**<Project Name>**



**Software Project Development Plan**

**Version <1.0>**

*[Note: Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]*

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

**1.**    **Introduction**

*1.1*      *Purpose*

*1.2*      *Scope*

*1.3*      *Definitions, Acronyms, and Abbreviations*

*1.4*      *References*

*1.5*      *Overview*

**2.**    **Project Overview**

*2.1*      *Project Purpose, Scope, and Objectives*

*2.2*      *Assumptions and Constraints*

*2.3*      *Project Deliverables*

*2.4*      *Delivery Plan*

**3.**    **Project Organization**

*3.1*      *Organizational Structure*

*3.2*      *External Interfaces*

*3.3*      *Roles and Responsibilities*

**4.**    **Management Process**

*4.1*      *Project Estimates*

*4.2*      *Project Plan*

4.2.1       Phase Plan

4.2.2       Iteration Objectives

4.2.3       Releases

4.2.4       Project Schedule

4.2.5       Project Resourcing

*4.3*      *Project Monitoring and Control*

**5.**    **Annexes**

**Software Development Plan**

**1.                  Introduction**

*[The introduction of the****Software Development Plan****provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of this****Software Development Plan****.]*

**1.1               Purpose**

*[Specify the purpose of this****Software Development Plan****. The text below is provided as an example****.****]*

The purpose of the *Software Development Plan* is to gather all information necessary to control the project. It describes the approach to the development of the software and is the top-level plan generated and used by managers to direct the development effort.

The following people use the *Software Development Plan*:

·         The **project manager** uses it to plan the project schedule and resource needs, and to track progress against the schedule.

* **Project team members** use it to understand what they need to do, when they need to do it, and what other activities they are dependent upon.

**1.2               Scope**

*[A brief description of the scope of this****Software Development Plan****; what Project(s) it is associated with and anything else that is affected or influenced by this document. The text below is provided as an example.]*

This *Software Development Plan* describes the overall plan to be used by the <project name> project, including deployment of the product. The details of the individual iterations will be described in the Iteration Plans.  
The plans as outlined in this document are based upon the product requirements as defined in the *Vision Document*.

**1.3               Definitions, Acronyms, and Abbreviations**

*[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the****Software Development Plan****. This information may be provided by reference to the project’s Glossary.]*

**Terminology Definitions**

|  |  |
| --- | --- |
| Terminology | Definition |
| Agile Framework | A framework that enables the incremental delivery of solutions and releases as the business is moving as opposed to a longer-term delivery of a larger product that may miss its intended market. |
|  |  |
| Scrum@Scale® Framework | Scrum is a process within an agile framework which is used to address complex adaptive problems, while productively and creatively delivering products of the highest possible value (Schwaber and Sutherland, 2017). Scrum@Scale is a framework that enables scrum to be scaled to the unique conditions of the organization to support the delivery of multiple projects through small development team while clear path for development and product life cycle. |
| Proof of Concept | A small tool that is used to test out a hypothesis, generally less than $150k a year and internal to a few EPA users. |
| Prototype | A proof of concept that starts to focus more on minimal viable product (MVP), user adoption, further development based on users feedback and formal development methods. The product may be public or internal to EPA for further user testing. |
| Production | **Production here is defined as environment and support provided by an IT team that is responsible for monitoring and management of production servers, scheduled jobs, incident management and receiving incidents and requests from end-users, analyzing these and either responding to the end user with a solution or escalating it to the other teams as and when needed** |
| Systems Administration | The process of managing the IT infrastructure including the deployment, monitoring, and maintenance of the servers including installing and updating the server’s operating system and installing components in the operating system, but not necessarily the configuration and management of this software. |
| DevOps | Development and Operations – Practices and tools that focus on delivering high quality software quickly on a routine basis that enables continuous delivery and continuous integration. |
| Release | The actual delivery of the software, features, hardware, and/or data to the end-users which may or may not be seen/noticed by the end-users. |
| Requirements | A formal document that describes user and functional requirements and is often used for testing purposes. |
| EPICs | Describes a larger set of needs that are subsequently broken down into a smaller subset of stories. |
| Tickets | Also referred to as “Issues”, combined with a requirements document, they breakdown the requirements into more granular needs which can then be prioritized. |
| Backlog | A list of features or requirements that have been prioritized by the Product Owner via tickets. |
| Sprint | A clear bounded amount of time (e.g., 2 weeks) where a clear amount of work is produced by the Scrum Team. |
| Sprint Planning | Discussion with the scrum teams to identify work to be completed during the Sprint. |
| Grooming | Preparing tickets from the backlog to be selected for the next three series of Sprints. |
| Daily Scrum (aka Stand Ups) | A daily check in between the scrum master and technical team to discuss activities for the day and share blockers, but solve at a later time. |
| Retrospective | Discussions between the Scrum Master and the Development Team to openly perform a “pulse check” for the team including what is working, what isn’t, and the path forward. |
| Jira | Atlassian Tool used for CCTE’s Ticketing System for tracking work which integrates with Slack, Confluence, and BitBucket. |
| Confluence | Atlassian Tool used for CCTE’s Product and Project Documentation (e.g., requirements, wireframes, etc.) system which integrates with BitBucket and Jira |
| BitBucket | Atlassian tool used for CCTE’s Code Repository for code reviews and testing. |
| Slack | Tool used for project discussions via topic and/or products which integrates with associated Jira projects to track work. |
| RACI | A diagram or table used to identify who is Responsible, Accountable, Consulted, and Informed. |
| Responsible | The individuals that are actually executing the work that needs to happen for a particular function, project, etc. |
| Accountable | The person who is ultimately held accountable in the event of a failure. It is the decision maker that can also allocate resources and usually represents the effort as whole. |
| Consulted | The individual who needs to be involved in the process or delivery of a product, but not necessarily is as hands on such as providing general direction. |
| Informed | The individual who needs to be briefed on progress or other technical decisions for some sort of contingency purpose. |

