Attachment #6

SW/PLN

Stat Profile Prime
(Codename FirepHOx)
Software Development Plan

Authors:

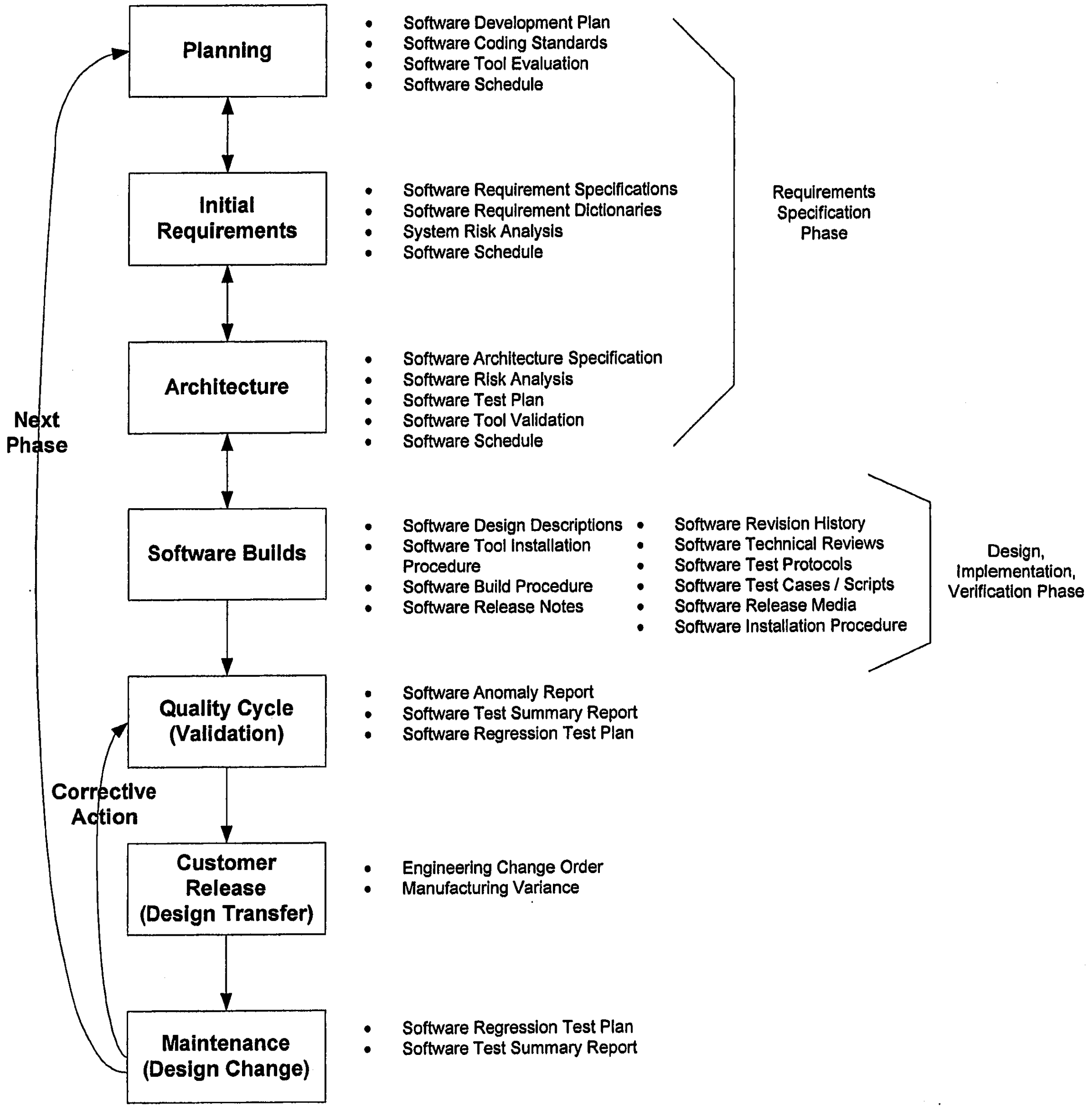
Name	Position	Signature	Date
Tim Kelleher	Principal Software Engineer		01/25/2017

Approvals:

Name	Position	Signature	Date
Tom Peterson	Program Manager		1/25/17.
Dharmin Patel	SQA Manager		01/25/17
Paul MacDonald	Sr. Director, Regulatory Affairs		1/26/17
Bill Jacques	Director, Quality		1/25/2017
Charlie Dodai	Lead Software Engineer		1/25/17

4.1 Software Life Cycle

This section contains the software life cycle for the design, verification, and validation. The following is a diagram depicting the life cycle flow. As depicted by the arrows, some flows are uni-directional and others are bi-directional.


