

Medical Interface Guidance.



Figure 1: Maddison's work on a cohesive graphical user interface (GUI) designed to offer a unified experience for United Imaging. The software was required to operate seamlessly across the entire range of imaging technologies and equipment.

First, do no harm

If you are a developer of medical interfaces, the requirement for you to consider human factors has increased over the last decade since it was acknowledged that human-machine interaction is responsible for a large portion of medical errors and adverse events. A good guiding principle is the Hippocratic oath – 'First, do no harm'. This should be at the core of your development process for any medical device.

Developing a medical interface is different from developing other product apps and interfaces

The core difference in the development of medical devices is the concept of harm. The onus is on you, the developer, to demonstrate that you have identified all potential use risks and categorised them according to their probability of occurrence and likely severity.

- You must document how you have designed out the risks or, if you couldn't eliminate them, reduced them 'as far as possible.'

- You must also demonstrate that you have tested your solutions (called mitigations) with users in a realistic setting.

The Holy Trinity

Whilst, you need to consider lots of regulations when developing a medical device interface. You must integrate three key, interlinked governing standards at your inception stage:

1. **ISO 14971:2019 Medical devices** – Application of risk management to medical devices
2. **ISO 13485:2016 Medical devices** – Quality management systems – Requirements for regulatory purposes or FDA 21 CFR PART 820
3. **ISO 62366-1: 2015** - Application of usability engineering to medical devices

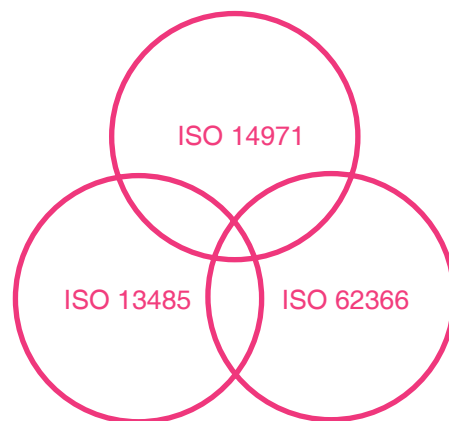


Figure 2: *Three fundamental governing standards*

You must develop your design in line with these standards from the start of the design process. You cannot retrospectively fit your design to them or factor them in at the end. Trying to fit your design to the process can be very expensive as you may find the interface will not pass all the relevant standards.

Integration of usability engineering is mandated by the governing bodies

Although the requirements of ISO 62366 and the US Food and Drug Administration's (FDA) 2011 draft guidance on human factors are not the same, they both require evidence and documentation of a usability engineering process. You must therefore satisfy usability requirements to gain a CE mark or get FDA approval for any medical device.

Our process for developing a medical device interface

Please remember that the word 'interface' refers to all interactions between the user and the device, not just the graphical user interface (GUI).

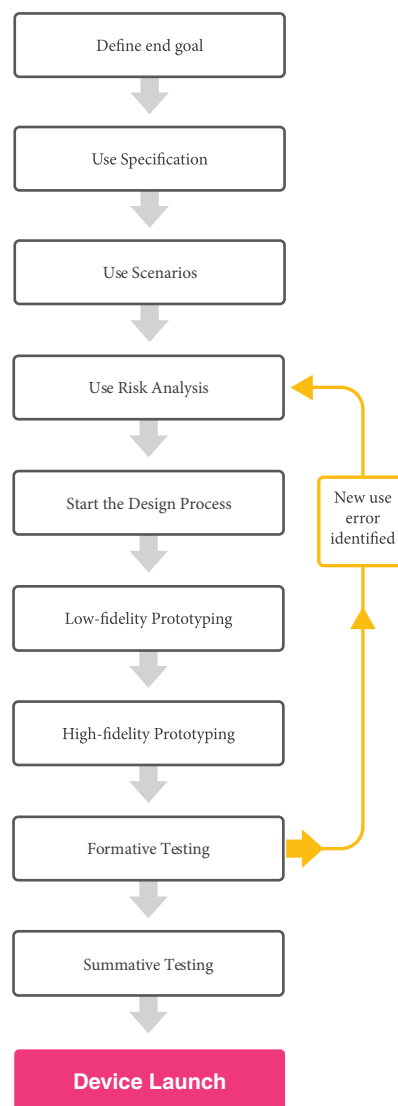


Figure 3: Example of the usability engineering process

Stage 1: Define the end goal

The documented design requirements should (but don't always) state what certifications are going to be sought, for example CE in Europe, URKA in the UK, or FDA in the USA. Each country has its own certifications and each imposes different design requirements that you must follow.