**4               Role Definitions**

|  |  |
| --- | --- |
| Terminology | Definition |
| Leadership | This includes CCTE’s Director, Deputy Director, Associate Director, Chief of Staff, Staff Chiefs, Division Directors, and Associate Division Directors. |
| Management | 2nd line supervisor and above. |
| Supervisor | 1st line or immediate supervisor. |
| SCDCD Division Director | Individual accountable for overall strategic delivery of technology solutions that are necessary to enable CCTE’s research. Individual also provides the technology strategic direction to the technical staff in SCDCD. |
| SCDCD Associate Director | With the division director, this individual is providing the strategic direction necessary for the delivery of high quality and high impact technology solutions. |
| ADB Branch Chief | Development and Operations – Practices and tools that focus on delivering high quality software on a rapid basis that enables continuous delivery and continuous integration. |
| DMMB Branch Chief | A clear bounded amount of time (e.g., 2 weeks) where a clear amount of work is produced by the Scrum Team. |
| DEQEB Branch Chief | A list of features or requirements that have been prioritized by the Product Owner via tickets. |
| Product Owner | Individual who is actively engaged with the development process by representing the voice of the customer, documents requirements, user stories, and tickets and prioritizes the product backlog. This individual is ultimately accountable for the product’s delivery. |
| Scrum Master | The individual who manages the scrum process for each project including leading the development team, serving as the primary interface with the product owner and represents the needs of the development team, and understands the capabilities of the development team. |
| Development Team | The individuals who are responsible for the development of the product within associated timeframes. |
| DevOps Lead | Individual who is developing the processes and tools to enable the quick delivery of high quality software at the fastest speed as possible. |
| Systems Administrator | Individual who is managing the IT infrastructure including the deployment, monitoring, and maintenance of the servers including installing and updating the server’s operating system and installing components in the operating system, but not necessarily the configuration and management of this software. |
| Scientific Analyst | Commonly referred to as a “Business Analyst” in industry, this person is the scientific partner to the product owners and scrum master by facilitating the requirements gathering process. |

**1.4               References**

*[This subsection provides a complete list of all documents referenced elsewhere in the****Software Development Plan****. Identify each document by title, report number if applicable, date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.*

*For the****Software Development Plan****, the list of referenced artifacts includes:*

·         *Iteration Plans*

·         *Development Case*

·         *Vision*

·         *Glossary*

·         *Any other supporting plans or documentation.]*

**1.5               Overview**

*[This subsection describes what the rest of the****Software Development Plan****contains and explains how the document is organized. The text below is provided as an example.]*

This *Software Development Plan* contains the following information:

Project Overview — provides a description of the project's purpose, scope, and objectives.  It also defines the deliverables that the project is expected to deliver.

