**Software Development Plan (SDP)**

**for the**

**Predictive Waste Water Management System**

**(PWMS)**

**Phase 1.0**

Prepared by

Fairleigh Dickinson Software Development - Analysis Team

**Security Notice**: The information contained within this document is Restricted. Unauthorized disclosure is prohibited. Failure to observe policy regarding proprietary information can result in disciplinary action, including dismissal, as well as result in a violation of proprietary rights and subject you and/or third parties to legal liability.

Purpose

This document is intended to be the software development plan for Predictive Waste Water Management System software.

Document Control

Document History

|  |  |  |
| --- | --- | --- |
| Date | Version | Comments |
| 2/14/2015 | 1.0II | After Internal Control Meeting |
|  | 2.0II | Revision with the release of Functional Baseline |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Authors

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Organization | Phone |
| Sirisha Venkannagari | Project Lead | FDU | 201 657 6175 |
| Bill Phillips | General Manager | FDU | 201-692-2258 |
|  |  |  |  |
|  |  |  |  |

Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Organization | Phone |
| Mark LaPlante | Product Manager | Eastech | 918-664-1212 |
| Bill Phillips | General Manager | FDU | 201-692-2258 |
| Amitha Peddi | SQA Engineer | FDU | 336-848-8346 |
| Kanchana | Project Manager | FDU | 732-306-0581 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sign Off List

|  |  |  |
| --- | --- | --- |
| Name | Title | Phone |
| Mark LaPlante | Product Manager | 918-664-1212 |
| Alfredo Tan | CS Department Chair | 2324 |
| Amitha Peddi | SQA Engineer | 336-848-8346 |
| Bill Phillips | General Manager | 201-692-2258 |
| Kanchana | Project Manager | 732-306-0581 |
|  |  |  |

Document Sign Off

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SECTION 1. SCOPE**

**1.1 INTRODUCTION**

This Software Development Plan (SDP) establishes the plan to be used during the development of software components of the Predictive Waste Water Management System (PWMS) for Eastech Flow Controls Company.

**1.2 SYSTEM OVERVIEW**

PWMS gathers (Data Gathering Subsystem) and stores data (Data Storage Subsystem) from the participatory network formed by a collection of Flow cell sensors and stores that data in order. The stored data is periodically monitored for anomalous or emergent situations in the waste water system being monitored (Audit and Alert subsystem). A framework shall be provided for use by the Data Analyst in order to create, package, and run predictive, optimization algorithms (Data Analysis Subsystem) to head off potential problems in the waste water network.

Since, we are using scenario/use case based approach, modeling of the system shall begin with the creation of scenarios and these shall evolve into the use cases for the PWMS. The roles (actors) of the PWMS shall be implemented by a security system that handles authentication and authorization for PWMS.

**1.3 DOCUMENT OVERVIEW**

This SDP describes the organization and procedures to be used by the Fairleigh Dickinson University, New Jersey (FDU) in performing software development for PWMS.

This plan is intended to be used by the sponsor, Eastech Flow Controls Company, Fairleigh Dickinson University, New Jersey (FDU) to monitor the procedures, management and contract work effort for the development of PWMS.

This SDP identifies applicable policies, requirements, and standards for PWMS project software development. It defines schedules, organization, resources, and processes to be followed for all software activities necessary to accomplish the development. This SDP contains no privacy considerations pertaining to the PWMS Project.

The SDP is organized as follows:

Section 2 provides an overview of the required work.

Section 3 describes plans for general software development activities.

Section 4 describes the details of all software planning, design, development, reengineering, integration, test, evaluation, product evaluation, Software Quality Assurance (SQA), and preparation for delivery activities.

Section 5 defines the project schedule and activity network.

Section 6 describes the project organization and the resources required to accomplish the work.

**1.4 PROJECT DELIVERABLES**

The project deliverables for PWMS include,

* Deliverables for the initial phase of this project shall be the RAD's for the PWMS. The RAD(s) will contain use cases, requirements, the functional, dynamic, and analysis models expressed in Unified Modeling Language (UML). Additional information shall include requirements for the proposed PWMS along with the test method used to verify each requirement. This establishes the Functional baseline.
* As part of the next phase, High Level Design documents shall be created. In these documents allocation of requirements from the RAD would be allocated to the PWMS subsystem's hardware and software components.
* Upon completion of the RAD(s), the subsystems would be arrived at by either procuring Commercial off the Shelf (COTS) systems or by software development. These decisions along with rationale for the decisions would be captured in the High Level Design Documents when the Allocated baseline is established.

