DEPARTMENT OF HEALTH AND HUMAN SERVICES
ENTERPRISE PERFORMANCE LIFE CYCLE FRAMEWORK

<OPDIV Logo>

PRACTICES GUIDE

REQUIREMENTS DEFINITION

Issue Date: <mm/dd/yyyy>
Revision Date: <mm/dd/yyyy>

Document Purpose

This Practices Guide is a brief document that provides an overview describing the best practices, activities, attributes, and related templates, tools, information, and key terminology of industry-leading project management practices and their accompanying project management templates. This Guide describes HHS EPLC required practices for requirements development.

Background

The Department of Health and Human Services (HHS) Enterprise Performance Life Cycle (EPLC) is a framework to enhance Information Technology (IT) governance through rigorous application of sound investment and project management principles and industry's best practices. The EPLC provides the context for the governance process and describes interdependencies between its project management, investment management, and capital planning components. The EPLC framework establishes an environment in which HHS IT investments and projects consistently achieve successful outcomes that align with Department and Operating Division goals and objectives.

The Enterprise Performance Life Cycle (EPLC) Requirements Analysis Phase validates and further analyzes the business requirements that were documented in the earlier phases. The activities include defining the functional requirements, non-functional requirements, the business model, and logical data model. The complete set of Requirements serves as a blueprint for the acquisition planning and the subsequent design activities.

Projects often encounter major difficulties because they lack clearly defined and documented requirements. The practice of requirements definition is part of an overarching requirements management practice. This overarching practice is a systematic approach to finding, documenting, organizing, and tracking requirements and the changes that may occur throughout the life of a project and is described in greater detail in the Requirements Management Practices Guide.

According to the Butler Group ~40% of effort in an average software project is rework. One way of avoiding these issues is to properly define project requirements. The further along the project is in its life cycle, the more costly it becomes to correct requirement errors. The cost of correcting a requirement error increases exponentially as the project matures. For example, to correct a requirement error in the operation stage could cost a multiple of 100-times or more than if that same error was fixed earlier in the project's life. Defining requirements correctly at the start of the project is often the single most important practice that prevents costly errors and increases the potential for project success. The challenge however, is to get business stakeholders, end-users, and project teams all on the same page in regards to those requirements.

Practice Overview

The practice of defining requirements provides a solid foundation for project success and proper delivery of the end product. Properly defined requirements provide the first view of what the intended product must do and how it should perform. They also provide a basis for product design and serves as a foundation for testing and user acceptance of the end product by identifying the goals, needs, and objectives of the project by asking questions such as:

- What problem are we trying to solve?
- What do we need to do to solve the problem?
- Why are we trying to solve the problem?
- How do we accomplish solving the problem?

Requirements definition is often the main practice that serves as a bridge between project teams and business stakeholders. The practice should define both product and project requirements as well as related functional and non-functional requirements. Requirements definition should begin early in the analysis phase. Requirements should then be managed throughout the life of a project from their high level, through detailed requirements, design, build, and test. Requirements captured at all business and product levels help to ensure that the project meets its objectives within the agreed upon limitations of time, scope, resources, and quality. Actual requirement types will be dependant upon the project and will also vary base on the methodology used to define them. Some major requirement types include:

- *Project Requirements* define how the work will be managed. This includes the budget, communication management, resource management, quality assurance, risk management, and scope management. Project requirements focus on the, who, when, where, and how something gets done and are generally documented in the Project Management Plan.
- *Product Requirements* include high level features or capabilities that the business team has committed to delivering to a customer. Product requirements do not specify how the features or the capabilities will be designed.
  - *Functional Requirements* address what the system does. They define any requirement that outlines a specific way a product function or component must perform.
  - *Non-Functional Requirements* (also referred to as Quality of Service by the International Institute of Business Analysts, Business Analysis Body of Knowledge) address items such as the technical solutions, topics that address the number of people who need to use the product, where the product will be located, the types of transactions processed, and types of technology interactions.

There are three main perspectives that need to be considered when properly defining requirements. They are the business perspective, user perspective, and the technical perspective.

- *Business Perspective* – Business and marketing stakeholders identify higher-level goals and objectives of the organization and will answer questions about why we are trying to solve the problem, the target market, what business need must be satisfied, and what metrics identify that the project is successful.
- *Client/User Perspective* – Most significant are client/end-user requirements. The client will answer questions about what problem needs to be solved, how they will interact with the product.
- *Technical Perspective* – Engineers and developers provide the technical perspective required to answer questions about how the project's objectives will be accomplished.