Project Organization — describes the organizational structure of the project team.

Management Process — explains the estimated cost and schedule, defines the major phases and milestones for the project, and describes how the project will be monitored.

Applicable Plans and Guidelines — provides an overview of the software development process, including methods, tools and techniques to be followed.

**2.                  Project Overview**

**2.1               Project Purpose, Scope, and Objectives**

*[A brief description of the purpose and objectives of this project and a brief description of what deliverables the project is expected to deliver.]*

**2.2               Assumptions and Constraints**

*[A list of assumptions that this plan is based and any constraints, for example. budget, staff, equipment, schedule, that apply to the project.]*

**2.3               Project Deliverables**

*[A list of the artifacts to be created during the project, including target delivery dates. The text below is provided as an example.]*

Deliverables for each project phase are identified in the Development Case.  Deliverables are delivered towards the end of the iteration, as specified in section *4.2.4 Project Schedule*.

**2.4               Evolution of the Software Development Plan**

*[A table of proposed versions of the****Software Development Plan****, and the criteria for the unscheduled revision and reissue of this plan. The text below is provided as an example.]*

The *Software Development Plan* will be revised prior to the start of each Iteration phase.

**3.                  Project Organization**

**3.1               Organizational Structure**

*[Describe the organizational structure of the project team, including management and other review authorities.]*

**3.2               External Interfaces**

*[Describe how the project interfaces with external groups. For each external group, identify the internal and external contact names. This should include responsibilities related to deployment and acceptance of the product.]*

**3.3               Roles and Responsibilities**

*[Identify the project organizational units that will be responsible for each of the disciplines, workflow details, and supporting processes. The text below is provided as an example.]*

|  |  |
| --- | --- |
| **Person** | **Roles** |
| Amar Singh, Director for SCDCD | Requirements Reviewer Architecture Reviewer Change Control Manager |
| Norman Adkins, Product Owner  Jyothi Tumkur, Scrum Master | Project Manager Deployment Manager  Project Reviewer Requirements Reviewer |
| Jason Brown, Scientific Analyst  Terry Brown, Scientific Analyst | System Analyst Requirements Specifier User Interface Designer Software Architect  Designer (Wire Diagrams) Implementer Code Reviewer Integrator Test Designer Tester Clowder Automator |
| TBD, Software Engineer | Designer Implementer Code Reviewer Integrator Test Designer Tester Technical Writer |
| Researchers and Research Technicians | Uploading of data, configuring their space, controlling changes in data, and adding users they wish to share their information with. controlling changes to artifacts. |

Anyone on the project can perform Any Role activities.

**4.                  Management Process**

**4.1               Project Estimates**

*[Provide the estimated cost and schedule for the project, as well as the basis for those estimates, and the points and circumstances in the project when re-estimation will occur.]*

**4.2               Project Plan**

*[This section contains the schedule and resources for the project.]*

*4.2.1          Phase Plan*

*[Include the following:*

·     *Work Breakdown Structure (WBS) — optional for small projects*

·     *a timeline or Gantt chart showing the allocation of time to the project phases or iterations*

·     *identify major milestones with their achievement criteria*

*Define any important release points and demos.]*

*4.2.2          Iteration Objectives*

*[List the objectives to be accomplished for each of the iterations.]*

*4.2.3          Releases*

*[A brief description of each software release and whether it’s demo, beta, and so on.]*

*4.2.4          Project Schedule*

*[Diagrams or tables showing target dates for completion of iterations and phases, release points, demos, and other milestones.]*

*4.2.5          Project Resourcing*

*[Identify the numbers and type of staff required here, including any special skills or experience, scheduled by project phase or iteration.*

*Describe how you will approach finding and acquiring the staff needed for the project.*

*List any special training project team members will require, with target dates for when this training should be completed.*

*Allocation of costs against the WBS and the Phase Plan.]*

**4.3               Project Monitoring and Control**

*[The following is a checklist of items to consider:*

·         *Requirements Management : Specify the information and control mechanisms which will be collected and used for measuring, reporting, and controlling changes to the product requirements.*

·         *Schedule and Budget Control:Describe the approach to be taken to monitor spending against the project budget and progress against the planned schedule. Describe how to take corrective action when required.*