**1.5 RELATIONSHIP TO OTHER PLANS**

The Software Configuration Management Plan (SCMP) which describes Configuration Identification, Control, Auditing and Status Accounting of the software components providing traceability of changes and visibility of process throughout the software development life cycle, is maintained as a separate document for PWMS.

# ORGANIZATIONAL STRUCTURE

The SCM organization will report directly to the Application Owner. The Organizational structure for Functional Baseline of PWMS subsystem

Project Manager

Kanchana

High Level RAD

Project Engineer

Sirisha

SDP, DG

General Manager

Bill Phillips

Engineer Sowmya( Data Storage RAD)

Engineer Amitha (SQA)

Engineer Santhosh (Access Control)

Engineer

Arun (Allocated BL, SCMP)

Engineer Santhosh

(Audit & Alert RAD)

**SECTION 2. OVERVIEW OF REQUIRED WORK**

Development activities in this plan are subject to the following requirements.

**2.1 SYSTEM AND SOFTWARE DEVELOPMENT REQUIREMENTS AND CONSTRAINTS**

a. Creating RAD documents to begin the object oriented analysis and design (OOA/OOD processes. Deliverables for the initial phase of this project shall be the RAD(S) for the PWMS.

b. Upon successful completion of the Requirements phase (i.e. the RAD(s) created are accepted), a set of High Level Design (HLD) Documents would be created in order to begin the design phase of the project.

c. Functional baseline review shall be scheduled to acknowledge agreement between the client and the developer

**2.2 SELECTED PROGRAM/ACQUISITION STRATEGY**

The PWMS is a new development. Software development will be accomplished and managed by FDU.

**2.3 PROJECT SCHEDULES AND RESOURCES**

Anticipated time of delivery for the PWMS RAD(s) (Functional Baseline) shall be

in February/March 2015 timeframe.

**2.4 OTHER REQUIREMENTS AND CONSTRAINTS**

None

**SECTION 3. PLANS FOR PERFORMING GENERAL SOFTWARE DEVELOPMENT ACTIVITIES**

**3.1 SOFTWARE DEVELOPMENT PROCESS**

The PWMS software development is tailored from the guidelines of IEEE/EIA 12207 and MIL-STD-498. Major project development phases and activities related to these standards are shown in Table 3-1.

**Table 3-1. Major PWMS Project Activities related to IEEE/EIA 12207 and MIL-STD-498**

|  |  |  |
| --- | --- | --- |
| **PWMS Activity** | **IEEE/EIA 12207.0 Activity** | **MIL-STD-498 Activity** |
| Project Planning and  oversight | Process implementation  Management Process | Project planning and oversight  Establishing a software development environment |
| Phase 1: Software  Requirements | Software reqts analysis | Software requirements analysis |
| Phase 2: Software Design HLD  Detailed Design | Software arch. design  Software detailed design | Software design |
| Phase 3: Software unit development, test, and integration of COTS products | Software coding and testing  Software integration | Software implementation and unit testing  Unit integration and testing |
| Phase 4: System Qualification  Test and Delivery | Software qualification testing  System integration  System qualification testing | CSCI qualification testing  CSCI/HWCI integration and testing  System qualification testing |
| Phase 5: Support of  installation and use | Software installation  Software acceptance support | Preparing for software use  Preparing for software transition |
| Software Quality Assurance | Quality assurance process | Software product evaluation  Software quality assurance |

The development process will integrate RAD(S) to begin the object oriented analysis and design (OOA/OOD) . As a part of the design phase, High Level Design (HLD) documents shall be developed capturing the information related to allocation of requirements from RAD to PWMS subsystems, procuring subsystems either by Commercial off the Shelf( COTS) systems or developing the software internally, and deployment decisions(i.e. on what machine(s) the PWMS will execute). Software design and coding will be performed using an object oriented design approach along with an agile development methodology to deliver work items as soon as possible after the Allocated Baseline is established.

