Failure Mode and Effects Analysis (FMEA): Risk Table

ISO 14971:2019 Section	Document Section
5.2	(all; entries about reasonably foreseeable misuse)
5.4	3
5.5	3, 4
6	3
7.1	4
7.2	4
7.3	4
7.5	4

IEC 62366-1:2015 Section	Title	Document Section
4.1.2	Risk Control as it relates to User Interface design	4
5.3	Identify known or foreseeable Hazards and Hazardous Situations	1,3

This is a Failure Mode and Effects Analysis (FMEA) of the device. It is separated into multiple sections:

- **Failure Modes** lists everything which can go wrong
- **Hazards and Analysis** lists everything (harms) which can subsequently happen, including an analysis of probability and severity
- The list of **Risk Control Measures** contains all control measures which were implemented for risk reduction, either reducing probability or severity, or both.

> Ugh, this became more complex than I initially expected. I am so sorry. From a teaching perspective, it's actually simpler to explain if I throw everything into one table. Instead, we have three here now, because I can't trivially upload spreadsheets to the website, so I had to reduce the column count to make them fit. Painful. I'll try to explain as we go along.

1. Preliminary Hazards Analysis (PHA)

> A Preliminary Hazards Analysis (PHA) is simply a list of stuff which can go wrong. Typically, you come up with that stuff when you think about your product. Like, when you do Covid predictions, you come up with the thought that a wrong prediction will result in

a bad outcome. Makes sense. So, here's a table in which you can collect those ideas. Besides the description it also has the column "source" which describes where the idea came from (typical options are: Intended Use, User Test, 14971 checklist (there is one in the 2012 version, otherwise there's TR 24971, I think)) and "Hazard ID(s)" which shows where you've continued the analysis (including probability and severity) of that risk. It refers to the ID(s) in the Hazards and Analysis table below.

ID	Source	Description	Hazard ID(s)
1	General Considerations	Wrong Covid Prediction	1
2	Intended Use	Wrong Covid Prediction	1
3	Usability Test	User misunderstands prediction result	1

2. Failure Modes

This is a list of stuff which can go wrong in your software. You should be able to come up with things while you write code and when you think about it. Typically, stuff which always can go wrong is 1) something becomes unavailable, 2) something returns invalid data, 3) something gets hacked.

I've written some examples for the Covid predictor application which I cover in my videos - you don't have to watch those for now. The idea is that it's an app which predicts whether a certain patient has Covid, pretty magical. An obvious failure mode would be that the app either calculates wrong predictions on the backend (ID 1), or the frontend displays the predictions wrongly (ID 2). They both lead to the same hazard (ID 1) which is listed in the Hazards and Analysis table below.

ID	Software System	Failure Mode	Hazard ID(s)
1	Backend	Wrong Covid Prediction	1
2	Frontend	Covid Prediction displayed wrongly	1

3. Hazards and Analysis

This is a list of stuff which will subsequently happen after your software has failed. It's more about what happens in the real world, not in your software. The 14971 wants you to analyze Hazards, Hazardous Situations and Harms, so that's what you'll find in the table :)

Here's what happens: In the beginning, there's a hazard, like a wrong Covid prediction. That hazard has a certain probability to lead to the (next) hazardous situation, in this case 1% (0.01), in which the user thinks he is healthy, but actually has Covid. You can estimate p1 with some medical knowledge, in this case maybe the prevalence of Covid in your target population.

Then, this hazardous situation may lead to a harm, in this case, disease progression - the user who got the wrong Covid prediction (healthy) may actually have Covid and now stay at home instead of going to the hospital. So the disease gets worse. Maybe only 10% (0.1, p2) of users will actually blindly trust my Covid app - so not all of them will get disease progression. Again, estimate this based on your own data.

Finally, you multiple p1 with p2, check your Risk Acceptance Matrix (see template for the risk management plan) which Probability Category it is (in this case P4). Also check that matrix which severity that harm would be, I'm just assuming it could be S2 here. Once you have your Probability (P) and Severity (S), check your matrix whether that's acceptable. In this case, it's not.

So we need a Risk Control Measure which is referenced here by ID (1). As we can see, it reduced the probability to P3, but not the severity. But that's fine, because the P3 and S2 - combination is acceptable, based on our Risk Acceptance Table.

The next table contains the list of Risk Control Measures.

ID	Hazard	p1	Hazardous Situation	p2	Harm	p1*p2	P	S	Acceptable?	Comments	Risk Control ID(s)	P	S	Acceptable?
1	Wrong Covid prediction	0.01	User thinks he is healthy, but has Covid	0.1	Disease progression	0.001	P4	S2	No		1	P3	S2	Yes

4. Risk Control Measures

This is the table of Risk Control Measures which was referenced from the Hazards and Analysis table above. We've used the Risk Control Measure with (ID 1) to reduce the risk of disease progression. For that, we came up with a procedure to check our prediction algorithm with test data before we ship it. That makes sense and that should probably reduce the probability of wrong predictions. In this case, by 0.01 (10ˆ-2, I'm coming up with numbers here). It doesn't reduce the severity, of course - the harm is still disease progression.

Note that the 14971 has three types of Risk Control Measures: * Inherent Safety by Design * Protective Measures * Information for Safety

Also note that, in simplified terms, Information for Safety must actually be displayed in your application to have any effect. Like, not in the user manual, because nobody reads the manual.

ID	Description	Type	Probability Reduction	Severity Reduction	Negative Influence on Device Safety / Performance Introduced by Mitigation Measure?	Verification Implementation	Verification Effectiveness
1	Check prediction algorithm with test data	Protective Measure	10ˆ-2	1	No	Link Software Test ID	Link Usability Test ID