIs of all Iterations. A Release acceptance can be subject to a successful UAT (User Acceptance Testing), where a set of users perform acceptance testing of the delivered application. The development team will provide support during the Build-UAT iterations to fix bugs.* |
| **Capgemini Verifiers** | *<Name the roles of the people on the team (or outside of team if appropriate) who will check/verify the deliverable (before it is delivered to the client or on receipt from the client or third party>* |
| **Capgemini Verification Method** | *<Document the ways in which quality of the deliverable will be ensured e.g. Review, Walkthrough, Inspection, Formal Inspection, Testing (state levels of testing e.g. unit testing, system testing, performance testing - should equate to tasks/activities on the global plan>* |
| **Capgemini Sign Off** | *<State the role of the person who is authorised to formally signing off the deliverable as meeting all specified acceptance criteria for the deliverable on behalf of Capgemini*  *It is possible that several people will sign the document following verification, but make it clear who have overall responsibility for its sign off>* |
| **Capgemini Sign Off Method** | <Signature on front page of deliverable / Signature on Acceptance Certificate / Not applicable *{delete appropriate}>* |
| **< The Client > Verifiers** | *<List the people that will review the deliverable against the quality criteria. The role of the person may be specified rather than the actual name e.g. Project Sponsor.*  *Not applicable if not a contractual deliverable>* |
| **< The Client > Sign Off** | *<State the name of the person who is authorised to formally signing off the deliverable as meeting all specified Acceptance Criteria {see above} for the deliverable on behalf of < The Client >*  *It is possible that several people will sign the document following verification, but make it clear who have overall responsibility for its sign off.*  *Not applicable if not a contractual deliverable>* |
| **< The Client > Sign Off Method** | <Signature on front page of deliverable / Signature on Acceptance Certificate / Not applicable *{delete as appropriate}>*  *Not applicable if not a contractual deliverable* |
| **Delivery Medium** | *< {E.g. paper, electronic (diskette or a specific file/directory location>* |
| **Number of Copies** | *<State number of copies of each media to be provided to the Destination>* |
| **Replication Methods** | *<If multiple copies are to be delivered state what process will occur to ensure all copies accurately reflect the master>* |
| **Delivery Arrangements** | *<State specific delivery arrangements e.g. issue by email to person, visit to client site, transit by road and how receipt will be checked>* |

* 1. Appendix B – UPM Deliverables

*Capgemini Unified Project Management method (UPM) consists of many artifacts that can be delivered in an engagement. Below is a minimal list of these mandatory UPM artifacts.*

* *Project Governance Plan*
* *Project Status Reports*
* *Client Acceptance Criteria*
* *Client Acceptance Procedures*
* *Project Plan*
* *Project Actuals/Forecast*
* *Client Invoices*
* *Staffing Plan*
* *Change Log*
* *Unified Risk Register* 
  1. Appendix C – Agile Tooling

|  |  |  |  |
| --- | --- | --- | --- |
| **CAF Iteration** | **Discipline** | **Tools** | |
| * Prepare * Build * Release | * Planning & Financial Management * Project Governance | Dashboard | * JIRA Agile / Confluence * Rally * Mingle * TeamForge \* * IBM JAZZ |
| Planning | * JIRA Agile * Visualstudio.com / Microsoft Team Foundation Server (TFS) * IBM JAZZ |
| Financials | * Jira plug-in for Clarity |
| * Prepare * Build | * Scope & Requirements Management | * Word / PowerPoint\*\* * Jira Agile / Confluence * IBM / Rational – RRC | |
| * Prepare * Build * Release | * Collaboration | * TeamForge\* * Google Hangouts\*\* * GIMS / Live Meeting / LVis * EBeam * VVM\*\* * IBM Jazz | |
| * Prepare * Build | * Architecture | * IBM / Rational Software Architecture * SPARX – Enterprise Architect | |
| * Build | * Design | * Modeling (UML / BPMN) | * SPARX – Enterprise Architect * IBM / Rational Software Architecture |
| * Build | * Development | * IDE | * Visual Studio / Resharper * Eclipse\*\* * IntelliJ |
| * Repositories | * Git * SVN * visualstudio.com / Microsoft Team Foundation Server (TFS) |
| * Build, * Release | * Testing | * Unit Testing | * Junit\*\* * MSUnit\*\* / Nunit\*\* |
| * UI Testing | * UI Testing: * Selenium\*\* * QTP * Rational Test Manager |
| * Performance Testing | * Visual Studio Test Edition * LoadRunner Quality Center / Test Director |
| * Functional Testing | * HP – QC, * IBM Rational Test Manager |
| * Build, * Release | * Deployment | * CI / CD | * Maven / Hudson\*\* * visualstudio.com / Microsoft Team Foundation Server (TFS) * FXcorp |

* 1. Appendix D – Glossary

Terms

This appendix defines key abbreviations and acronyms contained in this document.

