

Heuristics in UX Design

An introductory guide to UX heuristics and
how to succeed in using them

Table of Contents

Introduction: What is a heuristic anyways?

Heuristic #1 Findable

Heuristic #2 Accessible

Heuristic #3 Clear

Heuristic #4 Communicative

Heuristic #5 Useful

Heuristic #6 Credible

Heuristic #7 Controllable

Heuristic #8 Valuable

Heuristic #9 Learnable

Heuristic #10 Delightful

Conclusion: So, now what?

Introduction

What is a heuristic anyways?

Wikipedia defines heuristics as “strategies using readily accessible, though loosely applicable, information to control problem solving.” The origin of the word itself lies in curiosity and inquiry. In user experience design, this means applying a specific set of common principles to uncover problems and identify areas for improvement.

At DevFacto, we use them to evaluate, facilitate, and in some cases predict the strength, quality, and effectiveness of our work.

The set of heuristics we use are based on the well-regarded work of [Abby Covert](#)¹, whose list was built in consideration of [Nielsen & Molich](#)², [Peter Morville](#)³, [Lou Rosenfeld](#)⁴, [Ergonomics of Human System Interaction](#)⁵ and most recently [Resmini & Rosati](#)⁶.

Our 10 heuristics are:

- Findable
- Accessible
- Clear
- Communicative
- Useful
- Credible
- Controllable
- Valuable
- Learnable
- Delightful

¹ [abbytheia.com](#)

² [nngroup.com/articles/ten-usability-heuristics](#)

³ [semanticstudios.com](#)

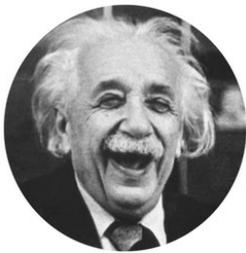
⁴ [louisrosenfeld.com](#)

⁵ [iso.org/standard/52938.html](#)

⁶ [pervasiveia.com](#)

Heuristics, huh?

An origin of curiosity and inquiry



"I have no special talents. I am only passionately curious."

- Albert Einstein

Your first consideration of heuristics should be in how they can drive you to ask the right questions. You can gain value simply by asking questions like "is (some important thing) findable in my current project?" or "does the solution I am building communicate the right things to people using it?"

When you can identify the most applicable heuristics against a particular task or user goal you get closer to achieving meaningful outcomes. Suddenly, you're honing in on what makes a great interaction for that user, and creating a really valuable piece of work.

At [DevFacto¹](https://www.devfacto.com/), we offer heuristic assessment, in which we use heuristics to break down issues in an existing solution and provide insight into what can be done to improve them. For each finding, we list applicable heuristics, severity of the issue, propose a solution, and list the benefits of that solution.

In this e-book we will discuss the top ten heuristics of UX design and how you can utilize them to design better, more usable software for any audience.

¹ <https://www.devfacto.com/>



“Information that’s hard to find will remain information that’s hardly found.”

- Peter Morville

Heuristic #1

Findable

The ease with which information can be located

Can users easily find what they’re looking for? Do you have analytics, or can you measure how long it takes them to find something with a stop watch? Before you can make any improvements, you need to know how long these actions take now. You should also make sure you never make things worse.

But findability isn’t just about time. Often what is intuitive for some isn’t intuitive for all. A user’s history, current task or mode, relative savviness, and context all contribute to how they go about finding things.

Is findability consistent across all channels and devices? Are search results consistently formatted?

Findability starts by understanding who is looking for what, and how. The results are measurable in successful finds in shorter times, but how you go about improving findability might depend on your knowledge of issues occurring outside of technology.

If you are trying to diagnose a findability problem, check out [Low Findability and Discoverability](#)¹ on Nielsen Norman Group.

¹ nngroup.com/articles/navigation-ia-tests



“The power of the web is in its universality. Access by everyone regardless of disability.”

