PURPOSE

This document provides guidance for the Configuration Management of software through the validation and manufacturing processes.

SCOPE

The software manufacturing process presented in this procedure applies to software developed for medical devices, which are used as either part of software validation, the Quality System, or to support product claims.

REFERENCES

ANSI/IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.
FDA – Quality System Inspection Technique (QSIT).

BACKGROUND

The Quality System Inspection Technique (QSIT) requires validation to be performed on the same product that came off the manufacturing line. While this is appropriate for physical products (since they have pilot production lines), it is not appropriate for software. In such cases, QSIT allows validation to be performed prior to production as long as it can be shown that the manufactured product is the same product that was validated. This is a simple process for software, involving a unique “fingerprint” of the software that can be tracked through the process.

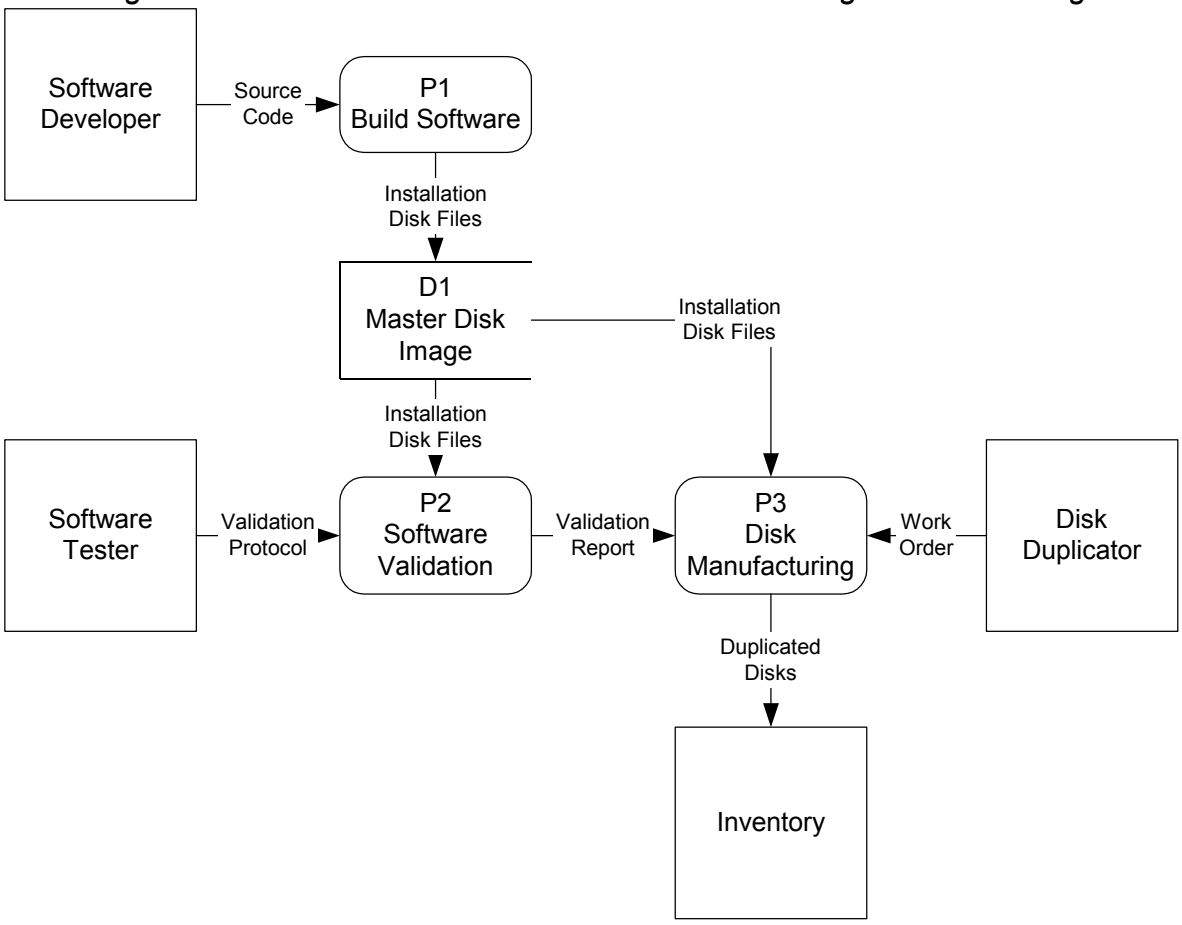
DEFINITIONS

Baseline	A specification or product that has been formally reviewed and agreed upon, that serves as the basis for further development, and that can be changed only through formal change control procedures. Any change to a baselined item produces a new baseline.
Distribution	Delivery of software to an intended user.
Manufacture	Any method of duplicating a software product onto distribution media.
Media	A distribution container for software, including but not limited to: floppy disks, CD-ROMs, ZIP disks, optical disks, tape, or network folders.

DATA FLOW FOR SOFTWARE DISTRIBUTION MEDIA

This section describes the general flow of data from the initial build through software manufacture, for the purpose of identifying the controls needed to ensure the integrity of the distribution media. The formality of each of the processes depends upon the intended use of the software.

Figure 1 – Media Information Flow From Build Through Manufacturing



PROCESS DETAILS

This process begins after the software has already been baselined per the project's Software Configuration Management Plan. To determine the level of formality required for each of the processes in figure 1, refer to the following table showing the process required by the intended use:

P1 Software development performs the build for the installation media.
Input: Source Code.
Output: Master copy of the installation media.

D2 Master copy of the installation media, archived for later retrieval.

