



**Software Development
And
Maintenance Plan**

PROTOCOL NO:

EFFECTIVE DATE:

PAGE NO.:

**Software Development
And
Maintenance Plan**

TITLE:

AUTHORING GROUP:

DATE:

SUPERSEDE PROTOCOL NO.:



**Software Development
And
Maintenance Plan**

PROTOCOL NO:

EFFECTIVE DATE:

PAGE NO.:

TABLE OF CONTENTS

Sr. No.	Contents	Page No.
1	Relevant Processes and Documents	
2	Required Resources	
	2.1 Team	
	2.2 Software	
	2.3 System Requirement /Target Runtime	
3	Design Phases	
4	Avoiding Common Software Defects Based on Selected Programming Technology	
5	Configuration Management and Version Control	
6	Documentation Activities	
7	Implementation Verification Activities	
8	Software System Test Activities	
9	Validation Activities	
10	Maintenance Activities	



Software Development And Maintenance Plan

PROTOCOL NO:

EFFECTIVE DATE:

PAGE NO.:

Software Development and Maintenance Plan

This document summarizes development and maintenance activities.

Mapping of Standard Requirements to Document Sections

ISO 13485:2016 Section	Document Section
7.3.2 Design and Development Planning	1, 2, 3, 7

1. Relevant Processes and Documents

Please see the relevant processes for the following activities:

- Risk management activities incl. SOUP risks: SOP Integrated Software Development.
- Problem resolution: SOP Problem Resolution
- Software development incl. deliverables, traceability, regular update of software development plan: SOP Change Management
- SOUP List
- SOP Usability Engineering

2. Required Resources

Role	Count	Responsibilities
Head of Development	1	Prioritizing tasks and technical oversight
Frontend Developer	2	Implementing Frontend Software Requirements
Backend Developer	1	Implementing Backend Software Requirements

2.1 Team

2.2 Software

Describe your software safety class according to IEC 62304 and your reasoning behind the classification.



Software Development And Maintenance Plan

PROTOCOL NO:

EFFECTIVE DATE:

PAGE NO.:

Programming Languages

List the languages using, including compiler and language versions.

Name	Version
pHp/Angular	XX

Development Software

List software used to support development, e.g.IDEs.

Name	Version
xxx	xxxyyy

2.3 System Requirement/Target Runtime

List your target runtime(s).

Name	Version
xxx	yyyz

Specify system requirements, e.g., the minimum specifications of the sever/compute instance you'll be running your software on

Minimum system Requirements:

- Server-grade dual-core CPU, e.g, *Intex Xeon E3-1230 v5 or higher*
- 4 GB of RAM
- 1 GBit/s up-and downlink
- 20 GB SSD storage



Software Development And Maintenance Plan

PROTOCOL NO:

EFFECTIVE DATE:

PAGE NO.:

3 Design Phases

The 13485 requires you to specify "Design Phases". Here are some suggestions which you could use.

	Title	Estimated Completion Date	Description	Review method
4	Specification			Software Requirements Checklist
A	Implementation			Code Reviews
v	Testing			System Test
o	Validation			Usability Evaluation
i	Release			Release Checklist

ng Common Software Defects Based on Selected Programming Technology

Discuss how your selected programming technology may introduce risks and how you plan to avoid them. With modern, dynamically typed languages, an obvious risk is that you encounter runtime exceptions. So you could argue that your test coverage is great and compensates for that. You could also link to your risk analysis here if you analyses those risks further.

5 Configuration Management and Version Control

Describe which version control software you're using, also describe your branching model, i.e., how your developers create branches during development, how you name them and how you merge them (pull requests? Merge commits? Squash before?).

Importantly, describe which things (code, build files, etc.) are put in version control. Describe how you name versions and how you tag them. Your goal should be that you can retrieve an old version and build it.

git is used as version control software. All source code and build files are committed to version control.

When implementing software requirements, developers create a new branch starting at master. During development, developers may create intermediate commits on this development branch.

When implementation is completed, a new merge commit to master is created.



Software Development And Maintenance Plan

PROTOCOL NO:

EFFECTIVE DATE:

PAGE NO.:

This is also the activity which constitutes integration of software units.

For each release, the goal is to be able to uniquely identify it and retrieve all relevant files (code, configuration files like build scripts, SOUPs, etc.) at any time in the future.

When a new software version is released, its commit is tagged in git. The tag is constructed by adhering to semver (semver.org) 2.0.0 which results in a version of format MAJOR,MINOR,PATCH,e.g,1.0.0

6 Documentation Activities

Describe your policy on what should be documented while you develop software. Maybe you want to keep an up-to-date software architecture diagram in the repository,etc. Make sure to mention how traceability between software requirements and Tests is maintained.

7 Implementation Verification Activities

Describe verification activities, e.g. code review.

For each pull request, a code review is performed by a team member who was not the main author of the code under review. The code review is only marked as "approved" if the code complies with the code review criteria. This is:

- *Code fulfills the software requirements*
- *Adherence to [PEP8 Style Guide](#)*

8 Software System Test Activities

Describe software system test activities. This could be continuous integration which is triggered by opening a pull request (e.g. Travis CI, Circle CI). Describe what is tested and how that automated system works.

Integration tests are included in software system tests.

9 Validation Activities

Validation is carried out as formative and summative usability evaluation as described in the software development process. A usability evaluation file (plan, protocol and report) will be prepared.



**Software Development
And
Maintenance Plan**

PROTOCOL NO:

EFFECTIVE DATE:

PAGE NO.:

10 Maintenance Activities

Describe how often you check SOUP issue trackers and how you document them. SOUP issue trackers are checked at least once every 6 months. The verification date is updated in the SOUP list accordingly.