Sources used to define requirements come from across the organization and include input from areas such as management, legal departments, end-users, enterprise architects, business analysts, system engineers, technology specialists, etc. Defining requirements effectively simply means figuring out what to make before starting to make it, keeping in mind that the end product should suit the needs of the client rather than the client suit the product. It is also good practice for the project team to map business processes with associated project objectives to ensure that the project's scope aligns with the organization's goals and objectives.

A well defined requirements definition document establishes a solid foundation for all stakeholders to agree upon the project's product requirements. Early understanding between all project participants, as to what needs to be accomplished, ultimately reduces development effort, provides a basis for estimating costs and schedules, serve as a baseline for enhancements, and can be used as a tool to verify/validate requirements.

Effective requirements definition also more closely aligns the organization's expectations of project deliverables resulting in fewer requirement errors, less rework, and an overall increase in successful project delivery. Successful requirements definition is often accomplished through an iterative practice of:

Requirements Gathering/Elicitation – Requirement gathering/elicitation is an iterative process that involves interacting with the client to gain consensus on the details of those requirements. There is no one perfect method for gathering requirements. The most appropriate method for gathering requirements differs from project-to-project. Some commonly used techniques for gathering requirements include:

- *Interviews* – Interviews are used to gather information however, the predisposition, experience, and skill of the person being interviewed needs to be taken into account. These elements have a tendency to prejudice information obtained during the interview process.
- *User Stories* – User Stories are a simple approach to requirements gathering/elicitation that shifts the focus from formal written requirements to conversation. User Stories are written by the

customer outlining functions that the system should perform. These stories are often only a few sentences long.

- *Workshops* – Requirements gathering workshops provide an opportunity for individual perspectives to be shared, refined, and combined in ways that builds upon business requirements.

Requirements Analysis – Requirements analysis is an iterative process that involves interacting with the client, project team, and other stakeholders to obtain a detailed understanding of each of the gathered requirements and how they will impact the project and product. The practice of requirement analysis delivers greater value to project stakeholders by reinforcing a stronger alignment between these and other groups. There is no one perfect method for analyzing requirements. The most appropriate method for analyzing requirements differs from project to project. Two high-level types of analysis include:

- *Qualitative Analysis* – Qualitative analysis includes methods for prioritizing the identified requirements for further action, such as quantitative analysis, scheduling, and actual product development. It assesses the priority of identified requirements using their level of importance to the client and end-users, the corresponding impact on other project objectives, as well as other factors such as the time frame and risk tolerance of the project constraints of cost, schedule, scope, and quality.
- *Quantitative Analysis* – Quantitative analysis is performed on requirements that have been prioritized by the qualitative analysis process as candidates for inclusion in the product to be developed. It analyzes those requirements and assigns a numerical rating indicating the priority order in which they should be delivered. When complete, it also presents a quantitative approach to decision making when uncertainty arises.

Some specific commonly used techniques for analyzing requirements include:

- *Use Case Analysis* – A Use Case is a narrative document that describes the sequence of events a user follows to complete a process. Use Cases are meant to capture the intended behavior of the product being developed without specifying how that behavior is to be implemented.
- *Prototyping* – Prototyping is a technique for building a quick and rough version of a desired system or parts of that system. A prototype can be used to illustrate the product's capabilities to users and designers. It can serve as a communications mechanism that allows reviewers to understand expected interactions with the final product.
- *Business Process Diagrams/Flow Charts* – Flow charts are graphical presentations that show the step-by-step sequence of activities and/or procedures that satisfy the requirement to be delivered in the final product.

An important element of this step is an Enterprise Architecture (EA) assessment, which is performed to ensure that the requirements are traceable to the HHS and OPDIV EA. The HHS has adopted an approach defined in terms of segments of functionality within a common business area. Under this approach, the business areas are grouped as communities of interest according to similarities in missions, goals, objectives, and commonality of services and business processes[1].

Conducted with the help of your EA team, the EA assessment involves validation of the requirements against the information content of the corresponding artifacts associated with the EA segment that the project in question will support. Among the artifacts, applicable to the Requirements Analysis Phase of the EPLC, are the Segment Information Exchange Matrix, Subject Area Mapping, Conceptual Data Model, Logical Data Model, Segment Business Process-Conceptual Data CRUD Matrix, Data Asset Inventory, Service Profile, Service-SRM Alignment, Technology Profile, Technology-TRM Alignment, Measurement Indicator-HHS PRM Alignment, and Performance Architecture.

