CMSC 447

Software Development Plan (SDP)

1 Scope

This section shall be divided into the following subsections.

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 document applies to the program identified as the Realty Map Filtering (RMF) application. The version number as of March 29, 2018 is 1.0.0.

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.

This application's purpose is to locate houses for sale across the United States (U.S.) that meet certain specifications defined by the user. These specifications include, but are not limited to, property data (e.g., market value), nearby schools, crime rates, and community types (e.g., urban, suburban, or rural). The results will be displayed on a map within the user interface (UI). The details of a property listing will be shown along with the specifications that it met from the user's query.

The developer team of this project is */* No Comment */*. The sponsor and primary user is Yatish Joshi from Cisco Systems. The system documentation will include this Software Development Plan (SDP), and these external documents: Software Requirements Specification (SRS), Software Design Description (SDD), Software Test Description (STD), Software Test Report (STR), and a Software User Manual (SUM).

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 contains an overview of the RMF application project. It allows the client to understand what we plan to create, how we plan to create it, and how we plan on testing it. This will allow the client to insure that we are creating what they intended and it is to their preferences.

We consider network traffic generated by this application will require security in order from being altered and returning false results. A countermeasure for this will require us to use an HTTPS connection as a query is made. Currently, we have no privacy concerns due to no intent of collecting information about the user in addition to the absence of a login account system.

This document contains requirement specifications, plans on development, preparations for software use and testing procedures. This will allow the client to insure that we are creating what they intended and it is to their preferences.

1.4 Relationship to other plans

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

NO

2 Referenced documents

This section shall list the number, title, revision, and date of all documents referenced in this plan.

Software Requirement Specification (version 1), March 28, 2018.

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:

These requirements and resources shall be described in greater detail in later parts of this document as well as in the Software Requirement Specification (SRS) document.

a. Requirements and constraints on the system and software to be developed

Requirements that the project must accomplish:

- Results shall be areas of interests
- Results shall be shown on a map
- Results shall be based upon a search from user inputs
- Primary return shall be houses to buy based on the specifications
- Search options will include local schools
- Local schools will include their ranking
- Search options will include states
- Search options will include counties
- Search options will include specific communities
- Search options will include local crime rates

Constraints of the Project:

- Only access to free APIs
- Each API has limited calls per day for free

b. Requirements and constraints on project documentation

Templates have been provided, however these templates are generic and not specific to this project. Therefore, it is up to the developers to filter

c. Position of the project in the system life cycle

The project is currently in the analysis phase as we haven't set our project requirements in detail yet.

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

The project will be written in Python.

e. Requirements and constraints on project schedules and resources

The project is to be completed at the end of the current semester, May 10th, 2018.

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.

Due to the nature of a classroom setting, we are forced to use agile development. This plan comes in six phases: Meet, Plan, Design, Develop, Test and Evaluate. This process will be repeated for requirement and may be repeated multiple times for any given requirement until it is satisfactory. The meet phase involves establishing requirements, and discussing changes that need to be made from the previous evaluation. This plan phase is where the development team will decide how to meet and integrate the requirement into the current state of the project. Those working on creating the code will record the exact purpose of each part of the software to make testing accurate and easier. The development phase will be the actual coding and creation of the software. The test phase will be where the testing member of the team will ensure that the purpose recorded has been achieved and the software is free of bugs. If any issues do arise, it will logged and progress will return back to the development phase to correct the problem. Once all bugs have been removed, the client will evaluate the software to ensure that it is as they expect it to be. If it is, progress will move on to the next requirement. Otherwize, the cycle will begin again still focusing on the particular specification.

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.

As this project is on a web browser, HTML, CSS, and Angular will be used to create the user interface. Javascript will be used to display map results. Python will be used to communicate with APIs to process filtering results. Lastly, gitHub will be used to synchronizing our progress on the code.

(programming languages and tools)

This project will be implemented as a web application. The front end with be implemented using Angular. The back end will be implemented using Python. Flask will be added to python to run the application. Github will handle source control. Travis CI will handle continuous integration and automatically build the project when new commits are pushed.

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:
   - Indents will be done through use of the tab key which is equivalent to 4 spaces.
   - All capitalization will be either in the format of lower camelcase or all lowercase with words separated with underscores
   - Spacing will be used to separate operation parts to make them more readable. Example: `a + b = c`
   - Tests will be run in its own file. Each test will run a separate requirement and insure that the requirement has been met.

b. Standards for header comments:

The following information will be provided at the very top of each file included in the project. Each of the following bullets will be on a new line

- Name of the file
- Primary file author(s)

   - Date of the file’s creation
   - Version number
   - A few sentences describing the purpose of the file and what it accomplishes

c. 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.)

d. Above each function will be a comment including the following information:
   - The name of the function followed by a short description of what the function accomplishes
   - The inputs of this function, each briefly described
   - The outputs of this function, each briefly described

e. Naming conventions for variables, parameters, packages, procedures, files, etc.
   - Variables will be named relevant to what their purpose is. They will be all in lower camelcase or all lowercase with words separated by underscores as necessary to improve readability. Example: `houseLocation` or `house_location.`
   - Constants will be in all uppercase and words will be separated by underscores. Example: `EARTH_GRAVITY`
   - Function names will be lowercase, with words separated by underscores as necessary to improve readability.