- Tim Berners-Lee

Heuristic #2

Accessible

The ease of approachability or entry

Can users easily approach or enter what you have built?

On the surface accessibility seems really straight forward, but that couldn't be further from the truth. In this age of responsive design, digital tools and services are accessed by all kinds of devices in infinitely different contexts.

Can users easily access what you have built with any device? How consistent is the experience across devices? Do the different form factors create barriers for specific tasks or functions?

How about users with disabilities? Is your markup formatted for screen readers used by people with sight issues? How about colour blindness, or limited motor skills? About 20% of people have some kind of disability, so accessibility should be part of the conversation of any project.

Improving accessibility can be as simple as ensuring your user interface and content have appropriately good contrast in each context they might be used in.



“Why do we deliberately build things that confuse the people who use them?”

- Don Norman

Heuristic #3

Clear

The ease with which a user can understand what is in front of them

Was their education or reading level known during the design process? Are they technical, process or business focused? Is the path to task completion obvious, and free from barriers?

A great test of clarity is listening to a user try and describe what you have built to a co-worker or friend. What parts did they get right? What did they miss completely? Was their tone vague, frustrated, happy, or empowered?

A lot of clarity issues can be found during hallway testing, but keep in mind the known and unknown differences between your coworkers and your users. Clarity can be gained or lost almost anywhere, but critical pieces are often information hierarchy, the visual design of interaction points like buttons and hyperlinks, and microcopy – the language used in those interaction points. You can learn more about microcopy [here](https://uxplanet.org/microcopy-tiny-words-with-a-huge-ux-impact-90140acc6e42)¹.

Just make sure you don't confuse clarity with simplicity. The two are related, but definitely not synonymous.

¹ <https://uxplanet.org/microcopy-tiny-words-with-a-huge-ux-impact-90140acc6e42>

Often times overly simple things profess clarity, when the reality is that they lack expected or even required features. There is a glorious intersection where the features you have built handle the business requirements, map directly to user needs, and do it in a crystal clear interface.

Sometimes clarity is lost simply because of the perceived value of complexity. Think about the 6 controls on a high-end kitchen appliance vs the many knobs, buttons and touch screens on one made by a popular brand. Which one takes you longer to figure out? Which one has you digging out the manual to get things done even after 6 months of use? Which of the many features of the popular model do you think you'll actually use in the first month, or the first year?



“If we want users to like our software, we should design it to behave like a likeable person: respectful, generous, and helpful.”

- Alan Cooper

Heuristic #4

Communicative

The information provided to users throughout their experience

Can users easily tell the status of a system, or their location in it?

A great experience is often supported by a sense of place, which is consistent across all channels and devices. Users feel more comfortable when they know where they are, and empowered when they understand what they can do there. Consistency is king.

Can users easily differentiate between actions and navigation? What about status? Am I signed in? Do I need to be? Is save an action, or a status? How is that communicated? What feedback do you provide for users who have completed an action? How about for those who have failed?

You also need to communicate with the user in a timely fashion. Untested solutions often give users what they need a step after they needed them. This can be worse than not providing a solution at all. Always consider: Have you prepared users for what is ahead of them? Do they know what to expect after each interaction? Does the messaging around tasks support users in doing them?

It is also possible to overwhelm users by over communicating. Think about how many features and tools software bombards you with vs how many you actually use. How many tabs did you have to use to complete what you considered a basic task?



“Ease of use may be invisible, but it’s absence sure isn’t.”

- Overheard at IBM

Heuristic #5

Usable

Facilitates users in producing desired or intended results

A well designed solution facilitates desired results, while removing frustration from difficult processes or previous tools. Does what you have built work well for both new and experienced users trying to complete their work?

How about related information? Are users still hunting around for information that will help them make informed decisions in completing tasks, or have you displayed it where appropriate? Sometimes usefulness is entirely dependent upon context or time.