Artifacts of the software development process shall be maintained in a repository identified in the SCMP. A status accounts system should be established to capture minutes of meetings, reviews, metrics, quality audits, product evaluation and preparation of project deliverables.

Software Configuration Management Plan (SCMP) will be developed as a separate document.

**3.2 GENERAL PLANS FOR SOFTWARE DEVELOPMENT**

PWMS software development shall follow the guidelines in IEEE/EIA 12207. The development approach will apply selected software engineering processes in accordance with the Software Engineering Institute (SEI) Capability Maturity Model (CMM).

**3.2.1 Software Development Methods**

The PWMS software development will apply the following general methods:

a. Phase 1: Requirement Analysis: Establishment of Functional Baseline. The project will follow the defined processes documented in Section 4 to conduct software requirements analysis. The requirement documents will contain use cases, requirements, the functional, dynamic, and analysis models expressed in Unified Modeling Language (UML). Deliverables to the Requirement phase are RAD(s)(Requirement Analysis Document) for the overall PWMS system and each identified subsystem. Satisfactory completion of a Software Requirements Review (SRR) will end Phase 1.l

b. Phase 2: High Level Design. The project will follow the defined processes documented in Section 4 to conduct High Level Design. During this phase the software and hardware components of the PWMS system will be identified. The requirements from RAD will be mapped to the subsystems, decisions regarding new software development and purchasing Commercial off-The Shelf products (COTS) for the subsystems will be made. Deployment decisions (i.e. on what machine(s) the PWMS will execute, along with rationale, are also captured in the HLD(s). A preliminary design review shall be scheduled to acknowledge agreement between the client and the developer. Satisfactory completion of a High Level Design Review will end Phase 2 and shall establish the Allocated Baseline.

c. Phase 3: Detailed Design : The software architecture will consist of reusable software components and components to be developed. The project will follow the defined processes documented in Section 4 to conduct object-oriented architectural and detailed software design of new software using an Agile Development approach. Sprints shall be performed through each phase following the establishment of the Allocated Baseline Emphasis will be placed on good software engineering principles such as information hiding and encapsulation, providing a complete description of processing, and the definition of all software and hardware component interfaces to facilitate software integration. Satisfactory completion of a Software Detailed Design Review will end Phase 3 and shall establish the Design Baseline.

The project will design and develop software to meet requirements that cannot be satisfied by reusable software. Software design will promote ease of future growth, specifically new application interfaces. Detailed design will also be expressed in a graphical form depicting classes, relationships, operations, and attributes.

1. The project will develop appropriate standards for design and coding methods for new software.

2. Software designers will select Commercial Off-The Shelf (COTS) reengineering tools and methods best suited to the structure of PWMS software and host/target processors. Reengineered design will be modified, as needed, to fully meet allocated requirements, and defined along with new software design, in the DDD.

d. Phase 4: Implementation: The project will develop new code , and document software following the processes in Section 5. New software will be based on principles of object-oriented design and exploit object-oriented features associated with the selected high-level language and Agile development environment. Code will be expected to conform to a single coding standard, source code must be supplemented with sufficient comments and standard code headers to meet commenting provisions of the coding standard and to promote understandability. Successful review of software code satisfying the coding standards will accomplish phase 4.

e. Phase 5: Testing: The project will conduct Qualification Testing according to Qualification Test Plans and Procedures, and document results in a Test Report.

f. Phase 6: Deployed PWMS System: After successfully analyzing the functionality of the software needed via multiple Agile Sprints, the software will be ready for deployment. More detail of this phase will be added to future updates of this SDP.

**3.2.2 Standards for the Software Products**

PWMS development will follow the documents that are listed in table 3-2 that impose standards that are applicable to software requirements, design, coding, testing, and deployment.

PWMS software documentation will comply with applicable directions contained in the documents listed in Table 3-2.