More specifically, the EA assessment performed as part of the Requirements Analysis Phase involves (but is not limited to) the following activities:

- Articulation of traceability between the requirements and the segment-specific goals, objectives, mission, and vision, HHS and OPDIV goals and objectives, as well as to particular capability gaps identified in the segment transition roadmap;
- Mapping of the solution-level use case and business process models to the models of the segment-supporting business processes;

[1] The HHS Architecture Development Methodology provides a detailed description of how development and analysis of EA segments is accomplished.

- Identification of the data assets affected by the proposed solution, and mapping of the solution-level conceptual and logical data models to the applicable segment-level models, data asset inventory, and information exchange matrix; and,
- Articulation of the alignment between the non-functional requirements and the relevant technology and service profiles, as well as the Federal Enterprise Architecture Service Component Reference Model (SRM) and Technology Reference Model (TRM).

For requirements to be effectively translated into a product they must be clearly understood by the individual building the product's functionality. This is accomplished documenting defined requirements in clear and consistent way. A well-documented requirement should describe the capabilities of the solution, any conditions that must exist for the requirement to operate, and any constraints that may prevent the solution from being able to fulfill the requirements. Some elements to consider when documenting requirements include:

- *Event/Condition* – Describes when a requirement must be fulfilled and the condition under which the solution is operating to fulfill that requirement. For example, a website user clicks the "submit" button which triggers an event that writes to a database.
- *Subject* – Describes who, or what performs an operation. This may be a user or a system responding to, or performing, an event. For example, writing data to the database is performed by the system opening a file and writing data to it.
- *Rules* – Describes any rules that govern the outcome of the requirements. For example, specifically formatting a data fields date field within a database as mm/dd/yyyy.

Requirements Verification – Requirements verification is an iterative process that improves the accuracy and completeness of project requirements as they are being added as part of the final product. The practice decreases the need for rework and increases overall stakeholder satisfaction because requirements are verified through interaction with the client, project team, and other stakeholders. There is no one perfect method for verifying requirements. The most appropriate method for verifying requirements differs from project-to-project and even requirement-to-requirement. Some commonly used techniques for verifying requirements include:

- *Peer Reviews* – Peer reviews use subject matter experts to review and evaluate defined requirements.
- *Lessons Learned* – Lessons learned document the cause of issues and the reasoning behind any corrective action taken to address those issues. It also includes the processes necessary for identification, documentation, validation, and dissemination of lessons learned.
- *Federal Regulations/Mandates* – Compliance with federal regulations and mandates issued by federal agencies and departments.

Requirements Attributes - To aid tracking and decision making, it is important to also track the requirements' attributes. Examples of the attributes include but not limited to: Source of the requirement, priority, status, complexity, requirement type, etc. The attributes can be maintained using a requirements management tool, as part of the Requirements Definition document, or the Requirements Traceability Matrix.

Best Practices

- **Communicate** – The most important step in defining product requirements is talking to people. Communication between and within teams, and with other stakeholders should be frequent, effective, and efficient, reinforced by the project manager and other organizational leaders.
- **Iterative Process** – Requirements management is an ongoing, iterative process conducted throughout the project lifecycle. Manage requirements throughout the entire life of the project.
- **Review and Approve** – Defined requirements should be reviewed and approved by business owners and other project stakeholders.
- **Document** – Defined requirements should be centrally documented using some type of tracking system or log.
- **Unique Identifier** – Each requirement should have a unique identifier and should be recorded as a single line entry.
- **Traceability** – Requirement traceability should be centrally documented using some type of tracking system or log.

- **Review** – Regular reviews of requirements and their traceability is good project management practice. Depending on the complexity of the project, the review process can occur daily; but should happen at least weekly for even the simplest projects.

Practice Activities

- **Start Early** – Begin defining requirements early in the project life cycle.
- **Agree** – Get business stakeholders, end-users, and project teams all on the same page in regards to requirements.
- **Define** – When defining requirements capture all levels of requirements some of which may include product and project requirements, as well as business, functional, and technical requirements.
- **Perspective** – Take into account many perspectives when defining requirements. Such as business perspective, user perspective, and technical perspective.
- **Multiple Sources** – When defining requirements use sources from across the organization and include input from areas such as management, marketing, legal departments, end-user, business analysis, system engineers, technology hardware specialists, etc.
- **Analyze** – Analyze requirements using a process that involves the client, project team, and other stakeholders and also utilizes qualitative and quantitative analysis.
- **Verify** – Verify requirements using a process that involves the client, project team, and other stakeholders to ensure completeness.
- **Iterative Process** – Defining requirements is an iterative process conducted throughout the life of the project. Requirements should get more detailed as more project information becomes known.
- **Manage Requirements** – Requirements management ensures that the end product meets the needs and expectations of project stakeholders.