·         *Quality Control:Describe the timing and methods to be used to control the quality of the project deliverables and how to take corrective action when required. Include techniques, metrics, criteria, and procedures used for evaluation— this will include walkthroughs, inspections, and reviews. Note that this is in addition to the Test Plan, which is not enclosed in the Software Development Plan.*

·         *Reporting and Measurement: Describe internal and external reports to be generated, and the frequency and distribution of publication. Specify which metrics should be collected and why.*

·         *Risk Management: Describe the approach that will be used to identify, analyze, prioritize, monitor and mitigate risks. Include a list of risks and their current status.*

·         *Project Close-out: Describe the activities for the orderly completion of the project, including staff reassignment, archiving of project materials, post-mortem debriefings and reports, and so forth.*

·         *Configuration Management: Describe the process by which problems and changes are submitted, reviewed, and dispositioned. Describe how project or product artifacts are to be named, marked, and numbered, including hardware, system software, Commercial-Off-The-Shelf (COTS), plans, models, components, test software, results and data, executables, and so on. Describe retention policies, and the back-up, disaster, and recovery plans. Also describe how the media is to be retained—online, offline, media type, and format.*

·         *Problem Resolution: Describe the approach to be taken to resolve disagreements with the customer, including how to handle schedule slips, scope, and contractual disagreements.*

·         *Subcontractor Management: Describe how subcontractors will be managed.*

·         *Process Improvement Plan: Describe how the effectiveness of the process will be assessed and improved.*

*The text that follows is provided as an example.]*

**Requirements Management**

The requirements for this system are captured in the Vision document. Requested changes to requirements are captured in Change Requests, and are approved as part of the Configuration Management process.

**Schedule and Budget Control**

Expenses are monitored by the project manager, and reported and assessed monthly. (See Reporting and Measurement below).

The project manager maintains a schedule showing the expected date of each milestone. The line items in the schedule include work packages assigned to individuals. Each individual who is assigned a work package provides %completion information to the project manager on a weekly basis. Changes in the schedule will be escalated to the project sponsors, who will then decide whether to alter scope in order to preserve target completion dates.

**Quality Control**

Defects will be recorded and tracked as Change Requests, and defect metrics will be gathered (see Reporting and Measurement below).

 All deliverables are required to go through the appropriate review process, as described in the Development Case. The review is required to ensure that each deliverable is of acceptable quality, using guidelines described in the ELMS BOARD for Small Projects review guidelines and checklists.

Any defects found during review which are not corrected prior to releasing for integration must be captured as Change Requests so that they are not forgotten.

**Reporting and Measurement**

Updated cost and schedule estimates, and metrics summary reports, will be generated at the end of each iteration.

The Minimal Set of Metrics, as described in the ELMS BOARD Guidelines: Metrics, will be gathered on a weekly basis.  These include:

Earned value for completed tasks. This is used to re-estimate the schedule and budget for the remainder of the project, and/or to identify need for scope changes.

Total defects open and closed – shown as a trend graph. This is used to help estimate the effort remaining to correct defects.

Acceptance test cases passing – shown as a trend graph. This is used to demonstrate progress to stakeholders.

In addition, overall costs will be monitored against the project budget.

**Risk Management**

Risks will be identified in Inception Phase using the steps identified in the ELMS BOARD for Small Projects activity “Identify and Assess Risks”. Project risk is evaluated at least once per iteration and documented in this table. The risks of the greatest magnitude are listed first in the table.

|  |  |  |
| --- | --- | --- |
| **Risk Ranking (High, Medium, Low)** | **Risk Description and Impact** | **Mitigation Strategy and/or Contingency Plan** |
|  |  |  |

**Configuration Management**

Appropriate tools will be selected which provide a database of Change Requests and a controlled versioned repository of project artifacts.

All source code, test scripts, and data files are included in baselines. Documentation related to the source code is also included in the baseline, such as design documentation. All customer deliverable artifacts are included in the final baseline of the iteration, including executables.

The Change Requests are reviewed and approved by one member of the project, the Change Control Manager role.

Full backups are performed monthly and incremental are performed nightly.

**5.                  Annexes**

*[Additional material of use to the reader of the****Software Development Plan****. Reference or include any project technical standards and plans which apply to this project. This typically includes the Development Case, plans for infrastructure, and product acceptance. It also typically includes Programming Guidelines, Design Guidelines, and other process guidelines. The text that follows is provided as an example.]*