**Table 3-2. PWMS Documentation Guidelines**

|  |  |
| --- | --- |
| **PWMS Document** | **Primary Guideline** |
| Software Development Plan (SDP) | MIL-STD-498 DID: Software Development Plan |
| Software Requirements Description (SRD) | Requirement Analysis Document (RAD) |
| User Documentation Description (UDD) | 12207.1 6.30: User documentation description |
| Software Design Description (SDD) | 12207.1 6.16: Software design description |
| Software Interface Design Descr. (SIDD) | 12207.1 6.19: Software interface design description |
| Software Development Files (SDF) | SEPO’s Software Development File (SDF) Template V1.1 |
| Integration Test Plan/Procedures | 12207.1 6.27/28: Test or validation plan/procedures |
| Integration Test Report | 12207.1 6.29: Test or validation results report |
| Source Code Record (SCR) | 12207.1 6.24: Source code record |
| Executable Object Code Record (EOCR) | 12207.1 6.7: Executable object code record |
| Qualification Test Plan/Procedures | 12207.1 6.27/28: Test or validation plan/procedures |
| Qualification Test Report | 12207.1 6.29: Test or validation results report |

**3.2.3 Reusable Software Products**

This section identifies and describes the planning associated with software reuse during development of the PWMS project and provisions to promote future reuse of newly- developed software.

**3.2.4 Handling of Critical Requirements**

Compatibility of interfaces with other hardware and meeting the requirements mapped to the subsystems are important in successful development of the PWMS software. These processes will be continually monitored to identify, track, and evaluate potential risks.

**3.2.5 Recording of Rationale**

Software development processes for development of the PWMS Project software described in Section 4 of this SDP identify specific program decision information to be recorded. Additional rationale required for software development will be provided in future SDP updates. The rationale for utilizing COTS (Commercial off-The Shelf Products) and deployment hardware for PWMS will be captured. Test management decisions and rationale will be recorded.

Decisions and rationale on software design, coding, and unit testing will be recorded as well as system, software, and interface requirements; software engineering, development, and test environments; system and test cases in the configuration status accounting system.

**3.2.6 Access for Acquirer Review**

The Software Project Manager will arrange for periodic reviews of PWMS Project software processes and products at appropriate intervals. The Software Project Manager will direct consolidation of discrepancies and report the status of corrective actions taken in response to reported discrepancies.

**SECTION 4. PLANS FOR PERFORMING DETAILED SOFTWARE DEVELOPMENT ACTIVITIES**

The overall life cycle to be used for PWMS Formatter development is shown in Figure 4-1. A waterfall strategy with six major phases and supporting activities will be used. Major documents and management reviews are also shown in the figure. Detailed plans for individual activities are contained in the following subsections.

**Deliverables**

Requirement

Analysis

Documents

High Level Documents

Detail Design Documents

PWMS Subsystems

Sprints

**Phases**

Requirement Gathering &

Modelling

Testing

Implementation

Detailed Design

High Level Design

Real Time

Testing

**Functional Design**

**Baseline Allocated Baseline Product Operational**

Requirement Analysis Model Review

**Baseline Baseline Baseline**

Procure

Develop

Passing this review is the goal

of the work being proposed.

Integrate

**4-1. Proposed Software Development Lifecycle**

**4.1 PROJECT PLANNING AND OVERSIGHT**

Good software project management principles shall be followed throughout the PWMS life cycle. Software development activities shall be planned in advance and monitored carefully throughout the life cycle.

**RESPONSIBILITY:** the Software Project Manager is responsible for preparing and executing this

Software Development Plan.

**ENTRANCE CRITERIA:** High Level RAD (Requirement Analysis Document), Software Development Plan (SDP), Software Configuration Management Plan(SCMP).

**INPUTS:** High Level RAD, SDP, SCMP

**TASKS:**

1. Clarify the requirements

Baseline the requirements documents (RAD(s)) and control changes.

Select and tailor standards, processes, and policies to be used

2. Identify the processes

Identify the life cycle to be used on the project (see Figure 4-1 and section 3) Establish processes for each phase of the life cycle (see sections 4.2 to 4.6)

3. Document the plans

Document software plans in this Software Development Plan. Participation by appropriate parties in the SDP Peer Review and acceptance demonstrates agreements to commitments in this SDP.