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

g. Restrictions, if any, on the complexity of code aggregates

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.

This project will use Angular, jQuery, Bootstrap, Redfin, Zillow, and the Google Maps API. Angular will be used for front end development. It enhances javascripts abilities in manipulating the HTML. This will improve the development process and results. jQuery is necessary for Angular. Bootstrap improves the aesthetics of the web application. Redfin and Zillow are APIs that will be the data that will be filtered to provide the results to the user. Lastly Google Maps will be used to display and provide and interactable map to visualize the location of the houses that meet the criteria of the search.

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.

Due to the way that this project is being developed, each part of it is separated and therefore can be used in many different ways. The code base will be easy to navigate and read. When possible, existing software packages that have extensive documentation will be utilized. With the exception of this use, this development team takes no responsibility for how the code in this application is used.

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

4.2.4.2 Security assurance

4.2.4.3 Privacy assurance

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.

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.

5.1.1 Software development planning (covering updates to this plan)

The team members interface via groupme. The version control of the code base is handled over github. Travis CI is used to handle continuous integration and automate testing of the web application when new commits are made. Documentation is recorded using Google Docs. If this plan is updated, we will record it using Google Docs.

5.1.2 CSCI test planning

5.1.3 System test planning

5.1.4 Software installation planning

Install Flask and Python. The software will not run without both of these installed. Git also needs to be installed to access the version control.

5.1.5 Software transition planning

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

5.2 Establishing a software development environment

This paragraph shall be divided into the following subparagraphs to describe the approach to be followed for establishing, controlling, and maintaining a software development environment. The planning in each subparagraph shall cover all contractual clauses regarding the identified topic.

5.2.1 Software engineering environment

The software is developed and run on a Macintosh laptop.

5.2.2 Software test environment

The test environment is Travis CI as well as a Macintosh laptop. First, any new changes will be tested on the local machine (Macintosh laptop) they are developed on. When verified by the developer, they will be pushed to Github. Travis CI will then pull the code from Github and test it in the cloud.

5.2.3 Software development library

The libraries used are Angular, jQuery, Bootstrap, and Flask. The purpose of each of these were identified in Section 4.2.3.1

5.2.4 Software development files

- Index.html: lays out a webpage
- app.js: controls the webpage
- style.css: stylesheet for webpage
- map_of_jobs.py: backend of web application

5.2.5 Non-deliverable software

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

The user input will consist of the following:

- State and zipcode will be entered through text
- Crime rates
- School ratings
- Housing cost
- House features

5.3.2 Operational concept

User inputs desired parameters. Based upon those parameters, results are calculated from the various APIs. The results are a collection of houses, with their location. The Google Maps API then uses these locations to display where the houses are located. Finally these homes are shown to the user to show their address and location.

5.3.3 System requirements

System must be able to run a web browser with Internet connection.

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

Development was performed on a Macintosh Laptop and code was hosted on gitHub. The software will be tested on Travis CI.

5.4.2 System architectural design

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.

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

Existing software packages will be used when possible along with existing APIs to process the specifications.

5.6.2 CSCI architectural design

5.6.3 CSCI detailed design

5.7 Software implementation and unit testing

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

5.7.1 Software implementation

How are we gonna implement the software (IDEs and other tools we will use)

5.7.2 Preparing for unit testing

We will have the software development members write a chunk of the program document list what the inputs and outputs are supposed to be

5.7.3 Performing unit testing

We will implement the unit tests using Pycharm. We will follow the documentation given by the people who wrote the code we are testing and write up meaningful tests to cover the majority of the functionality of the code.

5.7.4 Revision and retesting

Everytime we run a unit test, we will run the rest of the unit tests to make sure one section of code doesn't break another. If we make substantial changes to a section of code, we will look over its unit tests to be sure the tests are still meaningful and provide adaquest coverage.

5.7.5 Analyzing and recording unit test results

5.8 Unit integration and testing

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

5.8.1 Preparing for unit integration and testing

5.8.2 Performing unit integration and testing

5.8.3 Revision and retesting

5.8.4 Analyzing and recording unit integration and test results

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.

5.12.1 Preparing the executable software

5.12.2 Preparing version descriptions for user sites

5.12.3 Preparing user manuals

A user manual will be created by one of the members of the team. It will include a description of all functions as well as instructions on how to operate the software. the manual will also include screenshots of the software during the step by step instructions to make it easier to understand and help the user follow the steps.

5.12.4 Installation at user sites

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

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

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

5.14.2 Configuration control

5.14.3 Configuration status accounting

5.14.4 Configuration audits

5.14.5 Packaging, storage, handling, and delivery

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

During development, a regular meeting will be held every two weeks. Any documentation written or requirements coded will be reviewed by the client. The client will ensure that their expectation are met. If the client notices something unsatisfactory, the development team will make note and correct it and follow up on it during the next meeting. This cycle will repeat until all requirements have been completed. A final overall evaluation and correction cycle will occur, completing the RMF software.

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

As the software is being coded, the team member will perform tests to ensure that the software is performing correctly and accurately. Once a specification has been fully coded, a second team member will perform a test as well to provide an outside opinion. If any bugs arise, the original coder will be notified and will be the one to repair the program. Prior to final submission, an overall test will be performed to ensure that all the parts work well together.

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

NONE REQUIRED

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

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.

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.