Think about the last report you produced for a client. Was it easy for users to pick out the key information in a quick scan, or did even simple questions require them to focus for a period of time? We don’t often think about reports as functional, but we need to consider how and why each report is produced, and what each user that receives it actually needs. Is this report part of a deep process with lots of moving parts and years of data, or a timely checkup that leads to other quick actions? Was the formatting appropriate to the use?

The key to building something people love is to make sure they find it useful in the first place.



“Do you support content where it is needed? This is so important when asking for personal or sensitive data.”

- Abby Covert

Heuristic #6

Credible

The quality of being trusted and believed in

A credible tool is one that users have confidence in.

This starts by planning your work around the context of its users. Gamification might not go over very well at a law firm, and users might immediately write off what you have built if it doesn't suit their business. Gamification of a tool built for K-6 students and teachers will likely go much further.

Beyond that initial response you need to retain credibility and the key to that is dependability. Imagine a tool used by hospital staff breaking down at all the wrong moments. Or, an inspection tool for oilpatch tradespeople that rarely has enough bandwidth to sync the large amounts of data they need to collect?

Finally, credibility also comes in a form of timely updates. You can quickly lose trust if users experience problems and are unable to resolve them or find support resources. Always consider layers of help appropriate to the solution, the last of which is easily contacting helpdesk for immediate assistance.



“As far as the customer is concerned, the interface is the product.”

— Jef Raskin

Heuristic #7

Controllable

Puts the user in the driver’s seat

Are tasks and actions users want available and easily found?

This starts with available actions. Does a user have enough power to accomplish their goals? Can they efficiently complete something without causing other problems? A controllable tool does everything to help users avoid errors and gives the ability to quickly recover.

How often do you simply restart if you’ve made a mistake, rather than going through the painful recovery process in a system that doesn’t give you enough control? At some point, we’ve all cancelled online transactions because of a lack of control.

Also, consider the difference in the control a first time user needs against what a long time user gets the most benefit from. Depending on the context, a new user with too much control might ruin things for other users. Similarly, power users, who know exactly what needs to get done, require controls that help them do their job quickly.

You can take controllability a step further by letting users customize their own experience. This could mean allowing users to set up an interface in a way that makes the most sense to them.

Another consideration is offering the user control of how the system remembers information. This could mean letting the user specify cache size or number of undo states.

Heuristic #8

Valuable

Being of great use, service and importance



“Design is not just what it looks like and feels like. Design is how it works.”

- Steve Jobs

Do people even want to use what you have built?

How would someone describe your tool to a friend or co-worker? Do you HAVE to have this? Will it replace four other things? Did it remove that work-related splinter from their soul and clear up the infection in a matter of hours?

If the business case is sound, you know your work will affect the bottom line. That’s why validating business case before kick-off is so critical.

On the flip side, you can deliver value to a business and still fail if the tool you’ve built causes users pain. Users rarely care about the 10% increase in operational efficiency if you’ve given them another soul-destined splinter. And the problem with causing pain is that pain begets pain. Anyone you create pain for is likely going to pass that pain on to the next person, inevitably creating a cycle that eventually lands on the customer. It’s not likely that customers will hang around if your solutions are causing them pain.

This is precisely why software done right is an investment, and done poorly is an expense or liability.

Heuristic #9

Learnable

The quality of being quickly understood with minimal training



“When a design is not intuitive, our attention moves away from what we’re trying to accomplish to how we can get the interface to accomplish what we want.”

- Jared Spool

How quickly can users grasp what you have built?

If you have spent enough time with users to understand their goals and tasks, and have built a solution supporting those things, you will naturally land at an intuitive result. When information is structured after your best understanding of the user model, the gap is so small it shouldn’t even be noticed. By using the model they already know, you’re helping users quickly understand what you’ve built.

But the user model isn’t the only factor in building something learnable. Other design features such as icons can be helpful sign posts, as long as there aren’t too many of them. Colour works the same way. Colour coding is great, until nobody can recall if reports are purple or plum. Or eggplant.