The device will be classified as either:

- a medical device (MD) with a risk classification of 1 to 4 or
- an in vitro diagnostic device (IVD) with a risk classification of A to D.

As a rule of thumb, the higher the risk, the more work you will be required to do to prove that your interface does no harm. As your interface will be integrated into software, it must adhere to ISO 62304 Medical device software – Software life cycle processes. If it can be classified as software as a medical device (SaMD), it must also conform to ISO 82304-1 Health software. Software has a risk classification from A to C, with A the lowest class.

Top tip: Get this straight from the start - changing the standard your interface aligns to late in the development process can be very expensive.

Stage 2: Use specification

The use specification may already exist. If so, it will be documented in the user requirements documentation. If not, you will need to develop your own. Once you have a use specification, talk to your regulatory expert to make sure that it aligns with all of the requirements.

User profiles (user groups) – are descriptions of the different types of user. It is critical that you define your intended users' expertise, their qualifications, training, experience, knowledge, and potential limitations. User groups will be differentiated based on the tasks they perform; some users will conduct different tasks and look for different feedback from the same device.

Use environments are descriptions of the different types of environment in which the device will be used. At a high level, these may be descriptions of places – GP consulting rooms, pathology labs, cardiac catheter laboratories – but they should also include environmental considerations – sterility/cleanliness requirements, appropriate personal protective equipment (e.g., some gloves don't work with touchscreens), permissible levels of exposure to bodily fluids, lighting, noise, temperature, and vibration. Use environment descriptions should also capture any other devices that your device may be used in concert with.

Top tip: Go through Stages 1 and 2 with your regulatory expert. Each user type and environment must be tested and each test costs an additional fee.

Stage 3: Use scenarios

Use scenarios are descriptions of the interactions between users and your device that will result in the required outcome. Each use scenario will comprise multiple tasks/steps and decisions – you will analyse these from a risk perspective in Stage 4.

We recommend developing a workflow diagram or decision tree describing all of the required user inputs and device feedback/outputs for each scenario.

Check this workflow with several different users to ensure that it correctly reflects what is required of the interface. Make sure that you understand the operational or clinical decisions that users are making based on the information the interface is feeding them.

Top tip: Use a simple and accessible flow-chart software that can be easily shared and edited by users.



Figure 4: Example use of a flow chart to help define user interaction

Stage 4: Use risk analysis

The purpose of use risk analysis is to identify potential use errors that could lead to potential patient or user harm. You need to complete the work in Stages 1–3 before you can start this task and create your risk matrix.

Top tip: There are numerous templates and software packages you can use to help you build your risk matrix.

The purpose of use risk analysis is to identify potential use errors that could lead to potential patient or user harm. You need to complete the work in Stages 1–3 before you can start this task and create your risk matrix.

You should follow the risk process set out in ISO 14971:2019 Medical devices – Application of risk management to medical devices.

Your matrix should list and describe each risk, the task it is associated with, and the consequence of any error. You will then rank the likelihood of this risk occurring and the likely severity of the consequence if the error does occur. You should inform this process by researching the history of use errors for pre-existing or similar devices. Your risk matrix will be a living document that you will continue to amend and update as you conduct the remaining stages. You will need to integrate it into your development cycle.

User task	Task requirements	Potential use errors	Consequences & risk severity
Check voltage display	Ensure the voltage is set to the correct level	Faliure to notice incorrect power intensity	Potential error in reading accuracy resulting in burn to skin [High risk]
Set pulse interval	Use radial to input setting	Faliure to deliver correct power setting	Input area resulting in burn to skin [high risk]
Power on Laser	Depress Power button and read power status	Faliure to power up laser	No active laser [Low risk]

Figure 5: Example U/FMEA matrix

Stage 5: User interface specification

Create a draft user interface specification. The purpose of this document is to describe to the development team how the interface should work. This is another living document that will be updated as the design develops.

Share the user interface specification with the team responsible for releasing the final interface – or better still, co-develop it with them.

Top tip: Make sure you physically have the technical constraints – e.g. processor, screen type, software environment (To be use defined, they come with their own critical design constraints that must be incorporated into the design.)