Planning
Software Development Plan
Software Coding Standards
Software Tool Evaluation
Software Schedule
Initial Requirements
Software Requirement Specifications
Software Requirement Dictionaries
System Risk Analysis
Software Schedule
Requirements Specification Phase
Architecture
Software Architecture Specification
Software Risk Analysis
Software Test Plan
Software Tool Validation
Software Schedule
Next Phase
Software Builds
Software Design Descriptions
Software Tool Installation Procedure
Software Build Procedure
Software Release Notes
Software Revision History
Software Technical Reviews
Software Test Protocols
Software Test Cases / Scripts
Software Release Media
Software Installation Procedure
Design, Implementation, Verification Phase
Quality Cycle (Validation)
Software Anomaly Report
Software Test Summary Report
Software Regression Test Plan
Corrective Action
Customer Release (Design Transfer)
Engineering Change Order
Manufacturing Variance
Maintenance (Design Change)
Software Regression Test Plan
Software Test Summary Report

11/25/2013 Rev 1.2

Figure 4.1-1

The above lifecycle fully complies with Software Design Control, OP 145. The primary difference is the following:

- Requirements Specification Stage consists of Planning, Initial Requirements, and Architecture
- Software Builds consist of Design, Implementation, and Verification Stages

4.1.1 Planning

Planning will be performed at the beginning of each development phase for a project. The following software activities are performed as part of the planning phase:

- Work with marketing to define software requirements for development phase
- Participate in the development / update of system architecture and physical (computer) architecture
- Create / update software development plan with schedule
- Identify team members
- Identify / evaluate / purchase tools
- Document / train process and tools for team development

Software Deliverables: Software Development Plan, Software Coding Standards, Software Tool Evaluation, Software Schedule

4.1.2 Initial Requirements

At the beginning of a development phase, system, software, and user interface requirements are created and reviewed with marketing. Prototyping / feasibility (paper or software) and Human Factors Testing will be performed to test hypotheses.

When the requirements for the phase are reviewed and considered at least 80% complete, they are placed under control of the Requirements Management Tool. It is not required to identify all requirements for all milestones and phases at the beginning of the project.

Initial requirements can be the general user interface guidelines and essential screens to perform analytical testing.

Program Deliverable: System Risk Analysis
Software Deliverables: Software Requirement Specifications, Software Requirement Dictionaries, Updated Software Schedule

4.1.3 Architecture

Software architecture is started during requirements gathering and concludes after review with other disciplines in order to comply with system architecture and product specifications.

Detailed requirements are not required to perform initial software architecture.

Software Deliverables: Software Architecture Specification, Software Risk Analysis, Software Test Plan, Software Tool Validation, Updated Software Schedule

4.1.4 Software Builds

After software architecture and tool procurement / validation, the software team will produce a functional software build as specified by the Lead Software Engineer (typically once a week). The contents of the build will be identified by the Lead Software Engineer. In order to perform work in parallel, software development engineers will be assigned separate software components to avoid overlap. It is also intended, if possible, that software verification engineers should create test protocols / cases / scripts for different functional areas or levels of test to avoid overlap.

In addition to new functionality, a review of open issues (software defects and change requests) by the Lead Software Engineer will identify items that require resolution for an upcoming build.

Software Deliverables: Software Design Descriptions, Software Tool Installation Procedure(s), Software Build Procedure, Software Release Notes (if needed), Software Revision History, Software Technical Reviews, Software Test Protocols, Software Test Cases / Scripts, Software Release Media (if needed), Software Installation Procedure

4.1.5 Quality Cycle

Formal software verification and validation will execute all identified test cases/scripts in the test plan to uncover any latent defects. During this phase, all test results will be signed by the executor of the test and archived in the DHF as evidence. This formal testing will be denoted as the "Quality Cycle".

If a defect(s) is detected after the quality cycle which requires resolution, then the software will be branched. Only the fix to the defect(s) will be applied to the application branch. A regression analysis will be conducted by software development and SQA to create a regression test plan. Regression planning and testing will iterate until the software quality meets the release criteria. All corrections to a branch must also be reflected back into the main line (trunk).

At the conclusion of software testing, the Lead Software Engineer and SQA will generate a Test Summary Report and Software Anomaly Report.

Software Deliverables: Signed Test Cases, Software Regression Test Plan(s) if needed, Software Anomaly Report, Software Test Summary Report

4.1.6 Customer Release

At the completion of the Software Quality Cycle, when the software meets quality criteria for release, the software is transferred to manufacturing with an Engineering Change Order or Manufacturing Variance. Once released to manufacturing, the instrument can be used by an external customer. The customer can be any internal or external organization outside of R&D. Internal customers would be manufacturing, quality assurance, marketing, etc. External customers can be field trials or clinical use in medical offices, laboratories, or hospitals.

Software Deliverables: Software Archive Media, Manufacturing Variance (MV), Engineering Change Order (ECO)

4.1.7 Maintenance

After all planned software releases of the analyzer to the market based on customer needs, software development will enter a maintenance phase. During the maintenance phase, parts obsolescence and the corporate CAPA system will determine if and when an additional software release is required. If the software change is large in scope, then a software project should be created and the entire life cycle is repeated. If the software change is localized and small in scope, then the change can be performed with software regression testing.

Software Deliverables: Software Regression Test Plan(s), Software Test Summary Report