Development Plan

Purpose

This document outlines how software engineers will develop Clearview.

Software Development Life Cycle Model

Clearview will be developed using an evolutionary software development life cycle model.

The "evolutionary" strategy develops the software system using a sequence of builds. Customer needs and software system requirements are partially defined up front, then are refined in each succeeding build.

Development Process

Requirements Analysis Activity

If they have not been recorded already, the first task wil be to decide on system requirements and record them. The system requirements may be recorded in external software (e.g. Greenlight Guru). Each system requirement requires a unique identifier so that we can trace our software requirements back to the system requirements.

The initial set of software requirements should be gathered and recorded in the "requirements file." See Appendix A for the format of the requirements. To the extent possible, all software requirements should be enumerated at the start of the project. If new requirements are added during the project, they should be added to the requirements file.

When requirements are added or are changed, the developer must:

1. Re-evaluate the medical device risk analysis and update it as appropriate
2. Ensure that existing requirements, including system requirements, are re-evaluated and updated as appropriate
3. Verify that the software requirements implement the system requirements, and that the software requirements are properly linked to the system requirements
4. Verify that the software requirements implement all risk controls
5. Verify that the software requirements don't contradict each other
6. Verify that the software requirements are unambiguous
7. Verify that the software requirements are stated in terms that permit establishment of test criteria and performance of tests to determine whether the test criteria have been met

Unit Implementation and Verification Activity

Software Release Activity

When a new version of the software is released, the git commit corresponding to the state of the code should be tagged with the version number.

Maintenance Process

Risk Management Process

Configuration Management Process

Problem Resolution Process

Traceability

Documents

1. Software Development Plan (this document)
2. Risk Management File

References

1. ISO62304:2006

Appendix A

Requirements File

The requirements file is a YAML file containing a list of all software project requirements.

Each requirement will have a unique id beginning with the letter `"r-"`, a description, an optional list of software requirement ids, and a type, according to the following format:

```
r-1:
  type: functional
  description: A concise, unambiguous, verifiable description of the requirement.
  system_requirements: 23, 45, 46
r-2:
  type: input
  description: Another description.
  system_requirements: 3, 45
```

Software requirements must be categorized as one of the following types:

a. Functional and capability requirements (`"functional"`)

- performance (e.g., purpose of software, timing requirements),
- physical characteristics (e.g., code language, platform, operating system),
- computing environment (e.g., hardware, memory size, processing unit, time zone, network infrastructure) under which the software is to perform, and
- need for compatibility with upgrades or multiple SOUP or other device versions.

b. Sofware system inputs and outputs (`"input"` or `"output"`)

- data characteristics (e.g., numerical, alpha-numeric, format) ranges,
- limits, and
- defaults.

c. Interfaces between the software system and other systems (`"interface"`)

d. Software-driven alarms, warnings, and operator messages (`"alert"`)

e. Security requirements (`"security"`)

- those related to the compromise of sensitive information,
- authentication,
- authorization,
- audit trail, and
- communication integrity.

f. Usability engineering requirements that are sensitive to human errors and training (`"usability"`)

- support for manual operations,
- human-equipment interactions,
- constraints on personnel, and
- areas needing concentrated human attention.

g. Data definitions and database requirements (`"data"`)

h. Installation and acceptance requirements of the delivered medical device software at the operation and maintenance site or sites (`"installation"`)

i. Requirements related to methods of operation and maintenance (`"maintenance"`)

j. User documentation to be developed (`"user-documentation"`)

k. User maintenance requirements (`"user-maintenance"`)

l. Regulatory requirements (`"regulatory`)