Stage 6: Start the design process

If you're familiar with normal interface design or web design, this is where you will see the design process really change, as mandated by the regulatory requirements.

To design your interface, you need to use the design control process – sometimes referred to as the waterfall process – that is mandated in ISO 13485:2016 Medical devices – Quality management systems – Requirements for regulatory purposes.

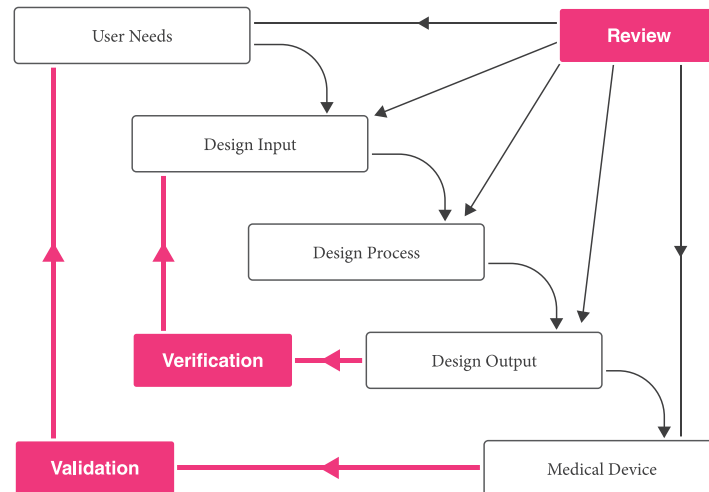


Figure 6: Design process flowchart example

Again, we suggest that you use an existing template or software to track and document this process. Ideally, use one that has revision control to track the changes. Each risk that you identified becomes a user need, which translates into a design input. Your subsequent design results in a potential solution.

You then test the design using a prototype. This step is known as verification (formative testing.) The process is an iterative loop that you follow until you have verified that all of the user's needs are met. Once you are confident in the design, you can integrate the solution into the final device to test. This step is known as validation (summative testing)

The above, is an overly concise summary. If you want to know more, we suggest that you read the Medicines and Healthcare products Regulatory Agency (MHRA) human factors guidelines or the FDA guidelines for human factors.

Stage 7: Low-fidelity prototyping

Develop prototype solutions using build, test and learn loops starting with quick low-fidelity mock-ups (wireframes) that you can test and refine.

Top tip: Start with a simple hand-drawn or PowerPoint wireframe. This allows you to share and test the interface with users. A low-fidelity prototype enables those users to quickly give you input, allowing you to add features and implement any changes as and when they become apparent.

The interface is designed to reduce risks ‘as far as possible’. You need to test that the solutions that have been designed into the interface mitigating the risks captured in your risk analysis.

For each use scenario, you must consider the whole of the interaction between the user and your device. Increasingly, graphical user interfaces (GUIs) – especially touchscreens – make up the lion’s share of user–device interactions, but don’t neglect the user’s interaction with physical inputs and outputs such as knobs, dials, and connectors.

The types of activity that can be used in the testing and development of your interface are laid out in **ANSI/AAMI HE75:2009 Human factors engineering – Design of medical devices**.