The hierarchy of information on a page or screen also plays a key role. Headings and subheadings, when used appropriately, help users understand what they are looking at and find things quickly. While simple, this is often overlooked.

Ultimately, learnability often comes down to what the user can remember. The trick is in designing things in such a way, that users have to remember very little to be really powerful. By making something impeccably consistent you can ensure that users have predictable and intuitive experiences.

All of these techniques lower cognitive load, decreasing the mental effort on person's working memory. It's a lot like RAM of the human brain.

Think of cognitive load in three layers. The top layer is presentation – how hard must I work to parse this information? The middle layer is learning – how hard must I work to understand this information? The bottom layer is processing – how hard must I work to apply or automate this information?

By reducing the load in the top presentation layer, we allow users to spend more time and effort on learning and processing, thus empowering them to focus on what's important.

There are many tools you can use to build something learnable, some are optional, and some you can't afford to do poorly. For example, you don't always need icons, but you are always modeling information. I urge you to make sure you get that right.

Heuristic #10

Delightful

Exceeding expectations and creating a 'WOW' experience



“User experience is everything. It always has been, but it’s undervalued and underinvested in. If you don’t know user-centered design, study it. Hire people who know it.”

-Evan Williams,
Co-Founder of Twitter

By far the hardest heuristic to deliver on, delight makes things personal. You can begin to harness the elusive creature of delight by differentiating your tool from a competitor’s, crushing user expectations, and providing a great experience across all channels. You will have created something good, maybe even great. But it still might not bring delight.

The first challenge on the path towards delight is that you must deliver on the rest of these heuristics. Any misstep or frustration along the path will deflate a great experience. So truly focus on them. The first nine heuristics can (and should) be measured. They start with science, and may just as well end there. But if you’ve been paying attention, something has been quietly brewing on these pages, and it is that initial spark required to bring delight: Empathy.

Until you can understand what your users are experiencing and see their world from their perspective, you will have a very hard time trying to provide delight.

Work hard to remove pain. Once you do, don't stop working. Keep smoothing things out, keep having conversations with your users, keep chasing this elusive creature. You might catch glimpses of it before harnessing it four iterations later. And it is always worth it.

Why? Because it brings with it an emotional connection, something long thought impossible in conversations between computers and humans. Everyone expects things to work, but the initial bar is often much higher than that.

Good is not good enough anymore, and so you have to ask the question "is what I have built ordinary?" Or have you created something extraordinary?

Conclusion

So, now what?

UX heuristics in the real world

Throughout this paper, we have reviewed our top 10 UX heuristics in software design. While it's easy to understand that these broad, mercurial concepts provide a lot of value – the question that remains is: how?

Ultimately, the goal of heuristics is to improve whatever project we are working on. Because they are rooted in human behavior, these heuristics provide a structure for improving the experience a human will have when using our software.

Note however, that solutions encompassing these heuristics do not necessarily promise perfection. While rooted in research, without a focused application, heuristic analysis on its own does not lead to quantifiable results.

Key Takeaways

1. Consider the lens

Heuristics act as lenses through which you can view your work with fresh eyes. Find the lenses that are the most valuable to your project, and start asking questions.

2. Consider the output

You might need this list to help frame discussions with your team during sprint reviews, or you might need to produce a report to sell change to leadership. In doing so, ensure that you consider: who is your audience, and do they value speed or rigor?

3. Do your research

Heuristic review is not a replacement for user research. As you map out a plan, don't be shy about seeking advice. At DevFacto, we're always happy to hear more about your situation and share recommendations that are best fit for your solution.



The foundation of any successful solution is a great user experience. Applications must be usable, accessible and enjoyable to interact with in order to survive.

At DevFacto, we build and manage software that humans love to use. Our UX specialists design applications to minimize learning curves, reduce user frustration and increase productivity.

[Find Out More](#) ¹

[Get in Touch](#) ²

¹ devfacto.com/services

² devfacto.com/contact-us