Identify work products

Establish schedules for the timely completion of tasks

Estimate the size of software work products

Estimate the project effort, costs, and resources

Identify project risks

Establish the project organization

4. Track the progress

Monitor execution of planned activities

Analyze status and take action to modify processes, change resources, adjust schedules, or amend plans to satisfy requirements. Conduct Management Reviews and Status Reviews to determine status of ongoing operations

Review status of all activities periodically with PWMS Senior Management

5. Control the products

Establish a software configuration management activity to control base lined work products

Establish a software quality assurance activity to monitor project products and processes

6. Cultivate teamwork

Assemble and manage the required project staff and resources

Provide training needed for project staff

7. Apply appropriate technology

Consider reuse during all life-cycle phases

Satisfy guidelines of Capability Maturity Model

Strive to continually improve project processes

**OUTPUTS:** this Software Development Plan, PWMS system

**EXIT CRITERIA**: Satisfactory execution of this Software Development Plan.

**4.2 PHASE 1: SOFTWARE REQUIREMENTS**

The PWMS Team shall establish and document software requirements, including the quality characteristics, specifications, as described below.

**RESPONSIBILITY:** System engineering team

**ENTRANCE CRITERIA:** Approved System Level Requirement Analysis Document (RAD)

**INPUTS:** RAD(s)

**TASKS:**

1. For the PWMS subsystem there are five subsystems; a data gathering (DG) subsystem, a data storage (DS) subsystem, a data analysis (DA) subsystem, an Access Control (AC) subsystem, and an audit and alert (AAS) subsystem. RAD(s) for all the above mentioned subsystems are developed in this phase.

2. Document the requirements in Software Requirement Analysis Document (RAD), which contains use cases, the functional, dynamic, and analysis models which would be expressed in Unified Modeling Language (UML). Additional information shall include requirements for the proposed PWMS along with the test method used to verify each requirement.

3. Evaluate the requirements against criteria: internal consistency, testability, flexibility of software design, operation, and maintenance.

4. Develop a preliminary Requirement Analysis Document (RAD) for each subsystem.

5. Conduct Peer Reviews on RAD(s).

6. Document a checklist specifying all the requirements to be met in Software Requirements Review (SRR) of the Functional Baseline.

7. Conduct SRR.

**OUTPUTS:** Approved RAD(s)

**EXIT CRITERIA**: Successful completion of RAD checklist.

**4.3 PHASE 2 : HIGH LEVEL DESIGN**

The PWMS Team shall transform the requirements for the software item into an architecture that describes its top-level structure and identifies the software components.

**RESPONSIBILITY:** System engineering team

**ENTRANCE CRITERIA:** Successful completion of RAD (Requirement Analysis Document) checklist

**INPUTS:** Approved RAD(s)

**TASKS:**

1. Design for each of the subsystem will be prioritized.

2. The developers shall transform requirements into a top-level architecture.

3. The developer will be assigned a reporter to report the status of the work assigned during the sprint.

4. Document top-level design for interfaces; design for each component and document it in the High Level Design (HLD) document.

5. Document the decisions about procuring COTS systems, and deployment decisions in the High Level Design (HLD) document.

6. The developer can take inputs from the reporter, ask questionnaire and present the work to the client for feedback during the design phase.

7. Evaluate architecture against criteria: traceability to requirements, external consistency

with requirements, internal consistency between software components, appropriateness

of design methods and standards, feasibility of design, operation, and maintenance.

8. After successful completion of the work items Peer Reviews on HLD must be conducted

and HLD must be updated as required.

9. Document a checklist specifying the requirements to be met called Software Design Review (SDR).

10. Conduct Software Design Review (SDR).

**OUTPUTS:** High Level Design Documents (HLD(s)).

**EXIT CRITERIA**: Successful completion of SDR checklist.

**4.4 PHASE 3: DETAILED DESIGN**

The PWMS Team shall develop a detailed design for each software component. The software architecture will consist of reusable software components and detail design of components to be developed.

We shall follow Agile Methodology from the Detailed Design phase. It facilitates continuous customer interaction providing visibility to the client allowing all the phases of the lifecycle

(Planning, implementation and unit testing) to be executed during the time frame of a sprint, achieving high productivity.

Agile methodology will be followed at this phase.

**RESPONSIBILITY:** System engineering team

**ENTRANCE CRITERIA:** Successful completion of SDR checklist

**INPUTS:** Approved HLD(s)

**TASKS:**

1. The work items (software components) to be developed will be prioritized based on requirements and scheduled for agile sprints.

2. The developers need to transform top-level architecture into detail design for each software configuration item identified. Identify the requirement of COTS system for PWMS and replace the subsystems as per necessity with the COTS products.

3. Document detail design for interface, develop the design for each component and document it in the Detail Design Document (DDD).

4. The developer need to report about the status of the work to the SCM person during

each sprint.

5. Evaluate architecture against criteria: traceability to requirements, external consistency

with requirements, internal consistency between software components, appropriateness of design methods and standards, feasibility of design, operation, and maintenance.

6. Update the DDD as necessary based on the software design.

7. Conduct Peer Reviews on DDD, and update DDD as per suggestion.

8. Update the Software Design Review (SDR) checklist as per detailed design

specifications and requirements.

9. Conduct Software Design Review (SDR)

**OUTPUTS:** DDD, updated DDD for each sprint

**EXIT CRITERIA**: Successful completion of checklist for reviewing detail design.

**4.5 PHASE 4: IMPLEMENTATION**

The PWMS Team shall transform the software design into executable code and conduct unit testing of each unit following agile methodology.

**RESPONSIBILITY:** Software Development Team

**ENTRANCE CRITERIA:** Approved SDR and DDD

**INPUTS:** Approved DDD(s)

**TASKS:**

1. Development of code for each subsystem will be prioritized based on requirement. Each requirement should be specified with necessary test cases. The time frame for the implementation phase will be divided into "Sprints".

2. The developer will be assigned development of software modules/units and data for testing each software unit. The developer shall report to the project lead regarding the status of the work items during each sprint.

3. The developer must develop each software unit following coding standards.

4. The developer can take inputs from the project lead, communicate with the client regarding the requirements and demonstrate the work items to the client.

5. Evaluate code and unit test results considering this criteria: traceability to requirements, external consistency with requirements, internal consistency, test coverage of units, appropriateness of coding methods and standards used, feasibility of integration, testing, operation, and maintenance.

6. Prepare Software Development Files (SDF) which records the specification of requirements developed and the SCI label name assigned while developing each requirement and the unit test results.

7. Conduct Peer Reviews on software code and unit test results.

8. Document a checklist called Software Development Review (SDR) which specifies meeting of requirements by individual software components, coding standards to be verified like proper documentation and labeling of code.

**OUTPUTS:** Integrated, executable code meeting the requirements and successfully executing unit test cases.

**EXIT CRITERIA**: Successful completion of checklist (SDR) for reviewing software development and successful implementation of test cases.

**4.6 PHASE 5: TESTING**

The PWMS team shall conduct integration testing of all units by integrating software configuration items with hardware configuration items, and conduct system qualification testing.

The testing needs to be performed in two stages, where the software and hardware components of PWMS system are integrated and tested for their functioning and the other phase where the PWMS system is operated in real time environment and tested for its functioning.

**RESPONSIBILITY:** Software Development Team, Testing Team

**ENTRANCE CRITERIA:** Approved SDR (Software Development Review) and DDD for Integration Test Plans/Procedures; successful completion of SDR checklist to begin Testing

**INPUTS:** Approved SDR

**TASKS:**

**Phase I**

1. Integrate the software with hardware configuration items, manual operations, and other systems as necessary, into the PWMS system.

2. Document integration plans and procedures in the Integration Test Plan/Procedures (ITP/P) which tests the functioning of the system when interacted with system hardware.

3. Conduct integration tests and document results in the Integration Test Report (ITR).

4. Evaluate plans and tests against criteria: traceability to requirements, external consistency with requirements, internal consistency, test coverage of requirements, appropriateness of test standards and methods used, conformance to expected results, feasibility of software qualification testing, operation, and maintenance.

**Phase -II**

5. The integrated PWMS system is deployed in the real time environment and tested for its functioning under the supervision of Eastech and FDU.

6. Evaluate plans and tests against criteria: Test coverage of requirements, appropriateness of test standards and methods used, conformance to expected results, feasibility of software qualification testing, operation, and maintenance.

7. Document the results in Testing Report (TR).

**OUTPUTS:** Integrated PWMS system deployed in real time environment

**EXIT CRITERIA**: Successful completion of Testing Review checklist

**4.7 PHASE 6: DEPLOYMENT OF PWMS SYSTEM**

The PWMS Team shall install the completed system at the designated user sites.

**RESPONSIBILITY:** PWMS Project Manager, supported by members of the software development team, Eastech and testing team

**ENTRANCE CRITERIA:** Approved RAD(s), DDD for Qualification Test Plans/Procedures; successful completion of TRR.

**INPUTS:** Integrated, executable system, DDD

**TASKS:**

1. Assist PWMS in installation and acceptance test planning

2. Support acceptance testing as needed.

3. Provide life-cycle support through implementation of Engineering Change Requests and

Software Trouble Reports.

**OUTPUTS:** Executable system in place

**EXIT CRITERIA**: Successful working of the PWMS system meeting all the requirements.

**4.8 SOFTWARE QUALITY ASSURANCE**

The PWMS Project will apply a quality assurance process to provide adequate assurance that the software products and processes in the project life cycle conform to their specified requirements and adhere to their established plans.

The Quality Assurance Team will prepare and execute a PWMS SQA Plan containing quality standards, methodologies, procedures, tools, resources, schedules, and responsibilities for performing the following activities.

**4.8.1 Product Assurance**

SQA will assure that each required software product exists and has undergone software product evaluations (e.g., Peer Reviews specified in 4.9), testing, and problem resolution, as required.

**4.8.2 Process Assurance**

SQA will assure that each required process, activity, and task is performed in accordance with this plan and the SQA Plan.

**4.8.3 Independence in Software Project Evaluation**

Individuals responsible for conducting SQA evaluations shall not be the developers of the software product, performers of the process, or responsible for the software product or process. This does not preclude such individuals from taking part in these evaluations. The individuals responsible for assuring compliance shall have the resources, responsibility, authority, and organizational freedom to permit objective SQA evaluations and to initiate and verify corrective actions. Specifically, the PWMS SQA Team has the authority and freedom to report discrepancies and deviations directly to the Sponsor and to the management above the PWMS Project Manager.

**4.9 PROJECT REVIEWS**

The purpose of peer reviews and management reviews is to provide management with tracking and oversight of the progress of software development undertaken by the PWMS Project and fulfillment of requirements. Timely technical and management reviews at the appropriate level of detail facilitate information reporting and interchange that tracks progress against plans, identify and resolve action items, and verify appropriate expenditure of assigned resources.

**4.9.1 Peer Reviews**

Peer Reviews (Joint Technical Reviews) will be held on work products based on Table 4.9-1.

|  |  |
| --- | --- |
| **PWMS Work Product** | **Type of Peer Review** |
| Software Development Plan (SDP) | Formal Inspection |
| Requirement Analysis Documents (RAD) FB | Formal Inspection |
| High Level Design Document (HLD) AB | Formal Inspection |
| Detail Design Document (DDD) per Sprint DB | Formal Inspection |
| Software Development Repository | Define in SCMP |
| Integration Test Plans/Procedures | Technical review |
| Integration Test Report | Walkthrough |
| Qualification Test Plan/Procedures | Technical review |
| Qualification Test report | Walkthrough |

**4.9.2 Management Reviews**

Reviews will be held using guidelines in IEEE/EIA 12207.2 and IEEE Std 1028. Attendees shall include representatives from Eastech, the PWMS Project Manager, the Hardware Manager, the Software Project Manager, and other stake holders designated by the PWMS Project Manager. The focus will be on completion of review checklists derived from the Software Management, and metrics collected in accordance to the Project Measurement Plan.

**4.9.3 Status Reviews**

Project Status Reviews among all project members will be held at least weekly. Monthly status meetings with Eastech will be held by the PWMS Project Manager and other designated individuals. The focus of these informal reviews will be the status of current processes and completion status of Management Review checklists. It is anticipated that Eastech will participate in the development of PWMS.

**4.10 RISK MANAGEMENT**

The project risks are identified in Table 4.10-1. They will be tracked by the PWMS Project

Manager and reported in the monthly status meetings and in each Joint Management Review.