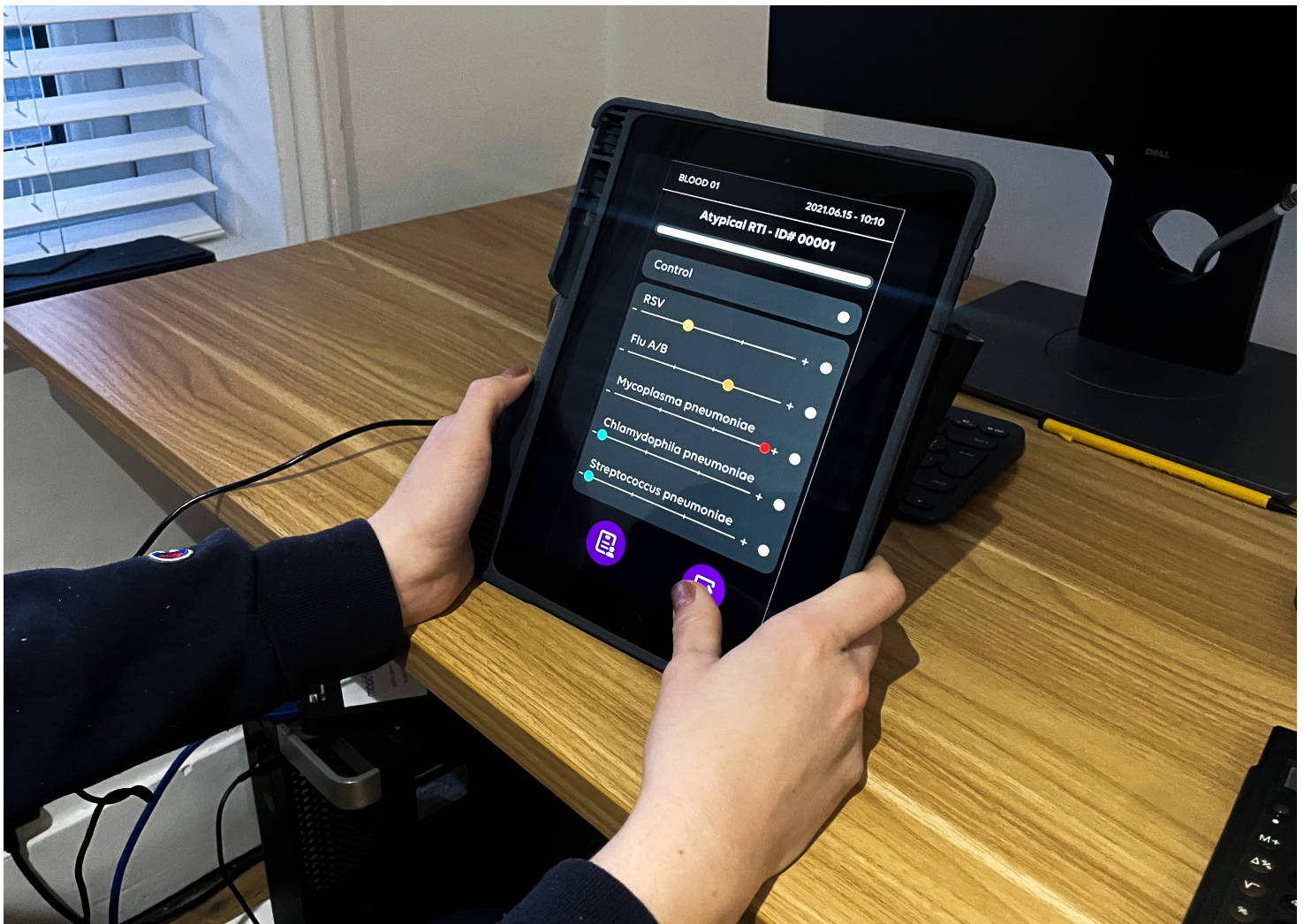


Figure 7: The image shows a low fidelity clickable prototype created for Genomtec. This is an early prototype being used to check and improve the clarity and intuitiveness of the design.

Stage 8: High-fidelity prototyping

As the build, test, learn loops improve your interface, you should find both your confidence and the fidelity of your prototype increasing. Your efforts should culminate in a high-fidelity prototype interface that looks and interacts like the final product. Be sure to document all activity as evidence of your work towards reducing harm.



Figure 8: *The image shows a high fidelity clickable prototype created for Creavo. The image shows a clinical team testing the prototype interface.*

Stage 9: Formative testing

Once you have produced your final prototypes (digital and/or physical) you need to put them through formative testing using on average 5–8* representative end-users for each of your defined user groups.

**Confirm the required number with your human factors or regulatory expert.*

Develop a test procedure (called a protocol) that lays out who and how your testing is to be conducted. You should conduct your formative testing as a simulated use study to assess the overall usability of the entire system.

Top tip: Make sure that you have in place all of the instructions and training that users need to be able to use the product. As it's a simulated use test, you can't interrupt partway through to instruct, prompt, or spoon-feed the user.

To run the test, you will typically need to develop the following:

- Draft instructions for use (IFU) – you may need to test these separately first
- A mock training video explaining how to use the device (if required)
- Labelling
- Packaging (if required)

All labelling, packaging, and any symbols used must adhere to ISO 15223-1:2021 Medical devices – Symbols to be used with information supplied by the manufacturer – Part 1: General requirements.

As described in Stage 6, this is an iterative process. As you go through the above stages, new risks will reveal themselves, which you and your team need to add to the risk matrix you created in Stage 4 and incorporate into your design inputs, outputs, and verification testing. It is easiest to think of Stages 3–9 as a loop – new risks are identified and mitigations are designed, tested, and integrated.

