

SOUP List (Software of Unknown Provenance)

The 62304 requires you to document your SOUP, which is short for Software of Unknown Provenance. In human language, those are the third-party libraries you're using in your code, typically in your `requirements.txt` or `Gemfile`.

Classes	IEC 62304:2006 Section	Document Section
B, C	5.3.3 (Functional and Performance Requirements)	2
B, C	5.3.4 (Hardware and Software Requirements)	2
B, C	7.1.2 (Hazardous Situations)	2
B, C	7.1.3 (SOUP Anomaly Lists)	2
A, B, C	8.1.2 (Identify SOUP)	2

1 Risk Level Definitions

The 62304 requires you to assess risks associated with SOUP. The simplest way to do this is to classify each SOUP as a certain risk level. Unless you're developing software which shoots radiation at patients, it's likely that your SOUP risk levels remain "low" or "medium".

Risk Level	Definition
Low	Malfunction in SOUP can't lead to patient harm.
Medium	Malfunction in SOUP can lead to reversible patient harm.
High	Malfunction in SOUP can lead to irreversible patient harm.

2 SOUP List

This is the actual SOUP list. For each third-party library you use, add an entry in the table below. The idea is to only have one "global" SOUP list for your medical device even though the code may actually live in multiple repositories. That's what the "software system" column is for; you could also mention your (git) repository there.

When specifying requirements, the 62304 requires you to mention functional, performance, hard- and software requirements. However, you may not have to re-state certain requirements if they apply to all SOUP, e.g. "runs on linux". So prefer to keep the requirements simple, in a way in which you would communicate it to colleagues on your development team when answering the question "why did we import this library?".

As with all templates: It's more about the content (i.e. the columns

you see below) than the tool (filling this out in google sheets / markdown / wherever). Nobody says that you have to maintain this as a google sheet. If you can find a way to integrate this in your workflow in a better way, e.g. in a markdown file in your git repository, go for it! Just keep in mind that you need to be able to export it to send it to auditors.

ID	Software System	PackageName	Programming Language	Version	Website	Last verified at	Risk Level	Requirements	Verification Reasoning
1	Mobilereact-App	react-native	JavaScript	0.61	Link	23.10.2020	Low *	Runs JS on Android / iOS	Commonly used, maintained by large organisation, sufficient test coverage

Template Copyright openregulatory.com. See template license.

Please don't remove this notice even if you've modified contents of this template.