**TÀI LIỆU LẬP KẾ HOẠCH PHÁT TRIỂN PHẦN MỀM**

**CHO**

***DỰ ÁN GRAVIAD***

**GRVD-SDP01-V1.0**

**Soạn bởi:**

**Hoàng Đức Bách**

**Đại học công nghệ - Đại học quốc gia Hà Nội**

O.

**DOCUMENT CONVENTIONS**

Các quy ước tiêu chuẩn được sử dụng trong tài liệu này để hướng người đọc đến các phần cụ thể của văn bản. Các phần này cung cấp hướng dẫn và giải thích, đồng thời yêu cầu người dùng thay thế thông tin dành riêng cho dự án của họ bằng thông tin chung được cung cấp hoặc “điền vào chỗ trống”. Các quy ước được sử dụng trong tài liệu này được hiển thị dưới đây.

|  |  |
| --- | --- |
| *Văn bản* | Global changes. Items that appear in italics represent changes that can be made globally throughout the document. |
| <Text> | Unique changes. Items that appear in <angled brackets> represent items that need to be changed on an individual basis. The <angled brackets> are not meant to appear in the completed version of the document. |
| *Italics* | Instructions and explanations. Each section of the template has been annotated with a guidance box, derived from the MIL-STD-498 Data Item Description (DID) DI-IPSC-81427, to assist the reader in drafting the content. For example: |

***Hướng dẫn***

*The guidance box provides instructions and explanations, in italics, as required to assist the user in drafting their own information.*

This page intentionally left blank.

|  |  |
| --- | --- |
|  |  |
| **Quản lý dự án Quản lý chương trình**  **Hoàng Đức Bách Hoàng Đức Bách** | **Senior** |
|  |  |

**RECORD OF CHANGES**

\***A** - THÊM **M** - THAY ĐỔI **D** - XÓA

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PHIÊN BẢN | THỜI GIAN | VỊ TRÍ SỬA ĐỔI | A\*  M  D | DÒNG MÔ TẢ NGẮN | YÊU CẦU THAY ĐỔI SỐ |
| 1 | 04/24 |  |  | Khởi tạo |  |

**TABLE OF CONTENTS**

**Section Page**

[SECTION 1. SCOPE .................................................................................................................. 1 1](#_Toc159909)

[1.1 IDENTIFICATION ....................................................................................................... 1 1](#_Toc159910)

[1.2 SYSTEM OVERVIEW ................................................................................................. 1 1](#_Toc159911)

[1.3 DOCUMENT OVERVIEW .......................................................................................... 1 1](#_Toc159912)

[1.4 RELATIONSHIP TO OTHER PLANS ........................................................................ 1 2](#_Toc159913)

SECTION 2. REFERENCED DOCUMENTS ........................................................................... 2-1

SECTION 3. OVERVIEW OF REQUIRED WORK ................................................................. 3-1

SECTION 4. PLANS FOR PERFORMING GENERAL SOFTWARE DEVELOPMENT

ACTIVITIES ................................................................................................................................ 4-1

4.1 SOFTWARE DEVELOPMENT PROCESS ................................................................ 4-1

4.2 GENERAL PLANS FOR SOFTWARE DEVELOPMENT ......................................... 4-4

4.2.1 Software Development Methods ............................................................................ 4-4 4.2.2 Standards for Software Products ............................................................................ 4-5 4.2.3 Reusable Software Products .................................................................................. 4-5 4.2.4 Handling of Critical Requirements ........................................................................ 4-6 4.2.5 Computer Hardware Resource Utilization ............................................................. 4-8 4.2.6 Recording of Rationale .......................................................................................... 4-8

4.2.7 Access for Acquirer Review .................................................................................. 4-8

SECTION 5. PLANS FOR PERFORMING DETAILED SOFTWARE DEVELOPMENT

ACTIVITIES ................................................................................................................................ 5-1

5.1 PROJECT PLANNING AND OVERSIGHT ............................................................... 5-1

5.1.1 Software Development Planning ........................................................................... 5-1 5.1.2 CSCI Test Planning ................................................................................................ 5-2 5.1.3 System Test Planning ............................................................................................. 5-3 5.1.4 Software Installation Planning ............................................................................... 5-3 5.1.5 Software Transition Planning ................................................................................ 5-3

5.1.6 Following and Updating Plans, including Intervals for Management Review ...... 5-4

5.2 ESTABLISHING A SOFTWARE DEVELOPMENT ENVIRONMENT ................... 5-4

5.2.1 Software Engineering Environment ....................................................................... 5-4 5.2.2 Software Test Environment ................................................................................... 5-5 5.2.3 Software Development Library .............................................................................. 5-5 5.2.4 Software Development Files .................................................................................. 5-7

5.2.5 Non-Deliverable Software ..................................................................................... 5-8

5.3 SYSTEM REQUIREMENTS ANALYSIS ................................................................... 5-8

5.3.1 Analysis of User Input ........................................................................................... 5-9 5.3.2 Operational Concept .............................................................................................. 5-9 5.3.3 System Requirements ............................................................................................. 5-9

5.4 SYSTEM DESIGN ..................................................................................................... 5-10

5.4.1 System-Wide Design Decisions ........................................................................... 5-10

5.4.2 System Architectural Design................................................................................ 5-10

5.5 SOFTWARE REQUIREMENTS ANALYSIS ........................................................... 5-10

5.5.1 Software Requirements Development Process .................................................... 5-11

5.6 SOFTWARE DESIGN ................................................................................................ 5-12

5.6.1 CSCI-Wide Design Decisions .............................................................................. 5-13 5.6.2 CSCI Architectural Design .................................................................................. 5-13

5.6.3 CSCI Detailed Design .......................................................................................... 5-17

5.7 SOFTWARE IMPLEMENTATION AND UNIT TESTING ..................................... 5-19

5.7.1 Software Implementation ..................................................................................... 5-20 5.7.2 Unit Testing ......................................................................................................... 5-22

5.7.3 Test Case/Procedure Implementation .................................................................. 5-22

5.8 UNIT INTEGRATION AND TESTING .................................................................... 5-24

5.8.1 Purpose ................................................................................................................. 5-24 5.8.2 Roles and Responsibilities ................................................................................... 5-24 5.8.3 Entry Criteria ........................................................................................................ 5-24 5.8.4 Inputs .................................................................................................................... 5-25 5.8.5 Process Activities ................................................................................................. 5-25 5.8.6 Outputs ................................................................................................................. 5-26 5.8.7 Exit Criteria .......................................................................................................... 5-26

5.8.8 Process Measurements ......................................................................................... 5-26

5.9 CSCI QUALIFICATION TESTING ........................................................................... 5-26

5.9.1 Independence in CSCI Qualification Testing ...................................................... 5-27 5.9.2 Testing on the Target Computer System .............................................................. 5-27

5.9.3 Performing CSCI Qualification testing ................................................................ 5-27

5.10 CSCI/HWCI INTEGRATION AND TESTING ......................................................... 5-29

5.10.1 Preparing for CSCI/HWCI Integration and Testing ............................................. 5-29 5.10.2 Performing CSCI/HWCI Integration and Testing ................................................ 5-30 5.10.3 Revision and Retesting ........................................................................................ 5-30

5.10.4 Analyzing and Recording CSCI/HWCI Integration and Test Results ................. 5-30

5.11 SYSTEM QUALIFICATION TESTING .................................................................... 5-31

5.11.1 Independence in System Qualification Testing .................................................... 5-31 5.11.2 Testing on the Target Computer System .............................................................. 5-32 5.11.3 Preparing for System Qualification Testing ......................................................... 5-32 5.11.4 Dry Run of System Qualification Testing ............................................................ 5-32 5.11.5 Performing System Qualification Testing ............................................................ 5-32 5.11.6 Revision and Retesting ........................................................................................ 5-33

5.11.7 Analyzing and Recording System Qualification Test Results ............................. 5-34

5.12 PREPARING FOR SOFTWARE USE ....................................................................... 5-34

5.12.1 Preparing the Executable Software ...................................................................... 5-35 5.12.2 Preparing Version Descriptions for User Sites .................................................... 5-35 5.12.3 Preparing User Manuals ....................................................................................... 5-35 5.12.4 Installation at User Sites ...................................................................................... 5-35

5.13 PREPARING FOR SOFTWARE TRANSITION ....................................................... 5-36 5.14 SOFTWARE CONFIGURATION MANAGEMENT ................................................ 5-36 5.15 SOFTWARE PRODUCT EVALUATION ................................................................. 5-37

5.16 SOFTWARE QUALITY ASSURANCE .................................................................... 5-37

5.17 CORRECTIVE ACTION ............................................................................................ 5-37

5.18 JOINT TECHNICAL AND MANAGEMENT REVIEWS ........................................ 5-37

5.18.1 Joint Technical Reviews ...................................................................................... 5-38

5.18.2 Joint Management Reviews ................................................................................. 5-38

5.19 OTHER SOFTWARE DEVELOPMENT ACTIVITIES ............................................ 5-39

5.19.1 Risk Management ................................................................................................ 5-40 5.19.2 Software Management Indicators ........................................................................ 5-40 5.19.3 Security and Privacy ............................................................................................ 5-41

5.19.4 Subcontractor Management ................................................................................. 5-41

5.19.5 Interface With Software Independent Verification and Validation (IV&V) Agents5-42

5.19.6 Coordination With Associate Developers ............................................................ 5-42

5.19.7 Improvement of Project Processes ....................................................................... 5-42

SECTION 6. SCHEDULES AND ACTIVITY NETWORK ..................................................... 6-1

SECTION 7. PROJECT ORGANIZATION AND RESOURCES ............................................. 7-1

7.1 PROJECT ORGANIZATION ....................................................................................... 7-1

7.2 PROJECT RESOURCES .............................................................................................. 7-5

SECTION 8. NOTES .................................................................................................................. 8-1

8.1 ACRONYMS ................................................................................................................ 8-1

APPENDIX A. PROJECT PLAN .............................................................................................. A-1

**LIST OF FIGURES**

**Figure Page**

Figure 1-1. RBC System Software Overview ............................................................................. 1-3

Figure 4-1. *XY Project* Software Engineering Process Model .................................................... 4-3

Figure 4-2. Test Roles and Responsibilities ............................................................................... 4-3

Figure 5-1 Project Planning and Oversight Process .................................................................... 5-2

Figure 6-1. Master Build Schedule ............................................................................................. 6-1

Figure 7-1. *XY Project* Organizational Structure ........................................................................ 7-2

**LIST OF TABLES**

**Table Page**

Table 3-1. Key Features of Three DoD Program Strategies ....................................................... 3-1

Table 4-1. Development Activities Group Allocation ................................................................ 4-4 Table 4-2. Standards and Specifications Applicable to Software Development ........................ 4-5

Table 5-1. The *XY Project* SEE Tools ........................................................................................ 5-4

Table 5-2. Software Test Environment ....................................................................................... 5-5

Table 5-3. XY Project's Non Deliverable Software .................................................................... 5-8

Table 7-1. Roles and Responsibilities ......................................................................................... 7-3

Table 7-2. Personnel Requirements (Person yrs) ........................................................................ 7-6

# SCOPE

*Kế hoạch phát triển phần mềm (SDP) mô tả kế hoạch của nhà phát triển để thực hiện nỗ lực phát triển phần mềm. Thuật ngữ "phát triển phần mềm" có nghĩa là bao gồm sự phát triển mới, sửa đổi, tái sử dụng, tái cấu trúc, bảo trì và tất cả các hoạt động khác tạo ra các sản phẩm phần mềm. SDP cung cấp cho bên mua sản phẩm cái nhìn sâu sắc và một công cụ để giám sát, các quy trình cần tuân thủ để phát triển phần mềm, các phương pháp được sử dụng, cách tiếp cận cần tuân thủ cho từng hoạt động cũng như lịch trình, tổ chức và nguồn lực của dự án.*

## NHẬN BIẾT

*Đoạn này phải bao gồm thông tin nhận dạng đầy đủ về hệ thống và phần mềm mà tài liệu này áp dụng, bao gồm, nếu có, (các) số nhận dạng, (các) tiêu đề, (các) chữ viết tắt, (các) số phiên bản và số phát hành ( S).*

## TỔNG QUAN HỆ THỐNG

Dự án Graviad liên quan đến việc phát triển ứng dụng trên 3 nền tảng **desktop, website** và **mobile**.

Ứng dụng cung cấp dịch vụ giúp các nhà bán hàng quảng cáo sản phẩm, dịch vụ cho khách bằng poster và không gian 3D trực tiếp trên ứng dụng, giúp các khách hàng tiếp cẩn sản phẩm một cách tổng quan và thu hút nhất. Ứng dụng cung cấp một giao diện mang thiết kế hiện đại, linh hoạt giúp giảm hầu hết các chi phí thiết kế, nhà bán hàng chỉ cần bỏ ra chi phí cự nhỏ để tạo một không gian quảng cáo trên ứng dụng, người bán hàng chỉ cần cung cấp các thông tin cần thiết và có thể cung cấp file 3D của sản phẩm, ứng dụng sẽ cung cấp giao diện thu hút khách hàng, cung cấp đường link để người bán có thể chia sẻ trên các nền tảng khác để thu hút khách hàng, thay thế cho các poster tĩnh. Ứng dụng còn cung cấp dịch vụ thiết kế 3D sản phẩm riêng.

Ngoài ra, ứng dụng sẽ cung cấp nền tảng mạng xã hội cho người dùng có thể khám phá các sản phẩm của các nhà bán hàng

## TỔNG QUAN TÀI LIỆU

SDP này xác định các chính sách, yêu cầu và tiêu chuẩn hiện hành để phát triển phần mềm Dự án Graviad. Nó xác định lịch trình, tổ chức, nguồn lực và quy trình cần tuân thủ cho tất cả các hoạt động phần mềm cần thiết để hoàn thành quá trình phát triển. SDP này không chứa các cân nhắc về quyền riêng tư liên quan đến Dự án Graviad

Figure 1-1. GRVD Tổng quan kế hoạch

Section 2 các tài liệu tham khảo

Section 3 tổng quan các công việc cần thiết.

Section 4 kế hoạch thực hiện hoạt động phần mềm tổng hợp.

Section 5 mô tả chi tiết tất cả hoạt động lập kế hoạch, thiết kế, phát triển, tái cấu trúc, tích hợp phân tích và kiểm thử, Software Configuration Management (SCM), đánh giá sản phẩm, Software Quality Assurance (SQA), và các hoạt động triển khai chính thức.

Section 6 xác định lịch trình và mạng lưới hoạt động.

Section 7 mô tả tổ chức dự án và các nguồn lực cần thiết để hoàn thành công việc.

Section 8 chứa các từ viết tắt được sử dụng trong SDP này.

## LIÊN QUAN ĐẾN TÀI LIỆU NÀY

# TÀI LIỆU THAM KHẢO

The documents listed below were either used to create this document or are referenced in it:

1. Software Development and Documentation, MIL-STD-498
2. Technical Reviews and Audits for Systems, Equipment, and Computer Software, MIL-

STD-1521

1. Software Development Plan, Data Item Description DI-IPSC-81427
2. Software Project Planning Process, SSC San Diego
3. Software Development Plan Template, SSC San Diego
4. *XY Project* Software Configuration Management Plan
5. *XY Project* Software Quality Assurance Plan
6. *XY Project* Software Measurement Plan
7. Risk Management Process, SSC San Diego
8. A Description of the SSC San Diego Software Process Assets (SPA), SEPO, dated April

2001. See <http://sepo.spawar.navy.mil/>

1. SSC San Diego COTS Evaluation, Selection and Qualification Process
2. Institute of Electrical and Electronics Engineers (IEEE)/Electronic Industries Association

(EIA) 12207 Series, IEEE and EIA, March 1998

1. Etc.

# TỔNG QUAN VỀ QUY TRÌNH PHÁT TRIỂN

Dự án Graviad sẽ áp dụng chiến lược tăng trưởng (Cải tiến sản phẩm được lập kế hoạch trước) để phát triển và phát triển các khả năng chức năng của Phần mềm Hệ thống GRVD.

## Mô hình phát triển tăng trưởng (Incremental Model)

Mô hình phát triển tăng trưởng là mô hình phát triển từng bước, bổ sung các tính năng mới vào phiên bản cũ cho đến khi đạt được phiên bản hoàn chỉnh. Nó cho phép phát triển dần dần và kiểm tra từng bước một cách độc lập

Các pha của mô hình tăng trưởng

### Xác định giai đoạn ban đầu

Trong các thời điểm, kế hoạch chỉ được lập cho lần tăng tiếp theo chứ không phải cho bất kì loại kế hoạch dài hạn nào

### Thiết kế và phát triển

Cần cam kết phát triển các tính năng cốt lõi. Sau khi các tính năng cốt lõi được phát triển, chúng sẽ được tinh chỉnh để tăng mức độ khả năng bằng cách thêm các chức năng mới trong các phiên bản kế tiếp. Mỗi phiên bản phát triển gia tăng thường được phát triển bằng cách sử dụng mô hình phát triển thác nước lặp lại

### Kiểm thử và đánh giá

Mỗi phiên bản đầu tiên của phần mềm được gọi là MVP (Minimum Viable Product) được triển khai sớm trong quá trình đầu tiên. Điều này giúp thu thập phản hồi từ người dùng càng sớm càng t

### Hoàn thiện và triển khai

Sau khi phát triển qua n phiên bản, phần mềm có thể được triển khai chính thức

Mô hình phát triển tăng trưởng yêu cầu lập kế hoạch và thiết kế tốt

Mô hình tăng trưởng sẽ giúp phát triển nhanh chóng, linh hoạt, ít tốn kém. Dễ dàng hơn trong việc kiểm tra và sửa lỗi

# **KẾ HOẠCH THỰC HIỆN HOẠT ĐỘNG PHÁT TRIỂN PHẦN MỀM TỔNG HỢP**

## QUY TRÌNH PHÁT TRIỂN PHẦN MỀM

**Sys Arch**

**Design**

**System**

**Reqts**

**Analysis**

**System**

**Qual**

**Test**

**System**

**Integra**

**-**

**tion**

**Software**

**Installation**

**Software**

**Acceptance**

**Support**

***Hardware items***

***Software Item***

**Development Process Activities**

**Software**

**Qual**

**Test**

**Software**

**Integra**

**-**

**tion**

**Software**

**Code**

**& Test**

**Software**

**Detailed**

**Design**

**Software**

**Arch.**

**Design**

**Software**

**Reqts**

**.**

**Analysis**

|  |
| --- |
| **Supporting Processes: Documentation, Configuration Management, Quality Assurance, Verification, Validation,**  **Joint Review, Audit, Problem Resolution** |

**Organizational Processes: Management, Infrastructure, Improvement, Training**

Figure 4-1. Quy trình phát triển phần mềm *Graviad*

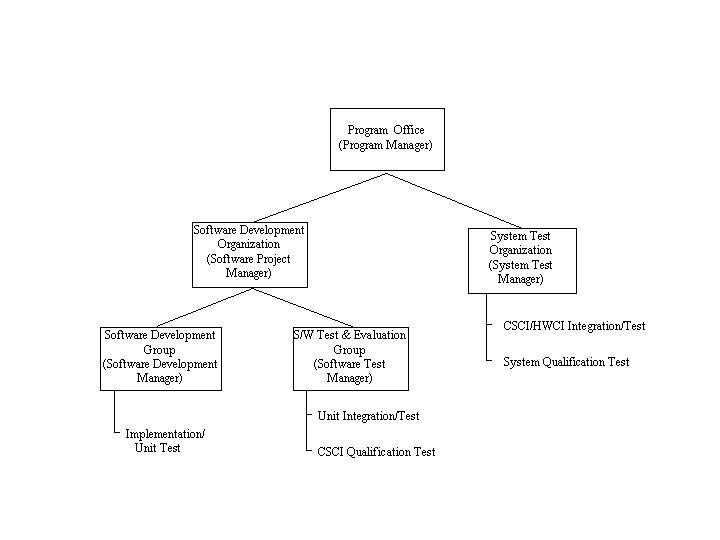


Figure 4-2. Kiểm thử và phản hồi

## KẾ HOẠCH TỔNG THỂ PHÁT TRIỂN PHẦN MỀM

### Software Development Methods

Việc phát triển phần mềm Dự án *Graviad* sẽ áp dụng các phương pháp chung sau::

1. Dự án sẽ tuân theo các quy trình đã xác định được ghi trong Phần 5 để tiến hành phân tích yêu cầu phần mềm và quản lý Đặc tả yêu cầu phần mềm (SRS). Thể hiện các yêu cầu phần mềm bằng ngôn ngữ nhằm giải quyết một mục tiêu hiệu suất duy nhất cho mỗi tuyên bố và thúc đẩy việc xác minh có thể đo lường được. Xây dựng một kiến trúc phần mềm bao gồm các thành phần phần mềm có thể tái sử dụng và các thành phần sẽ được phát triển. Phân bổ các yêu cầu phần mềm cho một hoặc nhiều thành phần của kiến trúc đó. Sử dụng công cụ cơ sở dữ liệu tự động để nắm bắt, tham chiếu chéo, theo dõi và ghi lại các yêu cầu.
2. Thiết kế phần mềm sẽ thúc đẩy sự phát triển dễ dàng trong tương lai, đặc biệt là các giao diện ứng dụng mới. Thiết kế cấp cao nhất sẽ được thể hiện dưới dạng đồ họa. Thiết kế chi tiết cũng sẽ được thể hiện dưới dạng đồ họa mô tả các lớp, mối quan hệ, hoạt động và thuộc tính. Dự án sẽ tuân thủ các tiêu chuẩn mà SDP yêu cầu về phương pháp thiết kế, mã hóa và thử nghiệm cho phần mềm mới.
3. Dự án sẽ sửa đổi (nếu cần), kiểm tra đơn vị, tích hợp và ghi lại phần mềm được sử dụng lại theo các quy trình tương tự được sử dụng cho phần mềm mới. Mặc dù mã được sử dụng lại sẽ không tuân theo một tiêu chuẩn mã hóa duy nhất, nhưng mã nguồn đã thay đổi phải được bổ sung đủ các nhận xét mới và tiêu đề mã tiêu chuẩn để đáp ứng các quy định về nhận xét của tiêu chuẩn mã hóa và để nâng cao tính dễ hiểu.

# KẾ HOẠCH THỰC HIỆN HOẠT ĐỘNG PHÁT TRIỂN PHẦN MỀM CHI TIẾT

## LẬP KẾ HOẠCH VÀ GIÁM SÁT DỰ ÁN

SDP này sẽ được duy trì và sửa đổi để phản ánh các kế hoạch, chính sách, quy trình, nguồn lực và tiêu chuẩn hiện tại ảnh hưởng đến Dự án Graviad. Trách nhiệm của Người quản lý dự án phần mềm là theo dõi những thay đổi về công nghệ trong ngành và chỉ đạo chương trình từ Người quản lý chương trình có thể yêu cầu sửa đổi kế hoạch này.

### Lập kế hoạch phát triển phần mềm

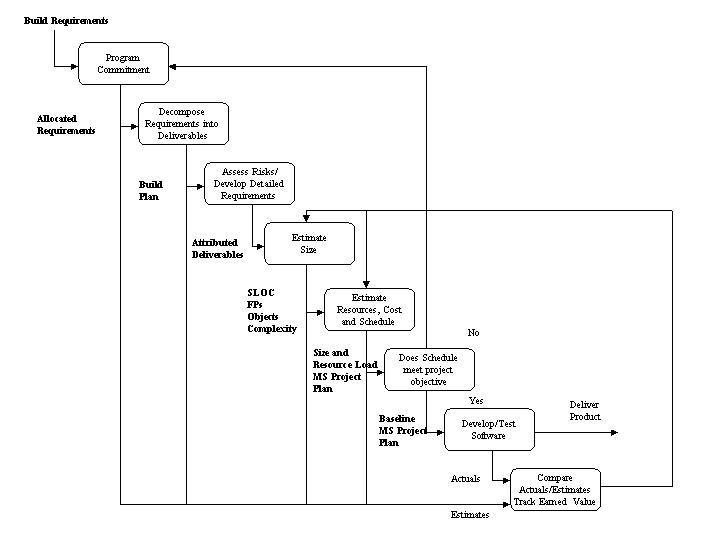


Figure 5-1 Quy trình lập kế hoạch và giám sát dự án

### Lập kế hoạch kiểm tra hệ thống

### Lập kế hoạch cài đặt phần mềm

## THIẾT LẬP MÔI TRƯỜNG PHÁT TRIỂN PHẦN MỀM

### Môi trường kỹ thuật phần mềm

### Môi trường kiểm thử phần mềm

### Thư viện phát triển phần mềm

### Thư viện phát triển chương trình

### Tập tin phát triển phần mềm

Nhóm Phát triển Phần mềm sẽ tổ chức các SDF để chứa các thông tin sau:

1. documents - các tài liệu đặc tả, phần mềm sẽ phát triển tuân theo các tài liệu này.
2. designs - các bản thiết kế cả đồ họa và hệ thống
3. sources - mã nguồn của phần mềm

## PHÂN TÍCH YÊU CẦU HỆ THỐNG

### Phân tích đầu vào của người dùng

1. Các yêu cầu làm rõ sẽ được đánh giá để xác định xem một hoặc nhiều yêu cầu có cần được diễn đạt lại hay không. Nếu việc làm rõ yêu cầu thay đổi một yêu cầu thì yêu cầu đó sẽ được xử lý như một yêu cầu thay đổi.
2. Các yêu cầu và lý do thay đổi sẽ được đánh giá theo tình trạng phát triển phần mềm để xác định tác động lên tiến độ, nỗ lực và chi phí cho việc phân tích, thiết kế, triển khai, tích hợp và thử nghiệm các yêu cầu phần mềm.
3. Các yêu cầu và lý do hỗ trợ cho việc từ bỏ/sai lệch các yêu cầu sẽ được đánh giá về tác động của chúng đối với tính toàn vẹn xử lý tổng thể của Dự án *Graviad*.

### Yêu cầu hệ thống

## THIẾT KẾ HỆ THỐNG

### Quyết định thiết kế toàn hệ thống

Hệ thống sẽ sử dụng mô hình MVC (Model-View-Controller) để phát triển. Dưới đây là mô tả chi tiết về cách hệ thống sử dụng mô hình này:

1. Model (Mô hình): Đại diện cho dữ liệu và logic liên quan đến dữ liệu trong hệ thống. Trong mô hình MVC, các đối tượng mô hình sẽ là nơi lưu trữ và xử lý dữ liệu, bao gồm các thao tác như truy vấn, cập nhật và xóa dữ liệu. Mô hình sẽ được thiết kế để đảm bảo tính toàn vẹn và nhất quán của dữ liệu.
2. View (Giao diện): Đại diện cho phần giao diện người dùng của hệ thống. Trong mô hình MVC, các đối tượng view sẽ hiển thị dữ liệu từ mô hình và cho phép người dùng tương tác với hệ thống thông qua các giao diện người dùng. Các view sẽ được thiết kế để đảm bảo sự trực quan và dễ sử dụng cho người dùng cuối.
3. Controller (Bộ điều khiển): Đại diện cho logic điều khiển và xử lý sự kiện trong hệ thống. Trong mô hình MVC, các đối tượng controller sẽ nhận các yêu cầu từ người dùng thông qua các giao diện người dùng, sau đó tương tác với mô hình để lấy và cập nhật dữ liệu và cuối cùng trả về kết quả cho người dùng thông qua các view tương ứng. Các controller sẽ được thiết kế để xử lý logic ứng dụng và điều hướng luồng điều khiển của hệ thống.

### Thiết kế kiến trúc hệ thống

## PHÂN TÍCH YÊU CẦU PHẦN MỀM

### Quy trình phát triển yêu cầu phần mềm

#### Quy trình hoạt động.

Các bước cần thiết để hoàn tất quá trình này được liệt kê dưới đây:

1. Thực hiện phân tích chi tiết và cập nhật các yêu cầu trong cơ sở dữ liệu để đảm bảo tính đầy đủ, có thể kiểm thử, nhất quán.
2. Thực hiện luồng dữ liệu chi tiết, luồng điều khiển và phân tích thuật toán.
3. Ghi lại phương pháp xác định chất lượng cho từng yêu cầu phần mềm và nhập vào cơ sở dữ liệu lập kế hoạch kiểm thử và làm thuộc tính cho từng yêu cầu trong cơ sở dữ liệu.
4. Xác định tiêu chí chấp nhận cho sửa đổi cơ bản (tức là % yêu cầu được nhắm mục tiêu vượt qua bài kiểm tra).
5. Ghi lại trong Kế hoạch xây dựng nội dung (tức là các yêu cầu, PC/R) của mức tăng và/hoặc bản sửa đổi hiện tại cũng như các tiêu chí chấp nhận liên quan.
6. Xây dựng dự thảo SRS
7. Cập nhật SRS dựa trên tổng hợp các phân tích và nhận xét tích lũy.
8. Xuất bản SRS hiện tại và Kế hoạch xây dựng.

#### Đầu vào.

1. Các yêu cầu thu nhập được
2. Các chi tiết, ý tưởng đã phác thảo

#### Đầu ra.

1. SRS
2. Xây dựng kế hoạch với các tiêu chí chấp nhận được ghi lại

#### Đo lường quy trình.

Lịch trình giá trị thu thập được

|  |  |
| --- | --- |
| Event | % Complete |
| Initial entry and attribution of build requirements into database | 10 |
| SRS Level requirements data flow, interfaces, and associated algorithms defined | 40 |
| Requirements allocated to build cataloguing and attribution complete in RM database. | 55 |
| SRS trace to SSS complete | 65 |
| Start Formal Inspection of SRS draft | 75 |
| Finish Formal Inspection and finalize SRS | 90 |
| SCCB Baseline Requirements | 100 |

## THIẾT KẾ PHẦN MỀM

Phương pháp thiết kế phần mềm cho Dự án *Graviad* sẽ là Thiết kế hướng đối tượng (OOP) cho server và Thiết kế hướng hàm (FP).

Thiết kế phần mềm sẽ diễn ra dần dần. Thiết kế cấp cao nhất cho phần mềm XY Project sẽ được phát triển trong giai đoạn thiết kế kiến trúc. Trong giai đoạn thiết kế chi tiết, phần mềm cần thiết để đáp ứng các khả năng được giao cho các bản dựng tăng dần của Dự án XY sẽ được thiết kế chi tiết đến mức đủ để triển khai nhằm hỗ trợ các khả năng được nhắm mục tiêu. Mô hình kiến trúc phần mềm của Dự án XY sẽ được đặt dưới sự quản lý cấu hình phát triển sau khi xem xét thiết kế cấp cao nhất. Việc cập nhật và cải tiến mô hình sẽ diễn ra dần dần trong quá trình thiết kế chi tiết. Một mô hình cập nhật sẽ được đặt dưới sự quản lý cấu hình phát triển sau mỗi lần kiểm tra chính thức thành công.

Bước thiết kế tạo ra một thiết kế dữ liệu, một thiết kế kiến trúc được cập nhật và một thiết kế theo thủ tục. Thiết kế dữ liệu biến đổi mô hình thông tin được tạo trong quá trình phân tích thành cấu trúc dữ liệu cần thiết để triển khai phần mềm. Thiết kế thủ tục chuyển đổi mô hình chức năng thành mô tả thủ tục của phần mềm.

### CSCI-Wide Design Decisions

### CSCI Architectural Design

### CSCI Architectural Modeling and Specification Process.

#### Mục đích.

Bước thiết kế phần mềm chuyển đổi mô hình được tạo trong quá trình phân tích yêu cầu thành các thành phần phần mềm riêng biệt sẽ được yêu cầu để triển khai phần mềm hệ thống. Thiết kế chi tiết chuyển mô hình kiến trúc thành phần phần mềm thành mô tả triển khai của phần mềm bao gồm Đơn vị phần mềm (SU), dữ liệu nội bộ, thuật toán, giao diện và luồng điều khiển của chúng.

Mô hình kiến trúc phần mềm của dự án sẽ được đặt dưới sự quản lý cấu hình phát triển sau khi Kiểm tra chính thức thiết kế kiến trúc. Các cập nhật và sàng lọc mô hình sẽ diễn ra tăng dần trong quá trình thiết kế chi tiết và mô hình cập nhật sẽ được đặt dưới sự quản lý cấu hình phát triển sau khi hoàn thành Kiểm tra chính thức đối với thiết kế chi tiết.

#### Đầu vào.

1. SRS

#### Quy trình hoạt động.

1. Xem xét tài liệu có sẵn về các yêu cầu, giả định và ràng buộc.
2. Xây dựng và đánh giá các phương án kiến trúc.
3. Lựa chọn mô hình kiến trúc. Xác định các thành phần phần mềm và đơn vị phần mềm nội bộ, sự tương tác, luồng dữ liệu và quy trình chức năng của chúng.
4. Phân bổ các yêu cầu đã biết cho các thành phần kiến trúc.
5. Lập tài liệu thiết kế kiến trúc theo phương pháp thiết kế chính thức.
6. Phát triển sơ đồ cơ sở dữ liệu ban đầu để hỗ trợ phần mềm hệ thống.

#### Đầu ra.

1. Tài liệu kiến trúc SSS
2. Tài liệu GUI
3. Lược đồ cơ sở dữ liệu.

#### Đo lường quy trình

1. Earned Value Schedule

|  |  |
| --- | --- |
| Event | % Complete |
| Start component level IDEF 0 diagrams | 10 |
| IDEF 0 at component level complete and associated algorithms defined | 40 |
| Each software component‟s SUs identified and structural relations defined | 55 |
| Architectural model trace to SRS complete | 65 |
| Start Formal Inspection of Architectural Model | 75 |
| Finish Formal Inspection and update of Architectural Model | 90 |
| CCB Baselines GUI Screens, initial database scheme, and Architectural Model (IDEF 0) | 100 |

1. Track % build allocated requirements traceable to architectural model components
2. Track actual vs planned staff hours expended
3. Track actual vs planned costs expended

## Thiết kế chi tiết

### Quy trình thiết kế chi tiết

#### Mục tiêu

Mục đích của Thiết kế chi tiết là xác định và mô tả từng bước các thành phần phần mềm CSCI theo các SU của chúng và các mối quan hệ bên trong của các SU đó bằng cách sử dụng biểu diễn đồ họa chi tiết.

Nếu việc phát triển thiết kế chi tiết xác định những thiếu sót trong thiết kế kiến trúc hoặc yêu cầu phần mềm, nhóm thiết kế phần mềm sẽ gửi các thay đổi yêu cầu và/hoặc cập nhật mô hình kiến trúc phát triển theo yêu cầu.

#### Đầu vào

1. SRS, kế hoạch xây dựng và cơ sở dữ liệu yêu cầu phần mềm được phân bổ
2. Tài liệu GUI
3. Lược đồ cơ sở dữ liệu.

#### Quy trình thiết kế

1. Thêm thuật toán, thuộc tính dữ liệu vào thao tác từng SU
2. Dự thảo tài liệu thiết kế chi tiết phần mềm SSD

#### Đầu ra.

1. SSD

#### Đo lường quy trình.

Earned Value Schedule

|  |  |
| --- | --- |
| **Event** | **% Complete** |
| Start analysis | 10 |
| Graphical representation complete and associated algorithms defined | 40 |
| SU structural relations, algorithms, and interfaces defined | 55 |
| Draft SDD | 65 |
| Start Formal Inspection of SDD | 75 |
| Finish Formal Inspection of SDD | 90 |
| LCCB Baseline SDD | 100 |

## TRIỂN KHAI PHẦN MỀM VÀ KIỂM TH

### Triển khai phần mềm

#### Purpose.

Mục đích của việc triển khai phần mềm và kiểm tra đơn vị là tạo ra các thành phần chức năng CSCI sẵn sàng cho kiểm tra trình độ chuyên môn bằng cách triển khai và kiểm tra từng SU thành phần như được xác định trong thiết kế chi tiết. Điều này có thể liên quan đến việc phát triển mã mới và/hoặc sửa đổi mã hiện có và tuân theo các nguyên tắc về phong cách lập trình được ghi.

#### Đầu vào

1. SRS
2. SĐ
3. GUI
4. Database.

#### Quy trình hoạt động.

1. Xem xét các yêu cầu được phân bổ cho SU, SDD và hướng dẫn về phong cách lập trình.
2. Duy trì nguồn làm việc cho nhiệm vụ được giao trong SDF.
3. Biên dịch phần mềm.
4. Thực hiện theo dõi từng hoạt động triển khai SU tới thành phần phần mềm gốc trong SDD.
5. Thực hiện kiểm thử theo kế hoạch kiểm thử đơn vị.
6. Yêu cầu về tài liệu và báo cáo và/hoặc lỗi thiết kế/sự không nhất quán.
7. Cập nhật thông tin số liệu, bao gồm cơ sở dữ liệu phân tích lỗi

#### Đầu ra.

1. SDFs

#### Quy trình hoạt động

|  |  |
| --- | --- |
| Event | Report % Complete |
| Start Code and Unit Test of component SUs | 10 |
| Complete Code and Unit Tests draft | 50 |
| Tracing of each SUs Unit Test to component allocated requirements complete | 60 |
| SU implementation technical review complete | 70 |
| Execution of Unit Test Complete, SU source code and Unit Tests updated | 85 |
| Component SDF validated by QA and IV&V | 90 |
| Component SU Code Checked into SDL | 100 |

**5.7.2 Unit Testing**

#### [Sample]

The purpose of the unit testing addressed in Section 5.7.1 is to identify and correct as many internal logic errors as possible. Problems not uncovered by unit testing are, in general more difficult to isolate when uncovered at the CSCI level. Unit testing will be conducted throughout the implementation process, first as part of the initial development process and later as changes to the unit are made. Unit tests will be repeatable and may be conducted at any point in the implementation process in accordance with the approved unit test plan. For the purposes of unit testing, a unit is an object. The goal for unit testing by developers is to perform path testing in which every affected branch is navigated in all possible directions at least once and every affected line of code is executed at least once. Unit test drivers and stubs will be developed as needed and will be placed under CM as part of the overall test utility. All unit test results will be recorded in the SDFs.

#### [End Sample]

**5.7.3 Test Case/Procedure Implementation**

##### Guidance

*The developer shall implement the test cases in terms of their test procedures and test data for verifying the software requirements corresponding to each CSCI.*

#### [Sample]

**5.7.3.1 Purpose**. The purpose of this process is to develop detailed steps for controlling tests, injecting inputs, recording results, and comparing actual to expected results. Document test case procedures in the final STD.

**5.7.3.2 Roles and Responsibilities**. The roles and responsibilities for this process are listed below:

1. *Software Test and Evaluation Group* performs analysis to implement the test case defined test procedures
2. *SQA* *Group* verifies process is performed
3. *IV&V Group* validates traceability
4. *SCM* *Group* performs CM of developed artifacts.

**5.7.3.3 Entry Criteria**. The entry criteria for this process are listed below:

1. Test cases designed
2. Software performance requirements allocated to test cases
3. GUI screens designed.

**5.7.3.4 Inputs**. The inputs for this process are listed below:

1. GUI screen and/or operator manual
2. Test case designs.

**5.7.3.5 Process Activities**. The steps needed to complete this process are listed below:

1. Review software GUI and/or operator manuals to identify methods of operator input, use of simulator/emulator tools, and software data recording.
2. Define and document detailed test procedure steps for providing inputs for test cases.
3. Prepare input data files to provide test stimuli. Prepare operator logs for the component under test as well as the test environment.
4. Define evaluation steps for conducting post-test analysis and comparing actual and expected test results.
5. Define and document the test procedures in the STD following the instructions of the assigned documentation standard.
6. (Recommended Option). Develop test harness with automated test procedures and test result analysis.
7. Trace specific test case procedures to requirements allocated to parent test case.
8. The *SQA Group* and *IV&V Group* provide analysis and comment on the test procedures.
9. Conduct a technical review to verify consistency of the STD test procedures (or test harness/procedures) with the test case descriptions, the STP, baselined requirements, and planning documents. Recommend revisions, as appropriate.
10. Revise draft STD (or test harness/procedures) to correct discrepancies and incorporate recommended changes to test case definitions and test procedures.
11. The LCCB baselines the test procedures in the STD placing them in the SDL.

**5.7.3.6 Outputs**. The outputs from this process are listed below:

1. Baselined STD containing test procedures
2. Test data files compiled.

**5.7.3.7 Exit Criteria**. The exit criteria for this process is that the STD is approved and under

CM.

**5.7.3.8 Process Measurements**. The following data shall be collected:

1. Earned Value Schedule

(Note: Earned Value will be tracked against each Test Case Procedure suite)

|  |  |
| --- | --- |
| Event | % Complete |
| Start Test Case Procedures development | 10 |
| Complete Test Case Procedures draft | 50 |
| Tracing of Test Case Procedures to test case allocated requirements complete | 60 |
| Test Procedure Technical Review complete | 70 |
| LCCB Baselines STD Test Procedures into SDL | 100 |

1. Track total number of test procedures required to implement all test cases
2. Track % test procedures completed
3. Track actual vs planned staff hours expended
4. Track actual vs planned costs expended

#### [End Sample] 5.8 UNIT INTEGRATION AND TESTING

##### Guidance

*This paragraph shall describe the approach to be followed for unit integration and testing.*

***Note****: Unit integration and testing means integrating the software corresponding to two or more SUs, testing the resulting software to ensure that it works together as intended, and continuing this process until all software in each CSCI is integrated and tested. The last stage of this testing is developer-internal CSCI testing.*

#### [Sample]

##### 5.8.1 Purpose

The purpose of integration and test is to incrementally integrate SUs into larger software components, and components into a complete system. Testing is performed to validate each component's ability to meet its stated requirements and to ensure interoperability of the major software components.

Integration continues until all software components are integrated with the system-level hardware suite into a single functioning system.

##### 5.8.2 Roles and Responsibilities

The roles and responsibilities for this process are listed below:

1. *Software Test* *and Evaluation Group* plans, integrates, and executes tests
2. *Software Development Group* performs updates in response to developmental P/CRs
3. *SQA* *Group* verifies process is performed and validates test results
4. *IV&V Group* validates traceability of successful tests runs
5. *SCM Group* performs configuration control of test reports, and updated project artifacts.

##### 5.8.3 Entry Criteria

The entry criteria for this process are listed below:

1. STP
2. Build Plan and schedule
3. Any previous integration test reports
4. Approved architectural design
5. Approved STD with Test Procedures. **5.8.4 Inputs**

The inputs for this process are listed below:

1. Updated SDL (includes qualified SUs)
2. Baselined STD (includes execution ready test cases with included test procedures)
3. Integration Test Report form
4. P/CR forms
5. Test drivers and test inspection tools as required by STD

##### 5.8.5 Process Activities

The steps needed to complete this process are listed below:

1. *Software Test and Evaluation Group* reviews the Build Plan and schedule, the list of software units/components to be included in the build, and the results of any previous integration tests to determine the software and software fixes to be tested. Determine a set of test cases to be used to test the build.
2. For the requirements identified for the build, develop an integration test plan. The integration test plan should identify the sequence of SU and component integration, traceability to the requirements to be validated by the test, test tools and drivers to be used, and the applicable test cases including procedures for conducting the test and analyzing the results. Document the integration test plan.
3. Perform a Technical Review of the integration test plan and resolve all comments.
4. *Software Test and Evaluation Group* places the integration test plan under developmental configuration control.
5. *Integration Test Team* requests and receives increment build, installs it in the integration test area.
6. Conduct the test in accordance with the integration test plan procedures. Record test results as they are observed.
7. Perform any required post test analysis or data reduction to determine pass/fail criteria as specified in the integration test plan.
8. Compare test results with expected results. If discrepancies are found, attempt to determine whether errors are associated with the software, test/test driver, or hardware.
9. Document all test results using the project integration test report form. This report should contain all data recorded from test tools, test results, and deviations from the test plan. Document any problems detected on a P/CR form. Document requirements satisfactorily tested in the test report.
10. Determine if another incremental build is required. Proceed with activity (e) above if needed. If not proceed to next step.
11. The *SQA Group* and *IV&V Group* provide analysis and comment on the test results.
12. *Software Test and Evaluation Manager* reviews the integration test report and any P/CRs for accuracy and thoroughness. File the test report in the project history file in the document library and submit a copy of the test report and P/CRs to the *Software Development Manager* for review and analysis.
13. TheLCCB baselines all updated developmental and test artifacts placing them in the SDL.

##### 5.8.6 Outputs

The outputs from this process are listed below:

1. Updated SDFs
2. Updated SDL (i.e., source code and test procedures)
3. Project History File updated with test results
4. P/CRs submitted for all detected problems.

##### 5.8.7 Exit Criteria

The exit criteria for this process are listed below:

1. Integration test report reviewed and approved
2. All build allocated testable requirements have been successfully tested
3. CCB baselines SDL with updated source code and test procedures
4. Project History File updated with test results.

##### 5.8.8 Process Measurements

The following data shall be collected:

1. Earned Value Schedule

|  |  |
| --- | --- |
| **Event** | **% Complete** |
| Start Analysis | 10 |
| Integration test plan drafted | 20 |
| Integration test plan completes Technical Review | 35 |
| Initial increment test begins | 50 |
| Incremental integration testing completed | 80 |
| Test results validated by QA and IV&V | 90 |
| LCCB Baseline updated SDL artifacts | 100 |

1. Track % build allocated testable requirements successfully completing integration test

(Note: Test Procedures trace to testable requirements. See Test Case Design)

1. Track SU successfully completing integration as a percent of total targeted for integration
2. Track open versus closed STRs
3. Track actual vs planned staff hours expended
4. Track actual vs planned costs expended.

#### [End Sample] 5.9 CSCI QUALIFICATION TESTING

##### Guidance

*This paragraph shall describe the approach to be followed for CSCI qualification which is performed to demonstrate to the acquirer that CSCI requirements have been met. It covers the CSCI requirements in SRSs and in associated IRS.*

#### [Sample]

The *CSCI Test Team* will conduct CSCI Qualification Testing on each of the *XY Project* CSCIs. The purpose of CSCI Qualification Testing is to verify satisfaction of CSCI performance requirements as documented in the SRS.

The following paragraphs describe CSCI Qualification Testing processes and assignment of responsibilities. The STP and STDs will provide the detailed plan and design for CSCI Qualification Testing in conformance with these processes.

Off-The-Shelf CSCIs, both COTS and GOTS, will be qualified in accordance with the SSC San Diego COTS Evaluation, Selection, and Qualification process, reference (g).

##### 5.9.1 Independence in CSCI Qualification Testing

The *Software Test and Evaluation Manager* will be responsible for CSCI Qualification testing, reporting directly to the *Software Project Manager*. The *Software Test and Evaluation Manager* will designate a Test Director and assign personnel to the *CSCI Test Team*.

##### 5.9.2 Testing on the Target Computer System

CSCI Qualification testing for *XY Project* will take place only on the target computer system, interfaced with hardware and software components of the software test environment. This restriction will assure that testing of timing, capacity, throughput, and responsiveness of *XY Project* CSCI components with respect to performance requirements can be accurately assessed.

##### 5.9.3 Performing CSCI Qualification testing

**5.9.3.1 Purpose**. The purpose of CSCI Qualification Testing is to validate satisfaction of requirements documented in the SRS and targeted for a specific build. CSCI Qualification Testing is to ensure suitability for release to an external test agency and/or user community. The

STP and STDs will provide the detailed plan and design for Software Qualification Testing **5.9.3.2 Roles and Responsibilities**. The roles and responsibilities for this process are listed below:

1. *CSCI Test Team* executes tests
2. *Software Development Group* performs updates in response to developmental P/CRs
3. *SQA Group* verifies process are performed and validates test results
4. *IV&V Group* validates traceability of successful tests runs
5. *SCM Group* performs configuration control of test reports, and updated project artifacts **5.9.3.3 Entry Criteria**. The entry criteria for this process are listed below:
6. Software components under baseline control
7. Unit, and Unit Integration Testing completed
8. High priority P/CRs resulting from integration testing closed
9. Software test materials completed
10. Software test environment certified
11. Approved Build Plan and schedule
12. Approved architectural design
13. Approved SDT with Test Procedures
14. Integration and test of system software completed and report approved.

**5.9.3.4 Inputs**. The inputs for this process are listed below:

1. Updated SDL (includes fully tested SUs)
2. Baselined STD (includes execution ready test cases with include test procedures) c. Integration Test Report
3. P/CR forms
4. Test drivers and test inspection tools as required by STD.

**5.9.3.5 Process Activities**. The steps needed to complete this process are listed below:

1. *Software Test and Evaluation Manager c*onducts Test Readiness Review (TRR) with the *Software Development Manager* to address the following issues:
   1. Verify high priority P/CRs resulting from integration testing are closed and corrections tested.
   2. Verify STP and STDs are complete and under control.
   3. Verify software test materials (e.g. data files, test environment, and Component operator logs) are complete and in conformance with STDs.
   4. Verify test environment is certified and operational. Verify that test environment configuration logs are up-to-date.
2. The *Software Project Manager* working with the *Software Test and Evaluation Manager* complete TRR. Authorize start of CSCI Qualification Testing or direct correction of discrepancies in STP, STD, test materials, and test environment configuration, as necessary and including return to the integration and test process.
3. *CSCI Test Team* loads and initializes software to meet prescribed test conditions for execution of complete test case suite.
4. Execute tests, following scripted test steps. Record results on operator logs and automated recording media.
5. Process recorded test data to reduce and format them in textual and graphic form.
6. Review test case purposes and expected results documented in the STD.
7. Evaluate actual test results by examining the Test History log, P/CRs, and recorded test data.
8. Ensure P/CRs opened during software testing have been tested and closed.
9. Ensure that adequate regression testing of the updated version has been conducted and any resulting P/CRs have been tested and closed.
10. Determine degree of compliance to acceptance criteria and record percent requirements satisfactorily tested.
11. Write test recommendations, promotion to more complex testing, retest, or system acceptance.
12. Prepare draft test report, following the instructions of the assigned documentation standard. Include calculated test metrics data. Append marked-up data from the STD, processed recorded test data to support test results and recommendations.
13. The *Software Test and Evaluation Manager, IV&V Manager* and the *SQA Manager* will review the draft test report to verify consistency of the report with recorded result data. Recommend revisions, as appropriate.
14. Revise draft test report to correct discrepancies and incorporate recommended changes.
15. The *Software Project Manager* will approve acceptance or direct corrective action as necessary. This may include P/CR repair and re-qualification testing, or a return to integration and test.
16. LCCB baselines updated SDL artifacts to establish final system build configuration.

**5.9.3.6 Outputs**. The outputs from this process are listed below:

1. Updated SDL
2. Final Test Report
3. Defect data collected for process analysis.

**5.9.3.7 Exit Criteria**. The exit criteria for this process are listed below:

1. Test results logged, including percent targeted requirements satisfactorily tested and degree of compliance to acceptance criteria.
2. Recorded test data processed.
3. Measurement database updated.
4. Test report indicating acceptance approved.
5. LCCB baselines components comprising the qualified system.

**5.9.3.8 Process Measurements**. The following data shall be collected:

1. Earned Value Schedule

|  |  |
| --- | --- |
| **Event** | **% Complete** |
| Start TRR | 10 |
| Initiate Qualification Test | 25 |
| Complete Qualification Test Cycle | 80 |
| Complete Review final Test Report | 85 |
| *Software Project Manager* approves Test Report recommendation for acceptance | 90 |
| LCCB Baselines Test Report and updates SDL | 100 |

1. Track % build allocated testable requirements successfully completing qualification test
2. Track open versus closed P/CRs
3. Track actual vs planned staff hours expended
4. Track actual vs planned costs expended.

#### [End Sample] 5.10 CSCI/HWCI INTEGRATION AND TESTING

##### Guidance

*This paragraph shall describe the approach to be followed for participating in CSCI/HWCI integration and testing. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.*

***Note 1****: CSCI/HWCI integration and testing means integrating CSCIs with interfacing HWCIs and CSCIs, testing the resulting groupings to determine whether they work together as intended, and continuing this process until all CSCIs and HWCIs in the system are integrated and tested.*

#### [Sample]

Upon completion of the testing and integration of all builds and the preliminary integration testing of all CSCI components comprising the *XY Project*, the software is ready for CSCI/HWCI integration testing. CSCI/HWCI testing will be conducted in accordance with the STP and the applicable STD after successful completion of a Test Readiness Review. The objective is to validate that the *XY Project* hardware and software components can individually be interfaced in accordance with the SRS and IRS/IDD requirements and that the components taken as a system are stable enough to proceed with System Qualification Testing. These tests are formally conducted by a test team composed of personnel from the *System Test Organization*, and in conjunction with the software developers. The *IV&V Group* will be present during these tests as an observer. A Software Test Report (STR) is prepared to document the results of the test, as delineated in the STP.

##### 5.10.1 Preparing for CSCI/HWCI Integration and Testing

Preparation for integration and testing of software and hardware will require the acquisition of the HW and/or HW simulators in configurations to support all planned platforms Test plans and procedures will need to be developed and implemented. The *System Test Organization* in preparing for CSCI/HWCI Qualification Testing will perform the following processes:

**5.10.1.1 Plan Software Tests Process**. The *System Test Organization* will plan CSCI/HWCI software tests, identify test resources and schedule, and prepare the inputs to the STP.

**5.10.1.2 Develop Test Cases Process**. The *System Test Organization* will develop a set of test cases in keeping with the overall test concept and objectives that adequately verify all allocated performance requirements for the CSCI.

**5.10.1.3 Develop Test Procedures Process**. The *System Test Organization* will develop detailed steps for controlling tests, injecting inputs, recording results, and comparing actual to expected results. Once verified, the *System Test Organization* will document test cases and steps in the STD.

**5.10.1.4 Prepare Test Environment Process**. The *System Test Organization* will define, develop, integrate, verify, and place a test environment under control that will support the CSCI test concept and objectives.

**5.10.1.5 Assuring Readiness for Hardware Configuration Item/Computer Software Configuration Item Integration and Testing**. The testing will be performed with the equipment at the *Program Manager’s* designated *System Test Organization.* The *System Test Organization* will verify that the software and hardware is under baseline control; informal testing of the CSCI component has been completed satisfactorily; informal testing of the hardware has been completed satisfactorily; test materials are completed; and test personnel and other resources are ready for the start of CSCI/HWCI testing.

**5.10.1.6 Conduct Test Readiness Review**. As a step preliminary to the conduct of

CSCI/HWCI integration and testing, a review of the individual components or activities which go into the makeup of the testing will be conducted by the *Program Manager* in the form of a Test Readiness Review (TRR)*.*

##### 5.10.2 Performing CSCI/HWCI Integration and Testing

The *System Test Organization* will execute CSCI tests in the controlled test environment and collect data recorded by operators and automated means during testing.

##### 5.10.3 Revision and Retesting

Revision and re-testing of a CSCI depends on analysis of test results and correct identification of problems detected during both test conduct and post-test analysis. Therefore, one of the processes described incorporates revision and re-testing.

##### 5.10.4 Analyzing and Recording CSCI/HWCI Integration and Test Results

The final results of the *XY Project* CSCI/HWCI integration and testing will be documented. The test report will be distributed for review and approval.

**5.10.4.1 Analyze and Evaluate Results, Revise Tests Process**. The *System Test Organization* will compare recorded and processed CSCI/HWCI integration and testing results data with expected results to identify, isolate, and assess the probable causes of errors in the CSCIs under test, hardware, test environment, or test materials. Assigned test personnel will revise tests or prepare P/CRs, as required.

**5.10.4.2 Report Test Results Process**. The *System Test Organization* will evaluate the results of CSCI/HWCI integration tests, determine that test results meet defined objectives, determine that P/CRs are closed and tested, and publish the STR.

**[End Sample]**

#### 5.11 SYSTEM QUALIFICATION TESTING

##### Guidance

*This paragraph shall describe the approach to be followed for participating in system qualification testing. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.*

***Note 1****: System qualification testing is performed to demonstrate to the acquirer that system requirements have been met. It covers the system requirements in the SSSs and in associated IRSs. This testing contrasts with developer-internal system testing, performed as the final stage of CSCI/HWCI integration and testing.*

***Note 2****: If a system is developed in multiple builds, qualification testing of the completed system will not occur until the final build. System qualification testing in each build should be interpreted to mean planning and performing tests of the current build of the system to ensure that the system requirements to be implemented in that build have been met.*

#### [Sample]

*XY Project* System Qualification Testing consists of complementary and progressive test phases. Once CSCI/HWCI integration and testing is completed by the *System Test Organization*, System Qualification Testing (SQT) begins.

A single STP will be generated to address the planning for all levels of software SQT. A STD will be generated for each CSCI component, documenting the test procedures to be run to verify each requirement in the SRS for that component. A cross reference matrix will be provided, using the project wide requirements traceability database, to document the test or tests that satisfy each SRS requirement. A STR will be generated for each CSCI component, documenting the results of each CSCI component test. The *System Test Organization* is responsible for generating the appropriate test documentation. The *Software Project Manager* is responsible for conduct of the tests. The software developers will be responsible for supplying test procedures for SUs they develop to the *System Test Organization* for each CSCI component so that they can be incorporated into the STD for system.

The System Qualification Testing (SQT) test will be used to validate the entire systems performance.

The System Qualification Testing (SQT) is the *Program Manager’s* approved and witnessed series of tests that demonstrate compliance with the requirements set forth in the SSS. The SQT is the acceptance mechanism for the developer‟s compliance with the terms of tasking by the *Program Manager.*

#### [End Sample]

**5.11.1 Independence in System Qualification Testing**

##### Guidance

*The person(s) responsible for fulfilling the requirements in this section shall not be the persons who performed detailed design or implementation of software in the system. This does not preclude persons who performed detailed design or implementation of software in the system from contributing to the process, for example, by contributing test cases that rely on knowledge of the system’s internal implementation.*

#### [Sample]

The SQT shall be accomplished by the *System Test Organization*. This is done to ensure that the product accepted by the government meets all system requirements. The *Program Manager* may approve on a case-by-case basis the use of the developer‟s test staff as SQT testers. However, the conduct of these tests shall remain the responsibility of the *System Test Organization*.

#### [End Sample]

**5.11.2 Testing on the Target Computer System**

##### Guidance

*The developer’s system qualification testing shall include testing on the target computer system or an alternative system approved by the acquirer.*

#### [Sample]

The target computer system shall be used for all SQT testing. In the event that commercial equivalents are the only system available, *System Engineering* shall certify that the commercial equivalent system has the same functional characteristics as the Target Computer System. If the systems are not equivalent, then the follow-on tests will include a test sample of those procedures that could not be run on the Target Computer System.

#### [End Sample]

**5.11.3 Preparing for System Qualification Testing**

##### Guidance

*The developer shall participate in developing and recording the test preparations, test cases, and test procedures to be used for system qualification testing and the traceability between the test cases and the system requirements. For software systems, the results shall include all applicable items in the STD. The developer shall participate in preparing the test data needed to carry out the test cases and in providing the acquirer advance notice of the time and location of SQT.*

#### [Sample]

SQT will be conducted at the *System Test Organization* facilities. Following SQT, the program then proceeds to a series of Demonstration Tests (DTs). An Operational Evaluation (OPEVAL) may or may not be conducted based upon the changes to the System Baseline. Since tests following SQT are performed by external agencies, the processes for these tests have not been discussed.

#### [End Sample]

**5.11.4 Dry Run of System Qualification Testing**

##### Guidance

*If system qualification testing is to be witnessed by the acquirer, the developer shall participate in dry running the system test cases and procedures to ensure that they are complete and accurate and that the system is ready for witnessed testing. The developer shall record the software-related results of this activity in appropriate SDFs and shall participate in updating the system test cases and procedures as appropriate.*

#### [Sample]

The system test procedures will be reviewed and approved prior to conduct of the SQT. This will allow the developers and testers an early view into problems and issues that might not otherwise be revealed until system testing.

#### [End Sample]

**5.11.5 Performing System Qualification Testing**

##### Guidance

*The developer shall participate in system qualification testing. This participation shall be in accordance with the system test cases and procedures.*

#### [Sample]

The SQT is intended to verify program performance in accordance with the SSS for the *XY Project* and those requirements specifications referenced from the SRS. The test will include all functional areas and interfaces to verify the functionality of a totally integrated program. Program reliability will be evaluated during independent functional and simultaneous operations, and in light and dense tactical environments. All functions and subsystem interfaces will be independently tested in a systematic manner. Approved test procedures will be developed to allow for repeatability of tests and to facilitate performance analysis. System performance will be visually analyzed, augmented by automated data collection, and results will be recorded by test personnel.

SQT components and objectives are listed below:

1. Functional Tests - Functional Tests comprise two parts: Functional Operability Testing (FOT) and Functional Stress Testing (FST). FOT and FST test the functional requirements and the functional stress requirements of the SRS. FOT and FST are combined within a single set of procedures.
2. Interface Validation Tests (IVT) - IVTs comprise three parts: Interface Message Tests (IMT), Interface Recovery Tests (IRT) and Interface Stress Tests (IST). These three components test all interface messages, software recovery from interface protocol errors, and software response to interface stress, respectively. All IVTs are run with simulators, and to the degree feasible, will be conducted prior to SQT.
3. Regression Tests - Regression tests are run to verify that program changes implemented following the beginning of SQT testing have not introduced program regression. System tests including a casualty test, is included in the Regression Test set.
4. Single Unit Tests - Single Unit Tests are performed for each of the *XY Project* functional areas to validate the program operation individually in a one-on-one link.
5. Multiple Unit Tests - Multiple Unit Tests are performed simultaneously for all of the *XY Project* functional areas to validate the program operation in a multi-unit environment.
6. Stress and Endurance Test - The Stress and Endurance Test is designed to satisfy the stress and endurance requirements for critical computer programs. Three periods of maximum stress are distributed throughout a 25 hour period. The program must operate continuously for 25 hours without resulting in a Priority 1 or 2 P/CRs to pass this test.
7. P/CR Correction/Closure Tests - These test are executed to verify fixes to problems and to concur with the decision to close.

Specific SQT requirements and processes will be specified in the STP.

#### [End Sample]

**5.11.6 Revision and Retesting**

##### Guidance

*The developer shall make necessary revisions to the software, provide the acquirer advance notice of retesting, participate in all necessary retesting, and update the SDFs and other software products as needed, based on the results of system qualification testing.*

#### [Sample]

The Regression Test (RT), a set of high level tests, will perform a representative sampling of critical *XY Project* functions. It will run against a newly delivered operational program during SQT to examine the possibility of regression between the new and previous program versions. The RT is intended to serve as "system checkout" and should retain a measure of simplicity to ensure that results may be compared from one run to the next. Requirements for this test will be derived from mission-critical functions and casualty requirements identified in the SSS. Specific mission-critical functions are chosen which ensure that failure among them does compromise the overall effectiveness of the *XY Project*. Testing will be done in a laboratory environment. **[End Sample]**

**5.11.7 Analyzing and Recording System Qualification Test Results**

##### Guidance

*The developer shall participate in analyzing and recording the results of system qualification testing. For software systems, the result shall include all applicable items in the STR.*

#### [Sample]

Test procedures shall be prepared for each event to be tested in SQT and shall contain clear identification to link it to its particular level of test, as well as to define test objectives. These test procedures shall contain the expected results, a pass/fail notation, and a summary, if applicable. Evaluations of test data shall provide the basis for a pass/fail determination leading to eventual acceptance or non-acceptance of the program. Problems in either software or design documentation or user manuals shall be documented as a P/CR.

A P/CR is a report describing an existing problem in a computer program or its support documentation. Some P/CRs may, in fact, report a design enhancement rather than a design problem, in which case that PR will eventually be closed-out by submission of an ECP. The P/CRs database contains the following kinds of data: P/CR name, P/CR number, P/CR description, a category specification, a severity specification, and a process state status code. In addition, the P/CR database incorporates a routing system to direct the handling of each P/CR from station to station such as entry, analysis, approval, design, code, test, acceptance, and closure. The P/CR database program retains comments generated by each station as it advances through the route. The P/CR database program generates a variety of reports by selected category in specified order. **[End Sample]**

#### 5.12 PREPARING FOR SOFTWARE USE

##### Guidance

*This paragraph shall describe the approach to be followed when preparing for software use. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.* ***Note****: If software is developed in multiple builds, the developer’s planning should identify what software, if any, is to be fielded to users in each build and the extent of fielding (for example, full fielding or fielding to selected evaluators only). Preparing for software use in each build should be interpreted to include those activities necessary to carry out the fielding plans for that build.*

#### [Sample]

The *Delivery Team* is responsible for correctly packaging the software consistent with the Build Plan and according to the *XY Project* documented processes.

#### [End Sample]

**5.12.1 Preparing the Executable Software**

##### Guidance

*The developer shall prepare the executable software for each user site, including any batch files, command files, data files, or other software files needed to install and operate the software on its target computer(s). The result shall include all applicable items in the executable software section of the* Build Plan*.*

#### [Sample]

The executable software will be prepared for each user site, including any batch files, command files, data files, or other software files needed to install and operate the software on its target computer(s). The result is included in the executable deliverable portion of the Program Package (PP) as documented in the Build Plan.

#### [End Sample]

**5.12.2 Preparing Version Descriptions for User Sites**

##### Guidance

*The developer shall identify and record the exact version of software prepared for each user site. The information shall include all applicable items in the SVD.*

**[Sample]**

The Software Version Description (SVD) identifies and describes a version of a CSCI component or interim change (i.e., changes that occur between CSCI versions) to the previously released version. The SVD records data pertinent to the status and usage of a CSCI version or interim change. It is used to release CSCI versions or interim changes to the customer and will be included in the Program Package (PP).

#### [End Sample]

**5.12.3 Preparing User Manuals**

##### Guidance

*The developer shall prepare user manuals in accordance with the following requirements.* ***Note****: Few, if any, systems will need all of the manuals in this section. The intent is for the acquirer, with input from the developer, to determine which manuals are appropriate for a given system and to require the development of only those manuals. The manuals in this section are normally developed in parallel with software development, ready for use in CSCI testing.*

#### [Sample]

User Manuals will be prepared by the *Software Development Group* and validated by the *Software Test and Evaluation Group.* The *Software Librarian* will baseline the User Manuals, and provide copies as part of the deliverable Program Package (PP).

#### [End Sample]

**5.12.4 Installation at User Sites**

##### Guidance

*The developer shall:*

1. *Install and check out the executable software at the user sites specified in the contract*
2. *Provide training to users as specified in the contract*
3. *Provide other assistance to user sites as specified in the contract.*

#### [Sample]

Installation and integration schedules must be developed and site surveys completed prior to development of the individual Program Packages (PP). The installation can begin at the completion of system development. All hardware and software components must be assembled and tested in a lab environment prior to being shipped to the user site.

The *XY Project* software and hardware systems will then be shipped to the user site and installed and checked out by the designated *Delivery Team*. Each *Delivery Team* willidentify needed training and prepare training materials. Training should be provided to users at the time of installation. Other assistance, such as user consultation, must be readily available after installation of each revision. Planning for the delivery of each build is contained in the Build Plan for that delivery.

#### [End Sample] 5.13 PREPARING FOR SOFTWARE TRANSITION

##### Guidance

*This paragraph shall describe the approach to be followed for preparing for software transition. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.*

***Note****: If software is developed in multiple builds, the developer’s planning should identify what software, if any, is to be transitioned to the support agency in each build. Preparing for software transition in each build should be interpreted to include those activities necessary to carry out the transition plans for that build.*

#### [Sample]

There are no current plans to transition the system software to another agency or contractor for support. **[End Sample]**

#### 5.14 SOFTWARE CONFIGURATION MANAGEMENT

##### Guidance

*This paragraph shall describe the approach to be followed for software configuration management. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.*

***Note****: If a system or CSCI is developed in multiple builds, the software products of each build may be refinements of, or additions to, software products of previous builds . Software configuration management in each build should be understood to take place in the context of the software products and controls in place at the start of the build.*

*Also refer to the SSC San Diego Software Configuration Management (SCM) Process and the Generic Software Configuration Management Plan (SCMP) Template. Available from the SSC San Diego PAL.*

#### [Sample]

Software Configuration Management will be performed under the direction of the *SCM Manager* according to the processes and procedures defined in the *XY Project*  Software Configuration Management Plan (SCMP) reference (f).  **[End Sample]**

#### 5.15 SOFTWARE PRODUCT EVALUATION

##### Guidance

*This paragraph shall describe the approach to be followed for software product evaluation. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.* ***Note****: If a system or CSCI is developed in multiple builds, the software products of each build should be evaluated in the context of the objectives established for that build. A software product that meets those objectives can be considered satisfactory even though it is missing information designated for development in later builds.*

#### [Sample]

The processes defined in the Software Quality Assurance Plan (SQAP), reference (g) address software product evaluations.  **[End Sample]**

#### 5.16 SOFTWARE QUALITY ASSURANCE

##### Guidance

*This paragraph describes the approach to be followed for software quality assurance. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.* ***Note****: If a system or CSCI is developed in multiple builds, the activities and software products of each build should be evaluated in the context of the objectives established for that build. An activity or software product that meets those objectives can be considered satisfactory even though it is missing aspects designated for later builds. Planning for software quality assurance is included in software development planning (see 5.1.1).*

*Also refer to the SSC San Diego Software Quality Assurance (SQA) Process and the Software Quality Assurance Plan (SQAP) Template. Available from the SSC San Diego PAL.*

#### [Sample]

The Software Quality Assurance Plan (SQAP), reference (g), is the guiding document for the conduct of SQA within the *XY Project.*  **[End Sample]**

#### 5.17 CORRECTIVE ACTION

##### Guidance

*This paragraph shall describe the approach to be followed for corrective action. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.*

#### [Sample]

The *XY Project* will use the processes for databasing, tracking, and directing correction of P/CRs as defined the SCMP, reference (f). **[End Sample]**

#### 5.18 JOINT TECHNICAL AND MANAGEMENT REVIEWS

##### Guidance

*This paragraph shall describe the approach to be followed for joint technical and management reviews. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.*

***Note:*** *Also refer to the SSC San Diego “Keys to a Successful Meeting/Review” brief and the Software Management for Executives Guidebook. Available from the SSC San Diego PAL.*

#### [Sample]

The purpose of technical and management reviews is to provide management with tracking and oversight of the progress of software development undertaken by the *XY 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.

##### 5.18.1 Joint Technical Reviews

The developer shall plan and participate in joint technical reviews at locations and dates proposed by the developer and approved by the acquirer. These reviews shall be attended by persons with technical knowledge of the software products to be reviewed. The reviews shall focus on in-process and final software products, rather than materials generated especially for the review. The reviews shall have the following objectives:

1. Review evolving software products, review and demonstrate proposed technical solutions; provide insight and obtain feedback on the technical effort; and surface and resolve technical issues.
2. Review project status and surface near and long-term risks regarding technical, cost, and schedule issues.
3. Arrive at agreed-upon mitigation strategies for identified risks, within the authority of those present.
4. Identify risks and issues to be raised at joint management reviews.
5. Ensure ongoing communication between acquirer and developer technical personnel.

##### 5.18.2 Joint Management Reviews

The *Software Project Manager* shall plan and participate in joint management reviews at locations and dates approved by the *Program Manager*. These reviews shall be attended by persons with authority to make cost and schedule decisions. The reviews will be scheduled in the MS Project Plan for the *XY Project.* Keep management informed about project status, directions being taken, technical agreements reached, and overall status of evolving software products. The objectives of management reviews are listed below:

1. Resolve issues that could not be resolved at joint technical reviews.
2. Arrive at agreed-upon mitigation strategies for near and long-term risks that could not be resolved at joint technical reviews.
3. Identify and resolve management-level issues and risks not raised at joint technical reviews.
4. Obtain commitments and acquirer approvals needed for timely accomplishment of the project.

Given below is a set of candidate joint management reviews that might be held during a software development project. There is no intent to require these reviews or to preclude alternatives or combinations of these reviews.

1. Software plan reviews. These reviews are held to resolve open issues regarding one or more of the items listed below: 1) Software Development Plan (SDP)

2) Software Test Plan (STP)

1. Operational concept reviews. These reviews are held to resolve open issues regarding the operational concept for a software system.
2. System/subsystem requirements reviews. These reviews are held to resolve open issues regarding the specified requirements for a software system or subsystem.
3. System/subsystem design reviews. These reviews are held to resolve open issues regarding one or more of the items listed below:
   1. The system or subsystem-wide design decisions
   2. The architectural design of a software system or subsystem.
4. Software requirements reviews. These reviews are held to resolve open issues regarding the specified requirements for a CSCI.
5. Software design reviews. These reviews are held to resolve open issues regarding one or more of the items listed below: 1) The CSCI-wide design decisions
   1. The architectural design of a CSCI
   2. The detailed design of a CSCI or portion thereof (such as a database).
6. Test readiness reviews. These reviews are held to resolve open issues regarding one or more of the items listed below:
   1. The status of the software test environment.
   2. The test cases and procedures to be used for CSCI qualification testing or system qualification testing.
   3. The status of the software to be tested.
7. Test results reviews. These reviews are held to resolve open issues regarding the results of CSCI qualification testing or system qualification testing.
8. Software usability reviews. These reviews are held to resolve open issues regarding one or more of the items listed below:
   1. The readiness of the software for installation at user sites
   2. The user and operator manuals
   3. The software version descriptions
   4. The status of installation preparations and activities.
9. Software supportability reviews. These reviews are held to resolve open issues regarding one or more of the items listed below:
   1. The readiness of the software for transition to the support agency.
   2. The software product specifications.
   3. The software support manuals.
   4. The software version descriptions.
   5. The status of transition preparations and activities, including transition of the software development environment, if applicable.
10. Critical requirement reviews. These reviews are held to resolve open issues regarding the handling of critical requirements, such as those for safety, security, and privacy.

**[End Sample]**

#### 5.19 OTHER SOFTWARE DEVELOPMENT ACTIVITIES

##### Guidance

*This paragraph shall describe the approach to be followed for other software development activities. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.*

#### [Sample]

These paragraphs describe the technical and management approach to coordinating and providing oversight of software development activities undertaken by the *XY Project*. Timely technical and management oversight at the appropriate level is necessary to accomplish the activities listed below:

1. Track progress against plans
2. Identify and resolve problems
3. Identify and verify appropriate expenditure of assigned resources
4. Depict activities and relationships associated with planning and conducting integration
5. Consider other related development actions and schedules that could impact *XY Project*. These paragraphs address those actions to be taken to maintain a broad management picture of *XY Project* software development.

#### [End Sample]

**5.19.1 Risk Management**

##### Guidance

*The developer shall perform risk management throughout the software development process. The developer shall identify, analyze, and prioritize the areas of the software development project that involve potential technical, cost, or schedule risks; develop strategies for managing those risks; record the risks and strategies in the software development plan; and implement the strategies in accordance with the plan.*

*Also refer to the Risk Management Process available from the SSC San Diego PAL.*

#### [Sample]

Risk analysis and risk management deal with concerns of project personnel regarding the capability of the designed system to achieve program objectives of technical performance, schedule and cost. Risk analysis is an iterative process. It attempts to identify what could go wrong, plan courses of action to mitigate the occurrence, and plan contingencies in the event problems arise. Risk analysis identifies potential problem areas, quantifies risks associated with these problems, assesses the effect of these risks, and generates alternative actions/mitigations to reduce risks and recover from their occurrence.

The major areas of risk for the software development will be managed following the process identified in the Risk Management Process, reference (i)

#### [End Sample]

**5.19.2 Software Management Indicators**

##### Guidance

*The developer shall use software management indicators to aid in managing the software development process and communicating its status to the acquirer. The developer shall identify and define a set of software management indicators, including the data to be collected, the methods to be used to interpret and apply the data, and the planned reporting mechanism. The developer shall record this information in the software development plan and shall collect, interpret, apply, and report on those indicators as described in the plan*

*Also refer to the Software Measurement Plan Template(SMP) available from the SSC San Diego PAL.*

#### [Sample]

The *XY Project* management team will use software metrics for measurement of cost and schedule variance, productivity and quality. The *XY Project* measurement program is defined in the Software Measurement Plan (SMP), reference (h).

#### [End Sample]

**5.19.3 Security and Privacy**

##### Guidance

*The developer shall meet the security and privacy requirements specified in the contract. These requirements may affect the software development effort, the resulting software products, or both.*

#### [Sample]

The *XY Project* software development is anticipated to contain a limited amount of classified data. Determination of classification will be in accordance with OPNAVINST S5513.3B, Security Classification Guide. Development of classified software will occur only in properly cleared spaces utilizing Automated Information Systems (AIS) approved computers. Transfer of classified programs between facilities shall utilize approved methods. The highest security level that should be encountered during software development is SECRET. Each software engineer or other software support person who requires access to classified material will be cleared to the appropriate level and supplied with an approved container in which classified material can be stored when not in use.

#### [End Sample]

**5.19.4 Subcontractor Management**

##### Guidance

*If subcontractors are used, the developer shall include in subcontracts all contractual requirements necessary to ensure that software products are developed in accordance with prime contract requirements.*

*Also refer to the SSC San Diego Contractor Acquisition and Monitoring (CAPM) Process. Available from the SSC San Diego PAL.*

#### [Sample]

The *Software Project Manager* will use the SEPO-developed CAPM Process for Software Contracts as a guide for contractor management. The *Software Project Manager* will augment the CAPM by initiating additional management actions described below.

Contractor personnel will participate in all development team activities including weekly status meetings, and informal and/or formal reviews as directed by *Software Project Manager*. Contractors will supply weekly measurements and monthly status reports on progress against their statement of work as called for in their individual Contract data requirements list (CDRLs). When deemed necessary, the *Software Project Manager* will establish working groups made up of government and contractor personnel to address major project areas such as architecture, integration, and display.

The *Software Project Manager* will also establish an overall *XY Project* software development, integration, and testing schedule. Included in this schedule will be all major project milestones that will enable a contractor to plan for delivery of their individual sections.

To manage and track *XY Project* status/progress, the *Software Project Manager* will direct the use of the software management tool, Microsoft Project.

#### [End Sample]

**5.19.5 Interface With Software Independent Verification and Validation (IV&V) Agents**

##### Guidance

*The developer shall interface with the software Independent Verification and Validation (IV&V) agent(s) as specified in a tasking statement.*

#### [Sample]

Coordination will be provided by having the *IV&V Group* participate/monitor activities in each phase of the development process including attending both formal and informal program reviews. In addition, the *IV&V Group* will be provided on-line access to source code, test scripts and documentation, and will participate in the SCCB.

During all phases of software development, the *IV&V Group* will be able to initiate change requests to the evolving baseline or to add their review comments into the active action items to be resolved as a part of each end-of-phase review. These comments will be treated like any comment or change request initiated by a member of the project organization. Review comments are expected to be supplied in written form, or electronically in *Microsoft Word*, for subsequent tracking by the *Software Project Manager* in the same manner review actions items are handled. All comments will be adjudicated and responses provided back to the *IV&V Manager* describing the intended action to be taken. Change requests to change something in the developmental baseline, after the phase has been completed, will be entered into the configuration control system as described in SCMP and addressed like any other change request. The *IV&V Group* will be able to monitor the status of change requests they submitted directly from the SCM system.

#### [End Sample]

**5.19.6 Coordination With Associate Developers**

##### Guidance

*The developer shall coordinate with associate developers, working groups, and interface groups as specified in the contract.*

#### [Sample]

The *Software Project Manager* and key staff members will meet weekly to clarify issues, obtain project status on deliverables, and elicit program comments. Updates will be provided and technical issues that affect the overall development effort will be discussed. In addition, the *Software Project Manager* will form sub-groups or committees to resolve complex technical and administrative issues.

To provide a documentation trail, E-mail will be used as the primary means of distributing data amongst participants.

#### [End Sample]

**5.19.7 Improvement of Project Processes**

##### Guidance

*The developer shall periodically assess the processes used on the project to determine their suitability and effectiveness. Based on these assessments, the developer shall identify any necessary and beneficial improvements to the process, shall identify these improvements to the acquirer in the form of proposed updates to the software development plan and, if approved, shall implement the improvements on the project.*

#### [Sample]

As the *XY Project* program matures, the SDP should undergo revision reflecting improvements to the processes. Quality and process improvement comes by analyzing and measuring the process in a structured, controlled manner and then changing the process. These improvements should be the product of lessons learned data, findings of audits, and process measurements collected and fed back into the processes.

The software development process described in this SDP generally follows MIL-STD-498. Where a process described in the SDP is deemed to be inadequate, cumbersome or nonresponsive, any *XY Project* team member may submit a recommendation to change or improve that process.

When compliance with specific software policies, processes, forms, or templates would impose severe impacts on project cost, schedule, resources, or customer relations, the implementing developer may request a waiver/deviation of specific provisions. They must propose reasonable and detailed alternatives during project planning. Requests for waivers/deviations must cite specific provisions of policies, processes, forms, or templates for which waivers/deviations are being requested, the impact of compliance, and alternatives to be implemented by the *XY project*. The Department *Software Engineering Process Group (SEPG),* interfacing with SEPO,shall be responsible for evaluating and acting upon suggestions for improvements in policies, processes, procedures, waivers/deviations, and standard forms and templates referenced in this SDP. In addition, process metrics collected during program development and lessons learned from this and other projects will be used to update the *XY Project* processes. The *Software Project Manager* is authorized to approve *SEPG* recommended changes to the *XY Project* processes described in the SDP.

The *Software Project Manager* is responsible for post mortem analysis of each *XY Project* build to identify candidate processes for improvement and for the submission of Project Data Forms (PDF) to the Organization Software Process Database (OSPD) maintained by SEPO.

**[End Sample]**

This page intentionally left blank.

**SECTION 6. SCHEDULES AND ACTIVITY NETWORK**

|  |
| --- |
| ***Guidance***  *This section shall present the items listed below:*   1. *Schedule(s) identifying the activities in each build and showing initiation of each activity,*   *availability of draft and final deliverables and other milestones, and completion of each activity. This can be created by using project management tools such as Microsoft Project.*   1. *An activity network, depicting sequential relationships and dependencies among activities and identifying those activities that impose the greatest time restrictions on the project. Activity networks* |

#### [Sample]

The *Program Office* has established a Master Build Schedule as defined in Figure 6-1. The supporting WBS, cost, schedule, and staffing requirements for the Master Build Schedule is contained in Appendix A, the Microsoft Project Plan for the *XY Project.*

**Requirements**

**Design**

**Develop**

**Integrate**

**Test**

***Version Release 1.0 Development Cycle***

***Version Release 2.0 Development Cycle***

***Version Release 3.0 Development Cycle***

**Q**

**00**

**3**

**4**

**00**

**Q**

**1**

**01**

**Q**

**Q**

**2**

**01**

**01**

**3**

**Q**

**4**

**Q**

**01**

**Q**

**1**

**02**

**2**

**02**

**Q**

**3**

**02**

**Q**

**4**

**02**

**Q**

**Q**

**03**

**1**

**Q**

**03**

**2**

**Q**

**03**

**3**

**4**

**03**

**Q**

**1**

**04**

**Q**

**Q**

**2**

**04**

**3**

**Q**

**04**

**04**

**Q**

**4**

**Q**

**1**

**05**

**Q**

**2**

**05**

Figure 6-1. Master Build Schedule

#### [End Sample

6-1

The page intentionally left blank.

6-2

**SECTION 7. PROJECT ORGANIZATION AND RESOURCES**

#### *Guidance*

*This section shall be divided into the following sections to describe the project organization and resources to be applied in each build.*

#### [Sample]

The *XY Project* and related organizations are described in Section 7.1. Staff requirements are tabularized in Section 7.2. **[End Sample]**

#### 7.1 PROJECT ORGANIZATION

##### Guidance

*This paragraph shall describe the organizational structure to be used on the project, including the organizations involved, their relationships to one another, and the authority and responsibility of each organization for carrying out required activities. Below is an example organization chart for XY Project.*

#### [Sample]

The *XY Project* Chain of Command and internal organization within SSC San Diego is depicted in Figure 7-1. The roles and responsibilities are contained in Table 7-1.

OPNAV

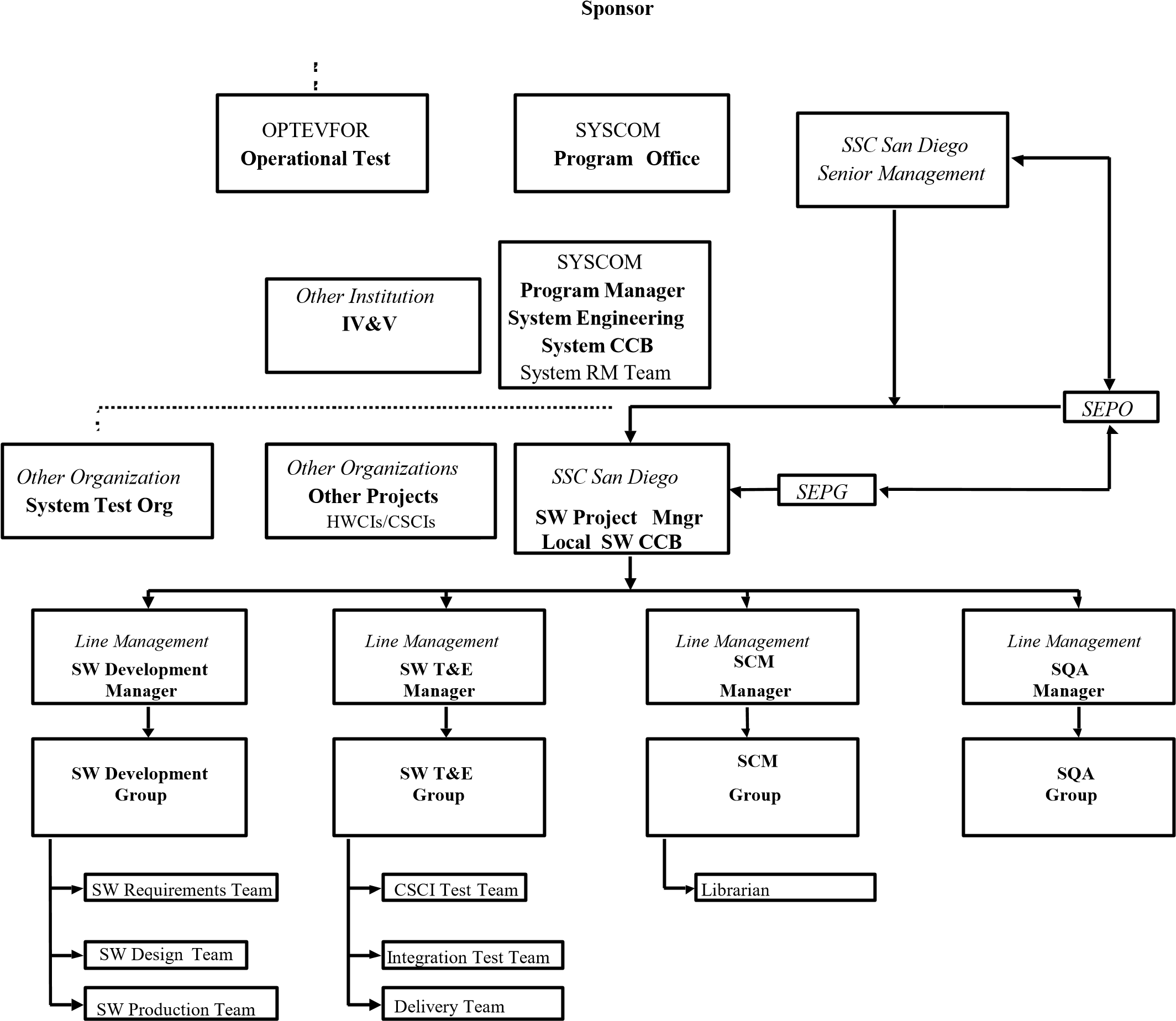


Figure 7-1. *XY Project* Organizational Structure

TABLE 7-1. ROLES AND RESPONSIBILITIES

|  |  |
| --- | --- |
| POSITION | **ROLES/RESPONSIBILITIES** |
| Sponsor | OPNAV level organization providing Mission Element Need Statement (MENS) and DoN budget. |
| Program Office | SYSCOM level organization tasked by OPNAV to perform system acquisition to meet MENS. |
| Program Manager | SYSCOM level manager assigned responsibility to direct system acquisition (hardware/software). |
| Configuration Control Agent | SYSCOM level manager responsible for overall system configuration. |
| System CCB | SYSCOM level configuration control board controlling the system configuration (hardware/software) a.k.a. SCCB. |
| Project | A major component of a Program, can be either a hardware or software configuration item. |
| System Test Organization | Organizational entity charged with system level acceptance testing. May, or may not, be at a separate geographic location but definitely has a separate chain of command. |
| System Test Manager | Office within the System Test Organization charged with responsibility for system testing. |
| System Test Group | Collective name of system test staff. |
| System Test Engineer | An individual in the System Test Group. |
| Project Manager | In this example, a SYSCOM level manager responsible to the Program Manager for a project, both hardware and software configuration items. However, it should be noted that this responsibility may also be placed within an organization such as SSC San Diego. |
| Systems Engineering Manager | Manager responsible to the Program Manager for the overall system engineering. A candidate for the role of System CCB Chairperson. |
| Systems Engineering Group | System-level engineering staff, responsible for the system configuration (hardware/software) and the production and maintenance of the System Specification (i.e., SSS). |
| Systems Engineer | Member of the Systems Engineering Group. |
| Systems Engineering Requirements Team | Member of the Systems Engineering Group charged with specific responsibilities for managing the system requirement‟s database. |
| Software  Development/Maintenance Organization | Organizational entity, such as SSC San Diego, charged with system software development/maintenance. |

|  |  |
| --- | --- |
| POSITION | **ROLES/RESPONSIBILITIES** |
| Senior Management | A senior manager (i.e., Executive Director, Department Head, or possibly a Division Head) providing infrastructure resources (i.e., facilities, admin., contracting, supply) to support an internal entity as a Software Development/Maintenance Organization. |
| Systems Engineering Process Office | SSC San Diego‟s senior level entity providing process improvement direction. |
| Software Engineering Process Group | Professional staff providing Software Process Improvement (SPI) leadership to an SSC San Diego internal organization, such as a Division. |
| Software Project Manager | An SSC San Diego „manager‟ (i.e., a Division Head, or possibly Branch Head) responsible to the Project Manager for assigned software CSCIs. A candidate for LCCB chairperson. |
| Software Development Manager | An organizational line manager (i.e., a branch head or group leader), responsible for software development to the Software Project Manager. |
| Software Development Group | Collective name of the software engineering staff responsible for CSCI development. |
| Software Design Team | Staff responsible for software architecture and design. |
| Software Production Team | Staff responsible for code, and/or reuse component selection, and unit test. |
| Software Requirements Team | Staff responsible for the software requirement‟s database. |
| Software Test and Evaluation Manager | An organizational line manager (i.e., a branch head or group leader) responsible for CSCI and Integration Test and Evaluation to the Software Project Manager. |
| Software Test and Evaluation Group | Collective name of the software engineering staff responsible for CSCI integration and testing. |
| CSCI Test Team | Software engineering staff responsible for Internal CSCI Testing. |
| Integration Test Team | Software engineering staff responsible for Internal CSCI/HWCI Testing. |
| Delivery Team | Software engineering staff responsible for on site deliveries, training, and testing. |
| SCM Manager | An organizational line manager (i.e., a branch head or group leader) responsible for Software Configuration Management (SCM) to the Software Project Manager. |
| SCM Group | Collective name of the software engineering staff responsible for SCM functions. |
| POSITION | **ROLES/RESPONSIBILITIES** |
| Documentation Team | Staff responsible for production and maintenance of project documentation, such as software specifications (i.e., SRS) and users manuals. |
| Librarian | Member of engineering staff who is keeper of document and program baselines (check in/out) |
| SQA Manager | An organizational line manager (i.e., a branch head or group leader), responsible for Software Quality Assurance (SQA) functions who reports to the Software Project Manager. |
| SQA Group | Collective name of the software engineering staff responsible for SQA functions. |
| IV&V Organization | Program Level organization charged with the responsibility of providing resources and management for IV&V functions. Most often organized such that they report to the Program Manager, typically an organizational entity with a chain of command separate from SSC San Diego‟s. |
| IV&V Manager | Office within the IV&V Organization charged with responsibility for IV&V functions. |
| IV&V Group | Collective name of IV&V staff. |

#### [End Sample]7.2 PROJECT RESOURCES

##### Guidance

*This paragraph shall describe the resources to be applied to the project. It shall include, as applicable, those listed below:*

1. *Personnel resources, including:* 
   1. *The estimated staff-loading for the project (number of personnel over time).*
   2. *The breakdown of the staff-loading numbers by responsibility (for example, management, software engineering, software testing, software configuration management, software product evaluation, software quality assurance).*
   3. *A breakdown of the skill levels, geographic locations, and security clearances of personnel performing each responsibility.*
2. *Overview of developer facilities to be used, including geographic locations in which the work will be performed, facilities to be used, and secure areas and other features of the facilities as applicable to the contracted effort.*
3. *Acquirer-furnished equipment, software, services, documentation, data, and facilities required for the contracted effort. A schedule detailing when these items will be needed shall also be included.*
4. *Other required resources, including a plan for obtaining the resources, dates needed, and availability of each resource item.*

*An example of a staffing plan is given in Table 7-2.*

#### [Sample]

Table 7-2 provides a breakdown of personnel requirements required by the *XY Project* to support development of the system software from Build 1 through Build 3 as illustrated in Figure 6-1. These resources have been allocated to the MS Project Plan contained in Appendix A.

TABLE 7-2. PERSONNEL REQUIREMENTS (PERSON YRS)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity Personnel | System Rqmts  Analysis/Desig n | Detailed Design | Code and Unit Test | System  Integration | Subtotals |
| Management | 1 | 2 | 2 | 1 | 6 |
| Systems Engineer | 5 | 3 | 2 | 1 | 11 |
| Hardware | 3 | 2 | 2 | 1 | 8 |
| S/W Engineer | 5 | 10 | 15 | 3 | 33 |
| CM | 2 | 2 | 4 | 2 | 10 |
| SQA | 1 | 1 | 2 | 1 | 5 |
| Testing | 1 | 2 | 1 | 5 | 9 |
| IV&V | 1 | 2 | 2 | 4 | 9 |
| Facilities | 1 | 1 | 2 | 2 | 6 |
| ILS | 1 | 1 | 2 | 2 | 6 |
| Subtotals | 21 | 26 | 34 | 22 | (Total) 103 |

**[End Sample]**

**SECTION 8. NOTES**

#### *Guidance*

*This section shall contain any general information that aids in understanding this document (e.g., background information, glossary, and rationale). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.*

#### 8.1 ACRONYMS

***Guidance***

*Alphabetically list all acronyms used in this document.*

#### [Sample]

|  |  |
| --- | --- |
| AIS | Automated Information System |
| AWG | Architecture Working Group |
| BCC | Black Controller CSCI |
| CAPM | Contractor Acquisition and Performance Monitoring |
| CASE | Computer-Aided Software Engineering |
| CC | Configuration Control |
| CDRL | Contract Data Requirements List |
| CI | Configuration Item |
| CM | Configuration Management |
| CM | Capability Maturity Model |
| COM | Computer Operation Manual |
| COTS | Commercial Off-the-Shelf |
| CPM | Computer Programming Manual |
| CR | Change Report |
| CSAR | Configuration Status Accounting Report |
| CSCI | Computer Software Configuration Item |
| DBDD | Database Design Description |
| DCR | Document Change Request |
| DDR | Detailed Design Review |
| DID | Data Item Description |
| DoD | Department of Defense |
| DoDI | Department of Defense Instruction |
| DoN | Department of Navy |
| ECP | Engineering Change Proposal |
| FOT | Functional Operability Testing |
| FQT | Formal Qualification Test |
| FSM | Firmware Support Manual |
| FST | Functional Stress Testing |
| GOTS | Government Off-the-Shelf |
| GUI | Graphical User Interface |
| HCI | Human Computer Interface |

|  |  |  |
| --- | --- | --- |
| HWCI | Hardware Configuration Item | |
| IDD | Interface Design Description | |
| IEEE | Institute of Electronics and Electrical Engineers | |
| ILS | Integrated Logistics Support | |
| IMT | Interface Message Tests | |
| IRS | Interface Requirements Specification | |
| IRT | Interface Recovery Tests | |
| IST | Interface Stress Tests | |
| IVT | Interface Validation Tests | |
| IV&V | Independent Verification and Validation | |
| KM | Cryptographic Module | |
| KPA | Key Process Area | |
| LCCB | Local Software Configuration Control Board | |
| LCM | Life Cycle Maintenance | |
| MENS | Mission Element Needs Statement | |
| MTP | Master Test Plan | |
| NDI | Non-Developmental Item | |
| OCD | Operational Concept Description | |
| OOD | Object-Oriented Design | |
| OPEVAL | Operational Evaluation | |
| PAL | Process Asset Library | |
| PAT | Program Acceptance Test | |
| PP | Program Package | |
| PPI | Pre Planned Product Improvement | |
| PR | Problem Report | |
| PSM | Practical Software Measurement | |
| PTR | Program Trouble Report | |
| QA | Quality Assurance | |
| RAM | Reuse Adaptation and Management | |
| RBC | Red/Black Controller | |
| RCC | Red Controller CSCI | |
| RM | Requirements Management | |
| R&R | Review and Response | |
| RT | Regression Test | |
| SCCB | System Configuration Control Board | |
| SCM | Software Configuration Management | |
| SCMP | Software Configuration Management Plan | |
| SCOM | Software Center Operator Manual | |
| SDD | Software Design Description | |
| SDF | Software Development File | |
| SDL | Software Development Library | |
| SDP | Software Development Plan | |
| SEE | Software Engineering Environment | |
| SEI | Software Engineering Institute | |
| SEN | | Software Engineering Notebook |
| SEPG | | Software Engineering Process Group |
| SEPO | | System Engineering Process Office |
| SIOM | | Software Input/Output Manual |
| SIP | | Software Installation Plan |
| SMP | | Software Measurement Plan |
| SOW | | Statement of Work |
| SPA | | Software Process Assets document |
| SPE | | Software Product Evaluation |
| SPI | | Software Process Improvement |
| SPIP | | Software Process Improvement Plan |
| SPM | | Software Project Manager |
| SPP | | Software Project Planning |
| SPS | | Software Product Specification |
| SPTO | | Software Project Tracking and Oversight |
| SQA | | Software Quality Assurance |
| SQAP | | Software Quality Assurance Plan |
| SQER | | Software Quality Evaluation Report |
| SQT | | System Qualification Test |
| SRS | | Software Requirements Specification |
| SSC | | SPAWAR Systems Center San Diego |
| SSDD | | System/Subsystem Design Description |
| SSS | | System/Subsystem Specification |
| STD | | Software Test Description |
| STE | | Software Test Environment |
| STP | | Software Test Plan |
| STR | | Software Test Report |
| STrP | | Software Transition Plan |
| SU | | Software Unit |
| SUM | | Software User Manual |
| SVD | | Software Version Description |
| SW | | Software |
| SYSCOM | | System Command |
| TECHEVAL Technical Evaluation  TD System level test description  TP System level test plan  TRR Test Readiness Review  WBS Work Breakdown Structure  **[End Sample]** | | |

**APPENDIX A. PROJECT PLAN**

#### *Guidance*

*Appendices may be used to provide information published separately for convenience in document maintenance (e.g., charts and classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendices may be bound as separate documents for ease in handling. Appendixes shall be lettered alphabetically (A, B, C…).*

**[Sample]**

*<XY Project* Microsoft Project Plan>

#### [End Sample]

A-1

*Project Name* SDP

*Document Identifier Date*

This page intentionally left blank.

A-2

**DOCUMENT CHANGE REQUEST (DCR)**

|  |  |
| --- | --- |
| Document Title: ***Project Name* Software Development Plan** | Tracking Number: |
| Name of Submitting Organization: | |
| Organization Contact: | Phone: |
| Mailing Address: | |
| DCR Description: | Date: |
| Change Location:  (use section #, figure #, table #, etc.) | |
| Proposed change: | |
| Rationale for Change: | |
| Note: For the <indicate appropriate authority> to take appropriate action on a change request, please provide a clear description of the recommended change along with supporting rationale.  Send to: Commanding Officer, Space and Naval Warfare Systems Center, Code [[2xx]], 53560 Hull Street, San  Diego, CA 92152-5001  Fax to: <indicate appropriate fax number>  Email to: <indicate appropriate email> Submit online: <indicate appropriate URL>  DCR Form 2/2005 | |