By the time you reach this point, you should have a high degree of confidence in your interface. If not, go back and make any changes then re-test. Keep in mind that after this stage your design will be translated or coded into the final device. It's much cheaper and less painful to make and test changes to prototypes than to the final product.

Stage 10: Summative testing

As a final stage, a summative study – typically required by IEC 62304:2006 Medical device software – Software life cycle processes, and the FDA – is conducted to assess the overall usability of the device and provide final validation of the design. The summative study focuses on critical tasks and overall use, and is performed on the final interface after integration into the final product.

As a guide, testing typically uses 10 to 15 users of each user type. For devices requiring FDA approval, the test must be conducted using US citizens.

If problems are found, a new summative test may be required. Hence, mistakes that make it through to summative testing can prove to be very expensive in terms of both cost and time. It is much more cost-effective to invest in Stages 3–9 than to troubleshoot once the product is finished.

The summative study is carried out in a user research facility with participants recruited to represent the target and anticipated end-users of the system. The results are submitted in a human factors summary report to the relevant authority.



Figure 9: SLE 6000 is a specialist infant respirator and a good example of the development of a critical medical Graphical User Interface. Maddison carefully developed and tested the interface alongside the client as well as clinical users.

Maddisons interface design principles.

1. First, do no harm

The overarching principle for developing a medical device is to use a process and practices that as far as possible reduce the risk of harm.

2. Define the governing regulatory requirements

This is based on the adage: 'Start with the end in mind'. Aligning a development to the wrong requirements can be a very expensive mistake and this can lead to reworking to put right.

3. Involve users and test the device with them early

When users are clinical specialists with expert knowledge, who have been trained on and are accustomed to using predicate or analogous devices. Therefore, they come to your interface with existing heuristics (mental models of likely interactions, operating methods and responses). Testing with users during all development phases, starting simply and adding complexity, ensures that users' existing mental models are harnessed to make your interface safe and intuitive.

4. Understand and use existing medical conventions

The medic about to use your interface for the first time in an emergency has probably been trained on or used one on a similar machine. They will assume that your interface works in the same way. The existing similar interface may not be intuitive to you but it is to the medic.

5. Develop the interface with successive build–test–learn loops

Use an iterative testing process, supporting users to test your interface – starting with a simple prototype that can be changed easily. As your confidence increases through testing, you can refine the prototype and its testing.

6. Design the interface to be intuitive

An intuitive interface reacts in the way that a user would expect. For a user to anticipate a reaction, they hold a heuristic. In the early phases of design, you must discover and map the heuristic. Mimicking the user's heuristic will make your interface intuitive. Where possible, make use of natural mappings – that is, natural arrangements of button or controls carry with them an intrinsic explanation of how they work or interact with a user. An intuitive interface reduces the need for any information from a user's memory to perform a task. (This is described as reducing cognitive load.)

7. Use poka-yoke principles

Mistake-proofing, that is, the addition of a design feature that makes it nearly impossible for a user to misuse the device. This principle can be illustrated in simple terms using a children's toy in which shaped blocks must be pushed through matching holes. It is impossible to put the wrong block in the wrong hole.



8. Design in redundancy

You should assume that your interface will be used in a busy, noisy environment and that it is probably not your user's primary focus. Use multiple simulations to design ways in which your interface communicates with the user, so ensuring that it delivers the feedback needed by the user. Design alternative systems for navigating your interface, thus supporting users with different expectations, especially where there is a legacy system or common competitor. Providing more than one way to complete a task will help your audience achieve their own goals.

9. Design in forgiveness

Humans make mistakes – especially when under pressure, as medics often are – no matter how well your user interface is designed. Use approaches that support users to prevent or recover from errors. This approach is generally called 'forgiveness', the key being to prompt or pre-empt a user when an action they are about to take may result in unexpected, irrevocable, or critical actions. Predict and explain the resulting action to the user, requiring them to confirm that the expected outcome is what they wish and expect. In addition, give them the ability to undo or stop the action if they realise they have made an error.

Disclaimer

Maddison Limited makes no representation and assumes no responsibility for the accuracy of the information contained in this guide, or its suitability for any purpose, and such information is subject to change without notice. You are encouraged to confirm any information obtained from or through this guide with other sources.