**Table 4.10-1. Risk Factors for the PWMS Project**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk Factor (and Reason for Significance)** | **Metrics**  **(and How Collected)** | **Risk Reduction/ Prevention Plan** | **Contingency**  **Plan** | **Entrance**  **Criteria** |
| **1. Requirements stability.**  (More people /project monitoring and  Organizations involved means more time, more requirements, and requirements. instability which leads to cost/schedule slippage) | Track number of requirements: original, new, changed, deleted (Count new, changed, and deleted requirements every month vs. previous month by automated tool which reads requirements database) | Report requirements stability with written estimated impact assessment to all parties every month and baseline (freeze) requirements at planned time in  schedule | Reduce functionality to meet cost and schedule or increase funds and schedule to  include new/changed requirements | If number of changed req. and number of new requirements per month exceeds 4 |
| **2: New Technology** (Learning curves for OOD design techniques and development language by the development team will take time | Count staff members qualified in each discipline (When staff member gets minimum score on OOD/ test or meets other objective criteria) | 1) Provide technical training in OOD  2) Use and enforce design and coding standards | Employ OOD expert consultant(s) to  assist development | If the design or coding milestones fall behind  schedule by more than  15%. |
| **3: Reliability of support organizations.** (Non-delivery of  required hardware and/or materials will directly impact on our ability to deliver the complete system) | Production, installation, and test milestones of each required outside element (Milestone progress  reports from all critical suppliers and vendors) | Provide a bonus structure that is delivering a required item for delivering on time | Identify alternative vendors and suppliers and purchase spares as backup | Delta time threshold over  2 weeks past scheduled delivery date |
| **4: Overall system testability.**  (No test facility exists  that matches the real system from beginning to end as it exists in the field.) | Percentage of Interfaces tested and verified as correct (Overall system interface requirements tracked through bi-weekly status reports until complete) | Develop a simulator and test environment that replicates as nearly as possible  the whole system, especially the interfaces | Move testing to field sites | If any required interfaces cannot be tested in a simulator environment  by their milestone due date |

**4.11 PROJECT MEASUREMENT PLAN**

4.11.1 Measurement Specifications

Effective management of the PWMS project will require ongoing measurement of project status.

**Table 4.11-1. PWMS Project Status Measurements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Issues** | **Core Measurement** | **Data Collection** | **Report Format** |
| Program  Size | Units/Objects planned vs. actual |  |  |
| Effort | Actual staff size vs. planned | Software Project  Manager |  |
| Cost | Actual costs expended vs. costs planned |  |  |
| Schedule | Actual dates vs. planned dates |  |  |
| Quality | Trouble Reports open vs. closed |  |  |
| Stability | Requirements Status/Traceability |  |  |
| Risks | As required if not covered above | As required |  |

**4.12 Program Size**

Size measurements are used to depict the magnitude of deliverable code and the status of code development on the project.

Functional size is measured in terms of the requirements.

The measure of the code production work necessary to implement the system can be measured in terms of source lines of code(shown here), or total objects, functions points, or software units based on the environment (e.g., language, code generation tools).

**4.12.1 Effort**

The object of this measurement is to illustrate management success in meeting staffing requirements for the software project.

**4.12.2 Costs**

This measurement shows expenditures of funds relative to the original plan.

**4.12.3 Schedule**

The milestone measurement shows tasks scheduled and those that have been accomplished.

**4.12.4 Quality**

Tracking the status of the program's trouble reports (TR's) shows insight into the quality of the product being developed.

**4.12.5 Stability**

The object of measuring the status of requirements is to demonstrate the stability of the implementation effort.

**4.12.6 Measurement Responsibilities**

Project status measurements will be collected under control of the Software Project Manager, with the assistance of the QA Team and inputs from the Software Development Team. Reports will be presented to the PWMS Project Manager and sponsors in the monthly status meetings and at each Joint Management Review. In addition, measurements reflecting problem areas or significant variances will be reviewed at the weekly Project Status Reviews and with the Project Manager on an as-needed basis.

**4.13 PROJECT TRAINING PLAN**

Project training will be provided by formal courses, informal “chalk talks,” or on-the-job training. Skills needed by the PWMS team include:

Project management/software project management skills (SCM (Software Configuration Management course)

The Capability Maturity Model and software process improvement

Computer program development

Peer reviews (Peer review Workshop)

Specific processes: planning and oversight, quality assurance, configuration management, etc.

**SECTION 5. PROJECT SCHEDULE PLAN**

