**Team Cursor**

**Encryption**

**Software Project Plan**

*[Authors Names]*

*Version 1*

*March 27, 2014*

|  |  |  |  |
| --- | --- | --- | --- |
| *Plan Version* | *Date* | *Authors* | *Comment* |
| 1.0 | 3/26/14 | Ricky Bifford | Starter version |

TABLE CONTENTS

[1 Introduction 1](#_Toc296529766)

[1.1 Identification 1](#_Toc296529767)

[1.2 Scope 1](#_Toc296529768)

[1.3 Document Overview 2](#_Toc296529769)

[1.4 Relationship to Other Plans 2](#_Toc296529770)

[2 Acronyms and Definitions 2](#_Toc296529771)

[2.1 Definitions 3](#_Toc296529772)

[2.2 Acronyms and Abbreviations 3](#_Toc296529773)

[3 References 3](#_Toc296529774)

[4 Overview 3](#_Toc296529775)

[4.1 Relationships 3](#_Toc296529776)

[4.2 Source Code 3](#_Toc296529777)

[4.3 Documentation 4](#_Toc296529778)

[4.4 Project Resources 4](#_Toc296529779)

[4.5 Project Constraints 5](#_Toc296529780)

[5 Software Process 5](#_Toc296529781)

[5.1 Software Development Process 5](#_Toc296529782)

[5.1.1 Life Cycle Model 5](#_Toc296529783)

[5.2 Software Engineering Activities 5](#_Toc296529784)

[5.2.1 Handling of Critical Requirements 5](#_Toc296529785)

[5.2.2 Recording Rationale 5](#_Toc296529786)

[5.2.3 Computer Hardware Resource Utilization 6](#_Toc296529787)

[5.2.4 Reusable Software 6](#_Toc296529788)

[5.2.5 Software Testing 6](#_Toc296529789)

[6 Schedule 7](#_Toc296529790)

[Appendix A Software Quality Assurance 8](#_Toc296529791)

[Appendix B Software Configuration Management 9](#_Toc296529792)

# Introduction

This document is designed to provide software identification, scope, primary objectives and test plan schedule to be adopted during the development of the Team Cursor Data Encryption Software. The goal of this document is to provide a framework that can be used by managers and developers to successfully test, plan, and implement the necessary software development processes in a timely and cost-effective manner.

The Team Cursor software development team will design and implement a data encryption software that is capable of encrypting a file for secure transmission. The schedule timeline for software planning is 3 weeks. Software development, testing and deployment for this project is scheduled for 5 weeks.

## Identification

This document applies to the software development effort in support of the development of TeamCursor\_DataEncryption Version 1.0. A detailed development timeline in support of this effort is provided in the supporting project Master Schedule.

## Scope

Team Cursor Data Encryption Software was developed for use by agents of S.H.I.E.L.D. to quickly and securely encode files for transmission across unsecure networks.

This overview shall not be construed as an official statement of product requirements. It will only provide a brief description of the system/software and will reference detailed product requirements outlined in the Team Cursor SRS.

Additionally, testing will be performed at several points in the life cycle as the product is constructed. Testing is a very dependent activity and will continue throughout the software development life cycle.

## Document Overview

The purpose of the Data Encryption Software Development Plan (SDP) is to guide Team Cursor project management during the development of Data Encryption Software version 1.0. Software Quality Assurance (SQA) procedures are captured in Appendix A of the SDP. Appendix B is used to document Software Configuration Management (SCM) activities where these activities deviate from Team Cursor SCM Plan. Project management and oversight activities (i.e., risk tracking, problem reporting) are documented in Appendix C of the SDP where these activities deviate from the Team Cursor Risk Management Plan. Requirements for the project will be captured in the Data Encryption Software Requirements Specification (SRS). This plan will be placed under configuration management controls according the Team Cursor Software Configuration Management Plan. Updates to this plan will be handled according to relevant SCM procedures and reviews as described in the Team Cursor Software Quality Assurance Plan.

## Relationship to Other Plans

There are several other S.H.I.E.L.D. documents which support the information contained within this plan. These documents include: S.H.I.E.L.D. Software Configuration Management Plan, S.H.I.E.L.D. Software Quality Assurance Plan, and [Organization Name] Metrics and Measurement Plan S.H.I.E.L.D. project specific documentation relating to this plan include: Team Cursor Data Encryption Software Requirements Specifications and Team Cursor Data Encryption Master Schedule. This plan has been developed in accordance with all S.H.I.E.L.D. software development processes, policies, and procedures.

# Acronyms and Definitions

## Definitions

|  |  |
| --- | --- |
| xxx | yyy |

## Acronyms and Abbreviations

|  |  |
| --- | --- |
| SRS  SQA | Software Requirements Specification  Software Quality Assurance |
| SDP  S.H.I.E.L.D. | Software Development Plan  Supreme Headquarters International Espionage Law Enforcement |

# References

*[This section should identify the specific references used within the SPP. Reference to all associated software project documentation, including the statement of work and any amendments should be included.]*

# Overview

*[This section of the SPP should list the items to be delivered to the customer, delivery dates, delivery locations, and quantities required to satisfy the terms of the contract. This list should not be construed as an official statement of product requirements. It will only provide an outline of the system/software requirements and will reference detailed product requirements outlined in the SRS, Appendix D. An example follows:*

*The project schedule in support of [Project Name] was initiated [Date] with a target completion date of [Date]. Incremental product deliveries may be requested, but none are identified at this time. The delivery requirements are to install [Project Name] on the current [Project Name] external website. [Project Name] will be delivered once deployment has occurred, and [Customer Name] acceptance of the product. [Customer Name] may request the installation of [Product Name] at one additional location. A [Project Name] users manual shall also be considered to be required as part of this delivery. Detailed schedule guidance is provided by [document name], located [document location]. Detailed requirements information is provided by [document name], located [document location].]*

## Relationships

*[This subsection should identify interface requirements and concurrent or critical development efforts which may directly or indirectly affect product development efforts.]*

## Source Code

*[This subsection should identify source code deliverables. An example follows:*

*There are no requirements for source code deliverables. If [customer name] requests the source code, it will be delivered on CD-ROM.]*

## Documentation

[*This subsection should itemize documentation deliverables and their required format. It should also contain, either directly or by reference, the documentation plan for the software project. This documentation plan may either be a separate document, stating how documentation is going to be developed and delivered, or may be included in this plan as references to existing standards with documentation deliverables and schedule detailed herein. An example is provided:*

*[Project Name] will have on-line help, a user’s manual will be created from this online help system. The user’s manual will be posted on the [Project Name] website and will be available for download by requesting customers. Please refer to the [Project Name] master schedule for a development timeline of associated users on line help and manual.]*

## Project Resources

*[This section should describe the project’s approach by describing the tasks (e.g., req. -> design -> implementation -> test) and efforts (update documentation, etc.) required to successfully complete the project. It should state the nature of each major project function or activity and identify the individuals who are responsible for those functions or activities.*

*This section should also describe the make-up of the team, project roles, and internal management structure of the project. Diagrams may be used to depict the lines of authority, responsibility, and communication within the project. Figure 1 is an example of an organizational chart.*

**

*Figure 1. Example Project Organization*

*The relationship and interfaces between the project team and all non-project organizations should also be defined by the SPP. Minimum interfaces include those with the customer, elements (management, SQA, SCM, etc.), and other support teams. Relationships with other contracting agencies or organizations outside the scope of the project that will impact the project require special attention within this section.]*

## Project Constraints

*[This section should describe the limits of the project including interfaces with other projects, the application of the program’s SCM and SQA (including any divergence from those plans), and the relationship with the project’s customer. This section should also describe the administrative and managerial boundaries between the project and each of the following entities: parent organization, customer organization, subcontracted organizations, or any organizational entities that interact with the project.*

*This section should capture the anticipated volume of the project through quantifiable measurements such as lines of code, function points, number of units modified, number of pages of documentation generated or changed, etc. It may be useful, if the project is well defined in advance, to breakdown project activities and perform size estimates on each individual activity. This information can then be tied to the project schedule as defined in Section 5.]*

# Software Process

## Software Development Process

*[This subsection of the SPP should define the relationships among major project functions and activities by specifying the timing of major milestones, baselines, reviews, work products, project deliverables, and signature approvals. The process model may be described using a combination of graphical and textual notations which include project initiation and project termination activities. The Waterfall Model, Fountain, and ETVX are examples of process models.]*

### Life Cycle Model

*[This subsection of the SPP should identify the life-cycle models used for the software development process. It should describe the project organizational structure, organizational boundaries and interfaces, and individual responsibilities for the various software development elements.]*

## Software Engineering Activities

### Handling of Critical Requirements

*[This subsection will identify the overall software engineering methodologies to be used during requirements elicitation and system design. An example is provided:*

*Please refer to the [Organization Name] Software Requirements Management Plan for information regarding the elicitation and management of customer requirements.]*

### Recording Rationale

*[This subsection should define the methodologies used to capture design and implementation decisions. An example follows:*

*All initial design and implementation decisions will be recorded and posted in the [Project Name] portal project workspace. All formal design documentation will follow the format as described in the [Organization Name] Design Documentation Template.]*

### Computer Hardware Resource Utilization

*[This subsection should describe the approach to be followed for allocating computer hardware resources and monitoring their utilization. Example text is provided:*

*All computer hardware resource utilization is identified in the [Project Name] Spend Plan. Resource utilization is billed hourly as an associated site contract charge. Twenty thousand dollars has been allocated to support the procurement of a development environment. Charges in support of project equipment purchases will be recorded by the Project Manager, [Name], and reported to senior management weekly.]*

### Reusable Software

*[This subsection should describe the approach for identifying, evaluating, and reporting opportunities to develop reusable software products. See the following example:*

*It is the responsibility of each developer to identify reusable code during the development process. Any item identified will be placed in the [Organization Name] Reuse library which is hosted on the [location]. It is the responsibility of each programmer to determine whether items in the reuse library may be employed during [Project Name] development.]*

### Software Testing

*[This subsection should be further divided to describe the approach for software implementation and unit testing. See below:*

*This project will use existing templates for software test plans and software unit testing when planning for testing. These separate documents will be hosted [location].]*

# Schedule

*[This section of the SPP should be used to capture the project’s schedule including milestones and critical paths. Options include Gantt Charts (Milestones Etc.TM, Microsoft ProjectTM), Pert Charts, or simple time lines. Figure 2 is an example schedule created with Milestones, Etc.TM for a short (two month) project.*

**

*Figure 2. Sample project schedule*

*]*

Appendix A Software Quality Assurance

*[This appendix should be divided into the following sections to describe the approach for Software Quality Assurance (SQA):*

*Software quality assurance evaluations*

*Software quality assurance records*

*Independence in software quality assurance*

*Corrective action*

*This appendix should reference the organizational, or program, level quality assurance policies, plans and procedures and should detail any deviations from the organizational standard. The following example text is provided:*

*Refer to the [Organization Name] Software Quality Assurance Plan for all applicable processes and procedures.]*

Appendix B Software Configuration Management

*[This appendix should be divided into the following sections to describe the approach for Software Configuration Management (SCM):*

*Configuration identification*

*Configuration control*

*Configuration status accounting*

*Configuration audits*

*Packaging, storage, handling, and delivery*

*This appendix should reference the organizational, or program, level configuration management policies, plans and procedures and should detail any deviations from the organizational standard. Please see example text below:*

*Refer to the [Organization Name] Software Configuration Management Plan for all applicable processes and procedures.]*