*{Delete any you don’t refer to}*

|  |  |
| --- | --- |
| **Term** | **Meaning** |
| Review (of a deliverable prior to delivery to < The Client >) | A formal documented review by one or more people, either from within or outside the project team |
| Walkthrough (of a deliverable prior to delivery to < The Client >) | A peer group review of any deliverable conducted by the project team members. This should normally include the originator of the deliverable |
| Inspection (of a deliverable prior to delivery to < The Client >) | A semi-formal check that the deliverable meets the specifications and conforms to standards. This may not include the originator of the deliverable |
| Formal Inspection (of a deliverable prior to delivery to < The Client >) | A formal methodical inspection (e.g. Gilb Inspection) |
| Testing (of a deliverable prior to delivery to < The Client >) | A series of formal tests against a set of documented expected results. The types and level of testing will depend on the stage of the project < and documented within a Test Strategy, drawn up in conjunction with < The Client > > |

Acronyms and Abbreviations

This appendix contains definitions of acronyms and abbreviations contained in this document, which may otherwise lead to incomprehension, misunderstandings or ambiguities.

|  |  |
| --- | --- |
| **Acronym or Abbreviation** | **Meaning** |
| CR | Change Request |
| EM | Engagement Manager |
| PGP | Project Governance Plan |
| TPS(n) | Third Party Supplier (n) |
|  |  |

* 1. Appendix E – Tailoring for Engineering Documents

If the project is following client provided templates for engineering activities then following information should be provided at least for the artifacts listed below -

Functional Specification, Technical Specification, Impact Analysis, Review log, Unit test cases and defect log, Integration Test Cases and Defect log, Query log, Delivery Note/Release Note

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IQ Template Name** | **Description** | **Client Template Name** | **Gaps if Any** | **How the gaps are addresses (if any)** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

* 1. Appendix F – Decision Analysis and Resolution Plan

Following are the potential scenarios for DAR in during software development lifecycle

|  |  |  |  |
| --- | --- | --- | --- |
| **DAR Scenarios** | **Project Phase** | **Evaluation Method selected** | **DAR Performed Y/N** |
| Make/Buy/Reuse Analysis or Make/Reuse Analysis | Analysis/Design | DAR WEIGHTED AVERAGE MATRIX (PUGH MATRIX) |  |
| Selection of Testing/Development tool | Analysis/Design/Testing | DAR WEIGHTED AVERAGE MATRIX (PUGH MATRIX) |  |
| Selection of the best design solution from multiple available solutions | Analysis/Design | DAR WEIGHTED AVERAGE MATRIX (PUGH MATRIX) |  |

Not all decisions will go through the DAR process, but the DAR may be used when:

·    The decision may have a significant impact on the project (plan, cost budget, quality of deliverables, etc.) and therefore on the overall success.

·    Situation is rather difficult, with multiple and complex alternatives to be assessed.

·    Many stakeholders with different viewpoints have to participate to the decision making process.

* 1. Appendix G - Document Control

|  |  |
| --- | --- |
| **DOCUMENT NAME** | Agile Project Governance Plan |
| **ABOUT THE DOCUMENT** | This document provides guidance to prepare PGP for Agile Projects |
| **SECURITY CLASSIFICATION** | Company Confidential |
| **LOCATION** | iQ |
| **TEMPLATE OWNER** | iQ Team |

|  |  |  |
| --- | --- | --- |
| **DOCUMENT AUTHORISATION** | | |
| Document Owner | Reviewer | Approver |
| iQ team | iQ Lead, SME | PEG Head |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEMPLATE AMENDMENT LOG** | | | | | |
| **Version** | **Date**  **Dd mmm yyyy** | **Source of Change** | **Section** | **A/M/D** | **Brief Description of change** |
| 1.0 | Apr,2015 | PIP0993 | All | A | Initial version of agile project governance plan |
| 1.1 | 9-Dec-2016 | PIP1166 | All | M | Updated as per CAF standards |

|  |  |  |  |
| --- | --- | --- | --- |
| **DOCUMENT AMENDMENT LOG** | | | |
| **Version** | **Date**  **dd mmm yyyy** | **Brief Description of change** | **Approval Mail** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |