CMSC 447

Software Development Plan (SDP)

1 Scope

This section shall be divided into the following paragraphs.

1.1 Identification

This paragraph shall contain a full identification of the system and the software to which this document applies, including, as applicable, identification number(s), title(s), abbreviation(s), version number(s), and release number(s).

This Software Development Plan (SDP) establishes the plans to be used for the development of the ROBBR system.

1.2 System overview

This paragraph shall briefly state the purpose of the system and the software to which this document applies. It shall describe the general nature of the system and software; summarize the history of system development, operation, and maintenance; identify the project sponsor, acquirer, user, developer, and support agencies; identify current and planned operating sites; and list other relevant documents.

The purpose of the ROBBR system is to take user input, use it to filter publicly available data on the United States, and output a list of the best possible zip codes for a jewel thieves syndicate to place their headquarters in a given state. The ROBBR system is being developed for customer, who will be the sole sponsor, acquirer, and user. The ROBBR system will run on a bootable usb that customer can carry with them at all times. The I am Root Software Engineering Team (hereafter referred to as the developer) shall be the sole developers of the software for privacy and security reasons.

1.3 Document overview

This paragraph shall summarize the purpose and contents of this document and shall describe any security or privacy considerations associated with its use.

This document is only to be seen by customer and developer and is meant to explain all of the functions of the product.

1.4 Relationship to other plans

This paragraph shall describe the relationship, if any, of the SDP to other project management plans.

This SDP and its companion documents, the Software Design Description (SDD), Software Requirements Specification (SRS), Software Test Description (STD), Software Test Report (STR), and Software User Manual (SUM) serve as guiding documents to develop the software for the ROBBR system. Additionally, the SUM is meant to serve as a manual for the customer on how to operate the ROBBR system. The SDP is meant to summarize those companion documents listed above.

2 Referenced documents

This section shall list the number, title, revision, and date of all documents referenced in this plan. This section shall also identify the source for all documents not available through normal Government stocking activities.

3 Overview of required work

This section shall be divided into paragraphs as needed to establish the context for the planning described in later sections. It shall include, as applicable, an overview of:

3.1 Requirements and constraints on the system and software to be developed

a. The system must run on a bootable usb.
b. The system must have access to the Internet.
c. The system must communicate with several APIs.

3.2 Position of the project in the system life cycle

ROBBR is in the beginning of its life cycle.

3.3 The selected program/acquisition strategy or any requirements or constraints on it

ROBBR will only be used by the customer.

3.4 Requirements and constraints on project schedules and resources

The ROBBR must be completed by the end of the semester and we will need a usb stick.

3.5 Other requirements and constraints, such as on project security, privacy, methods, standards, interdependencies in hardware and software development, etc.

None.

4 Plans for performing general software development activities

This section shall be divided into the following paragraphs. Provisions corresponding to non-required activities may be satisfied by the words "Not applicable." If different builds or different software on the project require different planning, these differences shall be noted in the paragraphs. In addition to the content specified below, each paragraph shall identify applicable risks/uncertainties and plans for dealing with them.

4.1 Software development process

This paragraph shall describe the software development process to be used. The planning shall cover all contractual clauses concerning this topic, identifying planned builds, if applicable, their objectives, and the software development activities to be performed in each build.

The developer will be using a variant on the waterfall method, which will include weekly meetings to determine and justify progress.

4.2 General plans for software development

This paragraph shall be divided into the following subparagraphs.

4.2.1 Software development methods

This paragraph shall describe or reference the software development methods to be used. Included shall be descriptions of the manual and automated tools and procedures to be used in support of these methods. The methods shall cover all contractual clauses concerning this topic. Reference may be made to other paragraphs in this plan if the methods are better described in context with the activities to which they will be applied.

The ROBBR software development will apply the following general methods:

a. Phase 1: Software Requirements. The project will follow the defined processes recorded in Section 5 to conduct software requirements analysis and develop the Software Requirements Description (SRD) and preliminary User Documentation Description (UDD).
b. Phase 2: Software Design. The software architecture will consist of reusable modules for performing individual tasks. The project will follow the defined processes documented in Section 5. An automated tool will be used to create documentation relating to the code description and outline.
c. Phase 3: Software Unit Development, Test, and Integration. The project will develop new code (or modify reused code), unit test, integrate, and document software following the processes in Section 5. While reused code will not be expected to conform to a single coding standard, changed source code must be supplemented with sufficient new comments and standard code headers to meet commenting provisions of the coding standard and to promote understandability.
d. Phase 4: System Qualification Test and Delivery. The developer will conduct Qualification and Testing according to Qualification Test Plans and Procedures, and document results in a Test Report.
e. Phase 5: Support Installation and Use. The I am Root project team will provide support, solely to the customer, to software installation, acceptance testing, and user operation, only until the end of the semester (approximately May of 2018).

4.2.2 Standards for software products

This paragraph shall describe or reference the standards to be followed for representing requirements, design, code, test cases, test procedures, and test results. The standards shall cover all contractual clauses concerning this topic. Reference may be made to other paragraphs in this plan if the standards are better described in context with the activities to which they will be applied. Standards for code shall be provided for each programming language to be used. They shall include at a minimum:

a. Standards for format (such as indentation, spacing, capitalization, and order of information)

- The indentation and spacing is dictated by the python language.
- Capitalization is defined by naming conventions, listed in 4.2.2.d

b. Standards for header comments (requiring, for example, name/identifier of the code; version identification; modification history; purpose; requirements and design decisions implemented; notes on the processing (such as algorithms used, assumptions, constraints, limitations, and side effects); and notes on the data (inputs, outputs, variables, data structures, etc.)

- Each file shall have a header including:
    a. File name
    b. File version number
    c. The developers
    d. A description of the file
    e. Required inputs and desired outputs
    f. Descriptions of all variables used
    g. State changes
    h. Description of all imports
    i. Any known bugs in the file

c. Standards for other comments (such as required number and content expectations)

- Each method shall have a header that tells what the function does and description of variables used within the method, as well as required inputs, desired outputs and state changes.
- Any algorithms not described in the file header or method header will be described with in line comments.

d. Naming conventions for variables, parameters, packages, procedures, files, etc.

- Functions and methods will be done in camel case, with leading lowercase characters (ex: camelCase)
- Variables and parameters will be separated by underscores with leading lowercase characters (ex: underscore_case)
- Objects will be separated by underscores with leading uppercase characters (ex: Upper_case)

e. Restrictions, if any, on the use of programming language constructs or features

- There shall be no restrictions on the use of programming language features.

4.2.3 Reusable software products

This paragraph shall be divided into the following subparagraphs.

4.2.3.1 Incorporating reusable software products

This paragraph shall describe the approach to be followed for identifying, evaluating, and incorporating reusable software products, including the scope of the search for such products and the criteria to be used for their evaluation. It shall cover all contractual clauses concerning this topic. Candidate or selected reusable software products known at the time this plan is prepared or updated shall be identified and described, together with benefits, drawbacks, and restrictions, as applicable, associated with their use.

The project will follow the processes defined in Section 5 to identify, evaluate, and incorporate reusable software products in the ROBBR system.

4.2.3.2 *Developing reusable software products*

This paragraph shall describe the approach to be followed for identifying, evaluating, and reporting opportunities for developing reusable software products. It shall cover all contractual clauses concerning this topic.

Emphasis will be placed on good software engineering practices such as encapsulation and heavy and descriptive in line comments to provide a basis for future development.

4.2.4 Handling of critical requirements

This paragraph shall be divided into the following subparagraphs to describe the approach to be followed for handling requirements designated critical. The planning in each subparagraph shall cover all contractual clauses concerning the identified topic.

4.2.4.1 *Safety assurance*

Safety is unlikely to be a concern in this project.

4.2.4.2 *Security assurance*

Software engineers on the I am Root team will only discuss information regarding CUSTOMER privately. Additionally, the entire program will be available solely on a bootable usb given to CUSTOMER at termination of the project.

4.2.4.3 *Privacy assurance*

The program does not store previous searches and will be solely available on a bootable usb.

4.2.4.4 *Assurance of other critical requirements*

4.2.5 Computer hardware resource utilization

This paragraph shall describe the approach to be followed for allocating computer hardware resources and monitoring their utilization. It shall cover all contractual clauses concerning this topic.

4.2.6 Recording rationale

This paragraph shall describe the approach to be followed for recording rationale that will be useful to the support agency for key decisions made on the project. It shall interpret the term "key decisions" for the project and state where the rationale are to be recorded. It shall cover all contractual clauses concerning this topic.

4.2.7 Access for acquirer review

This paragraph shall describe the approach to be followed for providing the acquirer or its authorized representative access to developer and subcontractor facilities for review of software products and activities. It shall cover all contractual clauses concerning this topic.

After each phase of production, all relevant documents and code will be provided to the customer for review. After the review, necessary changes will be made. Before the end of the next phase, improvements based upon customer input will be shown to the customer.

5 Plans for performing detailed software development activities

This section shall be divided into the following paragraphs. Provisions corresponding to non-required activities may be satisfied by the words "Not applicable." If different builds or different software on the project require different planning, these differences shall be noted in the paragraphs. The discussion of each activity shall include the approach (methods/procedures/tools) to be applied to: 1) the analysis or other technical tasks involved, 2) the recording of results, and 3) the preparation of associated deliverables, if applicable. The discussion shall also identify applicable risks/uncertainties and plans for dealing with them. Reference may be made to 4.2.1 if applicable methods are described there.

5.1 Project planning and oversight

This paragraph shall be divided into the following subparagraphs to describe the approach to be followed for project planning and oversight. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.

Project planning will be recorded in this document, the SDP, as well as the the SRS. All members of the I am Root development team will be responsible for the oversight of the documentation and software, but members may have more knowledge about the sections that they personally worked on.

5.1.1 Software development planning (covering updates to this plan)

The software development team will employ current "best" practices of unit testing, peer reviews, and project tracking. They will utilize the JIRA system to keep track of productivity, upcoming work, and previously accomplished tasks.

5.1.2 CSCI test planning

5.1.3 System test planning

After unit testing is completed and passed for all functions, they will be slowly added to one another and tested again.

5.1.4 Software installation planning

The software will be installed on a usb drive. The usb drive is a live bootable drive on which the software can be run.

5.1.5 Software transition planning

The contract for ROBBR will only be maintained until May 2018

5.1.6 Following and updating plans, including the intervals for management review

Plan updates and management review will take place during weekly email chains with the customer.We need to talk about how denis is getting updates and how often

5.3 System requirements analysis

This paragraph shall be divided into the following subparagraphs to describe the approach to be followed for participating in system requirements analysis. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.

5.3.1 Analysis of user input

Any confusion or clarification will be addressed in email between the customer and the designated POC. User input

5.3.2 Operational concept

5.3.3 System requirements

5.4 System design

This paragraph shall be divided into the following subparagraphs to describe the approach to be followed for participating in system design. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.

5.4.1 System-wide design decisions

The overall System-wide design decisions shall be made by the development team based on the design requirements given by the customer.

5.4.2 System architectural design

The architectural design shall be entirely decided by the development team.

5.5 Software requirements analysis

This paragraph shall describe the approach to be followed for software requirements analysis. The approach shall cover all contractual clauses concerning this topic.

Every week the project will be compared to the requirements. Ones met will be pushed off the list and ones that are not will be discussed. A plan of how to attack them for the new week will be created and next week we meet again.

5.6 Software design

This paragraph shall be divided into the following subparagraphs to describe the approach to be followed for software design. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.

5.6.1 CSCI-wide design decisions

Design decisions will be made by the team during meetings.

5.6.2 CSCI architectural design

The final product will consist of a single executable program launched by flashdrive, that will connect to the internet to acquire needed data.

5.6.3 CSCI detailed design

5.7 Software implementation and unit testing

5.7.1 Software implementation

The ROBBR software will be written primarily in the Python programming language. Open source APIs will also be utilized. Software produced using this process will be stored on the team's GitHub. Code peer reviews will be held to ensure better quality.

5.7.2 Preparing for unit testing

The preparation of unit test cases will be the responsibility of the programmer who makes modifications to the ROBBR software. For each software unit, test cases will be devleoped using structural testing methods. This will be based on the structure of the code. The test cases will be recorded and maintained in the software development files (GitHub).

5.7.3 Performing unit testing

For each ROBBR version modification, the affected software units will be tested by the programmer in accordance with the unit test cases.

5.7.4 Revision and retesting

Based on the results of unit testing, any necessary revisions to ROBBR will be made by the programmer. The software unit will be retested and the software documentation will be updated as needed. This procedure will be repeated until all test cases have been satisfied.

5.7.5 Analyzing and recording unit test results

The I am Root software development team will evaluate and approve the results of unit testing. The test results will be recorded with the test cases and maintained in the software development files (GitHub).

5.8 Unit integration and testing

5.8.1 Preparing for unit integration and testing

Team members will prepare integration integration test cases and performing integration testing. For each build element, test cases will be developed using functional testing methods. The test cases will be recorded and maintained on the team's GitHub repository.

5.8.2 Performing unit integration and testing

Upon approval by the I am Root software development team, the affected software units will be integrated into the test version of the build element - which will then undergo test cases.

5.8.3 Revision and retesting

Problems detected during integration testing will be recorded on a document shared by the I am Root development team. It follows that necessary revisions to the software will be performed by the team. After revisions are made, the software will be retested and the software documentation will be updated.

This procedure will be repeated until all test cases have been satisfied and up to delivery of the software.

5.8.4 Analyzing and recording unit integration and test results

The I am Root development team will evaluate and approve the results of integration testing. The test results will be recorded along with the test cases - all of which will be maintained in the GitHub.

5.9 CSCI qualification testing

This paragraph shall be divided into the following sub- paragraphs to describe the approach to be followed for CSCI qualification testing. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.

5.9.1 Independence in CSCI qualification testing

5.9.2 Testing on the target computer system

5.9.3 Preparing for CSCI qualification testing

5.9.4 Dry run of CSCI qualification testing

5.9.5 Performing CSCI qualification testing

5.9.6 Revision and retesting

5.9.7 Analyzing and recording CSCI qualification test results

5.10 CSCI/HWCI integration and testing

This paragraph shall be divided into the following subparagraphs to 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.

5.10.1 Preparing for CSCI/HWCI integration and testing

5.10.2 Performing CSCI/HWCI integration and testing

5.10.3 Revision and retesting

5.10.4 Analyzing and recording CSCI/HWCI integration and test results

5.11 System qualification testing

This paragraph shall be divided into the following sub- paragraphs to 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.

5.11.1 Independence in system qualification testing

5.11.2 Testing on the target computer system

5.11.3 Preparing for system qualification testing

5.11.4 Dry run of system qualification testing

5.11.5 Performing system qualification testing

5.11.6 Revision and retesting

5.11.7 Analyzing and recording system qualification test results

5.12 Preparing for software use

This paragraph shall be divided into the following sub- paragraphs to describe the approach to be followed for preparing for software use. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.

The software development team will be responsible for putting the software onto the USB.

5.12.1 Preparing the executable software

The program will be compiled and all ready to run. All the user will do is execute the software.

5.12.2 Preparing version descriptions for user sites

Whole numbers will be saved for full releases with .1 .1 naming conventions for incremental developments.

5.12.3 Preparing user manuals

There will be no support site

5.12.4 Installation at user sites

Once software is installed on USB it is out of developers hands. Now if the program is to be updated the user must take steps to do this.

5.13 Preparing for software transition

This paragraph shall be divided into the following subparagraphs to 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.

5.13.1 Preparing the executable software

5.13.2 Preparing source files

source files will all be uploaded and downloaded through github. This will make sure that every version is up to date.

5.13.3 Preparing version descriptions for the support site

5.13.4 Preparing the "as built" CSCI design and other software support information

5.13.5 Updating the system design description

5.13.6 Preparing support manuals

There will be no support manual at this time.

5.13.7 Transition to the designated support site

5.14 Software configuration management

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

5.14.1 Configuration identification

The program will have a limited number of configuration items...

5.14.2 Configuration control

5.14.3 Configuration status accounting

5.14.4 Configuration audits

5.14.5 Packaging, storage, handling, and delivery

The project shall be stored, delivered on, and run from a flash drive.

5.15 Software product evaluation

This paragraph shall be divided into the following sub- paragraphs to describe the approach to be followed for software product evaluation. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.

5.15.1 In-process and final software product evaluations

5.15.2 Software product evaluation records, including items to be recorded

5.15.3 Independence in software product evaluation

5.16 Software quality assurance

This paragraph shall be divided into the following sub- paragraphs to describe the approach to be followed for software quality assurance. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.

5.16.1 Software quality assurance evaluations

5.16.2 Software quality assurance records, including items to be recorded

5.16.3 Independence in software quality assurance

5.17 Corrective action

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

5.17.1 Problem/change reports, including items to be recorded (candidate items include project name, originator, problem number, problem name, software element or document affected, origination date, category and priority, description, analyst assigned to the problem, date assigned, date completed, analysis time, recommended solution, impacts, problem status, approval of solution, follow-up actions, corrector, correction date, version where corrected, correction time, description of solution implemented)

5.17.2 Corrective action system

5.18 Joint technical and management reviews

This paragraph shall be divided into the following subparagraphs to 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.

5.18.1 Joint technical reviews, including a proposed set of reviews

5.18.2 Joint management reviews, including a proposed set of reviews

5.19 Other software development activities

This paragraph shall be divided into the following subparagraphs to 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.

5.19.1 Risk management, including known risks and corresponding strategies

5.19.2 Software management indicators, including indicators to be used

5.19.3 Security and privacy

5.19.4 Subcontractor management

5.19.5 Interface with software independent verification and validation (IV&V) agents

5.19.6 Coordination with associate developers

5.19.7 Improvement of project processes

5.19.8 Other activities not covered elsewhere in the plan

6 Schedules and activity network

This section shall present:

a. 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
b. An activity network, depicting sequential relationships and dependencies among

activities and identifying those activities that impose the greatest time restrictions on the project

Table 6-1. Development Activities

Task Name	**Duration**	**Start**	**Finish**
Phase 1: Software Requirements	x day(s)	y mo/day/year	y mo/day/year
Develop Software Development Plan (SDP)	1+ days	Wed 3/21/2018	???
…			
Phase 2: Software Design			
…			
Phase 3: Software unit development, test, and integration			
…			
Phase 4: System Qual Test and Delivery			
…			
Phase 5: ______?			
…			

Chet's note: This should be added to throughout the year. As of 3/21/2018 ~9:50 PM, this table is merely a placeholder. We should be able to change around the phases and tasks as we see fit. For example, I'm not sure if we need something beyond Phase 4 (Phase 5 being maintenance post-delivery).

Figure 6-1. ROBBR Schedule and Major Milestones

[Placeholder]

Chet's note: Do we need an activity network? (see: Figure 6.1 from the sample SDP). I feel that it is something we can only draw up once we finish.

7 Project organization and resources

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

7.1 Project organization

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.

Every member of the development team will work under CUSTOMER directly, with one point of contact who is in charge of communication between CUSTOMER and the development team.

7.2 Project resources

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

a. 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, manage- ment, 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
b. 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.
c. 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.
d. Other required resources, including a plan for obtaining the resources, dates needed, and availability of each resource item.

8 Notes

This section shall contain any general information that aids in understanding this document (e.g., background information, glossary, 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.

A.

Appendixes

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

DESCRIPTION/PURPOSE

The Software Development Plan (SDP) describes a developer’s plans for conducting a software development effort. The term "software development" is meant to include new development, modification, reuse, reengineering, maintenance, and all other activities resulting in software products.

The SDP provides the acquirer insight into, and a tool for monitoring, the processes to be followed for software development, the methods to be used, the approach to be followed for each activity, and project schedules, organization, and resources.

APPLICATION/INTERRELATIONSHIP

Portions of this plan may be bound separately if this approach enhances their usability. Examples include plans for software configuration management and software quality assurance.

The Contract Data Requirements List (CDRL) should specify whether deliverable data are to be delivered on paper or electronic media; are to be in a given electronic form (such as ASCII, CALS, or compatible with a specified word processor or other support software); may be delivered in developer format rather than in the format specified herein; and may reside in a computer-aided software engineering (CASE) or other automated tool rather than in the form of a traditional document.

PREPARATION INSTRUCTIONS

General instructions.

a. Automated techniques. Use of automated techniques is encouraged. The term "document" in this means a collection of data regardless of its medium.

b. Alternate presentation styles. Diagrams, tables, matrices, and other presentation styles are acceptable substitutes for text when data required can be made more readable using these styles.

c. Title page or identifier. The document shall include a title page containing, as applicable: document number; volume number; version/revision indicator; security markings or other restrictions on the handling of the document; date; document title; name, abbreviation, and any other identifier for the system, subsystem, or item to which the document applies; contract number; CDRL item number; organization for which the document has been prepared; name and address of the preparing organization; and distribution statement. For data in a database or other alternative form, this information shall be included on external and internal labels or by equivalent identification methods.

d. Table of contents. The document shall contain a table of contents providing the number, title, and page number of each titled paragraph, figure, table, and appendix. For data in a database or other alternative form, this information shall consist of an internal or external table of contents containing pointers to, or instructions for accessing, each paragraph, figure, table, and appendix or their equivalents.

e. Page numbering/labeling. Each page shall contain a unique page number and display the document number, including version, volume, and date, as applicable. For data in a database or other alternative form, files, screens, or other entities shall be assigned names or numbers in such a way that desired data can be indexed and accessed.

f. Response to tailoring instructions. If a paragraph is tailored out of this document, the resulting document shall contain the corresponding paragraph number and title, followed by "This paragraph has been tailored out." For data in a database or other alternative form, this representation need occur only in the table of contents or equivalent.

g. Multiple paragraphs and subparagraphs. Any section, paragraph, or subparagraph in this DID may be written as multiple paragraphs or subparagraphs to enhance readability.

h. Standard data descriptions. If a data description required by this document has been published in a standard data element dictionary specified in the contract, reference to an entry in that dictionary is preferred over including the description itself.

i. Substitution of existing documents. Commercial or other existing documents, including other project plans, may be substituted for all or part of the document if they contain the required data.