P2 Software validation.
Input: Validation Protocol; master copy of the installation media.
Output: Validation Report (pass or fail).

P3 Disk Duplication. This could be duplication of distribution disks or electronic transfer of distribution media.
Input: Validation Report (validation must have passed), Master copy of the installation media, work order or equivalent instructions to duplicate media.
Output: Duplicated media for inventory.

HAZARD ANALYSIS FOR ERRORS IN DUPLICATION

Rather than perform a Process FMEA for disk duplication, this analysis assumes that the various error conditions will always occur; therefore, Probability would always be 10. Since this process must be applicable to any level of concern software, the Criticality is also assumed to be 10. The only parameter remaining for a Process FMEA would be the Detectability, which for this analysis is assumed to be 10 – not detectable. This reduces the task to a Hazard Analysis and mandates using the most effective mitigations.

This section examines the places in the process where information could be altered or corrupted such that the duplicated disks would be incorrect. This also recommends mitigations for the process. This is conducted similar to a software hazard analysis. Since the information flows serially, the mitigations all add together; therefore the residual concern is for the entire process.

Tag #	Hazard/Cause	Mitigation	Mitigation Type	Residual Concern
MFGHAZ.001	Developer builds wrong files (problem in P1).	a) Configuration Management includes version control software.	Protective	Minor
		b) Configuration Management includes build on a separate PC from the developer's PC.	Protective	
MFGHAZ.002	Developer delivers wrong product (problem in P1).	a) Transfer to archive includes a software "fingerprint" as a unique identifier.	Protective	
		b) Configuration Management includes an archive location that is controlled by the software tester.	Protective	
MFGHAZ.003	Files corrupted, overwritten or altered (problem in D1).	a) Software "fingerprint" is recorded in a non-volatile media, such as a checklist for delivery to testing.	Protective	
MFGHAZ.004	Tester retrieves wrong product (problem in P2).	a) Tester verifies the software "fingerprint" and compares it to the "fingerprint" reported by software development.	By Design	

Tag #	Hazard/Cause	Mitigation	Mitigation Type	Residual Concern
MFGHAZ.005	Manufacturing retrieves wrong media for duplication (problem in P3).	a) Validation report includes the software "fingerprint." Mfg verifies the media's "fingerprint" and compares it with the validation report.	By Design	
MFGHAZ.006	Media manufactured incorrectly (problem in P3).	a) Process validation of the disk duplication line, or use of a vendor that uses an approved process.	Protective	
		b) Duplicated disks have their "fingerprints" verified in an inspection step, such as a first article inspection.	By Design	
MFGHAZ.007	Manufacturing duplicates unvalidated software (problem in P3).	a) Formal "release" process, ensuring that software from D1 is physically transferred to another location upon release. Manufacturing uses that "release" location, rather than D1, to retrieve software for duplication.	By Design	

MITIGATION METHODS

Hazard Tag	Mitigation Method
MFGHAZ.001.a	Identify version control software used in the Software Configuration Management Plan.
MFGHAZ.001.b	Setup a separate build PC. For each build, get the files from the version control software. This ensures that all required files have been checked into version control.
MFGHAZ.002.a	Use a program to calculate a unique "fingerprint," such as a hash code or CRC.
MFGHAZ.002.b	Setup a transfer directory or archive of disks. Software delivered to testing is dropped into this location. Note the existence of this archive and the transfer process in the project's Software Configuration Management Plan.
MFGHAZ.003.a	Use a transfer checklist to transfer software to testing. Checklist includes location in the archive and the media "fingerprint." Note the existence of this checklist and the minimum information in the project's Software Configuration Management Plan. Checklists should be kept in a project binder, not the Design History File.
MFGHAZ.004.a	Use a program to calculate a unique "fingerprint," such as a hash code or CRC. Testing verifies this when they receive software from Development.
MFGHAZ.005.a	Testing includes the media "fingerprint" in the validation report. Manufacturing instructions for the media should include use of the "fingerprint" program to verify the media.
MFGHAZ.006.a	Either use a cGMP compliant production line, or an approved vendor (such as Reel Pictures) to duplicate disks.
MFGHAZ.006.b	Use a program to calculate a unique "fingerprint," such as a hash code or CRC. Manufacturing verifies this when they receive duplicated media. This can be a first article inspection, sampling, of 100% inspection, as appropriate.
MFGHAZ.007.a	Establish a formal release process via Document Control, in which the media is placed in a QA controlled location. Manufacturing will only retrieve media for duplication from this controlled location.

The residual risk is Minor, because all processes, P1, P2, and P3, use a software "fingerprint" to track the software through the system. While this does not change the Probability or Severity of a hazard, it raises Detectability to zero – making it virtually impossible for a hazardous condition to go undetected and uncorrected.

Software Requirements for "Fingerprint" Software

The use cases are:

Use Case 1 – Obtain "Fingerprint" for Process P1

1. Developer builds the distribution media and places it into a particular directory.
2. Software allows developer to select the directory to "fingerprint".
3. Software calculates "fingerprint" of the selected directory, including all subdirectories.
4. Software displays the calculated "fingerprint."

Use Case 2 – Verify "Fingerprint" for Process P2 & P3

1. Tester/Manufacturing retrieve the media.
2. Software allows Tester/Manufacturing to enter the expected "fingerprint".
3. Software calculates "fingerprint" of the selected directory, including all subdirectories.
4. Software displays the calculated "fingerprint" and a determination that it either: a) matches, or b) fails to match.
5. Software allows results to be printed.